



ΕΘΝΙΚΟ ΜΕΤΣΟΒΕΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ &
ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

**Ανάλυση και Μοντελοποίηση Ανθρώπινων Δραστηριοτήτων στα
Πληροφοριακά Συστήματα, και Ανάπτυξη Πρότυπης Εφαρμογής για
Καταγραφή, Παρουσίαση και Παρακολούθηση Προσωπικών
Δραστηριοτήτων**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Ευάγγελου Αρβανιτάκη

Επιβλέπων: Ασκούνης Δημήτριος
Αν. Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2015



ΕΘΝΙΚΟ ΜΕΤΣΟΒΕΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ &
ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

**Ανάλυση και Μοντελοποίηση Ανθρώπινων Δραστηριοτήτων στα
Πληροφοριακά Συστήματα, και Ανάπτυξη Πρότυπης Εφαρμογής για
Καταγραφή, Παρουσίαση και Παρακολούθηση Προσωπικών
Δραστηριοτήτων**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Ευάγγελου Αρβανιτάκη

Επιβλέπων: Ασκούνης Δημήτριος
Αν. Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 9η Ιουλίου 2015.

Ασκούνης Δημήτριος
Αν. Καθηγητής

Ψαρράς Ιωάννης
Καθηγητής

Ασημακόπουλος Βασίλειος
Καθηγητής

Αθήνα, Ιούλιος 2015



ΕΘΝΙΚΟ ΜΕΤΣΟΒΕΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ &
ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό.

Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Ευάγγελος Αρβανιτάκης

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Ηλ. Υπολογιστών

Copyright © 2015 – Με επιφύλαξη παντός δικαιώματος. All rights reserved

Περίληψη

Οι γρήγοροι ρυθμοί στους οποίους κινούνται οι κοινωνίες αναγκάζουν τους ανθρώπους να πραγματοποιούν σε καθημερινή βάση μια πληθώρα δραστηριοτήτων. Παράλληλα, η εξελισσόμενη τεχνολογία καθιστά δυνατή τη δημιουργία πληροφοριακών συστημάτων και εφαρμογών με σκοπό την καταγραφή και την παρουσίαση στατιστικών αποτελεσμάτων για ένα σύνολο εξειδικευμένων δραστηριοτήτων της ζωής των ανθρώπων. Καθένα αναφερθέν σύστημα έχει υλοποιηθεί με σκοπό να ανταποκρίνεται βέλτιστα στο σύνολο της κατηγορίας δραστηριοτήτων που αναφέρεται, γεγονός που οδηγεί αυτόματα σε ένα μεγάλο αριθμό διαφορετικών μοντέλων και σχεδιασμών.

Σκοπός της παρούσας διπλωματικής εργασίας ήταν η δημιουργία ενός γενικού μοντέλου, ικανού για την γενική περιγραφή όλων δραστηριοτήτων και η χρήση του στην ανάπτυξη διαδικτυακής εφαρμογής με όνομα «Activity Tracker». Συγκεκριμένα, έγινε μελέτη των διαφορετικών γνωστών οντολογιών, αναπτύχθηκε πλήρες μοντέλο περιγραφής ανθρωπίνων δραστηριοτήτων γενικού τύπου, καθορίστηκαν οι λειτουργικές και μη-λειτουργικές απαιτήσεις του συστήματος, αναπτύχθηκε το μοντέλο οντοτήτων-συσχετίσεων της βάσης, σχεδιάστηκε η αρχιτεκτονική του συστήματος και τέλος, επιλέχθηκαν προγραμματιστικά εργαλεία και υλοποιήθηκε διαδικτυακή εφαρμογή βασισμένη στο προαναφερθέν μοντέλο.

Η εφαρμογή αυτή στοχεύει αφενός στην καταγραφή και παρακολούθηση καθημερινών δραστηριοτήτων των χρηστών και αφετέρου στη διαγραμματική παρουσίαση στατιστικών αποτελεσμάτων προσαρμοσμένων στα εισακτέα δεδομένα του χρήστη. Με τον τρόπο αυτό, ο χρήστης μπορεί να παρακολουθήσει, να ελέγξει και να βελτιώσει την καθημερινότητά του. Το προτέρημα της εφαρμογής «Activity Tracker» έναντι των υπολοίπων είναι η χρήση κοινού μοντέλου περιγραφής, ανεξάρτητου της κατηγορίας/τύπου της δραστηριότητας. Ακόμα, η συσσώρευση όλων των δραστηριοτήτων σε μια κοινή πλατφόρμα, δίνει τη δυνατότητα ευκολότερης παρακολούθησης και άμεσης σύγκρισης – διαγραμματικής και μη – κάθε πιθανού συνδυασμού δραστηριοτήτων. Τέλος, η ομοιομορφία στην αντιμετώπιση όλων των δραστηριοτήτων, προσφέρει στην εφαρμογή τη δυνατότητα επέκτασής της, ώστε να δράσει ως ενδιάμεσο σύστημα που θα αφαιρεί τη μοντελοποίηση του εκάστοτε εξειδικευμένου παρόχου δραστηριοτήτων και θα μοντελοποιεί με κοινό τρόπο κάθε ανθρώπινο γεγονός.

Λέξεις κλειδιά:

Ανθρώπινες Δραστηριότητες, Μοντελοποίηση Δραστηριοτήτων, Εφαρμογή Διαδικτύου, Γενικό Μοντέλο Περιγραφής, Ανάλυση Δραστηριοτήτων, Διαγραμματική Απεικόνιση Ανθρώπινων Ενεργειών, Σύστημα Διαχείρισης Δραστηριοτήτων, Καταγραφή Δραστηριοτήτων, Κοινωνικά Μέσα, Προγραμματιστικές Διεπαφές

Abstract

The fast-paced society in which we are living, forces people to perform an abundance of activities on a daily basis. Simultaneously, the ever evolving technology enables the development of applications and information systems, which monitor actions and present users with statistical results for a group of specialized, daily-life activities. These systems are developed in such a way, so as to perform and respond optimally only to a specific type of activities, which instantly leads to a mass development of different models and designs that try to encapsulate best, the context of each activity type.

The scope of this thesis was the development of a generic model, able to describe every single type of activity and its context in the best way possible, and a web application entitled “Activity Tracker” that utilizes the aforementioned model. Specifically, the known ontologies were studied, a generic activity model was developed, the functional and non-functional system requirements were specified, the entity-relationship diagram of the database was created, the architecture of the overall system was designed and finally, the optimal programming tools were selected and the web application was developed.

This application aims not only at the recording and monitoring of the user’s daily activities, but also at the diagrammatic depiction and presentation of analytics, based on the recorded data. This way, the user can monitor, control and improve his everyday life. The advantage of the “Activity Tracker” application compared to the rest, is the use of the same model, independent of the category or type of each activity. In addition, the accumulation of all of the user’s activities in the same platform, provides the opportunity of an easier observation and instant comparison – through diagrams or not – of every possible activity combination, thus enabling the user to check how his activities influence and affect each other.

Lastly, the uniformity in dealing with every kind of activity, enables the application to be extended, in order to act as a middleware that strips the modelization of each service and applies the generic modelization of this thesis, so that other third-party applications and systems can treat all activities in the same way.

Keywords:

Human Activities, Activity Modeling, Web Application, Generic Depiction Model, Activity Analysis, Diagrammatic Depiction of Human Actions, Activity Management System, Activity Tracking, Social Media, Application Programmable Interfaces

Ευχαριστίες

Με την ολοκλήρωση της διπλωματικής μου εργασίας αισθάνομαι την ιδιαίτερη υποχρέωση να εκφράσω τις θερμότερες ευχαριστίες μου στον επιβλέποντα και Αναπληρωτή Καθηγητή του ΕΜΠ κ. Ασκούνη Δημήτριο, για την επίβλεψη της παρούσας διπλωματικής εργασίας και για την ευκαιρία που μου έδωσε να την εκπονήσω στο εργαστήριο Συστημάτων Αποφάσεων & Διοίκησης.

Στη συνέχεια, θα ήθελα να ευχαριστήσω ιδιαίτερος τον συνεργάτη του εργαστηρίου Συστημάτων Αποφάσεων & Διοίκησης και υποψήφιο Διδάκτωρ κ. Αλβέρτη Ιωσήφ για την εξαιρετική συνεργασία και καθοδήγηση σε όλη την πορεία της διπλωματικής μου εργασίας. Η βοήθεια και οι συμβουλές του ήταν καταλυτικής σημασίας για την επιτυχή ολοκλήρωσή της και ελπίζω πραγματικά να μου δοθεί ξανά η ευκαιρία συνεργασίας μαζί του στο μέλλον. Επίσης, θα ήθελα να ευχαριστήσω τη συνεργάτη εργαστηρίου και υποψήφια Διδάκτωρ κα. Μπιλίρη Ευμορφία για τη βοήθειά της σε τεχνικά ζητήματα και θέματα υλοποίησης.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια και τους φίλους που με στήριξαν σε πολλά επίπεδα κατά την πολυετή μου πορεία στο Εθνικό Μετσόβιο Πολυτεχνείο, όπως και τους καθηγητές της σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Ηλ. Υπολογιστών που με καθοδήγησαν και με εισήγαγαν στον ευρύ και ξεχωριστό κόσμο του ηλεκτρολόγου μηχανικού.

Πίνακας Περιεχομένων

1 Εισαγωγή	19
1.1 Πληροφοριακά Συστήματα & Εφαρμογές Καταγραφής Ανθρωπίνων Δραστηριοτήτων	19
1.2 Αντικείμενο Διπλωματικής	20
1.2.1 Συνεισφορά	21
1.3 Οργάνωση Κειμένου	22
2 Θεωρητικό Υπόβαθρο	23
2.1 Πανταχού Παρούσα Πληροφορική & Διεσδυτικά Συστήματα	24
2.2 Θεωρία Δραστηριοτήτων (Activity Theory)	25
2.3 Μοτίβα Οντολογιών για Μοντελοποίηση Περίπλοκων Δραστηριοτήτων (Ontology Patterns for Complex Activity Modelling)	26
2.4 Οντολογική Αναγνώριση Δραστηριοτήτων σε Έξυπνα Διεσδυτικά Περιβάλλοντα (Ontology Based Activity Recognition in Intelligent Pervasive Environments).....	28
2.5 Εις Βάθος Κατανόηση και Χρήση Συμπεριφοράς: Η Υπολογιστική Προσέγγιση Συμπεριφοράς (In-depth behavior understanding and use: The behavior informatics approach)	29
2.6 Μοντελοποίηση Εφαρμογής «Activity Tracker»	30
3 Σχετικές Εργασίες	33
3.1 Εφαρμογή «Reporter»	33
3.2 Εφαρμογή «Gyroscope»	34
3.3 Εφαρμογή «MyActivities»	35
3.4 Εξειδικευμένες Εφαρμογές	38
4 Ανάλυση Συστήματος	39
4.1 Απαιτήσεις συστήματος	39
4.1.1 Λειτουργικές Απαιτήσεις.....	39
4.1.2 Μη-Λειτουργικές Απαιτήσεις	39
4.2 Αρχιτεκτονική Συστήματος.....	40
4.3 Περιγραφή Λειτουργιών Υποσυστημάτων	42

4.3.1 Εγγραφή ενός Χρήστη στο Σύστημα με Συμπλήρωση Πεδίων.....	43
4.3.2 Τροποποίηση Στοιχείων & Διαγραφή Χρήστη από το Σύστημα.....	44
4.3.3 Σύνδεση (και Αποσύνδεση) ενός Χρήστη στο Σύστημα με Συμπλήρωση Πεδίων.....	45
4.3.4 Εγγραφή & Σύνδεση Χρήστη μέσω Μέσων Κοινωνικής Δικτύωσης.....	46
4.3.5 Εισαγωγή Δραστηριότητας στο Σύστημα	49
4.3.6 Επισκόπηση Δραστηριότητας στο Σύστημα.....	51
4.3.7 Τροποποίηση Δραστηριότητας στο Σύστημα.....	52
4.3.8 Διαγραφή Δραστηριότητας από το Σύστημα	53
4.2.9 Εισαγωγή Δραστηριοτήτων μέσω Τρίτων Εφαρμογών & Παρόχων.....	54
4.3.10 Επισκόπηση μέσω Ημερολογίου (Calendar) & Πλέγματος (Grid)	57
4.3.11 Επισκόπηση μέσω Χρονοδιαγράμματος (Timeline)	60
4.3.12 Δημιουργία Διαγραμμάτων & Παροχή Στατιστικών Στοιχείων από τα Εισαχθέντα Δεδομένα ..	61
4.4 Μοντέλο Οντοτήτων Συσχετίσεων.....	63
5 Σχεδίαση Συστήματος	67
5.1 Ανάπτυξη με Γνώμονα το Σχεδιασμό (Design-Driven Development).....	67
5.2 Γλώσσες & Πλαίσια Προγραμματισμού	70
5.2.1 Ανάπτυξη Πλευράς Πελάτη (Front-End Development)	70
5.2.2 Ανάπτυξη Πλευράς Εξυπηρετητή (Back-End Development).....	70
5.3 Αρχιτεκτονική Συστήματος σε Επίπεδο Κλάσεων.....	72
5.3.1 Κλάση User	74
5.3.2 Κλάση Activity.....	75
5.3.3 Κλάση Performs.....	77
5.3.4 Κλάση Places	79
5.3.5 Κλάση UserVerification.....	79
5.3.6 Κλάση UserSocialAuth.....	80
5.3.7 Κλάση PerformsProviderInfo	81
5.3.8 Κλάση Friend.....	81
5.3.9 Κλάση Object	82
5.3.10 Κλάση hasSuperActivity.....	83

5.4 Αρχιτεκτονική Συστήματος Πλαισίου Django	83
5.4.1 Οθόνη Login	84
5.4.2 Οθόνη Forgot Password.....	86
5.4.3 Οθόνη Χειρονακτικής Εγγραφής στο Σύστημα	87
5.4.4 Βοηθητικές Οθόνες	89
5.4.5 Οθόνη Index.....	92
5.4.6 Οθόνη Goals	99
5.4.7 Οθόνη Timeline.....	102
5.4.8 Οθόνη Settings	106
5.4.9 Οθόνη Friend Analytics	111
5.4.10 Οθόνη Object Analytics	119
5.4.11 Οθόνη Goal Analytics.....	122
5.4.12 Οθόνη Place Analytics.....	131
5.4.13 Οθόνη Activity Analytics.....	137
5.4 Βάση Δεδομένων.....	145
6 Υλοποίηση Συστήματος.....	147
6.1 Πλατφόρμες, Βιβλιοθήκες & Προγραμματιστικά Εργαλεία	147
6.2 Λεπτομέρειες Υλοποίησης	150
6.2.1 Αλγόριθμος Υπολογισμού Απόστασης	150
6.2.2 Μονάδα Αυτόματου Συγχρονισμού Δεδομένων	151
6.2.3 Δομές Δεδομένων & Βελτιστοποιήσεις.....	152
6.3 Στιγμιότυπα & Καθοδήγηση	153
6.4 Μεθοδολογία & Αναλυτική Παρουσίαση Ελέγχου	166
7 Επίλογος	167
7.1 Σύνοψη & Συμπεράσματα.....	167
7.2 Μελλοντικές Επεκτάσεις.....	169
8 Βιβλιογραφία	173
9 Παραρτήματα.....	175

9.1 Παράρτημα Ι: Οδηγίες εγκατάστασης.....	175
--	-----

Πίνακας Εκθεμάτων

Κεφάλαιο 2

Σχήμα 2.1 Activity Theory - Διάγραμμα Engestrom (1987)	26
Σχήμα 2.2 Μοντελοποίηση Δραστηριότητας του DOA	27
Σχήμα 2.3 Μοντελοποίηση Δραστηριότητας κατά Chen - Nugent (2011).....	28
Σχήμα 2.4 Μοντελοποίηση Δραστηριοτήτων Activity Tracker	32
Εικόνα 3.1 Εφαρμογή Reporter	34
Εικόνα 3.2 Εφαρμογή Gyroscope	35
Εικόνα 3.3 Εφαρμογή MyActivities	36
Εικόνα 3.3 Εφαρμογή MyActivities	36
Εικόνα 3.4 Πλατφόρμες και Εφαρμογές Small Data	38
Σχήμα 4.1 Αρχιτεκτονική Διασύνδεση Συστήματος και Υπομονάδων	41
Σχήμα 4.2 Διάγραμμα Εγγραφής Χρήστη στην Εφαρμογή.....	44
Σχήμα 4.3 Διάγραμμα Διαγραφής ή Τροποποίησης Δεδομένων Χρήστη	45
Σχήμα 4.4 Διάγραμμα Σύνδεσης και Αποσύνδεσης Χρήστη Εφαρμογής.....	46
Σχήμα 4.5 Διάγραμμα Σύνδεσης/Εγγραφής Χρήστη στην Εφαρμογή με χρήση Μέσων Κοινωνικής Δικτύωσης.....	49
Σχήμα 4.6 Διάγραμμα Εισαγωγής Δραστηριότητας	51
Σχήμα 4.7 Διάγραμμα Επισκόπησης Δραστηριότητας	52
Σχήμα 4.8 Διάγραμμα Επισκόπησης Δραστηριότητας	53
Σχήμα 4.9 Διάγραμμα Διαγραφής Δραστηριότητας	54
Σχήμα 4.10 Διάγραμμα Εισαγωγής Δραστηριότητας μέσω Τρίτων Εφαρμογών και Παρόχων.....	57
Σχήμα 4.11 Διάγραμμα Διαδικασίας Επισκόπησης Δραστηριοτήτων μέσω Ημερολογίου και Πλέγματος.....	59
Σχήμα 4.12 Διάγραμμα Διαδικασίας Επισκόπησης Δραστηριοτήτων μέσω Χρονοδιαγράμματος	61
Σχήμα 4.13 Διάγραμμα Παρουσίασης Στατιστικών Στοιχείων και Διαγραμμάτων	63
Σχήμα 4.14 Μοντέλο Οντοτήτων Συσχετίσεων του Activity Tracker.....	64
Εικόνα 5.1 Διεπιφάνεια Σύνδεσης και Εγγραφής Χρήστη στο Σύστημα	68
Εικόνα 5.2 Επισκόπηση Πλέγματος και Ημερολογίου	68
Εικόνα 5.3 Διεπαφή Βασικών Λειτουργιών των Δραστηριοτήτων	69
Εικόνα 5.4 Μοντέλο Παρουσίασης Αναλυτικών Στοιχείων και Διαγραμματικών Αναπαραστάσεων.....	69

Σχήμα 5.2 Διασύνδεση Πλαισίου Django.....	72
Σχήμα 5.1 Διασύνδεση Προτύπου MVC	72
Σχήμα 5.3 Απλουστευμένο Block Diagram των Κλάσεων της Εφαρμογής.....	73
Πίνακας 5.1 Πεδία Κλάσης User κληρονομημένα από AbstractUser	74
Πίνακας 5.2 Επιπλέον Πεδία Κλάσης User ως Επέκταση AbstractUser	74
Πίνακας 5.3 Μέθοδοι Κλάσης User κληρονομημένοι από AbstractUser.....	75
Πίνακας 5.4 Υποστηριζόμενες Δραστηριότητες Συστήματος.....	76
Πίνακας 5.5 Κατηγορίες Δραστηριοτήτων και Αντιστοίχιση Χρωμάτων.....	77
Πίνακας 5.6 Πεδία Κλάσης Activity.....	77
Πίνακας 5.7 Πεδία Κλάσης Performs	78
Πίνακας 5.8 Μέθοδοι Κλάσης Performs.....	78
Πίνακας 5.9 Πεδία Κλάσης Places	79
Πίνακας 5.10 Πεδία Κλάσης UserVerification.....	79
Πίνακας 5.11 Πεδία Κλάσης UserSocialAuth	80
Πίνακας 5.12 Μέθοδοι Κλάσης UserSocialAuth.....	80
Πίνακας 5.13 Πεδία Κλάσης PerformsProviderInfo.....	81
Πίνακας 5.14 Πεδία Κλάσης Friend	82
Πίνακας 5.15 Πεδία Κλάσης Object.....	82
Πίνακας 5.16 Πεδία Κλάσης hasSuperActivity	83
Πίνακας 5.17 Όψεις Συσχετιζόμενες με την Οθόνη Login.....	85
Πίνακας 5.18 Μονάδες και Συναρτήσεις Χειρισμού Πελάτη Συσχετιζόμενες με την Οθόνη Login	86
Πίνακας 5.19 Όψεις Συσχετιζόμενες με την Οθόνη Forgot Password	87
Πίνακας 5.20 Μονάδες και Συναρτήσεις Χειρισμού Πελάτη Συσχετιζόμενες με την Οθόνη Forgot Password	87
Πίνακας 5.21 Όψεις Συσχετιζόμενες με την Οθόνη Register	88
Πίνακας 5.22 Μονάδες και Συναρτήσεις Χειρισμού Πελάτη Συσχετιζόμενες με την Οθόνη Register	89
Πίνακας 5.23 Όψεις Συσχετιζόμενες με τις Βοηθητική Οθόνες.....	90
Πίνακας 5.24 Μονάδες και Συναρτήσεις Χειρισμού Πελάτη Συσχετιζόμενες με τις Βοηθητικές Οθόνες.....	91
Πίνακας 5.25 Όψεις Συσχετιζόμενες με την Οθόνη Index	95
Πίνακας 5.26 Συναρτήσεις και Μονάδες Χειρισμού Πελάτη Συσχετιζόμενες με την Οθόνη Index.....	99

Πίνακας 5.27 Όψεις Συσχετιζόμενες με την Οθόνη Goals	100
Πίνακας 5.28 Συναρτήσεις και Μονάδες Χειρισμού Πελάτη Συσχετιζόμενες με την Οθόνη Goals	101
Πίνακας 5.29 Όψεις Συσχετιζόμενες με την Οθόνη Timeline	103
Πίνακας 5.30 Συναρτήσεις και Μονάδες Χειρισμού Πελάτη Συσχετιζόμενες με την Οθόνη Timeline ..	106
Πίνακας 5.31 Όψεις Συσχετιζόμενες με την Οθόνη Settings	107
Πίνακας 5.32 Συναρτήσεις και Μονάδες Χειρισμού Πελάτη Συσχετιζόμενες με την Οθόνη Settings....	111
Πίνακας 5.33 Ανάλυση Οθόνης Friends Analytics.....	116
Πίνακας 5.34 Όψεις Συσχετιζόμενες με την Οθόνη Friend Analytics.....	117
Πίνακας 5.35 Συναρτήσεις και Μονάδες Χειρισμού Πελάτη Συσχετιζόμενες με την Οθόνη Friend Analytics	119
Πίνακας 5.36 Όψεις Συσχετιζόμενες με την Οθόνη Object Analytics	121
Πίνακας 5.37 Συναρτήσεις και Μονάδες Χειρισμού Πελάτη Συσχετιζόμενες με την Οθόνη Friend Analytics	122
Πίνακας 5.38 Ανάλυση Οθόνης Goal Analytics.....	127
Πίνακας 5.39 Όψεις Συσχετιζόμενες με την Οθόνη Goal Analytics	129
Πίνακας 5.40 Συναρτήσεις και Μονάδες Χειρισμού Πελάτη Συσχετιζόμενες με την Οθόνη Goal Analytics	130
Πίνακας 5.41 Ανάλυση Οθόνης Place Analytics.....	134
Πίνακας 5.42 Όψεις Συσχετιζόμενες με την Οθόνη Place Analytics	135
Πίνακας 5.43 Συναρτήσεις και Μονάδες Χειρισμού Πελάτη Συσχετιζόμενες με την Οθόνη Place Analytics	137
Πίνακας 5.44 Ανάλυση Οθόνης Activity Analytics	142
Πίνακας 5.45 Όψεις Συσχετιζόμενες με την Οθόνη Activity Analytics.....	143
Πίνακας 5.46 Συναρτήσεις και Μονάδες Χειρισμού Πελάτη Συσχετιζόμενες με την Οθόνη Activity Analytics	145
Σχήμα 5.4 Σχεσιακό Διάγραμμα Βάσης Δεδομένων Εφαρμογής	146
Εικόνα 6.1 Στιγμιότυπο Οθόνης Login.....	154
Εικόνα 6.2 Στιγμιότυπο Οθόνης Register	154
Εικόνα 6.3 Στιγμιότυπο Οθόνης Forgot Password	155
Εικόνα 6.4 Στιγμιότυπο Οθόνης Index (πάνω τμήμα).....	156
Εικόνα 6.5 Στιγμιότυπο Οθόνης Index (κάτω μέρος).....	156
Εικόνα 6.6 Στιγμιότυπο Οθόνης Index κατά την Προσθήκη Δραστηριότητας	157

Εικόνα 6.7 Στιγμιότυπο Βοηθητικής Οθόνης showActivity.....	157
Εικόνα 6.8 Στιγμιότυπο Βοηθητικής Οθόνης showGroupActivity.....	158
Εικόνα 6.9 Στιγμιότυπο Βοηθητικής Οθόνης editActivity	158
Εικόνα 6.10 Στιγμιότυπο Οθόνης Goals	159
Εικόνα 6.11 Στιγμιότυπο Οθόνης Timeline.....	160
Εικόνα 6.12 Στιγμιότυπο Οθόνης Settings	161
Εικόνα 6.13 Στιγμιότυπο Οθόνης Settings κατά την Επεξεργασία Προσωπικών Στοιχείων	161
Εικόνα 6.14 Στιγμιότυπο Οθόνης Settings κατά την Προσθήκη Τοποθεσίας	162
Εικόνα 6.15 Στιγμιότυπο Οθόνης Friend Analytics.....	163
Εικόνα 6.16 Στιγμιότυπο Οθόνης Object Analytics	163
Εικόνα 6.17 Στιγμιότυπο Οθόνης Goal Analytics	164
Εικόνα 6.18 Στιγμιότυπο Οθόνης Location Analytics.....	164
Εικόνα 6.19 Στιγμιότυπο Χάρτη Οθόνης Location Analytics	165
Εικόνα 6.20 Στιγμιότυπο Οθόνης Activity Analytics.....	165

1

Εισαγωγή

1.1 Πληροφοριακά Συστήματα & Εφαρμογές Καταγραφής Ανθρωπίνων Δραστηριοτήτων

Στην εποχή μας, με την θεσμοθέτηση της τεχνολογίας και ειδικότερα των κινητών τηλεφώνων τελευταίας γενιάς (smartphones) στην καθημερινή μας ζωή, η τάση ανάπτυξης συστημάτων και εφαρμογών για την καταγραφή κάθε μορφής πληροφορίας γίνεται όλο και εμφανέστερη. Από το οργανόγραμμα μιας επιχείρησης μέχρι τον κατάλογο ενός καταστήματος, η ψηφιοποίηση της πληροφορίας σε συνδυασμό με τις δυνατότητες του παγκοσμίου ιστού, προσφέρουν πλέον πολλά πλεονεκτήματα στον κάθε χρήστη.

Ανάμεσα στις αμέτρητες καταγράψιμες πληροφορίες, ανήκουν και οι καθημερινές δραστηριότητες του ανθρώπου. Οι ρυθμοί ζωής του 21^{ου} αιώνα έχουν οδηγήσει στην αύξηση των ημερήσιων δραστηριοτήτων και ενεργειών που πραγματοποιεί κατά μέσο όρο ένας άνθρωπος, συνεπώς, τώρα περισσότερο από ποτέ, γίνεται χρήσιμη η καταγραφή τους. Για το σκοπό αυτό έχουν δημιουργηθεί εκατοντάδες εφαρμογές (κυρίως για κινητές συσκευές smartphones) στις οποίες εισάγονται αυτόματα ή χειροκίνητα οι ενέργειες των χρηστών. Καθώς οι ανθρώπινες δραστηριότητες ποικίλουν εξαιρετικά τόσο σε κατηγοριοποίηση όσο και σε αριθμό, κάθε μια εκ των εφαρμογών στοχεύει στη βέλτιστη καταγραφή – και κατ' επέκταση παρουσίαση στατιστικών και διαγραμμάτων – για μια συγκεκριμένη κατηγορία δραστηριοτήτων. Για παράδειγμα υπάρχουν εφαρμογές για την καταγραφή δραστηριοτήτων σχετικών με την άθληση (τρέξιμο, ορειβασία, κολύμβηση, ποδηλασία κ.ά.), με την υγεία (βήματα, ακινησία, κάπνισμα), ακόμα και εφαρμογές για την καταγραφή του ύπνου. Αυτό σημαίνει ότι η κάθε εφαρμογή μοντελοποιεί διαφορετικά τα δεδομένα, έτσι ώστε να ανταποκρίνεται βέλτιστα στους στόχους της. Η έννοια λοιπόν και το μοντέλο της δραστηριότητας μπορεί να παίρνει εξ' ολοκλήρου διαφορετική μορφή από εφαρμογή σε εφαρμογή, δημιουργώντας έτσι ένα πρόβλημα συσχέτισης.

Οι περισσότερες από αυτές τις εφαρμογές δίνουν τη δυνατότητα στο χρήστη να δει αναλυτικά στοιχεία για τις δραστηριότητες που πραγματοποίησε, όπως και να τις συγκρίνει μεταξύ τους. Για παράδειγμα, στις εφαρμογές που επικεντρώνονται στην άθληση, είναι αρκετά εύκολο να δει κανείς πόσο έτρεξε σε σχέση με το πόσο έκανε ποδήλατο σε ένα ορισμένο χρονικό πλαίσιο. Αυτό που ο χρήστης δεν έχει την ευκαιρία να παρατηρήσει είναι το πόσο έτρεξε σε σχέση με το πόσο κάπνισε ή εάν η μεταβολή στη δραστηριότητα του

καπνίσματος οδήγησε σε μια μεταβολή στις μετρικές του τρεξίματος. Υπάρχουν συνεπώς κοινά αποδεκτές συσχετίσεις μεταξύ των δραστηριοτήτων (π.χ. κάπνισμα – άθληση), όπως επίσης και συσχετίσεις, γνωστές ή άγνωστες για το χρήστη, που αναφέρονται υποκειμενικά σε αυτόν (π.χ. μελέτη – κατανάλωση καφέ), οι οποίες θα ήταν πολύ χρήσιμο να μπορούν να οπτικοποιηθούν. Είναι εμφανές λοιπόν ότι η σύγκριση δραστηριοτήτων διαφορετικών κατηγοριών και τύπων, μπορεί να αποδειχθεί εξαιρετικά αναγκαία για πολλούς χρήστες. Το πρόβλημα στην υλοποίηση του τελευταίου είναι η διαφοροποίηση στη μοντελοποίηση των δραστηριοτήτων. Αυτό αυτόματα οδηγεί την κάθε εφαρμογή που θέλει να συσχετίσει δεδομένα με μια άλλη, στην ανάγκη να δημιουργήσει ένα στάδιο «μετάφρασης» της μοντελοποίησης της δεύτερης στη δική της. Πολλές φορές ωστόσο κάτι τέτοιο δεν είναι εφικτό καθώς η επιλεγμένη μοντελοποίηση της εφαρμογής ανταποκρίνεται καθαρά και αποκλειστικά σε συγκεκριμένους τύπους δραστηριοτήτων και η «μετάφραση» αυτή απαιτεί την επέκταση του μοντέλου ώστε να μπορεί να ανταποκρίνεται στην επιπρόσθετη πληροφορία. Επιπλέον, η κλιμάκωση αυτού, σημαίνει την δημιουργία ενός τέτοιου σταδίου για κάθε άλλη εφαρμογή, πράγμα μη αποδοτικό για μια κερδοσκοπική επιχείρηση. Συνεπώς, η ύπαρξη ενός ενδιάμεσου μέσου που θα πραγματοποιούσε μια μετάφραση των πληροφοριών/δραστηριοτήτων κάθε παρόχου/εφαρμογής σε μια ενιαία κοινά αποδεκτή μορφή θα ήταν πολύ χρήσιμη καθώς η εκάστοτε εφαρμογή θα μπορούσε να λάβει δεδομένα από κάθε άλλη εφαρμογή σε μια σταθερή μορφή, μη μεταβαλλόμενη από δραστηριότητα σε δραστηριότητα, από κατηγορία σε κατηγορία και τελικά από εφαρμογή σε εφαρμογή.

1.2 Αντικείμενο Διπλωματικής

Η παρούσα εργασία έχει ως τελικό στόχο την κατασκευή μιας πλατφόρμας διαχείρισης όλων των καθημερινών δραστηριοτήτων ενός ανθρώπου, προσφέροντας παράλληλα προστιθέμενη αξία σε αυτόν μέσω κατάλληλων διαγραμματικών αναπαραστάσεων και στατιστικών στοιχείων. Ο χρήστης θα μπορεί να έχει μια πλήρη εποπτεία της καθημερινότητάς του, θα μπορεί να εξετάσει την αλληλεπίδραση των δραστηριοτήτων μεταξύ τους, θα έχει τη δυνατότητα ελέγχου και καλύτερης οργάνωσης του χρόνου του και θα μπορεί να λάβει ενδιαφέρουσες πληροφορίες για το κοινωνικό του πλαίσιο, τις τοποθεσίες δραστηριοποίησής του, το βαθμό επίτευξης των στόχων και εν τέλει την πορεία της ζωής του στην πάροδο του χρόνου. Η εισαγωγή των δραστηριοτήτων στο σύστημα υποβοηθάται από τη δυνατότητα υποστήριξης συγχρονισμού τους μέσω τρίτων υπηρεσιών καταγραφής (π.χ. Google Fit, RunKeeper, Sleepy, Moves, Rescue Time κ.ά.), ενισχύοντας έτσι τον αυτοματισμό της διαδικασίας και μειώνοντας την ανάγκη άμεσης αλληλεπίδρασης του χρήστη με το σύστημα στο κομμάτι της εισαγωγής δεδομένων.

Για την κατασκευή του συστήματος, η πρώτη σχεδιαστική απόφαση αφορά τη μοντελοποίηση των δεδομένων. Τα παραδείγματα εφαρμογών που αναφέρθηκαν υποστηρίζουν ένα περιορισμένο υποσύνολο των ανθρώπινων δραστηριοτήτων οι οποίες ανήκουν σε μια συγκεκριμένη κατηγορία (π.χ. ύπνος, δραστηριότητες άθλησης κ.ά.). Αυτό σημαίνει ότι οι δραστηριότητες αυτές περιλαμβάνουν κοινά χαρακτηριστικά, τα οποία

περιγράφονται μέσω της χρήσης συγκεκριμένου μοντέλου δραστηριότητας, το οποίο είναι σε θέση να αποτυπώσει ικανοποιητικά όλο το πλαίσιο (context) που τις συνοδεύει. Στο Activity Tracker, ωστόσο, απαιτείται η χρήση ενός και μόνο μοντέλου, το οποίο πρέπει να είναι σε θέση να ανταποκρίνεται σε κάθε δυνατή δραστηριότητα, ανεξαρτήτως κατηγορίας και τύπου. Η σχεδιάσή του πρέπει να ελαχιστοποιεί τις απώλειες πληροφορίας για κάθε δραστηριότητα, αλλά ταυτόχρονα, να είναι σε θέση να την περιγράψει ικανοποιητικά μέσω της χρήσης κοινών χαρακτηριστικών που θα αποδίδουν νόημα σε κάθε επιλογή αυτής. Στο επόμενο κεφάλαιο της εργασίας, μετά την ανάγνωση θεωρητικών εργασιών και την μελέτη ποικίλων μοντελοποιήσεων των ανθρώπινων ενεργειών, πραγματοποιείται θέσπιση μιας γενικής μοντελοποίησης ανταποκρινόμενης αποδοτικά στην καθημερινότητα του εκάστοτε χρήστη.

Η επόμενη απόφαση αφορούσε το ίδιο το σύστημα και συγκεκριμένα την επιλογή του τύπου της εφαρμογής, των απαιτήσεων που αυτή θα πρέπει να ικανοποιεί και της αρχιτεκτονικής υλοποίησής της. Η διαδικτυακή εφαρμογή είναι σχεδιασμένη για πρόσβαση μέσω φυλλομετρητών ηλ. υπολογιστή, ωστόσο θα μπορούσε εξίσου εύκολα να είναι εφαρμογή για κινητά. Η τελευταία θα ήταν και η πιο ενδεδειγμένη επιλογή, δεδομένου του ότι οι υποστηριζόμενες υπηρεσίες συγχρονισμού δραστηριοτήτων έχουν υλοποιηθεί με τον ίδιο τρόπο. Ωστόσο, κάτι τέτοιο θα στερούσε τη δυνατότητα πρόσβασης των ατόμων που είτε δεν έχουν καθόλου κινητό τελευταίας γενιάς, είτε έχουν κινητό δίχως το κατάλληλο λογισμικό. Για το λόγο απορρίφθηκε η συγκεκριμένη μορφή υλοποίησης. Οι λειτουργικές και μη-λειτουργικές απαιτήσεις που θα έπρεπε να υλοποιεί η εφαρμογή αποτέλεσαν δύσκολη απόφαση, καθώς έπρεπε να ισορροπούν ανάμεσα στα πλαίσια της διπλωματικής εργασίας (από πλευράς προγραμματιστικών απαιτήσεων) και της παροχής αρκετών επιλογών που θα κάνουν τους χρήστες να την προτιμήσουν. Δυστυχώς, σε μια εφαρμογή τόσο γενικού χαρακτήρα, ποτέ δεν μπορούν να υπάρξουν «αρκετές» υποστηριζόμενες λειτουργίες και επιλογές. Μετά από έρευνα των σχετικών εργασιών που παρουσιάζονται στο κεφάλαιο (3) καθορίστηκαν οι λειτουργίες αυτές που θεωρήθηκαν απαραίτητες για την βιωσιμότητα της εφαρμογής, όπως και αυτές που ενισχύουν την εμπειρία του χρήστη και διευκολύνουν την αλληλεπίδρασή του με το σύστημα. Οι λειτουργίες αυτές καθόρισαν την βέλτιστη αρχιτεκτονική σχεδίαση του συστήματος και των υπομονάδων του, μια επιλογή που απαίτησε την εξέταση όλων των πιθανών αρχιτεκτονικών σχεδίασης και υλοποίησης (π.χ. MVC, Client – Server κ.ά.). Με βάση την αρχιτεκτονική αυτή, επιλέχθηκε η γλώσσα και η πλατφόρμα που θα μπορούσε να διευκολύνει την επεκτασιμότητα του συστήματος και υλοποιήθηκε το Activity Tracker.

1.2.1 Συνεισφορά

Η συνεισφορά της διπλωματικής συνοψίζεται ως εξής:

1. Μελετήθηκαν και παρουσιάστηκαν οντολογίες δραστηριοτήτων και εργασίες μοντελοποίησης ανθρώπινων ενεργειών και συμπεριφοράς.

2. Αναπτύχθηκε μια μοντελοποίηση για τη γενική και αόριστη ανθρώπινη δραστηριότητα, ικανή να περιγράψει το πλαίσιο (context) που τη συνοδεύει, όντας θεσπισμένη με βάση τη θεωρία των δραστηριοτήτων (Activity Theory).
3. Παρουσιάστηκαν οι απαραίτητες απαιτήσεις και λειτουργίες ενός συστήματος διαχείρισης ανθρώπινων δραστηριοτήτων, τονίζοντας την ανάγκη και το στόχο καθεμίας. Επίσης, παρουσιάστηκε αναλυτικά η πλήρης υλοποίησή του, τονίζοντας τις εκάστοτε σχεδιαστικές αποφάσεις, επιλογές εργαλείων, αρχιτεκτονικές και τεχνικές υλοποίησης.
4. Αναπτύχθηκε από την αρχή ένα από τα λιγιστά πληροφοριακά συστήματα διαχείρισης καθημερινότητας (ικανό να υποστηρίζει πληθώρα διαφορετικών δραστηριοτήτων), το οποίο μπορεί να αποτελέσει εύκολα βάση για διάφορες επεκτάσεις, παραδείγματα των οποίων μπορεί να βρει κανείς στο αντίστοιχο κεφάλαιο της παρούσας εργασίας.

1.3 Οργάνωση Κειμένου

Στο κεφάλαιο (2) παρουσιάζονται οι θεωρίες και οι σχετικές εργασίες πάνω στις οποίες θεμελιώθηκε η διπλωματική, όπως και η τελική μοντελοποίηση. Το κεφάλαιο (3) παρουσιάζει παρεμφερείς εργασίες στον τομέα των δραστηριοτήτων, πολλές από τις οποίες επηρέασαν και την παρούσα. Στο κεφάλαιο (4) καθορίζονται οι λειτουργικές και μη-λειτουργικές απαιτήσεις του συστήματος, παρουσιάζονται οι λειτουργίες που αυτό πρέπει να επιτελεί και σχεδιάζεται το διάγραμμα οντοτήτων – συσχετίσεων που προκύπτει. Το κεφάλαιο (5) παρουσιάζει τις προγραμματιστικές γλώσσες και πλαίσια, ενώ ταυτόχρονα αναλύει την αρχιτεκτονική του συστήματος σε επίπεδο κλάσεων και επιλεγμένου προγραμματιστικού πλαισίου (framework). Το κεφάλαιο (6) κάνει αναφορά στα εργαλεία που χρησιμοποιήθηκαν, εστιάζει σε σημαντικά τμήματα προγραμματιστικού κώδικα και παρουσιάζει στιγμιότυπα απευθείας από την τελική έκδοση της εφαρμογής. Το κεφάλαιο (7) αναλύει τα συμπεράσματα που προέκυψαν από την εργασία και προτείνει μελλοντικές επεκτάσεις της. Το κεφάλαιο (8) παρουσιάζει τη βιβλιογραφία που χρησιμοποιήθηκε και τέλος, το κεφάλαιο (9) αφιερώνεται στη διαδικασία εγκατάστασης της εφαρμογής σε προσωπικό υπολογιστή.

2

Θεωρητικό Υπόβαθρο

Η έννοια των ανιχνευτικών συσκευών, ειδικά στους τομείς που σχετίζονται με την υγεία γίνονται όλο και δημοφιλέστεροι, μιας και εισάγουν την έννοια του αυτοματισμού στις δραστηριότητες περίθαλψης των ασθενών. Ανιχνευτές κίνησης, συστήματα που παρακολουθούν το άνοιγμα και το κλείσιμο μιας πόρτας τείνουν να χρησιμοποιούνται για τον αποτελεσματικό έλεγχο ασθενών, υπαλλήλων, βρεφών κ.ά. Ωστόσο, αυξάνοντας σημαντικά τον αριθμό των ανιχνευτικών συσκευών αυτών, δημιουργείται ένας κυκεώνας πληροφοριών από τους αισθητήρες, ο οποίος χρήζει ανάλυσης για την παραγωγή ενός αποτελέσματος. Ταυτόχρονα, δεν είναι λίγες οι περιπτώσεις όπου τα δεδομένα αυτά δεν συμφωνούν μεταξύ τους και συνεπώς δημιουργούνται οι λεγόμενες «συγκρούσεις». Σκοπός της ύπαρξης μιας οντολογίας είναι – ανάμεσα σε άλλα – ο ορισμός κατάλληλων κανόνων, οι οποίοι είναι σε θέση να «αποσαφηνίσουν» όσο το δυνατόν καλύτερα τη μαζική αυτή πληροφορία που τους διατίθεται και να επιστρέψουν μια μορφή αποτελέσματος, που στην προκειμένη περίπτωση είναι η δραστηριότητα που λαμβάνει χώρα.

Ο όρος οντολογία είχε αρχικά οριστεί ως η «προδιαγραφή και εξειδίκευση μιας σύλληψης», ωστόσο στο κομμάτι των πληροφοριακών συστημάτων και της τεχνητής νοημοσύνης η οντολογία είναι ένα σύνολο πληροφοριών όπου διάφοροι όροι καθορίζονται και συσχετίζονται μεταξύ τους. Αυτή ορίζει τις βασικές σχέσεις και όρους που συνθέτουν τόσο το λεγόμενο «λεξιλόγιο», όσο και τους κανόνες για το συνδυασμό τους, με σκοπό τον καθορισμό επεκτάσεων του εκάστοτε λεξιλογίου. Ως αποτέλεσμα δημιουργείται μια ταξινόμια όρων και ένα σύνολο συμπερασματικών κανόνων.

Με πιο απλά λόγια, μια οντολογία περιγράφει μια βάση για την κοινή και μονοσήμαντη κατανόηση των πραγμάτων, εξαλείφοντας την αμφισημία μέσω της χρήσης σαφώς ορισμένων όρων. Φυσικά, στην αντικειμενοστραφή πλευρά των οντολογιών όπου θα εστιάσει η παρούσα διπλωματική εργασία, μια οντολογία ορίζει κλάσεις αντικειμένων και σχέσεις με τις οποίες οι κλάσεις αυτές συνδέονται μεταξύ τους. Ακόμα, κάθε μια κλάση μπορεί να απαριτίζεται από πεδία τα οποία περιγράφουν χαρακτηριστικά της κλάσης αυτής. Συνεπώς αυτή μπορεί εύκολα να περιγραφεί με ένα διάγραμμα κλάσεων υπό την προϋπόθεση ότι κάθε κλάση, σχέση και πεδίο έχει πλήρως οριστεί και τεκμηριωθεί.

Η επιλογή οντολογίας και κατ' επέκταση η δημιουργία μιας μοντελοποίησης ικανής να περιγράψει ένα μεγάλο αριθμό ανθρωπίνων δραστηριοτήτων αποτέλεσε μια πρόκληση δεδομένου του μικρού αριθμού επίσημα θεσπισμένων οντολογιών και του μεγάλου αριθμού διάσπαρτων και μη τεκμηριωμένων ανεπίσημων οντολογιών. Αν και στην παρούσα

διπλωματική δεν κατασκευάστηκε μια νέα οντολογία προσαρμοσμένη στο κύριο άξονά της εργασίας, η μελέτη των ήδη υπαρχουσών κρίθηκε καθοριστική στη διαμόρφωση του τελικού μοντέλου που χρησιμοποιήθηκε. Κατά καιρούς έχουν προταθεί διάφορες οντολογίες περιγραφής μιας δραστηριότητας, ωστόσο από το μεγάλο σύνολο οντολογιών που υπάρχουν, η έρευνα εστιάστηκε μόνο σε αυτές που συμφωνούσαν με τη λεγόμενη «θεωρία των δραστηριοτήτων» (Activity Theory). Βασιζόμενοι στη θεωρία αυτή, οι G. Meditskos et al. (2013) πρότειναν την DOA, μια οντολογία για την περιγραφή των συνθέτων δραστηριοτήτων η οποία ήταν πλήρως ευθυγραμμισμένη με τις γνωστές σχετικές οντολογίες όπως η SEM, ενώ ταυτόχρονα ήταν βασισμένη στην CONON, μια οντολογία των X. Wang et al. (2004) θεμελιωμένη στην OWL.

Η DOA, ενώ μπορούσε να περιγράψει ικανοποιητικά μια δραστηριότητα, αμελούσε να χειριστεί σωστά την πρόσθετη πληροφορία που μπορεί να την συνόδευε. Σε αυτό, λύση έδωσαν οι L. Chen, C. Nugent (2011) με την ανάπτυξη ενός εννοιολογικού – οντολογικού μοντέλου περιγραφής δραστηριότητας, ικανό να καταγράψει πλήρως τις σχέσεις μεταξύ αυτής και των οντοτήτων ή αντικειμένων που τη συνοδεύουν. Το μοντέλο αυτό θα μπορούσε να θεωρηθεί μια επέκταση του οντολογικού μοντέλου της DOA, καθώς είναι σε θέση να περιγράψει πρόσθετη πληροφορία, όντας ταυτόχρονα ικανό να καταγράψει πλήρως αυτή του DOA. Τέλος, με τη μελέτη ενός εμπειρικού μοντέλου συμπεριφοράς από τον L. Cao (2010) αναγνωρίστηκε ο βαθμός στον οποίο το γενικό πλαίσιο που περικλείει η συμπεριφορά ενός ανθρώπου είναι σε θέση να αντιστοιχιστεί πλήρως με την πραγματοποίηση μιας δράσης. Συνεπώς, αυτό επαφίεται εύκολα στη μοντελοποίηση της έννοιας της δραστηριότητάς.

Στο υπόλοιπο κεφάλαιο ακολουθεί μια πιο εκτενής ανάλυση των θεωρητικών αυτών τεχνικών και μοντέλων που προτάθηκαν και τελικά επηρέασαν καθοριστικά το τελικό μοντέλο δραστηριότητας που αναπτύχθηκε.

2.1 Πανταχού Παρούσα Πληροφορική & Διεσδυτικά Συστήματα

Στην σύγχρονη εποχή, οι οντολογίες είναι άρρηκτα συνδεδεμένες με την έννοια των διεσδυτικών υπολογιστικών περιβαλλόντων (pervasive environments) και τις πανταχού παρούσες πληροφορικής (ubiquitous computing). Οι όροι αυτοί αναφέρονται σε περιβάλλοντα, στα οποία η πληροφορική και γενικά τα συστήματα πληροφορίας υπάρχουν παντού, ανεξαρτήτως συσκευής. Παραδείγματα είναι το διαδίκτυο, το λειτουργικό σύστημα, οι αισθητήρες, οι μικροεπεξεργαστές, οι διαπροσωπείες, τα πρωτόκολλα, η δορυφορική παρακολούθηση κ.ά. Οι όροι αυτοί θα μπορούσαν συνεπώς να αναφέρονται σε υποβόσκουσες τεχνολογίες, οι οποίες δρουν και λειτουργούν, δίχως ο χρήστης να αλληλεπιδρά άμεσα με αυτές. Στην εποχή μας, η έννοια των «έξυπνων σπιτιών» (smart houses) είναι ένα αρκετά ικανοποιητικό παράδειγμα τέτοιου περιβάλλοντος και τεχνολογίας, όπου αισθητήρες μπορούν να παρατηρούν τις κινήσεις και τις ενέργειες ενός ανθρώπου και να εξάγουν συμπεράσματα για το τι κάνει. Για παράδειγμα, το άνοιγμα και το κλείσιμο της πόρτας του ψυγείου, σε συνδυασμό με το άνοιγμα και το κλείσιμο ενός συρταριού της κουζίνας θα μπορούσε να σημαίνει ότι το υποκείμενο - άτομο πεινάει ή/και

τρώει ή/και μαγειρεύει. Οι οντολογίες είναι αυτές που θα επιλύσουν, κατά το δυνατό, την ασάφεια που εμφανίζεται παραπάνω.

2.2 Θεωρία Δραστηριοτήτων (Activity Theory)

Η θεωρία των δραστηριοτήτων ήταν αποτέλεσμα Ρώσων ψυχολόγων να αναπτύξουν μια θεωρία για την ψυχολογία βασισμένη στη Μαρξιστική φιλοσοφία και τρόπο σκέψης. Υποστηρίζει ότι το πλαίσιο στο οποίο εκτελείται μια δραστηριότητα πρέπει να παίξει εξέχοντα ρόλο στην ανάλυση των ανθρώπινων ενεργειών, το αντικείμενο των οποίων πρέπει να διαχωριστεί από το αποτέλεσμα. Τα κυρίαρχα σημεία στη θεωρία αυτή είναι το υποκείμενο, το αντικείμενο και η κοινωνία, ενώ τα εργαλεία, οι κανόνες και ο καταμερισμός της εργασίας αποτελούν μια μορφή επιπλέον «τεχνουργημάτων» για τον καθορισμό ενός γενικού πλαισίου (context). Η θεωρία υποστηρίζει ότι ένα υποκείμενο παράγει ένα αποτέλεσμα τροποποιώντας ένα αντικείμενο, μέσω της χρήσης ενός εργαλείου, στα πλαίσια μια κοινωνίας με τον καταμερισμό εργασίας και τους κανόνες που την διέπουν. Συνεπώς, έμφαση δίνεται στις αλληλεπιδράσεις μεταξύ των ανθρώπων – οι οποίοι, λόγω της ύπαρξης της συνείδησης, είναι τα υποκείμενα – και του περιβάλλοντός τους. Είναι σημαντικό να αναφερθεί ότι η θεωρία των δραστηριοτήτων δεν προσφέρει μια μεθοδολογία, αλλά μια προσέγγιση κλάσης επιπέδων για την επίτευξη της ανάλυσης.

Κατά συνέπεια, η θεωρία ορίζει ένα πλαίσιο στο οποίο οι άνθρωποι εκλαμβάνονται ως μια κοινωνία και όχι ως ατομικές οντότητες, και εκτελούν δραστηριότητες οι οποίες υπόκεινται ιεραρχικής ανάλυσης. Οι δραστηριότητες υποκινούνται από ένα αντικείμενο (στόχο), περιλαμβάνουν κοινωνικές και τεχνικές μεσολαβήσεις και αναλύονται σε στοχευμένες ενέργειες. Είναι πολύ σημαντικό να τονιστεί ότι οι δραστηριότητες διασπώνται σε 3 επίπεδα:

1. Δραστηριότητα για την επίτευξη ενός τελικού, συχνά αόριστου σκοπού, διεξαγόμενη από την κοινωνία.
2. Δραστηριότητα για την επίτευξη ενός συγκεκριμένου στόχου, διεξαγόμενη από ένα άτομο – υποκείμενο.
3. Λειτουργική ενέργεια μιας δραστηριότητας, η οποία συχνά εκτελείται αυτόματα και μηχανικά.

Το διάγραμμα του Engestrom, αν και δεν δείχνει τα επίπεδα που αναφέρθηκαν, προσφέρει μια αρκετά ικανοποιητική οπτικοποίηση του συστήματος που ορίζει η θεωρία.

Activity System (Engestrom)



Σχήμα 2.1 Activity Theory - Διάγραμμα Engestrom (1987)

Το παραπάνω σχήμα δείχνει πώς μια πληθώρα από παράγοντες συνεργάζονται για να επηρεάσουν μια δραστηριότητα. Για να προβεί ένα άτομο (subject) σε ένα αποτέλεσμα (outcome) είναι αναγκαίο να παράγει αντικείμενα (objects) όπως η γνώση, η εμπειρία κ.ά., δουλεύοντας ως μέρος μιας κοινωνίας. Ακόμα, η ανθρώπινη δραστηριότητα λαμβάνει χώρα σε έναν οργανισμό ή μια κοινωνία που διέπεται από κανόνες και απαιτεί τη χρήση διαμεσολαβητών (artefacts) όπως εργαλεία, έγγραφα κ.ά.

Η θεωρία αυτή έχει άμεση σχέση με την παρούσα διπλωματική, καθώς ορίζει έναν κατευθυντήριο άξονα για τη θέσπιση οντολογιών και μοντέλων περιγραφής της δραστηριότητας, χωρίζοντας την τελευταία σε επίπεδα και ορίζοντας τα χαρακτηριστικά τα οποία την απαρτίζουν. Αυτό γίνεται εμφανές στις παρακάτω υποενότητες, όπου και αναλύονται εργασίες (papers), των οποίων τα μοντέλα έπαιξαν καθοριστικό ρόλο.

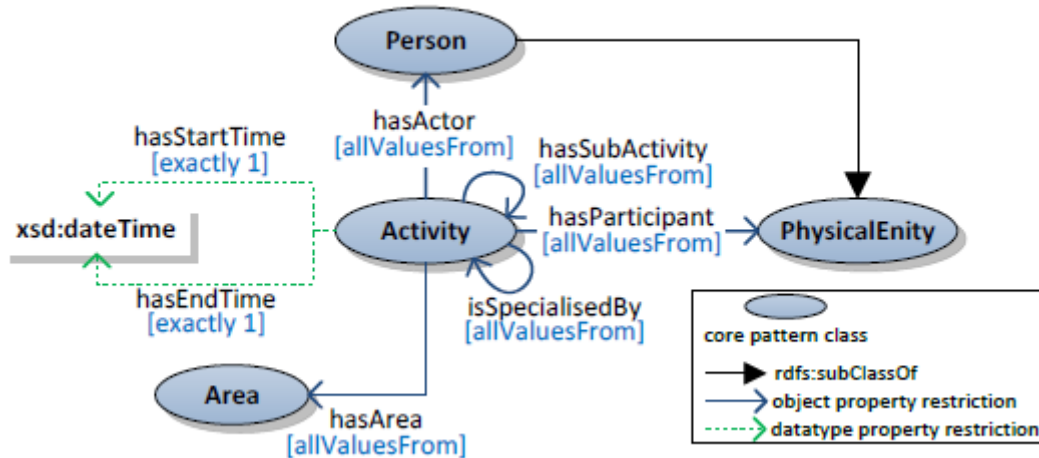
2.3 Μοτίβα Οντολογιών για Μοντελοποίηση Περίπλοκων Δραστηριοτήτων (Ontology Patterns for Complex Activity Modelling)

Στο υποκεφάλαιο αυτό θα γίνει ανάλυση της DOA (Domain Activity Ontology), μιας οντολογίας προτύπων δραστηριοτήτων που προτάθηκε από τους G. Meditskos et al. (2013). Αυτή έχει ως στόχο την παρουσίαση των σχέσεων που οδηγούν στην παραγωγή σύνθετων δραστηριοτήτων, τόσο σε ότι αφορά την κατηγορία τους, όσο και σε ότι αφορά τις χρονικές σχέσεις που θα πρέπει να ικανοποιούνται. Αποτέλεσμα είναι η επίσημη αποτύπωση των μοντέλων ερμηνεύσης δραστηριοτήτων μέσω κλάσεων χαρακτηριζόμενων από χρονικές συσχετίσεις, των οποίων η συνάθροιση μπορεί να υπονοήσει την ύπαρξη νέων σύνθετων δραστηριοτήτων.

Οι περισσότερες οντολογίες ως τώρα εστίαζαν στην παραγωγή βέλτιστων κανόνων για την δημιουργία μιας ορθής ανάλυσης εξαγωγής συμπερασμάτων, αντιμετωπίζοντας τα δεδομένα ως ένα αταξινόμητο σύνολο, αγνοώντας τη μετρική του χρόνου. Η εργασία αυτή

εστιάζει στην έννοια της χρονικής προτεραιότητας των απλών ενεργειών, καθώς και της μεταπληροφορίας που μπορεί να κρύβει αυτή. Ταυτόχρονα, εξετάζει τους τρόπους με τους οποίους η χρονική σειρά αυτή δομεί - και κατ' επέκταση αποδομεί - μια σειρά ενεργειών σε μια και μόνο σύνθετη ενέργεια, μια δραστηριότητα.

Προτείνει συνεπώς την ακόλουθη - κοινή για όλες τις δραστηριότητες - οντολογία :



Σχήμα 2.2 Μοντελοποίηση Δραστηριότητας του DOA

Στο παραπάνω σχήμα, είναι εμφανής η επίδραση του Activity Theory στη σχεδίαση της DOA. Μια δραστηριότητα έχει ένα υποκείμενο (Person), χαρακτηρίζεται από μια τοποθεσία (Area), πραγματοποιείται σε ένα χρονικό διάστημα περιγραφόμενο μέσω αρχής (hasStartTime) και τέλους (hasEndTime), και έχει συμμετέχοντες (Physical Entity) που αντιστοιχίζονται στην έννοια των εργαλείων του Α.Τ. Ωστόσο, η έννοια των συμμετεχόντων, όντας γενικά ορισμένη στην DOA, μπορεί να περιλαμβάνει και άλλες οντότητες Person, πράγμα που συνάδει με το βασικό πυρήνα της θεωρίας των δραστηριοτήτων, δηλαδή ότι το υποκείμενο παράγει ένα αποτέλεσμα, δρώντας στα πλαίσια μιας κοινωνίας. Ενδιαφέρον παρουσιάζει η εισαγωγή υπο- και υπερ-δραστηριότητας στην οντολογία, η οποία οδηγεί αυτόματα στην δημιουργία μιας πολυεπίπεδης ανάλυσης. Απομονώνοντας τις ενέργειες που συνθέτουν μια δραστηριότητα μπορεί και δημιουργείται μεταπληροφορία χρήσιμη τόσο για την απόφαση του συστήματος όσο και την δυναμική εκμάθηση του εκάστοτε χρήστη. Ο διαχωρισμός αυτός αντιστοιχίζεται πλήρως στα επίπεδα δραστηριοτήτων του Α.Τ. και συμβάλει στην κατασκευή ενός γράφου εξαρτήσεων, καθώς δημιουργείται άμεσα μια M-N σχέση αλληλεξάρτησης δραστηριοτήτων. Ωστόσο, παρότι η εργασία αυτή θέτει τα βασικά θεμέλια στη μοντελοποίηση που επιλέχτηκε στη διπλωματική, το μοντέλο του συμμετέχοντος δεν διαφοροποιείται κατάλληλα ανάλογα με την ανθρώπινη ή μη υπόσταση του. Επιπλέον, η οντολογία αυτή αποτυγχάνει να αποτυπώσει το στόχο και του αποτελέσματος που συνοδεύουν μια ανθρώπινη δραστηριότητα.

2.4 Οντολογική Αναγνώριση Δραστηριοτήτων σε Έξυπνα Διεσδυτικά Περιβάλλοντα (Ontology Based Activity Recognition in Intelligent Pervasive Environments)

Στόχος των L. Chen και C. Nugent είναι η ενημέρωση για τις ήδη υπάρχουσες τεχνικές αναγνώρισης δραστηριοτήτων όσο και η παρουσίαση μιας νέας προσέγγισης στη μοντελοποίηση οντολογιών με επίκεντρο τις δραστηριότητες, την αναπαράσταση και την εφαρμογή λογικής για την εξαγωγή συμπερασμάτων.

Παρότι η εργασία αυτή αναλύει διεξοδικά τους τρόπους και τις μεθόδους αναγνώρισης δραστηριοτήτων, ενδιαφέρον παρουσιάζει το οντολογικό μοντέλο το οποίο προτείνει για τη βέλτιστη ανίχνευση τους και την παραγωγή μεταπληροφορίας από τα απλά δεδομένα που παρέχονται. Όπως και το DOA, τονίζει την ανάγκη για ύπαρξη ενός διαχωρισμού επιπέδων στις δραστηριότητες, ο οποίος αποτελείται από τύπους δραστηριοτήτων και τις σχέσεις τους στο χώρο του προβλήματος. Συνεπώς, δημιουργείται μια ιεραρχική δομή μέσω ρητών εννοιών που συσχετίζουν μια δραστηριότητα με μια άλλη, γεγονός που έχει ως αποτέλεσμα την έμμεση κατασκευή μοντέλων κοινής γνώσης που αυξάνουν τις δυνατότητες της αυτόματης επεξεργασίας. Επιπροσθέτως, αυτό προσφέρει άμεση διαλειτουργικότητα μέσω της κοινής δομής και ορολογίας ανάμεσα στις δραστηριότητες. Για παράδειγμα η δραστηριότητα του καλλωπισμού μπορεί να αποτελείται από τις υπο-δραστηριότητες του ξυρίσματος και του πλυσίματος του προσώπου, ωστόσο η «σύνθετη» αυτή δραστηριότητα του καλλωπισμού, μοντελοποιείται και περιγράφεται με τον ίδιο ακριβώς τρόπο, όπως και οι υπο-δραστηριότητές της. Η καθοριστική διαφορά σε σχέση με το μοντέλο της DOA είναι η ικανότητα αποτύπωσης ενός ευρέως φάσματος πληροφορίας που το προηγούμενο αμελούσε. Αυτό είναι άμεσα εμφανές στο παρακάτω σχήμα:



Σχήμα 2.3 Μοντελοποίηση Δραστηριότητας κατά Chen - Nugent (2011)

Έχοντας ήδη παρουσιάσει τη DOA, το ενδιαφέρον εδώ μονοπωλεί η μορφή των πεδίων που απαρτίζουν μια δραστηριότητα. Εύκολα θα χαρακτηρίζε κανείς το παραπάνω οντολογικό μοντέλο ως μια οπτικοποίηση του Activity Theory, μιας και είναι σε θέση να αποτυπώσει κάθε πτυχή της εκτέλεσης μιας ενέργειας - δραστηριότητας. Περιγράφει, δηλαδή, ένα υποκείμενο (Actor) που μέσω μιας ενέργειας (Activity) παράγει ένα αποτέλεσμα (Effects) σε ένα συγκεκριμένο πλαίσιο τόσο χρονικό (Time – Duration) όσο και χωρικό (Location). Για την επίτευξη του ορίζει στόχους (Goal) και δρα με τους συγκεκριμένους κανόνες (Conditions) της κοινωνίας (Environment Entities), χρησιμοποιώντας κατάλληλα εργαλεία και μέσα (Resources). Επιπροσθέτως, υπάρχει μια πλήρης αντιστοιχία πεδίων ανάμεσα στο μοντέλο αυτό και στο αντίστοιχο του DOA, ιδιαίτερα στον τομέα της ιεράρχησης των ενεργειών μιας και ορίζει και αυτό τις έννοιες της υπο- και υπερ-δραστηριότητας. Συνεπώς θα μπορούσε να το εκλάβει κανείς ως μια μορφή επέκτασής του μοντέλου που εξετάστηκε στην προηγούμενη ενότητα. Αξίζει να σημειωθεί ότι το παραπάνω μοντέλο εισάγει την έννοια της θέσπισης ενός στόχου (goal) ο οποίος είναι άμεσα και άρρηκτα συνδεδεμένος με μια δραστηριότητα, γεγονός που παρουσιάζει ιδιαίτερο ενδιαφέρον, καθώς στις περισσότερες υλοποιήσεις ο στόχος είτε αποτελούσε ανεξάρτητο μέρος της οντολογίας είτε συνδεόταν με το δράστη, χρησιμοποιώντας τη δραστηριότητα καθαρά και μόνο ως μέσο επίτευξης του.

2.5 Εις Βάθος Κατανόηση και Χρήση Συμπεριφοράς: Η Υπολογιστική Προσέγγιση Συμπεριφοράς (In-depth behavior understanding and use: The behavior informatics approach)

Σε αντίθεση με τα δύο προηγούμενα, η εργασία αυτή εστιάζει στη μοντελοποίηση της ανθρώπινης συμπεριφοράς. Οι κλασσικές μοντελοποιήσεις βασίζονται σε ποσοστιαίες μεθόδους κοινωνικών επιστημών, ενώ παράλληλα η λεγόμενη «ανάλυση συμπεριφοράς» στηρίζεται σε δημογραφικές μετρήσεις και στοιχεία επιχειρήσεων, όπου η πληροφορία είναι κρυμμένη στα τακτικά συλλεγόμενα δεδομένα. Στόχος είναι η πρόταση ενός νέου είδους προσέγγισης για τη δυνατότητα εκμετάλλευσης τόσο των απλών και ξεκάθαρων δεδομένων που παρέχονται, όσο και αυτών που υπονοούνται μέσω της εφαρμογής μιας αλυσίδας σταδίων με ανάδραση. Τα στάδια αυτά αφορούν την κατασκευή δεδομένων συμπεριφοράς, τη μοντελοποίηση και αναπαράσταση της συμπεριφοράς, την ανάλυση των προτύπων και του «δικτύου» συμπεριφοράς που παράγεται και τέλος τη μέτρηση και αξιολόγηση της. Παρότι η εργασία ορίζει και αναλύει καθένα εκ των άνωθεν σταδίων, το κομμάτι που αφορά την παρούσα διπλωματική αναφέρεται στο μοντέλο συμπεριφοράς που κατασκευάστηκε και όχι στη χρήση του για την ανάλυση προτύπων και την παραγωγή κανόνων. Στόχος συνεπώς είναι η μοντελοποίηση και η στατιστική ανάλυση των δραστηριοτήτων, αδιαφορώντας για τα αίτια τα οποία τις προκαλούν.

Ο L. Cao προτείνει ότι η εκάστοτε συμπεριφορά ενός ανθρώπου μπορεί να περιγραφεί ικανοποιητικά μέσω χαρακτηριστικών και ιδιοτήτων όπως π.χ. η πράξη και ο χρόνος στον οποίο πραγματοποιήθηκε. Ωστόσο, για την αναπαράσταση της συμπεριφοράς, τα πεδία του μοντέλου θα πρέπει να είναι σε θέση να περιγράψουν πράξεις και ενέργειες που ένας άνθρωπος πραγματοποίησε, μιας και η συμπεριφορά κατά την E. Minton ορίζεται ως μια συλλογή ενεργειών και ιδιομορφιών ενός ατόμου. Συνεπώς η μοντελοποίηση της

συμπεριφοράς ανάγεται στην ουσία σε μια μοντελοποίηση των ενεργειών ενός ανθρώπου και τελικά συνδέεται στενά με το αντικείμενο μας. Τα πεδία που προτάθηκαν από την εργασία για το εμπειρικό αυτό μοντέλο συμπεριφοράς ήταν τα ακόλουθα:

- *Υποκείμενο*: Η οντότητα που εκτελεί την δραστηριότητα ή σειρά δραστηριοτήτων.
- *Αντικείμενο*: Η οντότητα πάνω στην οποία ασκείται η συμπεριφορά.
- *Πλαίσιο*: Το περιβάλλον στο οποίο η συμπεριφορά επενεργεί.
- *Στόχος*: Ο στόχος που το υποκείμενο θέλει να πετύχει.
- *Πεποίθηση*: Αναφέρεται στη γνώση και την πληροφόρηση του υποκειμένου για τον κόσμο.
- *Ενέργεια*: Η ενέργεια που το υποκείμενο αποφάσισε να εκτελέσει.
- *Επίδραση*: Τα αποτελέσματα της εκτέλεσης μιας συμπεριφοράς στο αντικείμενο.
- *Περιορισμοί*: Οι καταστάσεις που επιδρούν πάνω στη συμπεριφορά.
- *Χρόνος*: Το πότε λαμβάνει χώρα η συμπεριφορά.
- *Τοποθεσία*: Το που λαμβάνει χώρα η συμπεριφορά.
- *Κατάσταση*: Η κατάσταση στην οποία βρίσκεται η δραστηριότητα σε σχέση με την ολοκλήρωσή της. Για παράδειγμα τετελεσμένη, σε εξέλιξη κ.τ.λ.
- *Συσχέτιση*: Άλλες συμπεριφορές που συνδέονται με την παρούσα συμπεριφορά προερχόμενες από ένα δίκτυο συμπεριφορών.

Το παραπάνω μοντέλο μπορεί εμφανώς να παρομοιασθεί με τις οντολογίες και μοντελοποιήσεις των δύο προηγούμενων εργασιών. Τα πεδία υποκείμενο, στόχος, χρόνος και τοποθεσία αντιστοιχίζονται 1-1 με αυτά των προηγούμενων, με την κύρια διαφορά ότι η συμπεριφορά εδώ αντιμετωπίζεται ως μια στιγμιαία πράξη. Ακόμα, το πεδίο της ενέργειας στην περίπτωση μας και ταυτίζεται με τη έννοια της δραστηριότητας και η τιμή του μεταβάλλεται ανάλογα με την εκάστοτε δραστηριότητα που πραγματοποιείται.

Τα υπόλοιπα πεδία θα μπορούσαν εξίσου εύκολα να αντιστοιχιστούν στο υπό κατασκευή μοντέλο της παρούσας διπλωματικής, καθώς το πεδίο της κατάστασης αναφέρεται στο βαθμό περάτωσης μιας δραστηριότητας, το πεδίο της συσχέτισης παρέχει συσχέτιση μιας δραστηριότητας με μια άλλη (κάτι το οποίο στις προηγούμενες δραστηριότητες γινόταν μέσω της υπερ- και υπο-δραστηριότητας), το πεδίο των περιορισμών αναφέρεται στις συνθήκες κάτω από τις οποίες λαμβάνει χώρα μια δραστηριότητα κ.τ.λ.

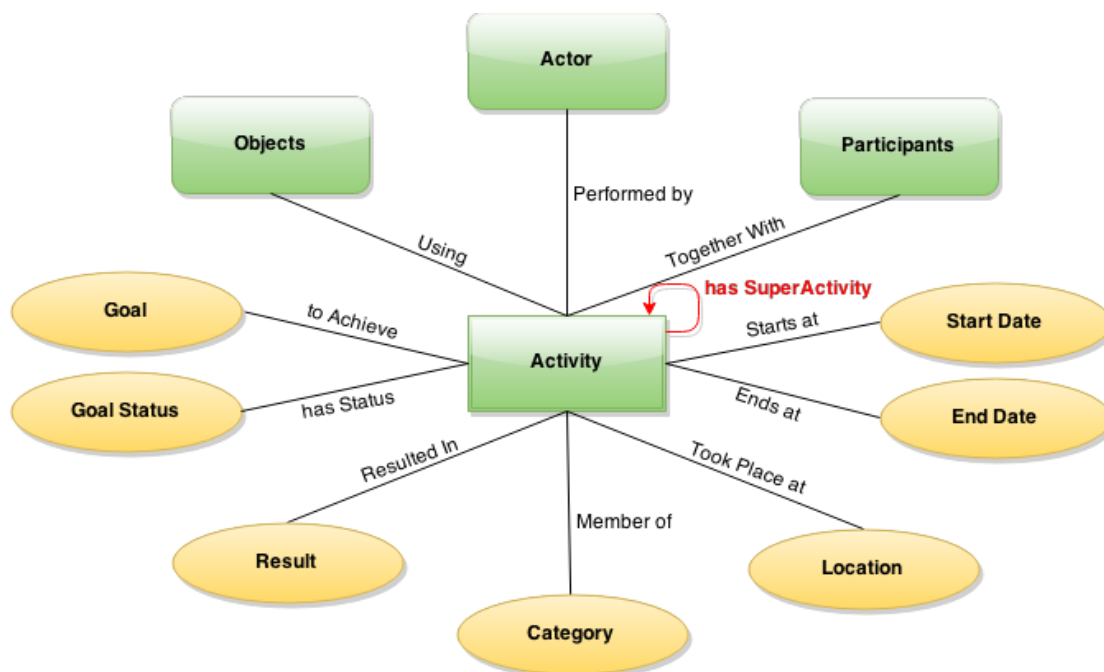
2.6 Μοντελοποίηση Εφαρμογής «Activity Tracker»

Οι σχετικές αυτές εργασίες άσκησαν καταλυτική επίδραση στην κατασκευή του μοντέλου. Αν και οι τρεις εργασίες που μελετήθηκαν προτείνουν παρεμφερή οντολογικά μοντέλα, ήταν πολύ χρήσιμες γιατί εστίαζαν σε διαφορετικά σημεία που χαρακτηρίζουν μια δραστηριότητα. Με την εργασία αυτή και τη συμβολή των υπολοίπων, είναι πλέον εύκολο να προβούμε στη θέσπιση των βασικών χαρακτηριστικών που θα πρέπει να απαρτίζουν το μοντέλο της αόριστης δραστηριότητας που θέλουμε να ορίσουμε. Η επιλογή μοντελοποίησης και όχι χρήσης ήδη υπαρχουσών οντολογιών, έγκειται στο στόχο της παρούσας διπλωματικής. Το αντικείμενό της είναι η κατασκευή μιας πρότυπης πλατφόρμας όπου η χρήση βάσης δεδομένων θεωρείται η βέλτιστη επιλογή για την επιτέλεση των

λειτουργιών που απαιτούνται. Ακόμα, ο συνδυασμός διαφορετικών οντολογιών (μιας και δεν υπάρχει γνωστή οντολογία που να περιγράφει μόνη της βέλτιστα τη γενική και αόριστη δραστηριότητα) θεωρείται κακή πρακτική, μιας και μπορεί να οδηγήσει σε ποικίλες μορφές ασάφειας και συγκρούσεων. Έτσι αποφασίστηκε ότι η καλύτερη μέθοδος περιλαμβάνει την κατασκευή ενός νέου μοντέλου περιγραφής της ζητούμενης αόριστης δραστηριότητας. Συγκεκριμένα, ορίσαμε τα ακόλουθα χαρακτηριστικά που θα απαρτίζουν το παραπάνω μοντέλο:

- **Actor:** Η ανθρώπινη οντότητα που πραγματοποιεί μια δραστηριότητα.
- **Activity:** Η δραστηριότητα που πραγματοποιείται.
- **Objects:** Τα αντικείμενα που μετέχουν κατά τη διεξαγωγή της δραστηριότητας
- **Participants:** Άλλοι άνθρωποι που συμμετέχουν στη δραστηριότητα.
- **Location:** Η τοποθεσία στην οποία η δραστηριότητα λαμβάνει χώρα.
- **Start Date:** Ο χρόνος εκκίνησης της δραστηριότητας.
- **End Date:** Ο χρόνος περάτωσης της δραστηριότητας.
- **SuperActivity:** Η δραστηριότητα ανώτερου επιπέδου, για την οποία η παρούσα δραστηριότητα αποτελεί υπο-δραστηριότητα. Προφανώς το πεδίο αυτό μπορεί να λάβει παραπάνω από 1 τιμές.
- **Category:** Η ομάδα/κατηγορία δραστηριοτήτων στην οποία ανήκει η δραστηριότητα.
- **Goal:** Ο θεσπισμένος από τον δράστη στόχος που συνοδεύει τη δραστηριότητα, ανεξάρτητα από το αν η επίτευξη του απαιτεί και άλλες δραστηριότητες.
- **Goal Status:** Η έκβαση του στόχου αυτού. Μπορεί να πάρει τρεις (3) δυνατές τιμές: Succeeded, In Progress, Failed που αντιστοιχούν στην επίτευξη, εν εξελίξει και αποτυχία του στόχου.
- **Result:** Το αποτέλεσμα που επήλθε με την περάτωση της δραστηριότητας. Αυτό είναι ανεξάρτητο του στόχου που θεσπίστηκε προηγουμένως.

Αυτά είναι εμφανή στο παρακάτω σχήμα, το οποίο προσφέρει μια απλουστευμένη οπτικοποίηση της παραπάνω μοντελοποίησης:



Σχήμα 2.4 Μοντελοποίηση Δραστηριοτήτων Activity Tracker

Στο σχήμα, τα μοντέλα (models) αποδίδονται με πράσινο χρώμα ενώ τα χαρακτηριστικά (attributes) με πορτοκαλί. Παρατηρούμε συνεπώς, τι συνδέεται άμεσα με τη δραστηριότητα και τι μπορεί να υπάρχει ανεξάρτητα από αυτή. Ο τρόπος που ορίσαμε τον στόχο, σηματοδοτεί ότι η έκβαση ενός στόχου δεν συνδέεται απαραίτητα με το αποτέλεσμα της δραστηριότητας. Για παράδειγμα, σε μια δραστηριότητα άθλησης ο στόχος ενδέχεται να είναι «Απώλεια 10 κιλών» ενώ το αποτέλεσμα της παρούσας δραστηριότητας «Καύση 500 θερμίδων». Ακόμα, η ίδια έννοια του στόχου δεν συνδέεται αποκλειστικά με την εκτέλεση της παρούσας δραστηριότητας, αλλά μπορεί να συνθέτει έναν τελικό σκοπό του οποίου η περάτωση απαιτεί την εκτέλεση πληθώρας δραστηριοτήτων. Παράδειγμα αυτού αποτελεί ο γενικός στόχος της απώλειας βάρους που αναφέρθηκε παραπάνω.

3

Σχετικές Εργασίες

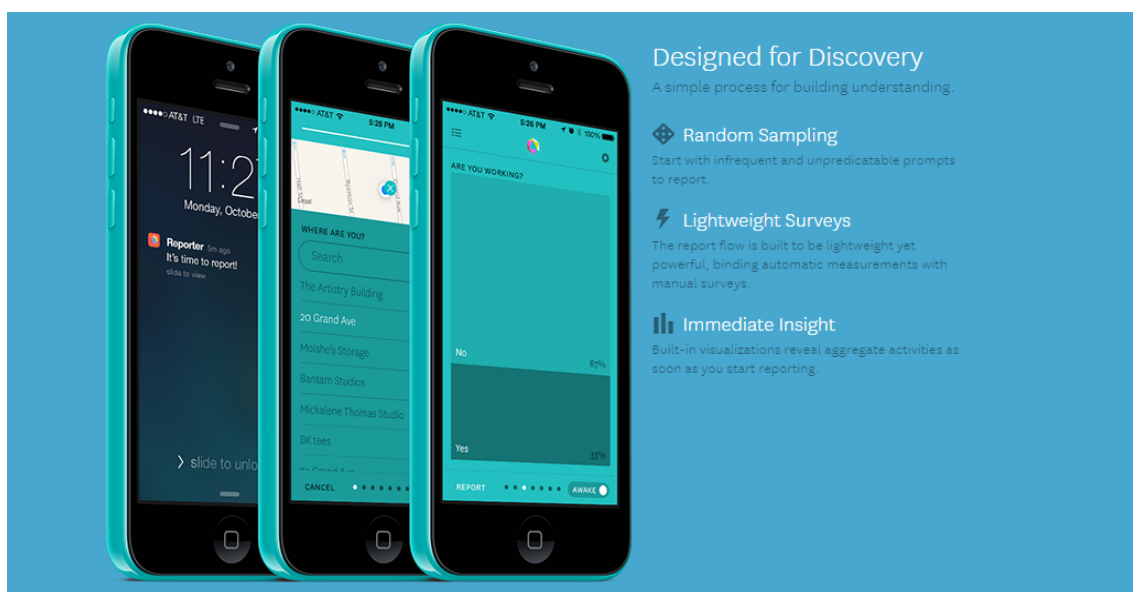
Τα τελευταία χρόνια έχουν αναπτυχθεί εκατοντάδες εφαρμογές για την καταγραφή και παρακολούθηση των καθημερινών ανθρωπίνων δραστηριοτήτων. Κάθε μια από αυτές υποστηρίζει μια σχετικά μικρή αλλά ικανοποιητική ομάδα δραστηριοτήτων χαρακτηριζόμενων από κοινό πλαίσιο, όπως για παράδειγμα ύπνος, άθληση, υγεία κ.ά. Παρόλα αυτά, μόνο ελάχιστες έχουν τη δυνατότητα υποστήριξης ενός μεγάλου συνόλου δραστηριοτήτων από διαφορετικές κατηγορίες, καθώς αυτό δημιουργεί πολλά προβλήματα τόσο στην υλοποίηση όσο και τη σχεδίαση του συστήματος. Στο κεφάλαιο αυτό, θα παρουσιαστούν μερικές εφαρμογές που – όπως και αυτή της παρούσας διπλωματικής – έχουν ως στόχο την παρακολούθηση και καταγραφή (tracking) ενός μεγάλου αριθμού ασυσχέτιστων δραστηριοτήτων. Οι εργασίες αυτές επηρέασαν τη σχεδίαση της εφαρμογής, όπως και τον τρόπο αλληλεπίδρασής της με τους τελικούς χρήστες. Τέλος, θα παρουσιαστούν συνοπτικά παραδείγματα δημοφιλών εφαρμογών υποστήριξης εξειδικευμένων συνόλων δραστηριοτήτων.

3.1 Εφαρμογή «Reporter»¹

Η εφαρμογή αυτή είναι στον πυρήνα της ένα ερευνητικό εργαλείο. Σε τακτά, αλλά εξατομικευμένα χρονικά διαστήματα, χρησιμοποιεί το σύστημα των ειδοποιήσεων του κινητού για να ρωτήσει το χρήστη τι ακριβώς κάνει την παρούσα στιγμή. Ταυτοχρόνως, μέσω του μικροφώνου και του GPS καταγράφει τόσο το επίπεδο θορύβου που υπάρχει όσο και την παρούσα τοποθεσία του χρήστη. Οι ερωτήσεις – οι οποίες μπορούν να επεκταθούν και από τον ίδιο το χρήστη – είναι έξυπνα διαμορφωμένες έτσι ώστε αυτός να μην ασχοληθεί πάνω από 10 δευτερόλεπτα για την συνολική απάντησή τους. Ερωτήματα όπως το αν δουλεύει αυτή τη στιγμή, αν είναι χαρούμενος ή κουρασμένος κ.λπ. βοηθούν το σύστημα που κρύβεται από πίσω να χτίσει συσχετίσεις μεταξύ των δραστηριοτήτων και στο τέλος – πέρα από τα κλασικά στατιστικά αποτελέσματα – να παρουσιάσει αλληλεπιδράσεις που ο χρήστης δεν είχε ποτέ παρατηρήσει.

¹ www.reporter-app.com

Κυρίαρχη διαφορά με την εφαρμογή της διπλωματικής, πέρα το λειτουργικό σύστημα στο οποίο αναπτύχθηκε (iOS), είναι ο τρόπος με τον οποίο εισάγονται τα στοιχεία. Το Activity Tracker έχει αναπτυχθεί ως ένα ημερολόγιο, το οποίο μπορεί να συγχρονίζει δεδομένα από άλλες εφαρμογές, ενώ το Reporter έχει αναπτυχθεί κατά κόρον ως ένα τακτικό ερωτηματολόγιο, ζητώντας ρητά από το χρήστη να εισάγει δεδομένα. Αρκετό ενδιαφέρον έχει και το γεγονός ότι τα δεδομένα του Reporter - σε αντίθεση με το Activity Tracker που είναι διαδικτυακή εφαρμογή - αποθηκεύονται στην κινητή συσκευή και δεν αποστέλλονται σε Server, κάτι το οποίο παρότι προσδίδει ασφάλεια δεδομένων, δεν βοηθάει την ίδια την εφαρμογή να εξελιχθεί μέσα από τους χρήστες της. Τέλος, η εξατομίκευση των ερωτήσεων που πραγματοποιεί η εφαρμογή, δίνει τη δυνατότητα προσαρμογής στις εκάστοτε δραστηριότητες που αφορούν το χρήστη, κάτι το οποίο στη διπλωματική κρίθηκε δύσκολο δεδομένης της ανάγκης για ύπαρξη ρητής ιεραρχίας ανάμεσα στις – προκαθορισμένες – δραστηριότητες.



Εικόνα 3.1 Εφαρμογή «Reporter»

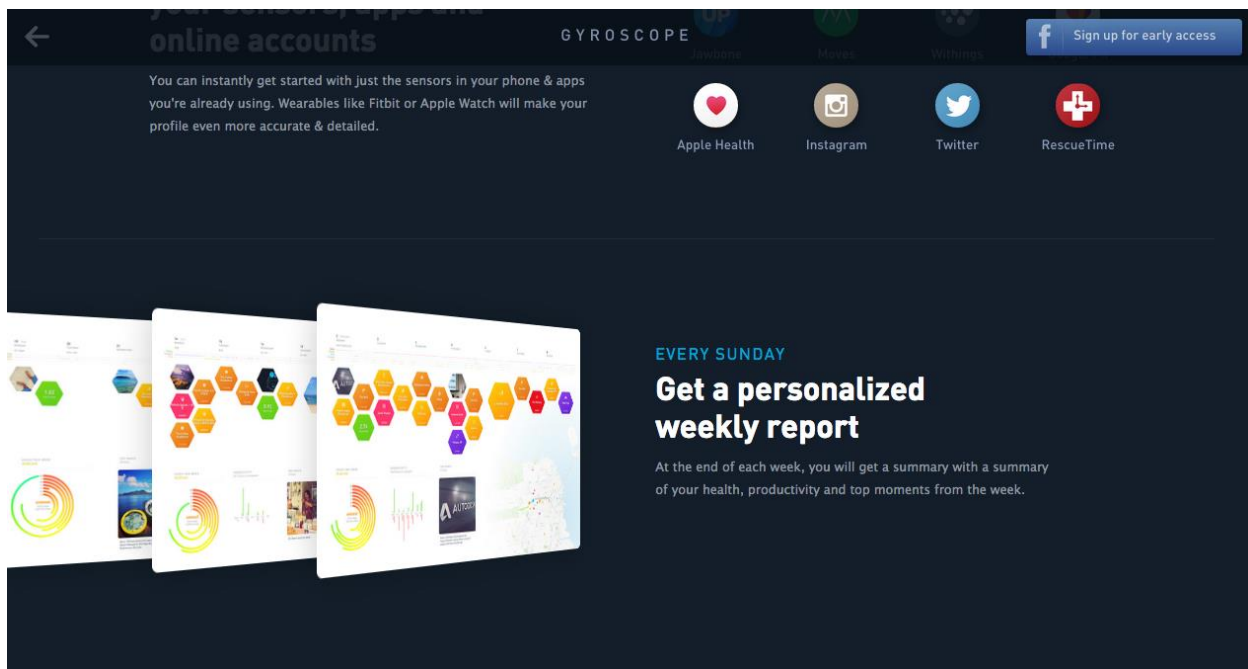
3.2 Εφαρμογή «Gyroscope»²

Το Gyroscope είναι μια νέα εφαρμογή για κινητές συσκευές smartphone που κατά την περίοδο συγγραφής της παρούσας διπλωματικής βρίσκεται σε beta στάδιο ελέγχου. Προσφέρει μια πιο ολοκληρωμένη μορφή παρακολούθησης (tracking) κάθε πτυχής της ζωής του καθενός, εστιάζοντας διαφορετικά σε κάθε είδος δραστηριότητας. Είναι μια όλα-σε-ένα εφαρμογή που συνδυάζει τα προτερήματα των εφαρμογών κοινωνικής δικτύωσης (μέσω της χρήσης φωτογραφιών, check-in κ.λπ.) με αυτά των εφαρμογών παρακολούθησης εξατομικευμένων δραστηριοτήτων, αποθηκεύοντας δεδομένα όπως οι τοποθεσίες, η παραγωγικότητα, η υγεία και η άθληση. Στο τέλος κάθε εβδομάδας, η εφαρμογή

² www.gyrosco.pe

παρουσιάζει αναλυτικά στατιστικά και διαγράμματα, όπως επίσης και κυρίαρχες στιγμές τις εβδομάδας. Παρά τις διάφορες παραμετροποιήσεις που επιδέχεται σε όλους τους βαθμούς της, το μεγάλο της πλεονέκτημα είναι – όπως και στο Activity Tracker – η δυνατότητα συγκέντρωσης και κατ’ επέκταση σύγκρισης όλων των δεδομένων ενός ανθρώπου σε ένα και μόνο χώρο. Αυτό ενισχύεται μέσω της πληθώρας εφαρμογών που μπορούν να συνδεθούν με αυτή, κάνοντάς την κατάλληλη για χρήστες που ήδη χρησιμοποιούν συγκεκριμένες εφαρμογές δραστηριοτήτων και δεν θέλουν να τις αποχωριστούν.

Αν και ο αριθμός υπηρεσιών που μπορούν να συνδεθούν με το Gyroscope είναι μεγάλος, ένα μειονέκτημα (το οποίο ισχύει κατά την περίοδο συγγραφής της διπλωματικής) σε σχέση με το Activity Tracker είναι η απουσία δυνατότητας άμεσης εισαγωγής δεδομένων στην εφαρμογή, δίχως τη χρήση άλλης εξωγενούς εφαρμογής. Η εφαρμογή αυτή συνεπώς δεν απευθύνεται σε άτομα που δεν είναι ήδη χρήστες άλλων εφαρμογών και που θέλουν οι ίδιοι να καταγράψουν με ημερολογιακό τρόπο τις καθημερινές τους δραστηριότητες.



Εικόνα 3.2 Εφαρμογή «Gyroscope»

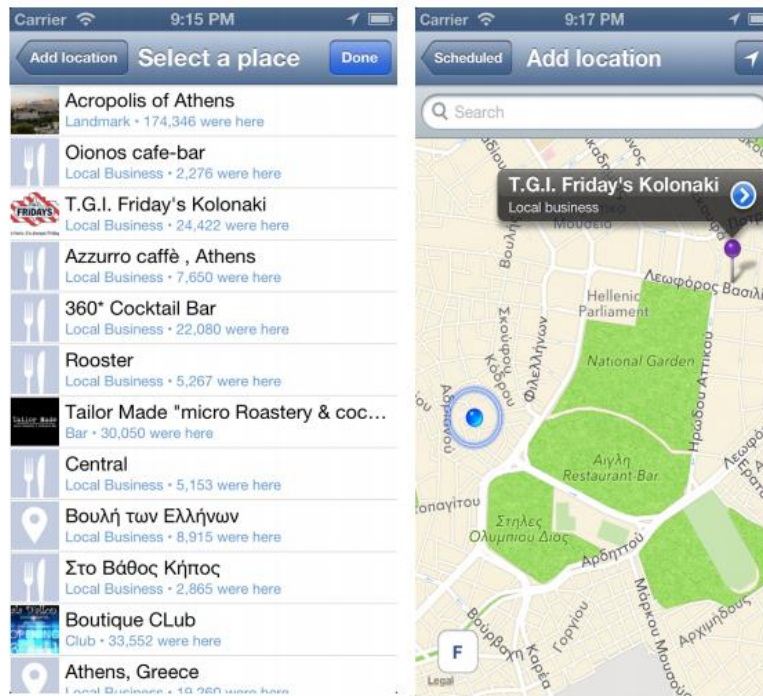
3.3 Εφαρμογή «MyActivities»³

Η εφαρμογή αυτή αναπτύχθηκε στα πλαίσια διπλωματικής εργασίας του Ε.Μ.Π και λειτουργεί ως ένας βοηθός για τη δημιουργία και οργάνωση δραστηριοτήτων σε ένα πρόγραμμα. Οι δραστηριότητες αυτές – στις οποίες ο χρήστης μπορεί να θέτει και τοποθεσία – χωρίζονται σε προγραμματισμένες και μη-προγραμματισμένες, όπου για τις πρώτες ο χρήστης λαμβάνει ειδοποιήσεις πριν τη στιγμή της προγραμματισμένης έναρξής τους, ενώ

³ <http://artemis.cslab.ntua.gr:80/Dienst/UI/1.0/Display/artemis.ntua.ece/DT2013-0091>

για τις δεύτερες ο χρήστης ειδοποιείται όταν βρίσκεται σε κοντινή απόσταση από την επιθυμητή τοποθεσία. Η εφαρμογή προσφέρει επίσης τη δυνατότητα αλληλεπίδρασης με το Facebook για την αναζήτηση ή διάθεση κοντινών τοποθεσιών στις οποίες άλλοι χρήστες ή ο ίδιος αντίστοιχα, πραγματοποιούν μια δραστηριότητα.

Η εφαρμογή υπερτερεί έναντι του Activity Tracker στο πλαίσιο της αλληλεπίδρασης μεταξύ των χρηστών, μέσω της διάθεσης των τοποθεσιών που προαναφέρθηκαν, ωστόσο μειονεκτεί στο πλαίσιο της ανάλυσης των δραστηριοτήτων του χρήστη. Η εφαρμογή δεν παρουσιάζει στατιστικά και μετρικές ούτε για τις τοποθεσίες που λαμβάνουν χώρα οι δραστηριότητες, ούτε για τις ίδιες τις δραστηριότητες που πραγματοποιήθηκαν σε ένα χρονικό πλαίσιο. Το τελευταίο οφείλεται στην αδυναμία του μοντέλου να περιγράψει πληροφορία που υπερβαίνει το όνομα, την τοποθεσία και το χρόνο εκκίνησης της δραστηριότητας, γεγονός που αντιβαίνει στους άτυπους κανόνες μοντελοποίησης και σχεδίασης ενός γεγονότος. Συμπερασματικά, ενώ η οργάνωση και εισαγωγή δραστηριοτήτων είναι εύκολη, τα οφέλη από την καταγραφή τους δεν είναι τόσα όσα στο Activity Tracker.



Εικόνα 3.3 Εφαρμογή «MyActivities»

3.4 Εφαρμογές «Small Data Lab»⁴

Οι εφαρμογές του small data lab είναι μια συλλογή από πλατφόρμες, κατάλληλα διαμορφωμένες για την λήψη κάθε χρήσιμης μορφής πληροφορίας από την καθημερινότητα ενός ανθρώπου, διατηρώντας πάντα την ανωνυμία και την ιδιωτικότητά του. Κάθε μια εξ'

⁴ www.smalldata.io

αυτών έχει την ευθύνη παρακολούθησης και καταγραφής εξειδικευμένης πληροφορίας, η οποία αφότου επεξεργαστεί είναι σε θέση να παράγει και να παρουσιάζει στατιστικά αποτελέσματα στον εκάστοτε χρήστη. Η πληροφορία αυτή είναι διαθέσιμη σε όλες τις μορφές πληροφοριακών συστημάτων με τα οποία αλληλεπιδρά ο άνθρωπος, όπως ηλ. υπολογιστές, smartphones κ.λπ. Δεδομένα όπως οι κινήσεις, οι αγορές τροφίμων, η λήψη ενός φαρμάκου, η αποστολή email κ.ά. είναι σε θέση να δημιουργήσουν πρότυπα, ικανά να περιγράψουν πλήρως τη ζωή ενός ανθρώπου, των οποίων οι μεταβολές μπορεί αυτόματα να μεταφραστούν σε πολυποίκιλες αλλαγές στα επίπεδα ενός ανθρώπου. Τα ίδια όμως δεδομένα μπορούν να χρησιμοποιηθούν και για την εξέταση της επίδρασης μιας θεραπείας σε έναν άνθρωπο ή τον εγκλιματισμό του σε ένα νέο περιβάλλον. Συνεπώς τα αποτελέσματα που συλλέγονται από τα small data δεν έχουν νόημα μόνο για τον ίδιο το χρήστη, αλλά και για άλλους ανθρώπους με τους οποίους αυτός αλληλεπιδρά. Κατά τη στιγμή συγγραφής της παρούσας διπλωματικής, οι ακόλουθες πλατφόρμες και εφαρμογές έχουν αναπτυχθεί - και συνεχίζουν να εξελίσσονται - με επίκεντρο τα small data και τους σκοπούς που αυτά επιτελούν:

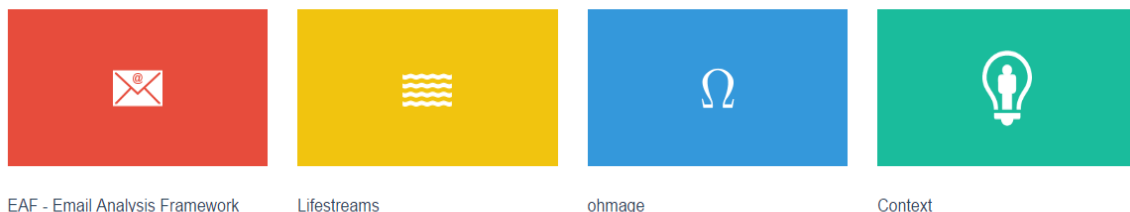
- **EAF - Email Analysis Framework**⁵: Δρα ως ενδιάμεσο λογισμικό (middleware), συλλέγοντας δεδομένα από κείμενα, email και άλλες πηγές με σκοπό να εξάγει συμπεράσματα για την εσωτερική κατάσταση του χρήστη σε ό,τι αφορά τη συχνότητα και το ύφος με τα οποία αυτός επικοινωνεί με άλλους.
- **Lifestreams**⁶: Συνεχούς ροής πλαίσιο για την επεξεργασία δεδομένων που κατηγοριοποιούνται ως small data. Συλλέγει δεδομένα από τρίτες πηγές και υπηρεσίες (όπως Moves, Gmail κ.ά.) και τα μεταφράζει ζωντανά σε χρήσιμη πληροφορία για το χρήστη. Η κύρια χρήση του αφορά τον έλεγχο ασθενών με χρόνιες παθήσεις.
- **Ohmage**⁷: Πλατφόρμα συμμετοχικής ανιχνευτικής τεχνολογίας η οποία μέσω των αντίστοιχων εφαρμογών της, συλλέγει δεδομένα από τις κινητές συσκευές των χρηστών, τόσο αυτόματα όσο και μέσω ερωτημάτων. Τα δεδομένα αυτά, μαζί με την τοποθεσία και το χρονικό στιγμιότυπο, οδηγούνται στην πλατφόρμα για επεξεργασία και η ανάλυση αυτή είναι διαθέσιμη πίσω στο χρήστη με τη μορφή ειδοποιήσεων και γραφικών απεικονίσεων.
- **Context**: Εφαρμογή για κινητά τηλέφωνα με λογισμικό android, η οποία συλλέγει συγκυριακά στοιχεία και πληροφορίες όπως το άνοιγμα και το κλείσιμο της οθόνης, τη χρήση εφαρμογών, τις κλήσεις, τα μηνύματα, τις περιηγήσεις κ.ά. Τα δεδομένα

⁵ <https://eaf.smalldata.io>

⁶ <https://github.com/ohmage/lifestreams>

⁷ <http://ohmage.org>

αυτά τροφοδοτούνται στο **Ohmage** και επεξεργάζονται από το **Lifestreams** για την εξαγωγή πληροφοριών που έχουν νόημα για το χρήστη.



Εικόνα 3.4 Πλατφόρμες και Εφαρμογές «Small Data»

3.4 Εξειδικευμένες Εφαρμογές

Ακολουθούν εφαρμογές εξειδικευμένης φύσης, μερικές από τις οποίες υποστηρίζονται και ως μέσο συγχρονισμού δεδομένων με την εφαρμογή της παρούσας διπλωματικής:

- **Runkeeper⁸**: Μια από τις πιο διαδεδομένες εφαρμογές για την καταγραφή των δραστηριοτήτων άθλησης, κίνησης ακόμη και υγείας μέσω ειδικού περιβραχιόνιου. Παρεμφερείς εφαρμογές είναι οι Google Fit, Nike +, Runtastic, Strava κ.ά.
- **Sleepy⁹**: Εφαρμογή για τη μέτρηση του ύπνου και των μετρικών του όπως η διάρκεια βαθύ ύπνου, το φαινόμενο R.E.M κ.ά. Πρόσφατα συνεργάστηκε με το Runkeeper, το οποίο πλέον υποστηρίζει τις δραστηριότητες ύπνου που καταγράφονται μέσω της εφαρμογής αυτής. Παρεμφερείς εφαρμογές είναι οι Sleep Cycle, Sleep Track, GK Sleep κ.ά.
- **Moves¹⁰**: Εφαρμογή καταγραφής της κίνησης και των τοποθεσιών του χρήστη, η οποία τρέχει στο παρασκήνιο. Ανά πάσα στιγμή ο χρήστης έχει στη διάθεση του ένα χρονογράφημα (timeline) με τα μέρη που επισκέφτηκε, την απόσταση μεταξύ τους, τον τρόπο μετακίνησης κ.ά. Παρεμφερής εφαρμογή είναι και το Foursquare, το οποίο κάνει χρήση διαφορετικών αναπαραστάσεων.
- **Fitbit¹¹**: Ίσως η πιο διαδεδομένη εφαρμογή υγείας, η οποία μέσα από τη χρήση ενός περιβραχιόνιου (wearable) μπορεί να ενημερώνει το χρήστη ανά πάσα στιγμή για μετρήσεις όπως πίεση αίματος, σφυγμοί, παλμοί καρδιάς, βάρος κ.ά. Παρεμφερείς εφαρμογές είναι το JawBone και το Withings.
- **RescueTime¹²**: Εφαρμογή που μετράει την αποδοτικότητα κάποιου που εργάζεται στον υπολογιστή και τον ενημερώνει για το βαθμό αποτελεσματικότητάς του, όπως και το χρόνο που σπαταλάει δίχως να πράττει οτιδήποτε.

⁸ www.runkeeper.com

⁹ <https://play.google.com/store/apps/details?id=it.mf.android.sleepy>

¹⁰ www.moves-app.com

¹¹ www.fitbit.com

¹² www.rescuetime.com

4

Ανάλυση Συστήματος

Στα υποκεφάλαια που ακολουθούν θα γίνει ανάλυση των λειτουργιών της εφαρμογής Activity Tracker. Μαζί θα αναλυθούν λειτουργικές και μη-λειτουργικές απαιτήσεις του συστήματος.

4.1 Απαιτήσεις συστήματος

Στο υποκεφάλαιο αυτό θα πραγματοποιηθεί μια πρώτη εισαγωγή στις γενικές απαιτήσεις που αναμένεται να ικανοποιεί το σύστημα. Ύστερα από συζητήσεις με τον υπεύθυνο της διπλωματικής, αναλύθηκαν οι απαιτούμενες ανάγκες της πλατφόρμας, καθώς και ο στόχος επίτευξης προστιθέμενης αξίας. Αυτές διαχωρίζονται σε λειτουργικές και μη-λειτουργικές και παρουσιάζονται συνοπτικά παρακάτω:

4.1.1 Λειτουργικές Απαιτήσεις

- Ανάπτυξη εφαρμογής διαδικτύου συλλογής προσωπικών δεδομένων με αυτόματες αλλά και χειροκίνητες μεθόδους.
- Παρουσίαση χρήσιμης πληροφορίας και συμπερασμάτων με τη μορφή στατιστικών στοιχείων και διαγραμματικών αναπαραστάσεων.
- Σύνδεση, εγγραφή και τροφοδότηση δεδομένων μέσω τρίτων υπηρεσιών όπως Facebook , Google+, Runkeeper.
- Μοντελοποίηση εισακτέων δεδομένων .

4.1.2 Μη-Λειτουργικές Απαιτήσεις

- Ευκολία χρήσης και εκμάθησης εφαρμογής.
- Ανταποκρισιμότητα Ιστοσελίδας (Responsiveness) με στόχο την πρόσβαση από όλες τις συσκευές.
- Ταχύτητα περιήγησης και εκτέλεσης βασικών λειτουργιών εφαρμογής.
- Χρήση βιβλιοθηκών ανοικτού κώδικα για υλοποίηση γραφιστικών οντοτήτων.
- Εποπτική παρακολούθηση πληθώρας διαγραμμάτων με νόημα για το χρήστη.
- Επιθυμία χρήστη να επιστρέφει και να επαναχρησιμοποιεί την πλατφόρμα.

4.2 Αρχιτεκτονική Συστήματος

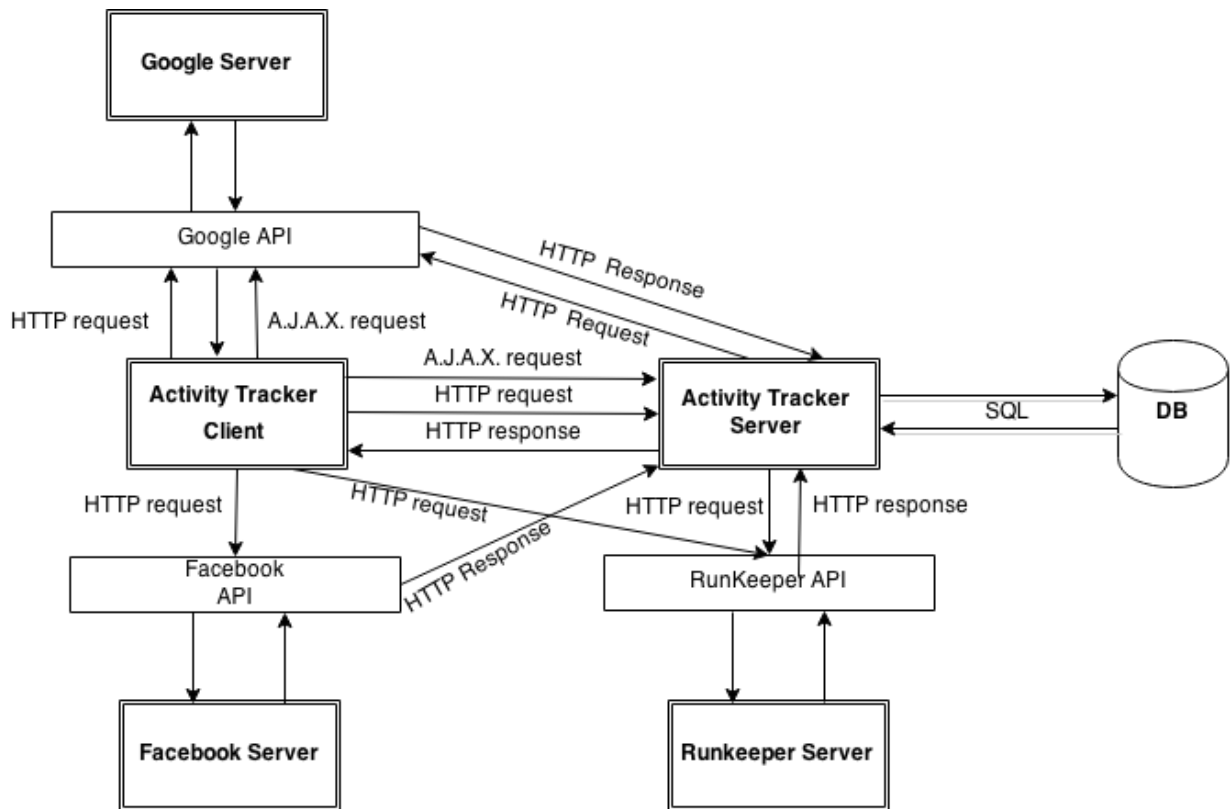
Από τα πολλά μοντέλα αρχιτεκτονικής συστημάτων που υπάρχουν, εκείνο το οποίο ταίριαζε καλύτερα στην παρούσα εφαρμογή, ήταν αυτό του πελάτη – εξυπηρετητή (client – server). Σύμφωνα με το μοντέλο αυτό, υπάρχουν δύο διαφορετικές πλευρές που αλληλεπιδρούν μεταξύ τους. Η client πλευρά ζητάει δεδομένα από τη server πλευρά και η server πλευρά της τα παρέχει πίσω. Στη διπλωματική μας, ο server είναι υπεύθυνος για τη διαχείριση της βάσης δεδομένων, στο βαθμό της εισαγωγής, αναζήτησης, διαγραφής και τροποποίησης των πληροφοριών, όπως και για την παροχή των δεδομένων της βάσης πίσω στην client πλευρά, ως μια μορφή απάντησης στο αίτημά της.

Στο κεφάλαιο αυτό θα εξεταστούν οι αρμοδιότητες και λειτουργίες κάθε υποσυστήματος, καθώς και ο τρόπος με τον οποίο καθένα από τα επιμέρους τμήματα αλληλεπιδρά με τα υπόλοιπα. Τέλος, θα παρουσιαστεί ένα διάγραμμα στο οποίο θα παρουσιάζεται η διασύνδεσή τους.

- **Βάση Δεδομένων (Database):** Η μονάδα αυτή είναι υπεύθυνη για την αποθήκευση και διατήρηση πληροφοριών, ώστε να είναι προσπελάσιμες και ανακτώμενες με χρήση κατάλληλων μεθόδων. Μέσω της αλληλεπίδρασής της με την πλευρά του server, προσφέρει την ικανότητα οργάνωσης, αναζήτησης, προσθήκης, τροποποίησης και διαγραφής δεδομένων, ενέργειες που είναι απαραίτητες για την ομαλή λειτουργία της εφαρμογής.
- **Πλευρά Πελάτη (Client Side):** Η υπομονάδα αυτή είναι υπεύθυνη για τη δημιουργία κατάλληλων διεπαφών για την αποστολή αιτημάτων, τα οποία επιτρέπουν στο χρήστη να αλληλεπιδρά με το server και κατ' επέκταση με τη βάση δεδομένων. Είναι υπεύθυνη για τη δημιουργία διεπαφών για την προσθήκη, τροποποίηση και διαγραφή δραστηριοτήτων, για την δημιουργία οπτικοποίησής τους μέσω της εισαγωγής τους σε ένα ημερολόγιο, για την κατασκευή διαγραμματικών αναπαραστάσεων, για την δημιουργία κατάλληλων χαρτών για την εισαγωγή τοποθεσιών, για την εγγραφή, εισαγωγή και τροποποίηση των χαρακτηριστικών ενός χρήστη, όπως και άλλες υπολειτουργίες οι οποίες θα αναλυθούν διεξοδικά στην επόμενη υποενότητα.
- **Πλευρά Εξυπηρετητή (Server Side):** Η υπομονάδα αυτή έχει την αποκλειστική ευθύνη της αλληλεπίδρασης με τη βάση δεδομένων, ως αποτέλεσμα ενός αιτήματος του client. Με το πέρας της αλληλεπίδρασης αυτής, στέλνει κατάλληλη απάντηση πίσω στην client πλευρά είτε με την μορφή δεδομένων, είτε με τη μορφή σήματος, περιγράφοντας την έκβαση της ενέργειας που αιτήθηκε να πραγματοποιήσει. Συγκεκριμένα, η υπομονάδα έχει την ευθύνη της εισαγωγής, τροποποίησης και διαγραφής δραστηριοτήτων από τη βάση δεδομένων, της εισαγωγής, διαγραφής και τροποποίησης των στοιχείων ενός χρήστη, της παροχής και κωδικοποίησης ζητούμενων πληροφοριών από τη βάση δεδομένων πίσω στην client πλευρά, της αλληλεπίδρασης με παρόχους κοινωνικών υπηρεσιών (Facebook, Google κ.ά.) για την λήψη πληροφοριών και λοιπών δεδομένων, όπως επίσης και άλλες υπολειτουργίες που θα αναλυθούν στην επόμενη υποενότητα του κεφαλαίου.

- Εξωγενείς Εξυπηρετητές και Προγραμματιζόμενες Διεπαφές Εφαρμογών:** Το Activity Tracker αλληλεπιδρά με άλλους servers για την λήψη πληροφοριών και δεδομένων μέσω διεπαφών (APIs) που αυτοί παρέχουν. Αυτές είναι υπεύθυνα για την επικοινωνία και κατ' επέκταση ανταλλαγή δεδομένων μεταξύ των εξωγενών συστημάτων, ορίζοντας τους τρόπους με τους οποίους αυτές υποστηρίζονται. Συγκεκριμένα, γίνεται ανταλλαγή δεδομένων με τρεις (3) διαφορετικούς παρόχους, το Facebook, το Google και το RunKeeper. Στις επόμενες υποενότητες θα εξεταστούν λεπτομερώς τα δεδομένα και οι αλληλεπιδράσεις με την εφαρμογή Activity Tracker. Εκεί, υπονοείται η χρήση των APIs για την πραγματοποίηση των λειτουργιών που περιλαμβάνουν τα εξωτερικά συστήματα αυτά.

Η αλληλεπίδραση των παραπάνω υπομονάδων είναι εμφανής στο παρακάτω block diagram:



Σχήμα 4.1 Αρχιτεκτονική Διασύνδεση Συστήματος και Υπομονάδων

Στο παραπάνω σχήμα παρατηρούμε τα υποσυστήματα που αναφέραμε, τις διασυνδέσεις τους, καθώς και τις μεθόδους αλληλεπίδρασης. Συγκεκριμένα, γίνεται χρήση του HTTP πρωτοκόλλου για την ανταλλαγή πληροφοριών, κάτι αναμενόμενο δεδομένης της υλοποίησης εφαρμογής διαδικτύου.

Το HTTP αν και αρκετά βερμπαλιστικό, έχει αρκετά προτερήματα όπως η δυνατότητα προώθησης αιτημάτων (forwarding), η χρήση ήδη ανοικτών θυρών, η ευκολία υλοποίησης, η αναγνωσιμότητα, η ανεξαρτητοποίηση σε επίπεδο φυλλομετρητή (browser) και πλατφόρμας, η δυνατότητα εύκολης ανταλλαγής δεδομένων μεταξύ άλλων διαδικτυακών

υπηρεσιών, η δυνατότητα εισαγωγής δεδομένων στην κρυφή μνήμη (caching) κ.ά. Συνεπώς, το πρωτόκολλο αυτό είναι η καλύτερη επιλογή για την επικοινωνία μεταξύ των υποσυστημάτων της εφαρμογής που κατασκευάστηκε. Από το σύνολο των HTTP αιτημάτων, η εφαρμογή έκανε χρήση μόνο του GET και του POST αιτήματος. Το πρώτο προσφέρει τη δυνατότητα αποστολής ενός αιτήματος ανάγνωσης πληροφορίας στο server και κατά συνέπεια αναμένει μια σειρά δεδομένων ως απάντηση σε αυτό, ενώ το δεύτερο προσφέρει τη δυνατότητα αποστολής (στο server) μιας σειράς από επιλογές ή δεδομένα που εισήχθησαν από το χρήστη και χρησιμοποιείται για την μεταφορά πληροφορίας. Ως απάντηση αναμένεται, συνήθως, μια οποιαδήποτε μορφή σήματος που να υποδηλώνει την έκβαση της διαδικασίας που τέθηκε εις ενεργεία από το αίτημα.

Θα πρέπει να τονιστεί ότι τα A.J.A.X. αιτήματα που εμφανίζονται στο παραπάνω σχήμα δεν είναι τίποτα άλλο παρά **ασύγχρονα** HTTP αιτήματα, τα οποία μέσω της javascript μπορούν να εκτελούνται στο παρασκήνιο χωρίς να παρεμβαίνουν στη συμπεριφορά και παρουσίαση της σελίδας. Με την A.J.A.X. καθίσταται δυνατή η ανταλλαγή πληροφορίας μεταξύ client και server, δίχως την ανάγκη για επαναφόρτωση της ιστοσελίδας. Η απουσία της, συνδυαζόμενη με την πληθώρα επιλογών που παρέχει η εφαρμογή στον χρήστη, θα καθιστούσε την εφαρμογή απίστευτα δύσχρηστη, καθώς αυτός θα βρισκόταν «παγιδευμένος» σε μια συνεχή κατάσταση ανανέωσης της ιστοσελίδας. Για το λόγο αυτό, γίνεται διαχωρισμός των αιτημάτων, έτσι ώστε να δοθεί έμφαση στην απουσία αλλαγής ιστοσελίδας, όπως και επαναφόρτωσης όλων των δεδομένων που απαιτούνται για την προβολή της.

Επιστρέφοντας στο σχήμα, παρατηρούμε ότι ο client αποστέλλει HTTP/A.J.A.X. αιτήματα (πάντα GET ή POST) στο server ενώ αυτός επιστρέφει αντίστοιχες HTTP απαντήσεις, οι οποίες περικλείουν δεδομένα ή κατάλληλα σήματα που ενημερώνουν την client πλευρά για την πορεία του αιτήματός της. Αξίζει να τονιστεί ότι πολλές φορές η απάντηση του server έπεται μιας αναζήτησης ή τροποποίησης της βάσης δεδομένων, καθώς μόνο αυτός είναι σε θέση να αλληλεπιδράσει μαζί της. Ενδιαφέρον παρουσιάζει ακόμα ότι τόσο η client όσο και η server πλευρά επικοινωνεί με κοινωνικούς παρόχους, οι οποίοι και παρέχουν δεδομένα. Η πρώτη πλευρά είναι υπεύθυνη για τη δημιουργία διεπαφών χορήγησης δικαιωμάτων πρόσβασης στον εκάστοτε πάροχο (με εξαίρεση τους χάρτες Google Maps), ενώ η δεύτερη είναι υπεύθυνη για την πραγματοποίηση ανάγνωσης των δεδομένων και την αποθήκευσή τους στην εφαρμογή μας. Συγκεκριμένα, το Facebook χρησιμοποιείται για την παροχή στοιχείων ενός χρήστη, το Google τόσο για την παροχή στοιχείων όσο και για τη χαρτογράφηση των συντεταγμένων μιας τοποθεσίας, ενώ το RunKeeper για την τροφοδότηση των δραστηριοτήτων και ενεργειών ενός χρήστη στην εφαρμογή μας. Προφανώς, στις εξωγενείς υπηρεσίες δεν υπάρχει γνώση των πρωτοκόλλων και των μεθόδων που χρησιμοποιούνται για την αλληλεπίδραση με τα υποσυστήματά τους, οπότε και δεν έχουν εξειδικευτεί στο παραπάνω σχήμα. Στα υποκεφάλαια που ακολουθούν, γίνεται ανάλυση των λειτουργιών του κάθε υποσυστήματος.

4.3 Περιγραφή Λειτουργιών Υποσυστημάτων

Στο κεφάλαιο αυτό θα περιγραφούν οι λειτουργίες που απαιτείται να εκτελεί το κάθε υποσύστημα και θα παρουσιαστούν διαγραμματικά οι ροές των δεδομένων που

πραγματοποιούνται, σε πρότυπο σχεδίασης Gane & Sarson. Επιγραμματικά, αυτές είναι οι εξής:

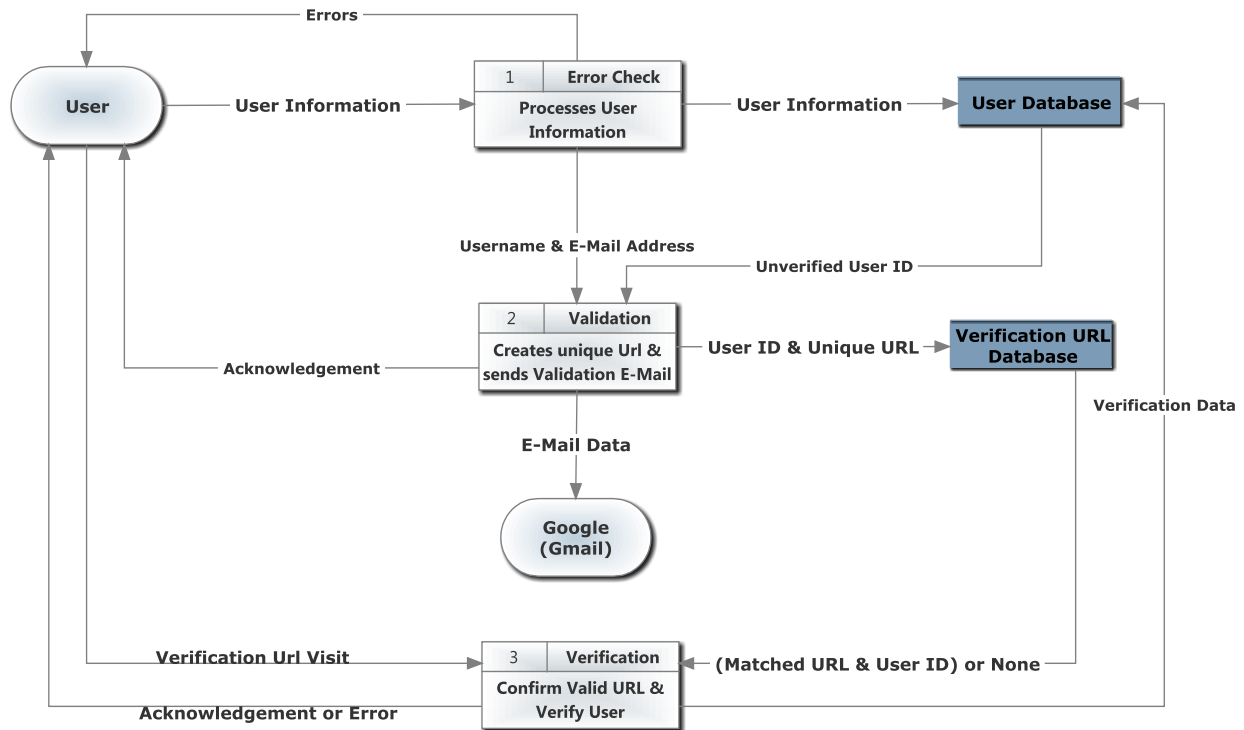
1. Εγγραφή (και τροποποίηση/διαγραφή) ενός χρήστη στο σύστημα με συμπλήρωση πεδίων.
2. Σύνδεση (και αποσύνδεση) ενός χρήστη στην εφαρμογή με συμπλήρωση πεδίων.
3. Εγγραφή και σύνδεση ενός χρήστη μέσω μέσων κοινωνικής δικτύωσης.
4. Εισαγωγή, επισκόπηση, τροποποίηση και διαγραφή μιας δραστηριότητας στο σύστημα. Για κάθε μία από αυτές, δυνατότητα προσθήκης τοποθεσίας, συμμετεχόντων, χρησιμοποιούμενων αντικειμένων, χρονικής διάρκειας, στόχου και αποτελέσματος αυτής.
5. Δυνατότητα εισαγωγής δραστηριοτήτων μέσω τρίτων εφαρμογών και κοινωνικών παρόχων.
6. Επισκόπηση μέσω χρονοδιαγράμματος (timeline) , πλέγματος (grid) και ημερολογίου (calendar).
7. Δημιουργία διαγραμμάτων και παροχή στατιστικών στοιχείων από τα εισαχθέντα δεδομένα.

Επειδή τα υποσυστήματα που ανήκουν στην εφαρμογή είναι πρακτικά λίγα σε αριθμό, επόμενο είναι, ότι καθένα από αυτά θα εκτελεί μια πληθώρα λειτουργιών. Επιπροσθέτως, επειδή υπάρχει client – server αρχιτεκτονική, η πλειονότητα των λειτουργιών απαιτεί και τα δύο υποσυστήματα, με τον client να παρέχει συχνά τη διεπιφάνεια και το server να πραγματοποιεί την υλοποίηση. Για το λόγο αυτό δεν θα αναλυθεί κάθε υποσύστημα χωριστά, αλλά θα γίνει ξεχωριστή εξέταση των λειτουργιών, τονίζοντας κάθε φορά τις αρμοδιότητες και ευθύνες της κάθε υπομονάδας.

4.3.1 Εγγραφή ενός Χρήστη στο Σύστημα με Συμπλήρωση Πεδίων

Η λειτουργία αυτή είναι η πιο βασική, καθώς είναι ο μόνος τρόπος να έχει κανείς πρόσβαση στο Activity Tracker, ενώ ταυτόχρονα απαιτείται για την καταχώρηση των ατομικών δραστηριοτήτων στο σύστημα. Ο client είναι υπεύθυνος για τη δημιουργία κατάλληλων πεδίων για την εισαγωγή δεδομένων που αφορούν το ονοματεπώνυμο, το όνομα χρήστη, τον κωδικό πρόσβασης, την ημερομηνία γέννησης, τη διεύθυνση e-mail και το γένος του κάθε ατόμου. Ακολουθεί HTTP αίτημα και ροή των δεδομένων αυτών στο server , όπου πραγματοποιείται επικύρωση και έλεγχος τόσο για ορθή εισαγωγή στοιχείων, όσο για και ήδη χρησιμοποιούμενη διεύθυνση email και όνομα χρήστη. Αν υπάρξει σφάλμα τότε επιστρέφεται μήνυμα στην πλευρά του client για ανάγκη διόρθωσης των πεδίων, διαφορετικά δημιουργείται νέος χρήστης με τα εισαγμένα στοιχεία και επιστρέφεται μήνυμα επιτυχίας στην πλευρά του client.

Για την επιτυχή ολοκλήρωση της εγγραφής απαιτείται επίσκεψη σε ένα μοναδικό σύνδεσμο που έχει σταλθεί στη διεύθυνση e-mail του νέου χρήστη. Αν ο χρήστης επισκεφτεί το μοναδικό αυτό σύνδεσμο, τότε γίνεται επαληθευμένος χρήστης (verified user) και έχει πλέον πλήρη πρόσβαση στην εφαρμογή. Αν ο σύνδεσμος που επισκεφτεί δεν είναι ορθός, τότε επιστρέφεται κατάλληλο μήνυμα σφάλματος στον client. Τα παραπάνω είναι εμφανή στο ακόλουθο διάγραμμα:



Σχήμα 4.2 Διάγραμμα Εγγραφής Χρήστη στην Εφαρμογή

4.3.2 Τροποποίηση Στοιχείων & Διαγραφή Χρήστη από το Σύστημα

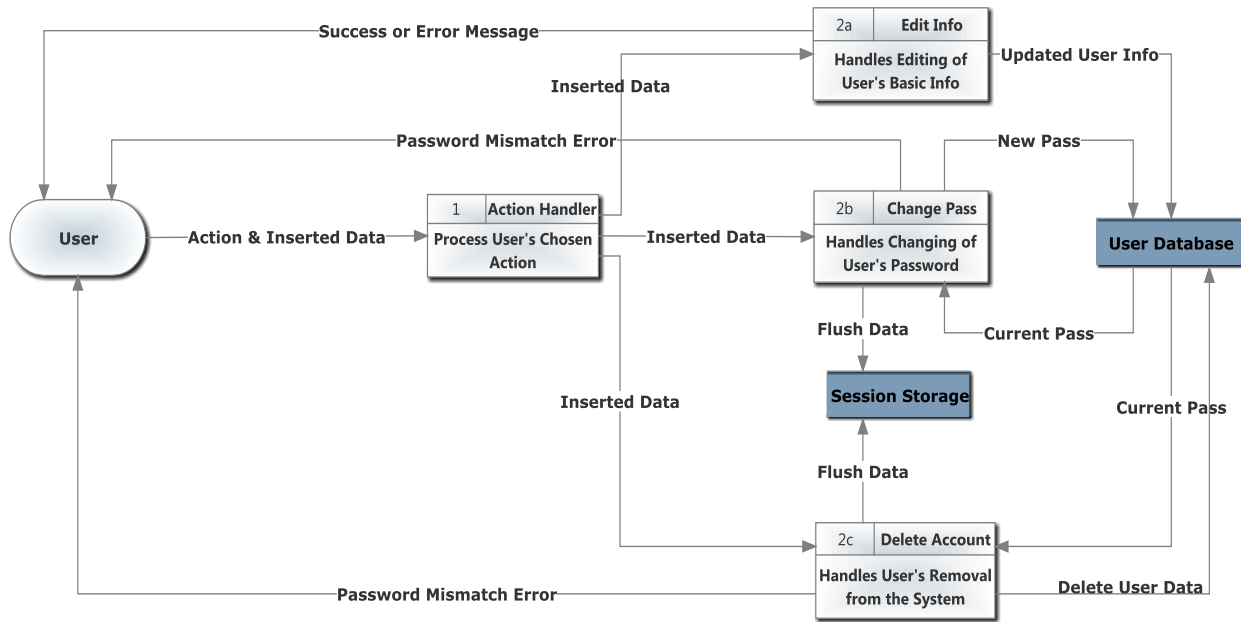
Ανά πάσα στιγμή, ο χρήστης πρέπει να έχει τη δυνατότητα να τροποποιεί τα δεδομένα του λογαριασμού του, όπως και να πραγματοποιήσει ολική διαγραφή του. Για το λόγο αυτό, ο client δημιουργεί κατάλληλες διεπαφές για τροποποίηση προσωπικών πληροφοριών, αλλαγή κωδικού πρόσβασης και διαγραφή λογαριασμού. Όπως και προηγουμένως, τα δεδομένα αυτά μεταφέρονται στο server (μέσω HTTP αιτήματος) όπου και πραγματοποιείται κατάλληλος έλεγχος ορθότητας των δεδομένων επιστρέφοντας αντίστοιχο μήνυμα επιτυχίας/αποτυχίας στο χρήστη. Συγκεκριμένα:

- Αν ο χρήστης επέλεξε να αλλάξει προσωπικά στοιχεία, τότε επιστρέφεται μήνυμα επιτυχούς τροποποίησης ή σφάλματος που υποδεικνύει τα προβληματικά πεδία.
- Αν ο χρήστης επέλεξε να αλλάξει τον κωδικό πρόσβασης του, τότε απαιτείται η εισαγωγή τόσο του παλιού όσο και του νέου κωδικού (του τελευταίου εις διπλούν). Αν αυτοί είναι ορθοί τότε επιστρέφεται μήνυμα επιτυχίας και ακολουθεί αποσύνδεση

του χρήστη από το σύστημα για λόγους ασφαλείας, διαφορετικά επιστρέφεται μήνυμα σφάλματος υποδεικνύοντας τα προβληματικά πεδία.

- Αν ο χρήστης επέλεξε να διαγράψει το λογαριασμό του, τότε καλείται να εισάγει τον κωδικό του. Αν αυτός είναι ορθός πραγματοποιείται ταυτόχρονη αποσύνδεση και διαγραφή του χρήστη από το σύστημα, διαφορετικά επιστρέφεται μήνυμα σφάλματος.

Τα παραπάνω γίνονται εμφανή στο ακόλουθο διάγραμμα ροής δεδομένων:



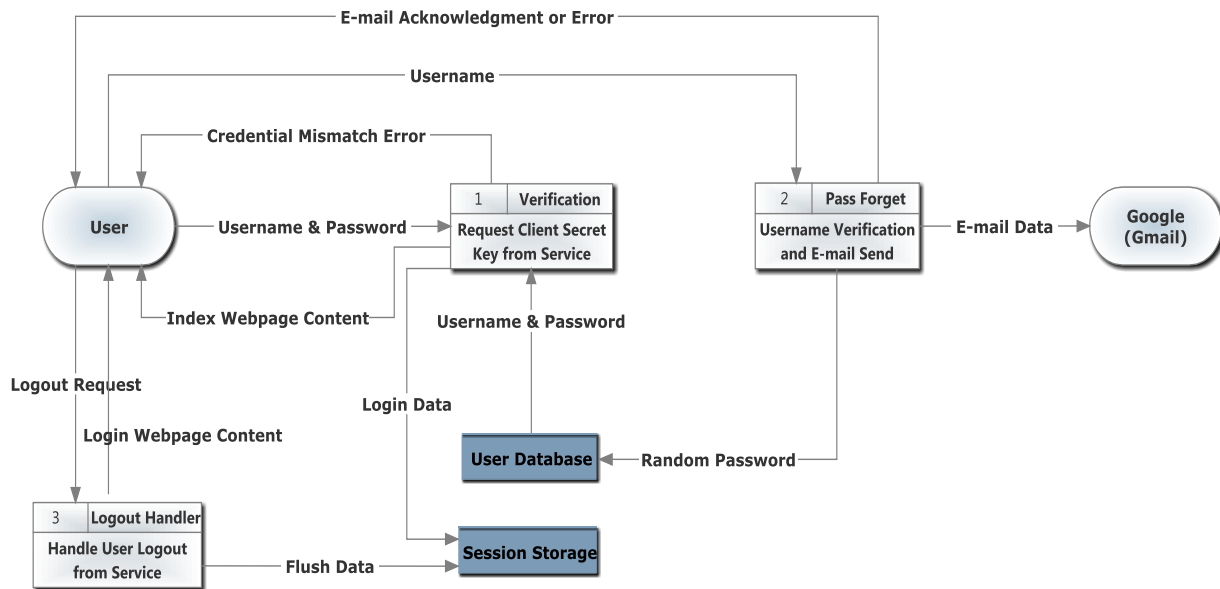
Σχήμα 4.3 Διάγραμμα Διαγραφής ή Τροποποίησης Δεδομένων Χρήστη

4.3.3 Σύνδεση (και Αποσύνδεση) ενός Χρήστη στο Σύστημα με Συμπλήρωση Πεδίων

Από τη στιγμή που ο χρήστης είναι εγγεγραμμένος, είναι απαραίτητη η υλοποίηση μιας διεπιφάνειας για τη σύνδεση (και κατ' επέκταση αποσύνδεση) του στην εφαρμογή. Για το λόγο αυτό, η client πλευρά είναι υπεύθυνη για την παρουσίαση μεθόδων εισαγωγής ονόματος χρήστη και κωδικού πρόσβασης. Τα δεδομένα αυτά αποστέλλονται στο server για επαλήθευση και είτε γίνεται σύνδεση του χρήστη στο σύστημα με ταυτόχρονη μεταφορά του στο βασικό μενού (σελίδα "index"), είτε επιστρέφεται μήνυμα λανθασμένων στοιχείων.

Για λόγους πληρότητας, θα ήταν σωστό να αναφερθεί και η λειτουργία "forgot password", που αφορά την επαναφορά του κωδικού για χρήστη που τον έχει ξεχάσει. Συγκεκριμένα, επιλέγοντας τη λειτουργία αυτή, ο χρήστης καλείται να συμπληρώσει το όνομα χρήστη που έχει επιλέξει και αυτόματα γίνεται ανάθεση νέου τυχαίου κωδικού στο χρήστη, ο οποίος και αποστέλλεται στη διεύθυνση e-mail που είναι συνδεδεμένη με το λογαριασμό αυτό. Αν το όνομα που συμπλήρωσε δεν είναι ένας εγγεγραμμένος χρήστης, τότε επιστρέφεται μήνυμα

σφάλματος στον client. Η όλη διαδικασία γίνεται εμφανής στο παρακάτω διάγραμμα, το οποίο περιλαμβάνει όλες τις δυνατές εκβάσεις και εκδοχές του αιτήματος:



Σχήμα 4.4 Διάγραμμα Σύνδεσης και Αποσύνδεσης Χρήστη Εφαρμογής

4.3.4 Εγγραφή & Σύνδεση Χρήστη μέσω Μέσων Κοινωνικής Δικτύωσης

Η εφαρμογή πρέπει να δίνει τη δυνατότητα σε ένα χρήστη να εγγραφεί στο σύστημα δίχως την ανάγκη χειρωνακτικής εισαγωγής των προσωπικών του στοιχείων, αλλά μέσω λογαριασμών που διατηρεί σε άλλες υπηρεσίες. Η παρούσα διπλωματική εστίασε στις υπηρεσίες των Facebook¹³ και Google¹⁴. Τόσο η Facebook όσο και η Google έχουν υλοποιήσει μια προγραμματιστική διεπαφή (API), η οποία – ανάμεσα σε άλλα – δίνει τη δυνατότητα σε εξωγενείς εφαρμογές να αλληλεπιδρούν με την υπηρεσία. Ανάμεσα στις πολλές αλληλεπιδράσεις συμπεριλαμβάνεται και η ανάγνωση των προσωπικών στοιχείων του χρήστη, όπως αυτά είναι καταχωρημένα στις υπηρεσίες αυτές. Συνεπώς, αν ο χρήστης παρέχει την απαραίτητη έγκριση, το Activity Tracker είναι σε θέση να κάνει ανάγνωση των στοιχείων αυτών και χρησιμοποιώντας τα να πραγματοποιήσει ταυτόχρονη εγγραφή χρήστη (αν αυτός δεν ήταν ήδη εγγεγραμμένος) και σύνδεσή του στο σύστημα. Επειδή η σύνδεση με οποιαδήποτε υπηρεσία περιλαμβάνει ακριβώς τα ίδια βήματα, θα γίνει περιγραφή τη διαδικασίας δίχως αναφορά σε συγκεκριμένη υπηρεσία.

Όταν ο χρήστης επιλέξει να συνδεθεί με μια υπηρεσία, αυτόματα ο server του Activity Tracker ζητάει από το server της υπηρεσίας ένα κλειδί πρόσβασης. Αυτό ονομάζεται Client Secret Key και είναι μοναδικό για κάθε χρήστη της υπηρεσίας. Το κλειδί αυτό, το οποίο δεν είναι τίποτα άλλο παρά μια αλφαριθμητική σειρά χαρακτήρων, είναι αντιστοιχισμένο με

¹³ www.facebook.com

¹⁴ www.google.com

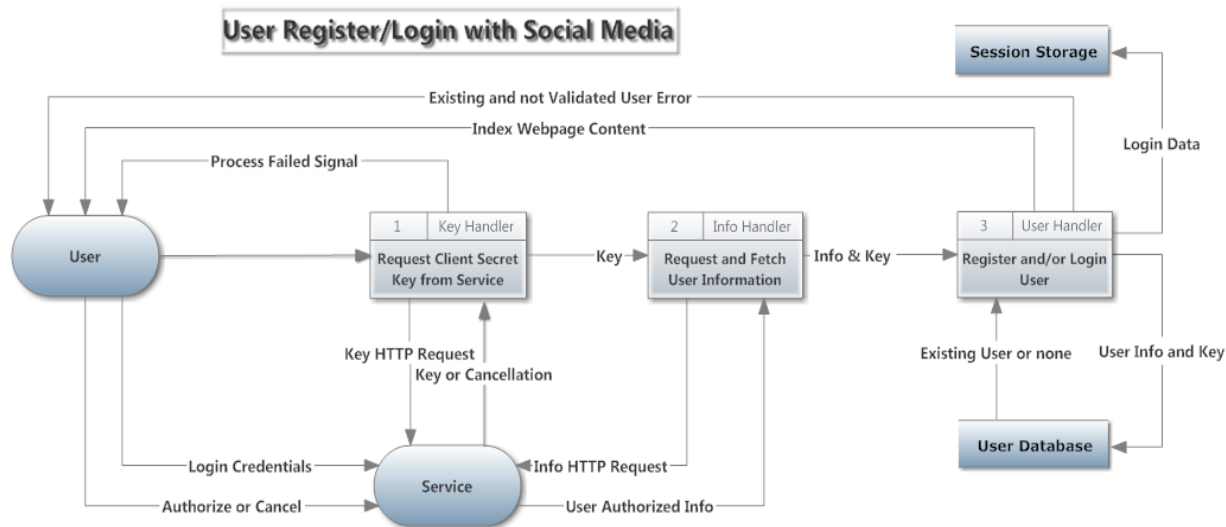
μια σειρά δικαιωμάτων πρόσβασης που θα έχει το Activity Tracker στα δεδομένα του χρήστη. Ωστόσο για την απόκτηση του κλειδιού αυτού ο χρήστης πρέπει να εξουσιοδοτήσει το Activity Tracker να λάβει το κλειδί αυτό. Από τη στιγμή που δοθεί η έγκριση, ο server της υπηρεσίας επιστρέφει το κλειδί που ζητήθηκε από το server του Activity Tracker, το οποίο και αποθηκεύεται στη βάση δεδομένων. Με τη χρήση του κλειδιού αυτού, ο server της εφαρμογής μπορεί να πραγματοποιεί HTTP αιτήματα στο server της υπηρεσίας πάνω στο σύνολο των δεδομένων στα οποία έχει πρόσβαση. Έτσι ο server αιτείται των προσωπικών στοιχείων του χρήστη και ελέγχει (μέσω της διεύθυνσης e-mail) αν αυτός είναι ήδη καταχωρημένος στο σύστημα. Αν όχι, τότε δημιουργεί νέο χρήστη (ο οποίος δεν χρειάζεται να επαληθεύσει τη διεύθυνση email του) και τον συνδέει στο σύστημα. Αν ωστόσο είναι ήδη καταχωρημένος, τότε τον συνδέει απλά στην εφαρμογή. Αξίζει να τονιστεί ότι πολλές φορές, τα κλειδιά που αναφέραμε έχουν χρόνο ζωής και συνεπώς γίνονται άχρηστα μετά από ένα πεπερασμένο χρονικό διάστημα.

Γίνεται συνεπώς κατανοητό, ότι το μόνο προαπαιτούμενο για την εγγραφή/σύνδεση χρήστη με χρήση μέσων κοινωνικής δικτύωσης είναι το κρυφό κλειδί του χρήστη (Client Secret Key). Ο server του Activity Tracker είναι υπεύθυνος για τις αιτήσεις κλειδιών και την αποθήκευσή τους στη βάση δεδομένων, για τις αιτήσεις δεδομένων χρήστη και προφανώς για την εγγραφή και σύνδεση του χρήστη στο σύστημα. Από την άλλη πλευρά, ο client είναι υπεύθυνος για την παροχή διεπαφής σύνδεσης στην υπηρεσία, εξουσιοδότησης της εφαρμογής και ενημέρωσης για την έκβαση του αιτήματος. Η διαδικασία θα περιγραφεί βήμα προς βήμα, μέσω του ακόλουθου παραδείγματος:

- *Ο Bob αποφασίζει να συνδεθεί μέσω της υπηρεσίας X μιας και το βρίσκει ταχύτερο και πιο αποδοτικό.*
 1. Επιλέγει συνεπώς «σύνδεση με X» και ανοίγει ένα νέο παράθυρο/σελίδα.
 2. Αποστέλλεται το αίτημα «σύνδεσης με X» στο server του Activity Tracker και ο τελευταίος (δρώντας ως client πλέον για την υπηρεσία X) πραγματοποιεί ένα HTTP αίτημα στη διεύθυνση της υπηρεσίας X που είναι υπεύθυνη για την παροχή κλειδιών, αναμένοντας την απάντηση του server της.
 3. Η υπηρεσία αυτή ελέγχει αν ο Bob είναι ήδη συνδεδεμένος σε αυτή (ελέγχοντας αν υπάρχει συνδεδεμένος χρήστης στο φυλλομετρητή). Αν ναι, τότε γίνεται μετάβαση στο βήμα (5). Αν όχι, ο Bob παρατηρεί ότι η σελίδα που είχε ανοίξει περιέχει διεπαφή για τη σύνδεση του στην υπηρεσία X.
 4. Ο Bob συνδέεται ως client και τα δεδομένα του μεταβαίνουν άμεσα στο server της X, όπου και πραγματοποιείται ταυτοποίηση στοιχείων.
 5. Ο server της X ελέγχει αν ο Bob έχει εξουσιοδοτήσει το Activity Tracker να λάβει τα προσωπικά του στοιχεία. Αν ναι, τότε γίνεται μετάβαση στο βήμα (7). Αν όχι τότε η σελίδα αλλάζει και περιέχει πλέον διεπαφή εξουσιοδότησης της εφαρμογής Activity Tracker στα δεδομένα του Bob.
 6. Ο Bob επιλέγει αν θέλει να επιτρέψει στο Activity Tracker να λάβει τα στοιχεία του. Αν ναι, γίνεται μετάβαση στο επόμενο βήμα. Αν όχι, τότε αποστέλλεται μήνυμα απόρριψης από το server της X στον server του Activity Tracker και αυτός σταματάει τη διαδικασία στον client και κατ' επέκταση στον Bob.

7. Από τη στιγμή που υπάρχει εξουσιοδότηση, ο server της υπηρεσίας X αποστέλλει στο server του Activity Tracker ένα μοναδικό, κρυφό κλειδί (μαζί με ενδεχομένως άλλες πληροφορίες, όπως π.χ. χρόνο ζωής κλειδιού) με το οποίο το Activity Tracker έχει πρόσβαση στα εξουσιοδοτημένα δεδομένα. Ο Bob παρατηρεί μια σελίδα να «φορτώνει», καθώς ο έλεγχος μεταβαίνει στους servers.
8. Ο server του Activity Tracker αποθηκεύει το κλειδί αυτό στη βάση δεδομένων.
9. Χρησιμοποιώντας το κλειδί, πραγματοποιεί HTTP αίτημα για τη λήψη των βασικών στοιχείων του Bob.
10. Ο server της υπηρεσίας X ταυτοποιεί το κλειδί και αν επαληθευτεί η ορθή του χρήση (σωστή τιμή κλειδιού, εντός χρόνου ζωής κ.λπ.), τότε αυτός «απαντάει» με τα εξουσιοδοτημένα στοιχεία που ζητήθηκαν.
11. Από τη στιγμή που ο server της εφαρμογής λάβει τα στοιχεία του Bob κλείνει η αλληλεπίδραση με την υπηρεσία X και πλέον γίνεται αλληλεπίδραση client-server εντός του Activity Tracker. Ο server ελέγχει αν ο Bob είναι ήδη καταχωρημένος στο σύστημα (μέσω email). Αν δεν είναι, τότε πραγματοποιείται νέα εγγραφή με αυθαίρετο όνομα χρήστη και γίνεται μετάβαση στο βήμα (12). Αν είναι ήδη εγγεγραμμένος, τότε γίνεται έλεγχος αν ο χρήστης είναι επαληθευμένος, δηλαδή αν έχει επαληθεύσει τη διεύθυνση email που χρησιμοποιεί όπως αναφέρθηκε στην λειτουργία της εγγραφής παραπάνω. Αν ναι, τότε γίνεται μετάβαση στο βήμα (12). Αν όχι, τότε ο server επιστρέφει κατάλληλο μήνυμα σφάλματος στον client, ο Bob ενημερώνεται για την ανάγκη επικύρωσης της διεύθυνσης email του και η διαδικασία σταματάει για όλα τα υποσυστήματα.
12. Ο server πραγματοποιεί σύνδεση, επιστρέφει μήνυμα επιτυχίας στον client, ο Bob μεταβαίνει στην κύρια σελίδα της εφαρμογής και η διαδικασία σύνδεσης με την υπηρεσία κοινωνικής δικτύωσης X λαμβάνει τέλος επιτυχημένα.

Θα μπορούσε κανείς συνεπώς να χωρίσει τη διαδικασία σε δύο (2) μέρη. Το 1^ο αφορά την απόκτηση του κρυφού κλειδιού όπου απαιτεί αλληλεπίδραση του server της υπηρεσίας X, του server του Activity Tracker (ο οποίος δρα ως client στο API του προηγούμενου server) και του χρήστη Bob (ο οποίος δρα ως client τόσο για το server της εφαρμογής, όσο και για το server της υπηρεσίας στο κομμάτι της σύνδεσης και εξουσιοδότησης). Το 2^ο αφορά την ανάγνωση των δεδομένων του χρήστη Bob, δεδομένης της ύπαρξης κρυφού κλειδιού. Το τμήμα αυτό απαιτεί αλληλεπίδραση μεταξύ των 2 servers (όπου ξανά ο ένας παίζει το ρόλο του client), με τον Bob να παραμένει πλήρως αμέτοχος. Τα παραπάνω είναι εμφανή στο ακόλουθο διάγραμμα ροής δεδομένων, το οποίο καλύπτει συνοπτικά όλα τα βήματα της παραπάνω διαδικασίας. Είναι σημαντικό να τονιστεί ότι το διάγραμμα δείχνει όλες τις δυνατές ροές, πράγμα που σημαίνει αυτόματα την ύπαρξη όλων των δυνατών καταστάσεων και εκβάσεων ενός γεγονότος.



Σχήμα 4.5 Διάγραμμα Σύνδεσης/Εγγραφής Χρήστη στην Εφαρμογή με χρήση Μέσων Κοινωνικής Δικτύωσης

4.3.5 Εισαγωγή Δραστηριότητας στο Σύστημα

Στο σημείο αυτό θα γίνει αναφορά στις λειτουργίες που σχετίζονται με το κύριο περιεχόμενο και άξονα της παρούσας διπλωματικής. Η πρώτη και πιο αναγκαία, είναι η δυνατότητα του χρήστη να εισάγει στο σύστημα τη δραστηριότητα ή ενέργεια που πραγματοποίησε. Με βάση τη μοντελοποίηση του υποκεφαλαίου (2.6), μια δραστηριότητα χαρακτηρίζεται από ένα σύνολο χαρακτηριστικών και αλληλεπιδράσεων με οντότητες, όπως αυτές είναι εμφανείς στο σχήμα (2.5). Προφανώς, είναι στην ευχέρεια του χρήστη να εισάγει όσα χαρακτηριστικά επιθυμεί, ωστόσο υπάρχουν κάποια τα οποία θεωρήθηκαν αναγκαία για την εισαγωγή μιας δραστηριότητας στο σύστημα, στα οποία και πραγματοποιείται έλεγχος τιμών. Σύμφωνα με το προαναφερθέν σχήμα, τα αντικείμενα και οι συμμετέχοντες μιας δραστηριότητας αντιμετωπίζονται ως ξεχωριστές οντότητες για λόγους επεκτασιμότητας, όσο και παραγωγής αναλυτικότερων διαγραμμάτων και στατιστικών. Για το λόγο αυτό καθορίστηκε αναγκαία η αποθήκευση των τιμών αυτών στο σύστημα, μέσω της εισαγωγής των ξεχωριστών τιμών στη βάση δεδομένων.

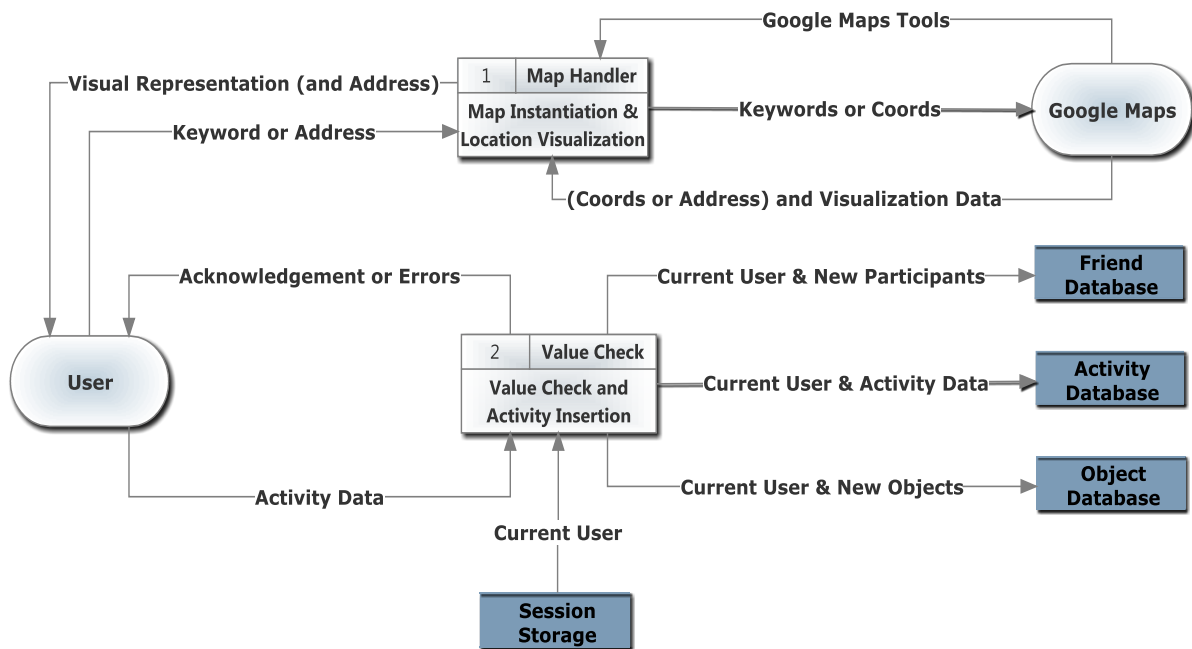
Ιδιαίτερο χαρακτηριστικό αποτελεί η τοποθεσία μιας δραστηριότητας, η οποία υλοποιείται με χρήση των χαρτών της Google. Ο client αρχικοποιεί τα απαραίτητα εργαλεία για τη δημιουργία του χάρτη και πραγματοποιεί αλληλεπίδραση με το server της Google για τον προσδιορισμό και τροποποίηση μιας τοποθεσίας.

Συγκεκριμένα, η διαδικασία που πραγματοποιείται γίνεται εμφανής μέσω του ακόλουθου παραδείγματος:

- Ο χρήστης Bob επιλέγει μια τοποθεσία ή διεύθυνση μέσω του client του Activity Tracker

1. Αυτόματα αποστέλλεται ένα ασύγχρονο HTTP αίτημα στο server της Google, το οποίο περιέχει τη διεύθυνση ή τοποθεσία που επιλέχθηκε.
2. Ο Google server επεξεργάζεται το αίτημα αυτό και επιστρέφει ένα αποτέλεσμα που περιέχει τις συντεταγμένες της τοποθεσίας που πιστεύει ότι αιτήθηκε, καθώς είναι δυνατή η ύπαρξη ταυτόσημων διευθύνσεων σε διαφορετικές τοποθεσίες.
3. Ο client σχεδιάζει ένα σύμβολο τύπου πινέζας (pin) στις συντεταγμένες αυτές, το οποίο είναι μεταφέρσιμο για την πραγματοποίηση μικρών διορθώσεων της ακριβούς τοποθεσίας. Αν ο χρήστης δεν μεταφέρει το pin, τότε η διαδικασία σταματάει εδώ.
4. Αν ο χρήστης «σύρει» την πινέζα (pin), τότε άμεσα ο client αποστέλλει ξανά ένα ασύγχρονο HTTP αίτημα στο server της Google, που αυτή τη φορά περιέχει τις συντεταγμένες της πινέζας (pin).
5. Ο Google server πραγματοποιεί μια αντίστροφη διαδικασία (reverse geolocation), παράγει τη διεύθυνση που αντιστοιχεί στις συντεταγμένες αυτές και την επιστρέφει στον client.
6. Ο client παρουσιάζει τη διεύθυνση αυτή στο Bob, ο οποίος κρίνει αν αυτή αντιστοιχεί στην τοποθεσία που έλαβε χώρα η δραστηριότητα. Αν αυτή είναι λανθασμένη, τότε ο Bob επαναλαμβάνει τη διαδικασία με διαφορετική ή πιο εξειδικευμένη είσοδο, ξεκινώντας είτε από το βήμα (1) είτε από το βήμα (4).

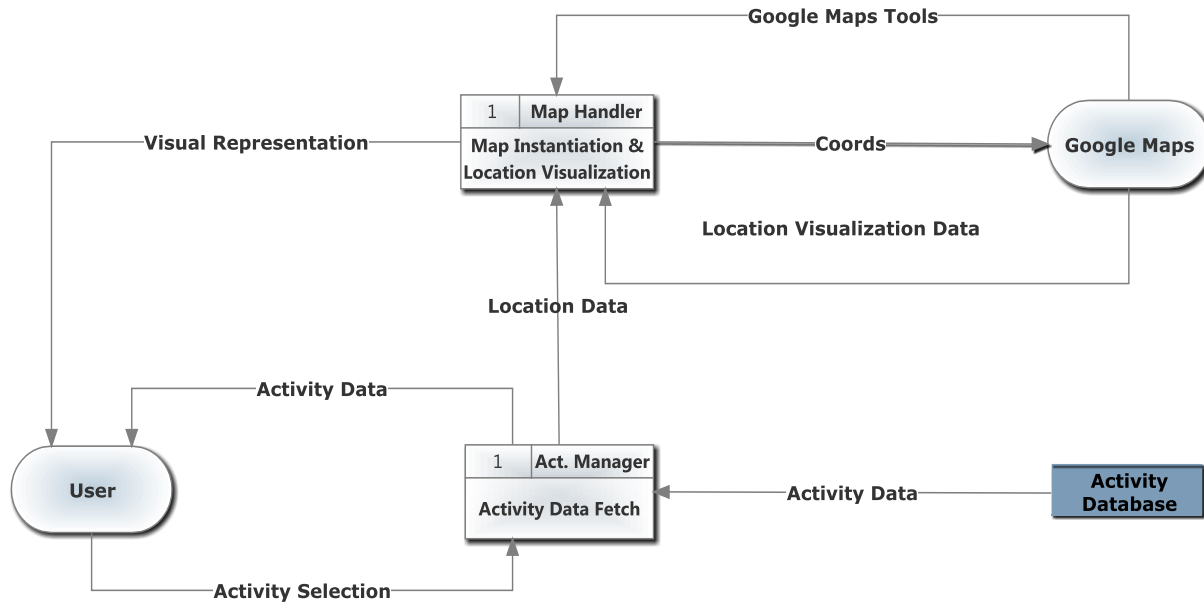
Συνοψίζοντας τα παραπάνω, η client πλευρά είναι υπεύθυνη για τη δημιουργία διεπιφάνειας για την εισαγωγή των χαρακτηριστικών που συνοδεύουν μια δραστηριότητα, όπως και για το σχεδιασμό και την αλληλεπίδραση με την υπηρεσία της Google στο κομμάτι καθορισμού της τοποθεσίας. Τα τελικά εισακτέα χαρακτηριστικά αποστέλλονται (ασύγχρονα) στο server, όπου και πραγματοποιείται ένας έλεγχος ορθότητας τιμών. Αν υπάρχουν προβληματικές τιμές, τότε ο server επιστρέφει μήνυμα σφάλματος στην client πλευρά και ο χρήστης καλείται να διορθώσει τις αντίστοιχες τιμές που εισήγαγε. Σε αντίθετη περίπτωση, ο server εισάγει τη δραστηριότητα στη βάση δεδομένων, εισάγει ενδεχομένως στο σύστημα τους νέους συμμετέχοντες και αντικείμενα που συνοδεύουν τη δραστηριότητα και επιστρέφει μήνυμα επιτυχίας στον client. Όλα τα παραπάνω είναι εμφανή στο ακόλουθο σχήμα:



Σχήμα 4.6 Διάγραμμα Εισαγωγής Δραστηριότητας

4.3.6 Επισκόπηση Δραστηριότητας στο Σύστημα

Ο χρήστης πρέπει να έχει τη δυνατότητα εποπτείας των δραστηριοτήτων που έχει εισάγει, συνεπώς είναι αναγκαία η δημιουργία κατάλληλης διεπιφάνειας. Ωστόσο, επειδή η επισκόπηση των δραστηριοτήτων μπορεί να πραγματοποιείται από διάφορα τμήματα (σελίδες) της εφαρμογής, όπως θα περιγραφεί σε επόμενα κεφάλαια, αναγκαία κρίθηκε η ύπαρξη μιας ενιαίας μεταφέρεσιμης δομής που θα μπορεί να παρουσιάζει στο χρήστη το σύνολο των χαρακτηριστικών που συνοδεύουν μια δραστηριότητα. Αυτή θα περιλαμβάνει τα ίδια χαρακτηριστικά ανεξάρτητα της δραστηριότητας (αφού όλες μοντελοποιούνται όμοια), με τις τιμές να αλλάζουν ανάλογα με την εκάστοτε δραστηριότητα. Ο client είναι συνεπώς υπεύθυνος για τη δημιουργία της επιφάνειας αυτής και για την επικοινωνία με το server με σκοπό την ανάκτηση των τιμών της δραστηριότητας. Ακόμα, αν στα χαρακτηριστικά αυτά είναι συμπληρωμένη και η τοποθεσία, τότε είναι υποχρέωση του client ο σχεδιασμός και η αναπαράσταση της τοποθεσίας στους χάρτες της Google.



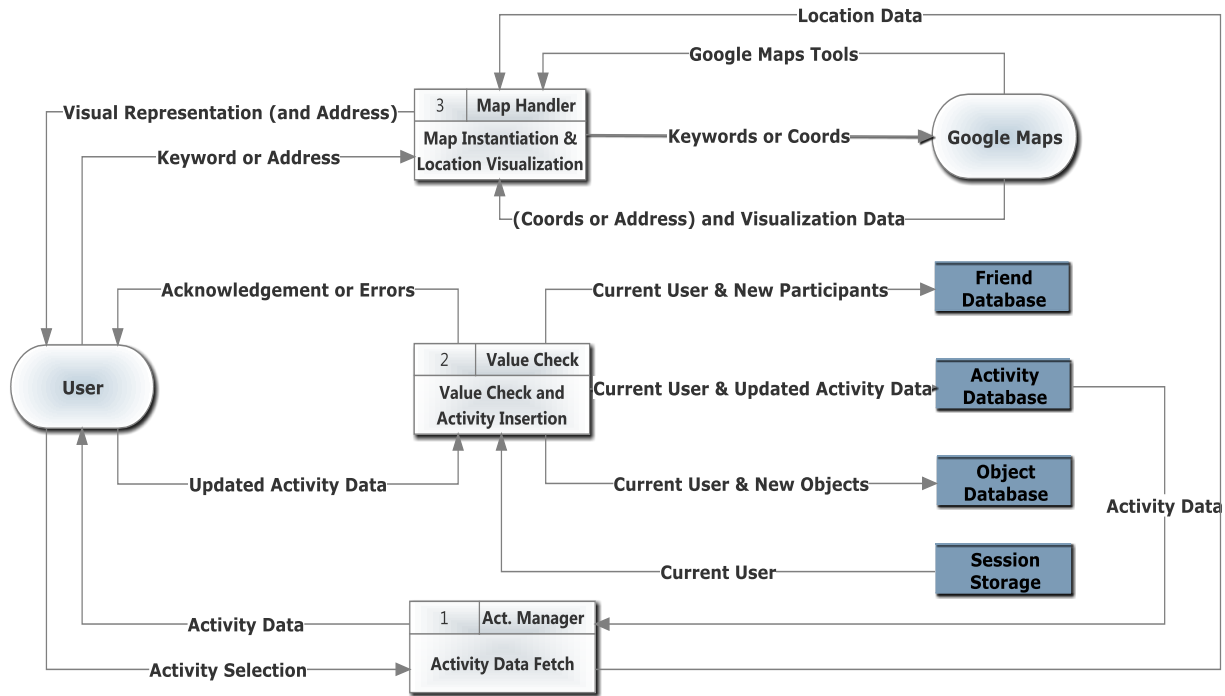
Σχήμα 4.7 Διάγραμμα Επισκόπησης Δραστηριότητας

4.3.7 Τροποποίηση Δραστηριότητας στο Σύστημα

Η τροποποίηση μιας δραστηριότητας περιλαμβάνει μια διεπιφάνεια όμοια με αυτής της εισαγωγής δραστηριοτήτων, με ταυτόχρονη ύπαρξη μιας αρχικοποίησης των τιμών των χαρακτηριστικών σε αυτά που συνοδεύουν την προς τροποποίηση δραστηριότητα. Η διαδικασία περιλαμβάνει τα εξής βήματα:

1. Δημιουργία μιας διεπιφάνειας.
2. Αρχικοποίηση χαρακτηριστικών μέσω ανάκτησης των τιμών τους από το server του Activity Tracker.
3. Ασύγχρονη αποστολή των τροποποιημένων χαρακτηριστικών πίσω στο server.
4. Πραγματοποίηση ελέγχου τιμών από το server.
5. Απάντηση μηνύματος σφάλματος στον client ή αποστολή μηνύματος επιτυχίας και ταυτόχρονη καταχώρηση των νέων τιμών στη βάση δεδομένων.

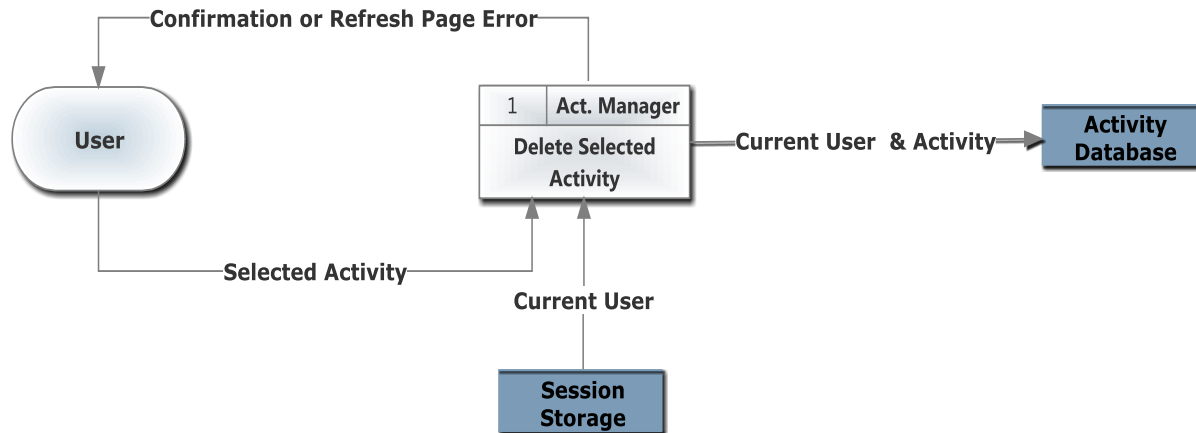
Παρατηρούμε ότι η διαδικασία είναι ένας συνδυασμός λειτουργιών επισκόπησης και προσθήκης νέας δραστηριότητας. Από τα παραπάνω βήματα παραλείψαμε την επικοινωνία με τη Google για τους χάρτες και την καταχώρηση των ενδεχόμενων νέων συμμετεχόντων και αντικειμένων του χρήστη στη βάση δεδομένων, μιας και αποτελούν υπολειτουργίες. Παρόλα αυτά είναι εμφανής η ύπαρξή τους μέσω του ακόλουθου διαγράμματος:



Σχήμα 4.8 Διάγραμμα Επισκόπησης Δραστηριότητας

4.3.8 Διαγραφή Δραστηριότητας από το Σύστημα

Ολοκληρώνοντας τις λειτουργίες που αφορούν τις δραστηριότητες, αναγκαία είναι και η ύπαρξη δυνατότητας διαγραφής μιας επιλεγμένης δραστηριότητας. Συγκεκριμένα, ο client πρέπει να παρέχει κατάλληλη διεπιφάνεια (όπως π.χ. ένα κουμπί) με την οποία ο χρήστης να μπορεί να επιλέγει τη δραστηριότητα που επιθυμεί να διαγράψει. Αυτόματα ο client πρέπει να αποστέλλει ασύγχρονο αίτημα στο server για διαγραφή της επιλεγμένης δραστηριότητας, ο οποίος με τη σειρά του ο server πρέπει να αλληλεπιδρά με τη βάση δεδομένων, να διαγράφει την καταχώρηση και να επιστρέφει μήνυμα επιτυχίας ή σφάλματος στον client. Θα πρέπει να τονιστεί, ότι ένα σφάλμα στη διαδικασία αυτή σημαίνει την απουσία της επιλεγμένης δραστηριότητας στη βάση δεδομένων, πράγμα που με τη σειρά του προδίδει ένα πρόβλημα συνέπειας (coherency problem) μεταξύ του συγκεκριμένου client και του server. Η πιο απλή λύση είναι η επαναφόρτωση της σελίδας και η ανανέωση του client σε περίπτωση μηνύματος σφάλματος του server.



Σχήμα 4.9 Διάγραμμα Διαγραφής Δραστηριότητας

4.2.9 Εισαγωγή Δραστηριοτήτων μέσω Τρίτων Εφαρμογών & Παρόχων

Όπως αναφέρθηκε και στο κεφάλαιο (3), έχουν δημιουργηθεί πολλές εφαρμογές για την καταγραφή και παρακολούθηση μεμονωμένων τύπων δραστηριοτήτων. Οι υπηρεσίες Runtastic, Runkeeper, Google Fit καταγράφουν αποδοτικά τις δραστηριότητες άθλησης, το Sleepy καταγράφει τα χαρακτηριστικά του ύπνου κ.λπ. Αυτές χρησιμοποιούνται ευρέως στη σημερινή κοινωνία, καθώς τα σύγχρονα κινητά τηλέφωνα με τη χρήση κάμερας, GPS, αισθητήρων κ.ά. θέτουν από μόνα τους τα βασικά θεμέλια για την υποστήριξη των εφαρμογών αυτών. Θα ήταν χρήσιμο, συνεπώς, το Activity Tracker να έχει τη δυνατότητα να λαμβάνει δεδομένα από άλλες εφαρμογές που τα έχουν καταγράψει, έτσι ώστε να δράσει ως μια αποθήκη παντός τύπου δραστηριοτήτων, οι οποίες μοντελοποιούνται με τον ίδιο ακριβώς τρόπο και μπορούν να συσχετίζονται μεταξύ τους. Στα πλαίσια της παρούσας διπλωματικής, επιλέχθηκε η υλοποίηση διεπαφών υποστήριξης δραστηριοτήτων άθλησης μέσω της εφαρμογής Runkeeper και δραστηριότητας ύπνου μέσω της εφαρμογής Sleepy. Η τελευταία, κατά το διάστημα συγγραφής της εργασίας ενοποιήθηκε με το Runkeeper και συνεπώς τα δεδομένα της είναι πλέον προσβάσιμα μέσω του τελευταίου. Στο ακόλουθο παράδειγμα θα παρουσιαστεί η διαδικασία που ακολουθείται για τη σύνδεση με την εφαρμογή X και την τροφοδότηση δραστηριοτήτων μέσω αυτής. Οι δύο διαδικασίες είναι αντίστοιχες για οποιονδήποτε πάροχο.

- Ο Bob, όντας επαληθευμένος χρήστης της εφαρμογής Activity Tracker επιλέγει να συνδέσει το λογαριασμό που κατέχει στην υπηρεσία/εφαρμογή X με το Activity Tracker, ώστε να μπορεί να τροφοδοτεί δεδομένα μέσω αυτού.
 1. Επιλέγει συνεπώς «σύνδεση με X» και ανοίγει ένα νέο παράθυρο/σελίδα.
 2. Αποστέλλεται το αίτημα «σύνδεσης με X» στο server του Activity Tracker και ο τελευταίος (δρώντας ως client πλέον για την υπηρεσία X) πραγματοποιεί ένα HTTP αίτημα στη διεύθυνση της υπηρεσίας X που είναι υπεύθυνη για την παροχή κλειδιών, αναμένοντας την «απάντηση» του server αυτής.

3. Η υπηρεσία αυτή ελέγχει αν ο Bob είναι ήδη συνδεδεμένος σε αυτή (ελέγχοντας αν υπάρχει συνδεδεμένος χρήστης στο φυλλομετρητή). Αν ναι, τότε γίνεται μετάβαση στο βήμα (5). Αν όχι, ο Bob παρατηρεί ότι η σελίδα που είχε ανοίξει περιέχει διεπαφή για τη σύνδεση του στην υπηρεσία X.
 4. Ο Bob συνδέεται ως client και τα δεδομένα του μεταβαίνουν άμεσα στο server της X, όπου και πραγματοποιείται ταυτοποίηση στοιχείων.
 5. Ο server της X ελέγχει αν ο Bob έχει εξουσιοδοτήσει το Activity Tracker να λάβει τα προσωπικά του δεδομένα. Αν ναι, τότε γίνεται μετάβαση στο βήμα (7). Αν όχι τότε η σελίδα περιέχει πλέον διεπαφή εξουσιοδότησης της εφαρμογής.
 6. Ο Bob επιλέγει αν θέλει να επιτρέψει στο Activity Tracker να λάβει τα στοιχεία του. Αν ναι, γίνεται μετάβαση στο επόμενο βήμα. Αν όχι, τότε αποστέλλεται μήνυμα απόρριψης από το server της X στον server του Activity Tracker και αυτός σταματάει τη διαδικασία στον client και τον Bob.
 7. Από τη στιγμή που υπάρχει εξουσιοδότηση, ο server της X αποστέλλει στο server του Activity Tracker ένα μοναδικό, κρυφό κλειδί (μαζί με ενδεχομένως άλλες πληροφορίες, όπως π.χ. χρόνο ζωής κλειδιού) με το οποίο πλέον το Activity Tracker έχει πλήρη πρόσβαση στα εξουσιοδοτημένα δεδομένα.
 8. Ο server του Activity Tracker αποθηκεύει το κλειδί αυτό στη βάση δεδομένων και το συνδέει με το χρήστη Bob. Όσο το κλειδί είναι έγκυρο και βρίσκεται στη βάση δεδομένων, η διαδικασία αυτή δεν θα πραγματοποιηθεί ξανά. Εξαίρεση αποτελεί η περίπτωση που ο ίδιος ο Bob επιλέξει την αποσύνδεση της υπηρεσίας X με το Activity Tracker, όπου και το κλειδί διαγράφεται από τη βάση δεδομένων, στερώντας από το Activity Tracker τη δυνατότητα ανάγνωσης από την υπηρεσία X.
- ο *Ο Bob, έχοντας εξουσιοδοτήσει την εφαρμογή Activity Tracker, επιλέγει να συγχρονίσει με αυτή τις δραστηριότητες που καταγράφηκαν στην υπηρεσία X.*
1. Ο Bob επιλέγει τη λειτουργία «συγχρονισμός δραστηριοτήτων από την υπηρεσία X».
 2. Άμεσα, ο client στέλνει ασύγχρονο αίτημα συγχρονισμού δεδομένων στο server του Activity Tracker, το οποίο εμπεριέχει και το όνομα της υπηρεσίας που επιλέχθηκε.
 3. Ο server αναλύει το όνομα της υπηρεσίας και βρίσκει – μέσω μιας κατάλληλης δομής δεδομένων – τους απαραίτητους συνδέσμους (URLs) και επικεφαλίδες(headers) για τη δημιουργία αιτήματος, και σε συνδυασμό με το κρυφό κλειδί του χρήστη πραγματοποιεί ένα HTTP αίτημα ανάγνωσης δεδομένων, δρώντας ως client για το server της υπηρεσίας X. Αξίζει να τονιστεί ότι τα δεδομένα που αιτούνται είναι είτε νέα δεδομένα, είτε παλιά αλλά τροποποιημένα μετά την τελευταία φορά που πραγματοποιήθηκε συγχρονισμός.
 4. Ο server της υπηρεσίας X ελέγχει την ορθή χρήση του κλειδιού και στέλνει «απάντηση» στο server του Activity Tracker. Αν υπάρχει πρόβλημα εγκυρότητας του κρυφού κλειδιού, η απάντηση περιέχει κωδικοποιημένο το σφάλμα, ο server του Activity Tracker το προωθεί στον client, ο τελευταίος ενημερώνει το Bob για

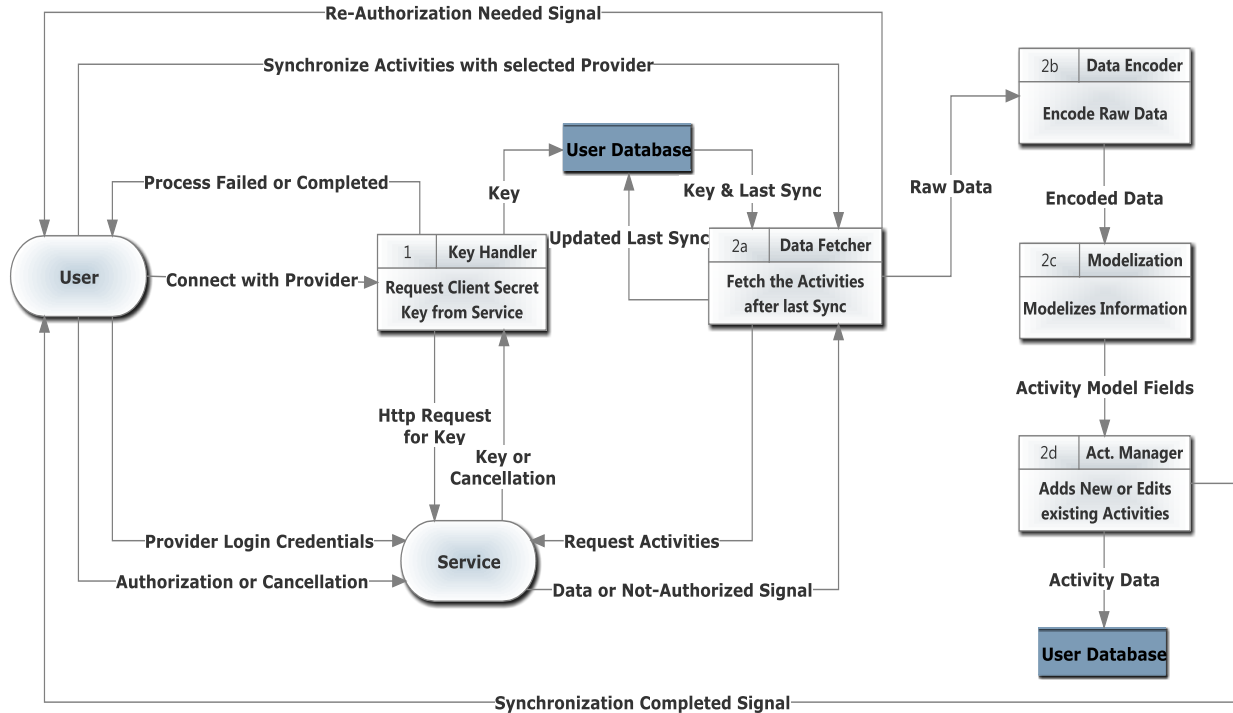
την ανάγκη επανάληψης της διαδικασίας εξουσιοδότησης και η λειτουργία σταματάει εδώ. Διαφορετικά, ο server του Activity Tracker λαμβάνει μια σειρά από δεδομένα, καταγράφοντας τη στιγμή που τα έλαβε (για την επόμενη φορά που θα πραγματοποιήσει το βήμα 3).

5. Τα δεδομένα κωδικοποιούνται σε μια διαχειρίσιμη μορφή.
6. Ακολουθεί το στάδιο της μοντελοποίησης, το οποίο χρησιμοποιεί τις τιμές των δραστηριοτήτων με σκοπό την αντιστοίχιση ή και την παραγωγή νέων τιμών για τη συμπλήρωση των πεδίων που απαιτούνται για τη μοντελοποίηση που περιγράφηκε στο υποκεφάλαιο (3.6). Το βήμα αυτό «μεταφράζει» όσο καλύτερα γίνεται την μοντελοποίηση της υπηρεσίας X, στη μοντελοποίηση που επιλέχθηκε από το Activity Tracker. Το στάδιο αυτό είναι διαφορετικό από υπηρεσία σε υπηρεσία, μιας και τα μοντέλα των δραστηριοτήτων είναι εξίσου διαφορετικά. Ο server είναι υπεύθυνος για την επιλογή των απαιτούμενων διαδικασιών του σταδίου αυτού, ανάλογα με τον πάροχο που χρησιμοποιήθηκε.
7. Ακολουθεί εισαγωγή ή τροποποίηση δραστηριοτήτων στη βάση δεδομένων. Συγκεκριμένα, για κάθε δραστηριότητα, ο server εξετάζει αν αυτή υπάρχει ήδη στη βάση δεδομένων ή αν αποτελεί νέα δραστηριότητα. Στην 1^η περίπτωση, η δραστηριότητα αποστάθηκε στο server του Activity Tracker, επειδή ο Bob τροποποίησε κάποιες τιμές των χαρακτηριστικών της στην υπηρεσία X, γεγονός που σημαίνει ότι οι ίδιες τιμές πρέπει να τροποποιηθούν και στο Activity Tracker. Στη 2^η περίπτωση, η δραστηριότητα εισάγεται στη βάση δεδομένων μαζί με ένα κωδικό που τη συνδέει με την αντίστοιχή της στην υπηρεσία X.
8. Αφού εισαχθούν ή τροποποιηθούν οι δραστηριότητες, αποστέλλεται HTTP απάντηση ολοκλήρωσης του συγχρονισμού στον client του Activity Tracker, ο οποίος με τη σειρά του ενημερώνει το Bob ότι ο συγχρονισμός με την υπηρεσία X ολοκληρώθηκε.

Παρατηρούμε λοιπόν ότι η διαδικασία συγχρονισμού δραστηριοτήτων έχει πολλές ομοιότητες με αυτή της σύνδεσης με μια υπηρεσία. Συγκεκριμένα, και στις δυο λειτουργίες απαιτείται αρχικά η εξουσιοδότηση του εκάστοτε χρήστη και η λήψη ενός κρυφού κλειδιού από το server του Activity Tracker. Έπειτα ακολουθεί η ανάγνωση δεδομένων και η αλληλεπίδραση με τη βάση που οδηγεί στη διαφοροποίηση των δύο λειτουργιών. Αμελώντας το κομμάτι που αφορά την απόκτηση του κλειδιού, η παραπάνω διαδικασία συνοψίζεται στην εύρεση απαραίτητων πληροφοριών HTTP ερωτήματος, αίτηση για δεδομένα, λήψη τους, ανανέωση της τελευταίας στιγμής συγχρονισμού, κωδικοποίηση δεδομένων, μοντελοποίηση δραστηριοτήτων και προσθήκη (ή τροποποίησή τους) στη βάση δεδομένων αντιστοιχίζοντάς τες στις αρχικές δραστηριότητες της εκάστοτε υπηρεσίας (για το βήμα 7 παραπάνω).

Είναι εύκολα αντιλήψιμο, ότι ο client είναι υπεύθυνος για τη δημιουργία διεπιφανειών σύνδεσης, εξουσιοδότησης και συγχρονισμού δραστηριοτήτων του χρήστη, για κάθε μια από τις υποστηριζόμενες υπηρεσίες, όπως επίσης και για την αποστολή της υπηρεσίας συγχρονισμού που επιλέχθηκε, στο server. Ο server είναι υπεύθυνος για όλα τα υπόλοιπα μέρη της διαδικασίας όπως περιγράφηκαν παραπάνω. Στο ακόλουθο διάγραμμα φαίνεται η

ροή των δεδομένων για την πραγματοποίηση του συγχρονισμού των δραστηριοτήτων. Η διαδικασία 1 περιγράφει την ενέργεια που απαιτείται για την απόκτηση του κρυφού κλειδιού, ενώ οι διαδικασίες 2a, 2b, 2c, 2d τις ενέργειες που απαιτούνται για το συγχρονισμό των δραστηριοτήτων της υπηρεσίας X.



Σχήμα 4.10 Διάγραμμα Εισαγωγής Δραστηριότητας μέσω Τρίτων Εφαρμογών και Παρόχων

4.3.10 Επισκόπηση μέσω Ημερολογίου (Calendar) & Πλέγματος (Grid)

Στο σημείο αυτό θα αναλυθούν οι επισκοπήσεις, ένα κομμάτι που κατά κύριο λόγο συνδέεται με τον client. Η πρώτη από αυτές είναι η επισκόπηση των δραστηριοτήτων μέσω ημερολογίου και μέσω πλέγματος, οι οποίες λόγω αλληλεπίδρασης και παρεμφερούς λειτουργικής συμπεριφοράς, παρουσιάζονται μαζί.

Ο χρήστης μέσω του ημερολογίου πρέπει να μπορεί να έχει μια επίβλεψη των δραστηριοτήτων που έχει εισάγει στο σύστημα ανά ημέρα και ώρα. Συγκεκριμένα, το ημερολόγιο αυτό πρέπει να έχει τουλάχιστον δύο (2) διαφορετικές όψεις, μια που να αφορά μηνιαία (ή και εβδομαδιαία) δεδομένα παρουσιάζοντας τις ημέρες ως υποδιαίρεση της όψης, και μια που να αφορά τα ημερήσια δεδομένα παρουσιάζοντας τις ώρες της συγκεκριμένης ημέρας ως υποδιαίρεση. Οι δραστηριότητες παρουσιάζονται στις υποδιαιρέσεις αυτές με τον τίτλο και την ώρα έναρξής τους, ενώ το μέγεθος που καταλαμβάνουν στην εκάστοτε υποδιαίρεση προδίδει τη διάρκειά τους. Για παράδειγμα στην μηνιαία όψη, η δραστηριότητα διάρκειας 48 ωρών καταλαμβάνει 2 ημερήσιες υποδιαιρέσεις, ενώ στην ημερήσια όψη η δραστηριότητα 5 ωρών καταλαμβάνει τα 5 ωριαίες υποδιαιρέσεις. Ο χρήστης επιλέγοντας μια δραστηριότητα από το ημερολόγιο πρέπει να μπορεί να έχει πρόσβαση στις λειτουργίες

αναλυτικής επισκόπησης, τροποποίησης και διαγραφής της δραστηριότητας (όπως αυτές περιγράφηκαν παραπάνω), ενώ επιλέγοντας μια υποδιαίρεση ημέρας (ή ώρας) πρέπει να έχει τη δυνατότητα εισαγωγής δραστηριότητας στην ημέρα (ή ώρα έναρξης) που επέλεξε. Συνεπώς το ημερολόγιο δρα ως μια γενική διεπιφάνεια διαχείρισης δραστηριοτήτων, καθώς πέρα από εποπτεία, δίνει πρόσβαση σε όλες τις λειτουργίες που αφορούν τις δραστηριότητες. Από την άλλη πλευρά, η οργάνωση των δραστηριοτήτων σε πλέγμα δίνει μια πιο γρήγορη και μαζική εποπτεία στις δραστηριότητες. Ωστόσο, όταν οι δραστηριότητες γίνουν πολλές το πλέγμα μπορεί να μεγαλώσει σε χαοτικά επίπεδα, μιας και θα πρέπει να περιλαμβάνει όλες τις δραστηριότητες. Για το λόγο αυτό, το πλέγμα χρησιμοποιεί το ημερολόγιο που αναφέραμε προηγουμένως, για να καθορίσει ποιες δραστηριότητες θα παρουσιαστούν και συνεπώς ανήκει στην ίδια οθόνη (με την έννοια της ιστοσελίδας) της εφαρμογής με αυτό. Πιο συγκεκριμένα, στην οργάνωση πλέγματος παρουσιάζονται μόνο οι δραστηριότητες που περιλαμβάνονται στην όψη του ημερολογίου που ο χρήστης έχει επιλέξει την παρούσα στιγμή. Για παράδειγμα αν ο χρήστης επιλέξει μηνιαία προβολή, τότε στο πλέγμα παρουσιάζονται οι δραστηριότητες που ανήκουν στον επιλεγμένο μήνα, ενώ αν επιλέξει ημερήσια προβολή παρουσιάζονται αυτές που ανήκουν στην τρέχουσα επιλεγμένη ημέρα. Παρατηρούμε συνεπώς ότι το ημερολόγιο, μέσω των όψεών του, επηρεάζει το πλέγμα. Η προστιθέμενη αξία του συγκεκριμένου τύπου εποπτείας έγκειται σε δυο (2) χαρακτηριστικά. Αφενός, στη δυνατότητα άμεσης εποπτείας της διάρκειας, μιας και κάθε δραστηριότητα πρόκειται να χαρακτηρίζεται από όνομα και διάρκεια (έναντι ονόματος και χρόνου έναρξης στο ημερολόγιο). Αφετέρου, στη δυνατότητα αναδιάταξης του πλέγματος με βάση ποικίλες επιλογές για την επίτευξη καλύτερης εποπτείας. Για παράδειγμα, ο χρήστης θα μπορούσε να οργανώσει αλφαβητικά ή ανά κατηγορία τις δραστηριότητες, να ομαδοποιήσει τις κοινές δραστηριότητες κ.ά. Όπως και στο ημερολόγιο ο χρήστης επιλέγοντας μια δραστηριότητα έχει πρόσβαση στις λειτουργίες πλήρους επισκόπησης, τροποποίησης και διαγραφής. Ο client είναι υπεύθυνος για τη γραφική υλοποίηση και σχεδιασμό τόσο του ημερολογίου όσο και του πλέγματος. Η διαδικασία που πραγματοποιείται τόσο για το ημερολόγιο όσο και το πλέγμα, γίνεται πιο κατανοητή μέσω του ακόλουθου παραδείγματος.

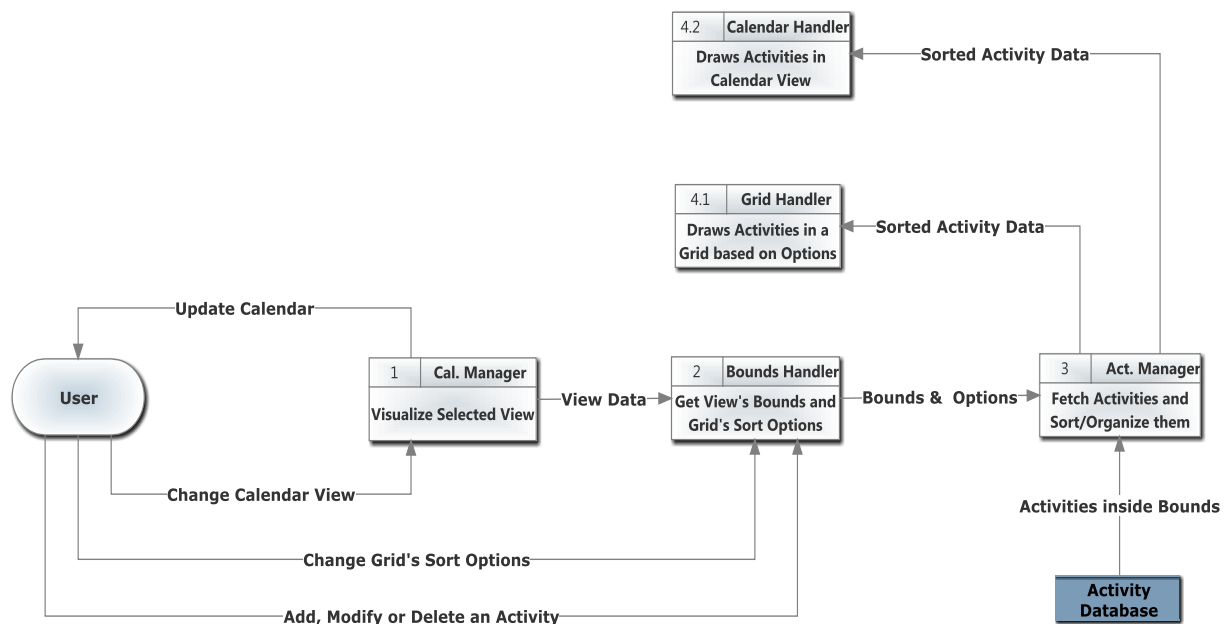
- ο *Ο Bob επιλέγει να δει τις δραστηριότητες του Ιουνίου του 2015*
 1. Ο client με την αλλαγή της όψης σε «Ιούνιο 2015» μεταβάλλει το ημερολόγιο ώστε να παρουσιάζει της υποδιαίρεσεις της επιλεγμένης όψης, οι οποίες στη συγκεκριμένη όψη είναι οι ημέρες του Ιουνίου.
 2. Ελέγχοντας τον τύπο της επιλεγμένης όψης (μηνιαία, εβδομαδιαία, ημερήσια κ.λπ.) βρίσκει τα όρια της σε επίπεδο ημέρας, ώρας, λεπτών και δευτερολέπτων και τα αποστέλλει ασύγχρονα στο server. Μαζί με τα όρια αυτά αποστέλλεται και η επιλεγμένη οργάνωση του πλέγματος, η οποία θα παίξει το ρόλο της ταξινόμησης των δραστηριοτήτων στο πλέγμα.
 3. Ο server με τη σειρά του αναζητά τις δραστηριότητες που λαμβάνουν χώρα εντός των χρονικών ορίων αυτών και τις ταξινομεί/οργανώνει με βάση την επιλογή του χρήστη. Έπειτα απαντάει πίσω στον client, παρέχοντάς του τα απαραίτητα χαρακτηριστικά των δραστηριοτήτων που απαιτούνται για την οπτικοποίησή τους σε ημερολόγιο και πλέγμα.

4.1.1. Ο client εισάγει και κατ' επέκταση σχεδιάζει τις δραστηριότητες αυτές στο ημερολόγιο συνδέοντας τις κατάλληλα με αυτές στη βάση δεδομένων.

4.1.2. Ο client σχεδιάζει τις δραστηριότητες αυτές σε σχήμα πλέγματος με βάση τη σειρά που αυτές επιστράφηκαν από το server και τις συνδέει κατάλληλα με τη βάση δεδομένων.

Αξίζει να τονιστεί, ότι η διαδικασία είναι ακριβώς ίδια αν ο Bob επέλεγε μια ημέρα αντί για ένα μήνα. Ακόμα, σε περίπτωση προσθήκης, διαγραφής ή τροποποίησης ημερομηνίας των δραστηριοτήτων ακολουθείται μια αντίστοιχη διαδικασία, (παρακάμπτοντας προφανώς το 1^ο βήμα), με την διαφορά του ότι απαιτείται η ολική διαγραφή και επανασχεδίαση του πλέγματος, καθώς απαιτείται ενδεχομένως αναδιάταξή του. Τέλος, όταν ο Bob επιλέξει διαφορετική οργάνωση πλέγματος, δίχως να αλλάξει όψη, τότε λαμβάνουν χώρα μόνο τα βήματα (2), (3) και (4.2).

Όπως εύκολα παρατηρεί κανείς, η συνολική διαδικασία λαμβάνει χώρα στην πλευρά του Client (στην οποία περιλαμβάνεται και η σχεδίαση/αρχικοποίηση του ημερολογίου) με εξαίρεση την αναζήτηση των κατάλληλων δραστηριοτήτων στη βάση δεδομένων, ενέργεια για την οποία είναι υπεύθυνος ο server. Αναλυτικότερα, οι ροές των δεδομένων είναι εμφανείς παρακάτω:



Σχήμα 4.11 Διάγραμμα Διαδικασίας Επισκόπησης Δραστηριοτήτων μέσω Ημερολογίου και Πλέγματος

Ενδιαφέρον παρουσιάζει η ροή που αφορά την προσθήκη, τροποποίηση ή διαγραφή μιας δραστηριότητας. Η διαδικασία που περιγράφηκε για το ημερολόγιο και το πλέγμα πραγματοποιείται μετά από κάθε εισαγωγή, τροποποίηση ή διαγραφή μιας δραστηριότητας, δεδομένου ότι αυτή πραγματοποιήθηκε στην όψη (ιστοσελίδα) που απεικονίζονται το ημερολόγιο και το πλέγμα. Άρα η ροή προσθήκης, τροποποίησης και διαγραφής του παραπάνω σχήματος, αντικαθίσταται από τα διαγράμματα ροής δεδομένων που αφορούν

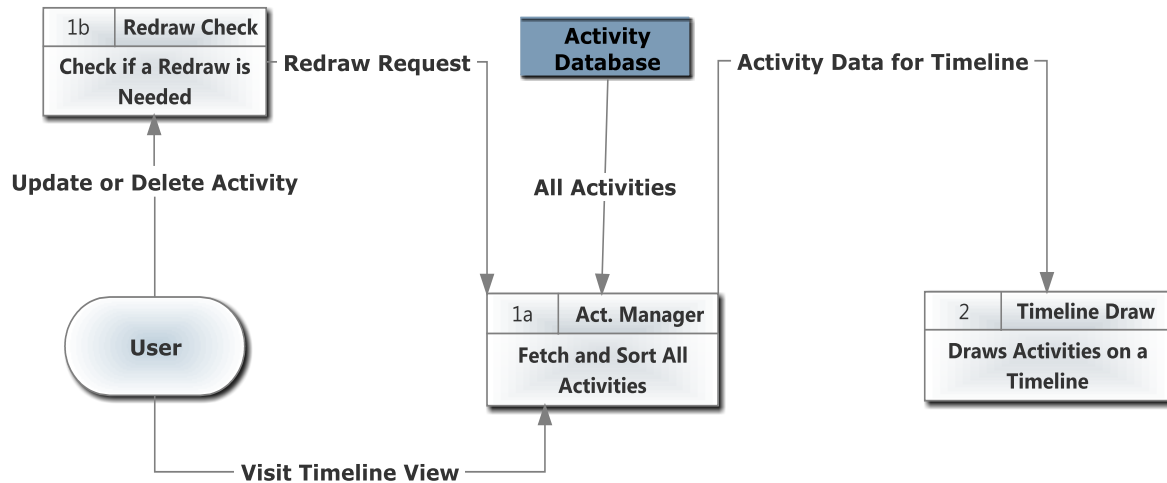
τις λειτουργίες αυτές, δημιουργώντας έτσι ένα πιο συνολικό διάγραμμα που συνδέει τις λειτουργίες που αφορούν τις δραστηριότητες με αυτές που αφορούν την εποπτεία τους.

4.3.11 Επισκόπηση μέσω Χρονοδιαγράμματος (Timeline)

Η εποπτεία μέσω ημερολογίου και πλέγματος είναι αρκετά ικανοποιητική, ωστόσο πολλές φορές δεν παρέχει καλή επισκόπηση των δραστηριοτήτων που λαμβάνουν χώρα εντός διαφορετικών όψεων. Ακόμα, δεν εστιάζει αποδοτικά στη φθίνουσα χρονική ταξινόμηση των δραστηριοτήτων, κάτι το οποίο προσφέρει η επισκόπηση με τη χρήση Χρονογραφήματος. Αυτή η μορφή εποπτείας, η οποία από μόνη της θα αποτελέσει ξεχωριστή οθόνη στην εφαρμογή, δίνει τη δυνατότητα στο χρήστη να ελέγχει αποδοτικά τη ροή των δραστηριοτήτων και ενεργειών του, ταξινομώντας τις παράλληλα σε ένα χρονικό άξονα, από τις πιο πρόσφατες στις πιο παλιές.

Ο client έχει την υπευθυνότητα σχεδίασης και υλοποίησης του χρονογραφήματος αυτού, όπως και των δραστηριοτήτων που το συνοδεύουν. Κάθε δραστηριότητα περιγράφεται εποπτικά από μερικά βασικά χαρακτηριστικά, ενώ επιλέγοντάς τη, ο χρήστης έχει τη δυνατότητα αναλυτικής εποπτείας, τροποποίησης και διαγραφής της. Για το λόγο αυτό ο client, για τη δημιουργία της συνολικής όψης, αποστέλλει αίτημα στο server, ζητώντας το σύνολο των δραστηριοτήτων ταξινομημένο κατά φθίνουσα σειρά. Ο server λαμβάνει και ταξινομεί τις δραστηριότητες, επιστρέφοντας στον client τα βασικά χαρακτηριστικά των δραστηριοτήτων, τα οποία απαιτούνται για τη σχεδίαση του Χρονοδιαγράμματος. Από τη στιγμή που ο client τα λάβει, έχει την πλήρη ευθύνη κατασκευής του. Ενδιαφέρον παρουσιάζουν οι λειτουργίες που αφορούν την τροποποίηση και διαγραφή μιας δραστηριότητας. Η διαγραφή μιας δραστηριότητας σημαίνει αυτόματα τον επανασχεδιασμό του συγκεκριμένου τμήματος του Χρονογραφήματος, μετά την αφαίρεση μιας δραστηριότητας. Ο client δεν γνωρίζει ωστόσο τις δραστηριότητες που βρίσκονται χρονικά εκατέρωθεν της διαγραμμένης δραστηριότητας, συνεπώς απαιτείται ασύγχρονο αίτημα από τον client στο server για την εύρεση των δραστηριοτήτων αυτών και τον επανασχεδιασμό του συγκεκριμένου τμήματος του Χρονοδιαγράμματος. Η διαδικασία τροποποίησης χωρίζεται σε δύο περιπτώσεις. Η 1^η αφορά την τροποποίηση των χαρακτηριστικών της δραστηριότητας που δεν σχετίζονται με τη σειρά εμφάνισής της στο Χρονογράφημα (όπως π.χ. αντικείμενα, συμμετέχοντες κ.λπ.), ενώ η 2^η αφορά την χρονική τροποποίηση όπου, όπως και με τη διαγραφή, απαιτεί επανασχεδιασμό του συγκεκριμένου τμήματος του Χρονοδιαγράμματος.

Συνεπώς, ο client είναι υπεύθυνος για την κατασκευή του Χρονοδιαγράμματος, όπως και τον – ενδεχόμενο - επανασχεδιασμό του σε περίπτωση διαγραφής ή τροποποίησης. Από την άλλη πλευρά, ο server απλά λαμβάνει και ταξινομεί τις αιτούμενες δραστηριότητες αλληλοεπιδρώντας με τη βάση δεδομένων. Αξίζει να σημειωθεί, ότι οι αρμοδιότητες τροποποίησης ή διαγραφής δραστηριότητας ανήκουν στην αντίστοιχη λειτουργία, όπως αυτή περιγράφηκε και αναλύθηκε στο υποκεφάλαιο (4.2.7) και (4.2.8) αντίστοιχα.



Σχήμα 4.12 Διάγραμμα Διαδικασίας Επισκόπησης Δραστηριοτήτων μέσω Χρονοδιαγράμματος

4.3.12 Δημιουργία Διαγραμμάτων & Παροχή Στατιστικών Στοιχείων από τα Εισαχθέντα Δεδομένα

Η τελευταία και πιο απαραίτητη λειτουργία είναι η ανάλυση των εισαγμένων δραστηριοτήτων στο σύστημα. Η λειτουργία αυτή έχει ως επίκεντρο το χρήστη, καθώς ο τελευταίος είναι σε θέση να τροποποιεί την ανάλυση που πραγματοποιείται, έτσι ώστε αυτή να προσαρμόζεται ικανοποιητικά στις ανάγκες του. Αυτό σημαίνει ότι ο χρήστης πρόκειται να έχει στη διάθεσή του διάφορα φίλτρα και επιλογές, των οποίων η μεταβολή θα μεταβάλλει και την ανάλυση που πραγματοποιείται. Το κυριότερο και πιο απαραίτητο φίλτρο είναι ο χρονικός ορίζοντας, ο οποίος καθορίζει τις δραστηριότητες αυτές που θα μετέχουν στην ανάλυση. Αλλα φίλτρα/επιλογές ενδέχεται να είναι η συγκεντρωτική ανάλυση, η επικέντρωση της ανάλυσης σε μια συγκεκριμένη δραστηριότητα, αντικείμενο ή συμμετέχοντα, η επιλογή για ανάλυση στόχων, ή τοποθεσιών κ.λπ. Συνεπώς, οι επιλογές στην συγκεκριμένη οθόνη της εφαρμογής είναι υπεύθυνες για τη μεταβολή του περιεχομένου της, εισάγοντας και αφαιρώντας δεδομένα ανάλυσης.

Τα αναλυτικά αυτά δεδομένα χωρίζονται σε δυο (2) μέρη, τα αναλυτικά στοιχεία και τα στατιστικά διαγράμματα.

Τα αναλυτικά στοιχεία αποτελούν μεμονωμένες στατιστικές τιμές οι οποίες συνήθως μεταβάλλονται ανάλογα με τις επιλογές του χρήστη. Διακρίνονται σε μέσες τιμές, ποσοστιαίες κατανομές και απόλυτους αριθμούς, έχοντας ως σκοπό την παρουσίαση μιας συνοπτικής εικόνας της ανάλυσης. Παραδείγματα είναι οι μέση χρονική διάρκεια μιας συγκεκριμένης δραστηριότητας, το ποσοστό που κατέχει μια δραστηριότητα σε σχέση με το

συνολικό αριθμό δραστηριοτήτων και ο συνολικός αριθμός δραστηριοτήτων, πάντα εντός του ορισμένου χρονικού πλαισίου. Οι τιμές αυτές δίνουν μια αρκετά γενική εικόνα, η οποία αναλύεται περαιτέρω μέσω των στατιστικών διαγραμμάτων.

Τα διαγράμματα αυτά, όπως και οι στατιστικές τιμές, τροποποιούνται ανάλογα με τα φίλτρα. Ενδέχεται να αυξηθούν ή να μειωθούν σε αριθμό, ακόμα και να αλλάξουν μορφή (διαγράμματα πίτας, ραβδογράμματα κ.ά.) για να απεικονίσουν βέλτιστα τις επιλογές του χρήστη. Παραδείγματα εξ' αυτών είναι η ανάλυση της πορείας μιας δραστηριότητας σε αριθμό εμφανίσεων και διάρκεια, ο συνολικός καταμερισμός των δραστηριοτήτων, η χαρτογράφηση των δραστηριοτήτων για ανάλυση τοποθεσιών κ.λπ.

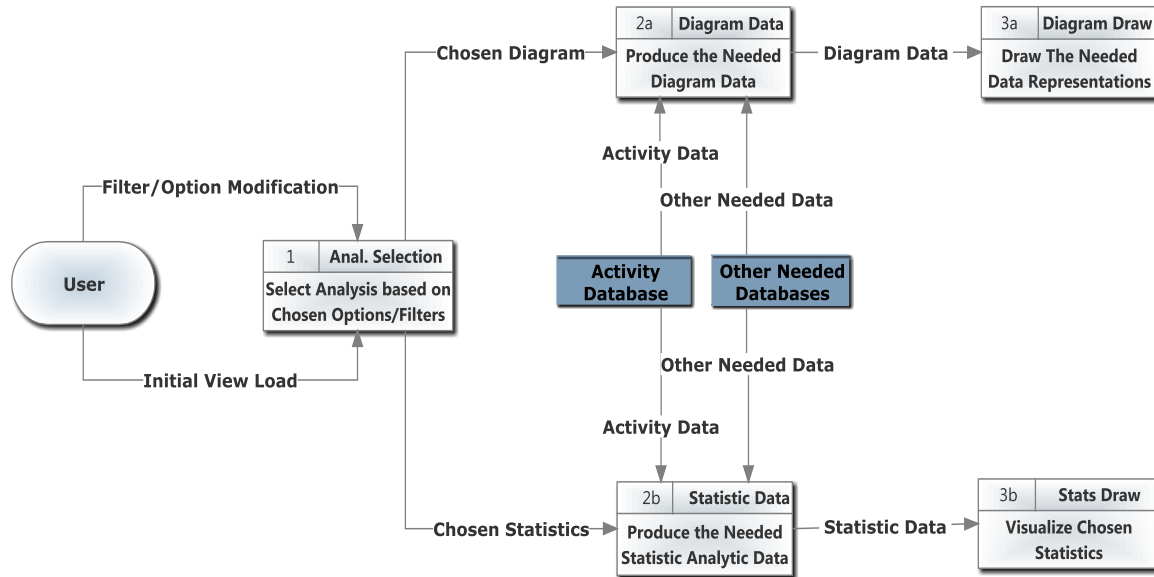
Ολοκληρώνοντας την περιγραφή, παρατηρεί κανείς την υλοποίηση μιας αρχιτεκτονικής σχεδίασης τύπου MVC (Model View Controller), κατά την οποία ένας ελεγκτής (Controller), μέσω διαφόρων προγραμματιστικών διαδικασιών (Models) είναι υπεύθυνος για την αλλαγή της όψης (View) που βλέπει ο χρήστης. Στη συγκεκριμένη περίπτωση, η έννοια του View αντιστοιχίζεται στα στατιστικά διαγράμματα και τα αναλυτικά στοιχεία, του Controller στα φίλτρα και τις επιλογές που ο χρήστης έχει στη διάθεσή του, και του Model στις διαδικασίες των υπομονάδων του συστήματος (πλευρά server) που είναι υπεύθυνες για τη μεταβολή της ανάλυσης.

Η συνολική διαδικασία περιλαμβάνει τα εξής βήματα:

1. Ο χρήστης επισκέπτεται την ιστοσελίδα ή τροποποιεί τις επιλογές/φίλτρα που του παρέχονται.
2. Μια μονάδα του client λαμβάνει τις προκαθορισμένες (default) τιμές ή τις τροποποιημένες από το χρήστη τιμές των επιλογών/φίλτρων και βρίσκει τα διαγράμματα και αναλυτικά στοιχεία που αντιστοιχούν στις επιλογές αυτές.
3. Ο client αποστέλλει ασύγχρονα δυο (2) HTTP αιτήματα στο server του Activity Tracker, ένα για την παροχή δεδομένων του συγκεκριμένου διαγράμματος και ένα για την παροχή των τιμών των αναλυτικών στοιχείων.
4. Ο server αλληλεπιδρά με τη βάση δεδομένων, υπολογίζει τις απαραίτητες τιμές και τις αποστέλλει σε κατάλληλη μορφή πίσω στον client. Στη συγκεκριμένη περίπτωση στέλνει δύο (2) απαντήσεις κατ' αντιστοιχία των αιτημάτων.
5. Ο client αφαιρώντας τα προηγούμενα διαγράμματα, σχεδιάζει και υλοποιεί τα ανανεωμένα διαγράμματα, ενώ παράλληλα παρουσιάζει τα αναλυτικά στοιχεία στο χρήστη.

Όπως είναι κατανοητό, για τη λειτουργία αυτή, ο client είναι αρμόδιος για την παροχή της διεπαφής επιλογής φίλτρων, την απόφαση για τη διαγραμματική και στατιστική αναπαράσταση των δεδομένων και για την τελική σχεδίαση και παρουσίαση των συνολικών δεδομένων στην όψη. Από την άλλη πλευρά, ο server είναι υπεύθυνος για τον υπολογισμό και μορφοποίηση των απαραίτητων τιμών που αντιστοιχούν στα εκάστοτε διαγράμματα και

στοιχεία, όπως αυτά καθορίστηκαν από την πλευρά του client. Αναλυτικότερα οι ροές των δεδομένων είναι εμφανείς παρακάτω:

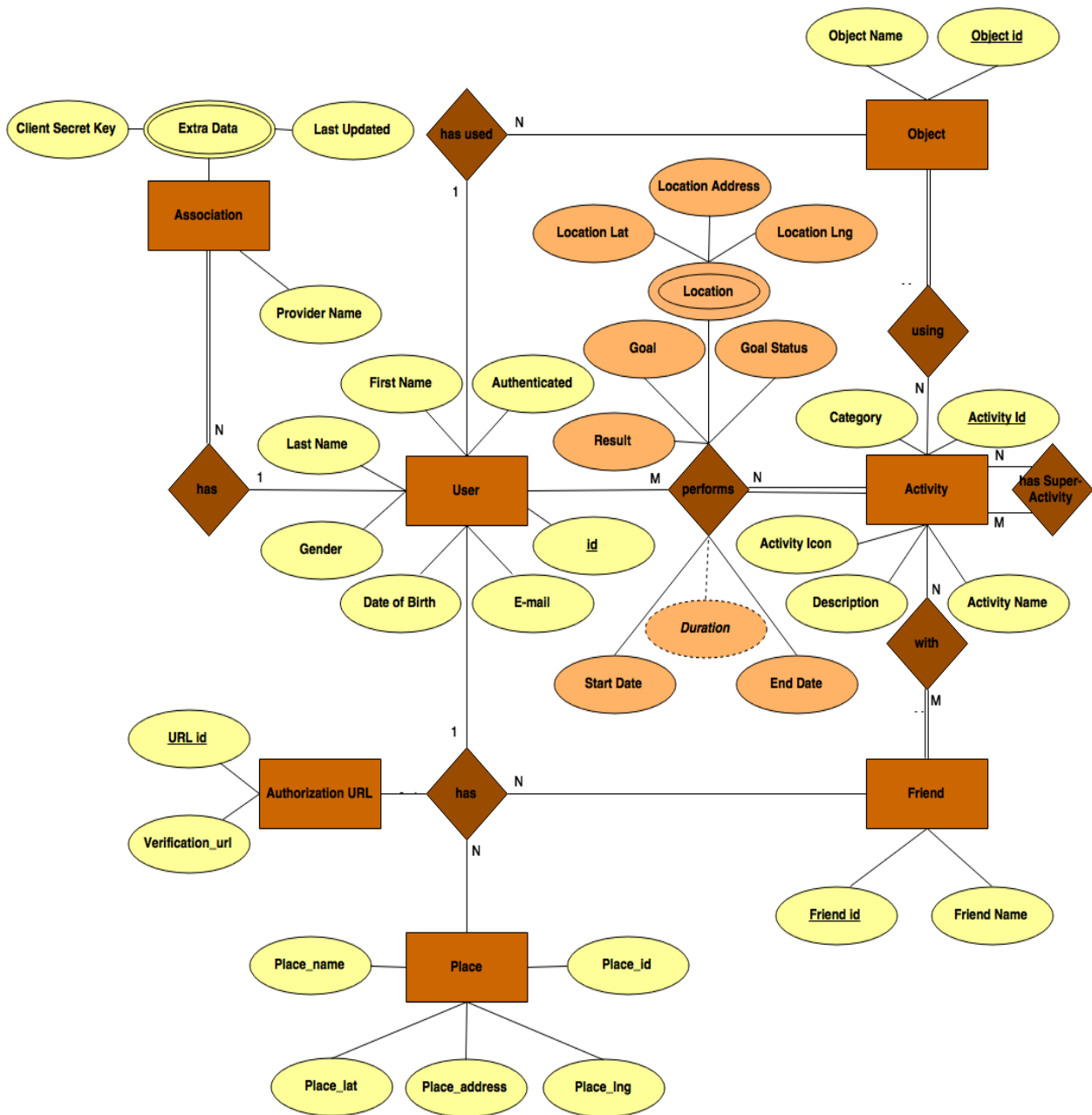


Σχήμα 4.13 Διάγραμμα Παρουσίασης Στατιστικών Στοιχείων και Διαγραμμάτων

Αξίζει να τονιστεί, ότι για τη δημιουργία αναλυτικών δεδομένων, ενδέχεται να χρειαστεί και η συμμετοχή άλλων τμημάτων της συνολικής βάσης δεδομένων του Activity Tracker, πέρα από το τμήμα που αναφέρεται στις δραστηριότητες. Για το λόγο αυτό, τα επιπλέον τμήματα αναπαριστώνται απλοϊκά μέσω της οντότητας «Other Needed Database», παρότι στην πραγματικότητα οι οντότητες αυτές ενδέχεται να είναι παραπάνω από μια ή ακόμα και πλήρως απύσες.

4.4 Μοντέλο Οντοτήτων Συσχετίσεων

Επειδή η διαδικτυακή εφαρμογή που υλοποίησε η παρούσα διπλωματική έκανε χρήση βάσης δεδομένων, κρίθηκε αναγκαία η παρουσίαση του μοντέλου οντοτήτων συσχετίσεων που χαρακτηρίζει τη βάση. Αυτό είναι το ακόλουθο:



Σχήμα 4.14 Μοντέλο Οντοτήτων Συσχετίσεων του Activity Tracker

Παρατηρώντας το παραπάνω σχήμα, το οποίο συμπύχθηκε λόγω ανάγκης παρουσίασής του σε μια ενιαία σελίδα, βλέπουμε μια πληθώρα συνδέσεων. Ο χρήστης ορίζεται ως οντότητα *απαρτιζόμενη* από διάφορα προσωπικά χαρακτηριστικά και συνδέεται μέσω συσχετίσεων με όλες τις υπόλοιπες οντότητες. Συγκεκριμένα:

- Συνδέεται «1-1» με την οντότητα των συνδέσμων (URL) επαλήθευσης μέχρι να γίνει επαληθευμένος χρήστης, όπου και παύει να υπάρχει η διασύνδεση (εξού και η σχέση «0..1»).

- Συνδέεται με την οντότητα που αφορά τις τοποθεσίες, ορίζοντας έτσι τις προσωπικές του τοποθεσίες (όπως π.χ. το σπίτι του, τη δουλειά του κ.λπ.), οι οποίες θα χρησιμοποιηθούν στην ανάλυση τοποθεσιών των δραστηριοτήτων του.
- Συνδέεται με άλλες έμβιες οντότητες οι οποίες απαρτίζουν τους φίλους και καθορίζονται από τους συμμετέχοντες που ορίζει ο χρήστης στις δραστηριότητές του. Η ύπαρξή τους βοηθάει την ανάλυση του κοινωνικού περιβάλλοντος του χρήστη.
- Συνδέεται, αντιστοίχως, και με άυλες οντότητες, δηλαδή αντικείμενα που συμμετέχουν στην διεκπεραίωση των δραστηριοτήτων. Με την καταγραφή τους ο χρήστης είναι σε θέση να εξερευνά το βαθμό αλληλεπίδρασης των εργαλείων και των αντικειμένων στις καθημερινές του δραστηριότητες.
- Συνδέεται με την οντότητα των Συνδέσεων, για την αποθήκευση όλων των δεδομένων σχετικών με τις εξωτερικές υπηρεσίες, όπως το κρυφό κλειδί, ημερομηνίες συγχρονισμού κ.ά. Τα APIs που ορίστηκαν στο υποκεφάλαιο (4.1) είναι τα χρησιμοποιούμενα μέσα για τη λήψη των προαναφερθέντων δεδομένων.
- Συνδέεται με την οντότητα που ορίζει τις διαθέσιμες δραστηριότητες μέσω μιας πολλαπλής M-N συσχέτισης. Συνεπώς, αυτή γίνεται πίνακας στην σχεσιακή βάση δεδομένων που καταγράφει τις εμφανίσεις δραστηριοτήτων συνδέοντας τις οντότητες δραστηριότητας και χρήστη, προσθέτοντας χαρακτηριστικά που αναφέρονται στη συγκεκριμένη εμφάνιση.

Αξίζει να σημειωθεί, ότι το παραπάνω διάγραμμα οντοτήτων-συσχετίσεων δεν είναι σε θέση να περιγράψει τη σχέση της πραγματοποιούμενης δραστηριότητας που προέρχεται από εξωτερική υπηρεσία, με την αντίστοιχή της στην υπηρεσία αυτή. Η σύνδεση αυτή απαιτεί τη «συσχέτιση μιας συσχέτισης» και συνεπώς είναι αδύνατο να παρουσιαστεί. Η ύπαρξή της, ωστόσο, γίνεται εμφανής στο διάγραμμα κλάσεων που θα παρουσιαστεί στο επόμενο κεφάλαιο (5).

Κλείνοντας, κοιτώντας τα ποικίλα χαρακτηριστικά που πλαισιώνουν τις οντότητες και τις συσχετίσεις, είναι εύκολο κανείς να παρατηρήσει τόσο τα συνολικά δεδομένα που καταγράφονται από την εφαρμογή, όσο και την υλοποίηση της μοντελοποίησης της δραστηριότητας, όπως αυτή ορίστηκε στο υποκεφάλαιο (2.6).

5

Σχεδίαση Συστήματος

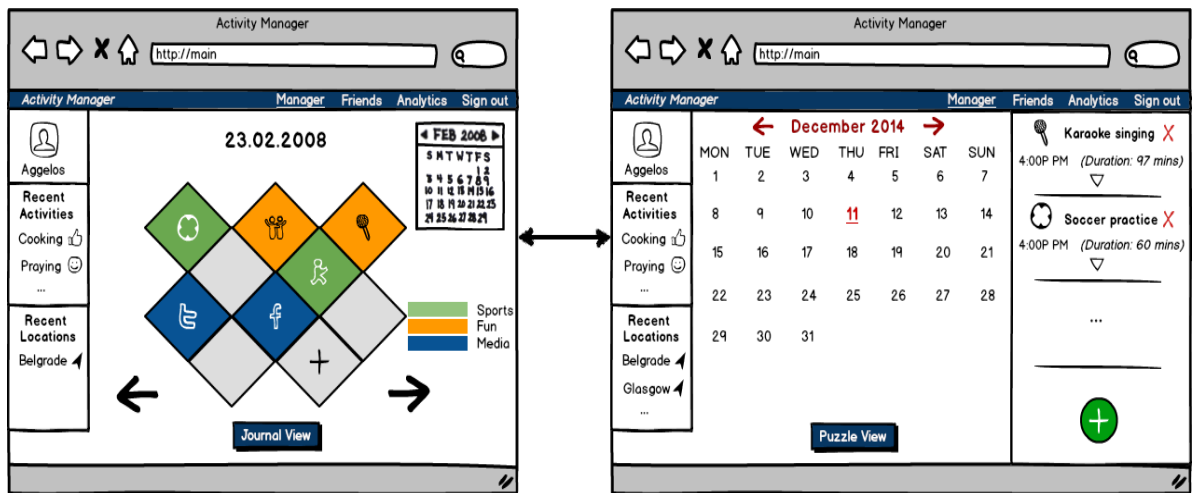
Στο κεφάλαιο αυτό θα πραγματοποιηθεί περαιτέρω εστίαση στα δομικά τμήματα της εφαρμογής, εξειδικεύοντας όλα όσα αναφέρθηκαν στο προηγούμενο κεφάλαιο. Θα παρουσιαστεί η αρχική σχεδίαση των οθονών του συστήματος ως Mock Up, θα περιγραφεί και θα αναλυθεί η αρχιτεκτονική σε επίπεδο κλάσεων, θα παρουσιαστεί το πλαίσιο (framework) και οι γλώσσες πάνω στα οποία βασίστηκε η ανάπτυξη της εφαρμογής και θα πραγματοποιηθεί δεύτερο επίπεδο αρχιτεκτονικής ανάλυσης βασιζόμενο στο επιλεγμένο πλαίσιο. Τέλος, θα γίνει αναφορά στη βάση δεδομένων που χρησιμοποιήθηκε για την αποθήκευση της πληροφορίας των χρηστών.

5.1 Ανάπτυξη με Γνώμονα το Σχεδιασμό (Design-Driven Development)

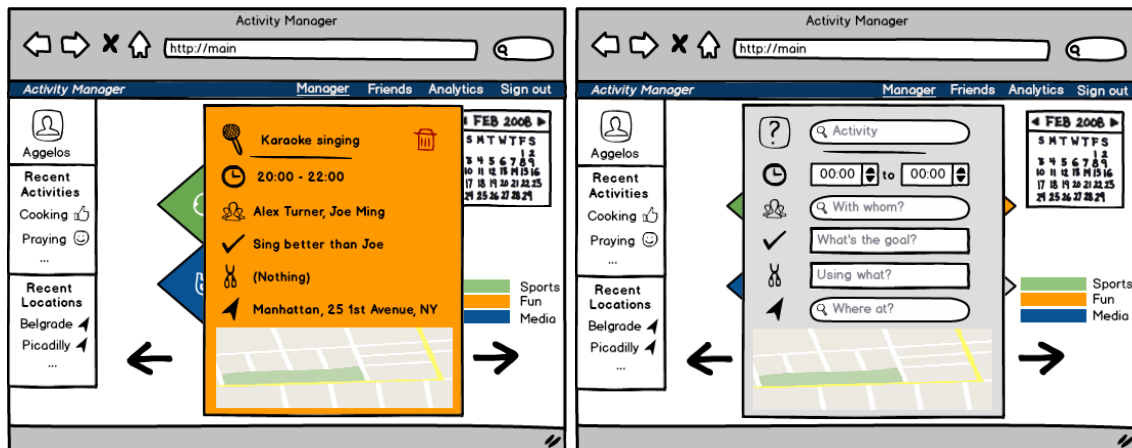
Μετά τον καθορισμό των απαιτήσεων του συστήματος, το επόμενο βήμα στην ανάπτυξη της πλατφόρμας ήταν η πρώτη σχεδίαση της εφαρμογής. Αποτέλεσμα της σχεδίασης αυτής ήταν ο καθορισμός των βασικών γλωσσών προγραμματισμού που θα αποτελούσαν τη βέλτιστη επιλογή για την ανάπτυξη της εφαρμογής, ο καθορισμός των εισόδων και επιλογών που ήταν απαραίτητο να παρέχονται στο χρήστη, καθώς και ο διαχωρισμός των προαναφερθέντων λειτουργιών της εφαρμογής σε οθόνες. Πραγματοποιήθηκε συνεπώς μια ανάπτυξη με επίκεντρο το σχεδιασμό, μέσω μιας πρώτης πρόχειρης σχεδίασης (Mock Up) που αποτέλεσε και βασικό άξονα για την τελική υλοποίηση. Στην υποενοότητα αυτή θα παρουσιαστούν οι βασικές οθόνες του συστήματος, όπως αυτές καθορίστηκαν εμμέσως από τις απαιτήσεις και τις λειτουργίες του συστήματος, ενώ η σχεδιάσή τους πραγματοποιήθηκε με χρήση του εργαλείου Balsamiq της ομώνυμης εταιρίας. Τέλος, στις ακόλουθες οθόνες η εφαρμογή παρουσιάζεται με το όνομα Activity Manager, ωστόσο καθώς αυτό ήταν ήδη δεσμευμένο, η εφαρμογή μετονομάστηκε τελικά σε Activity Tracker.



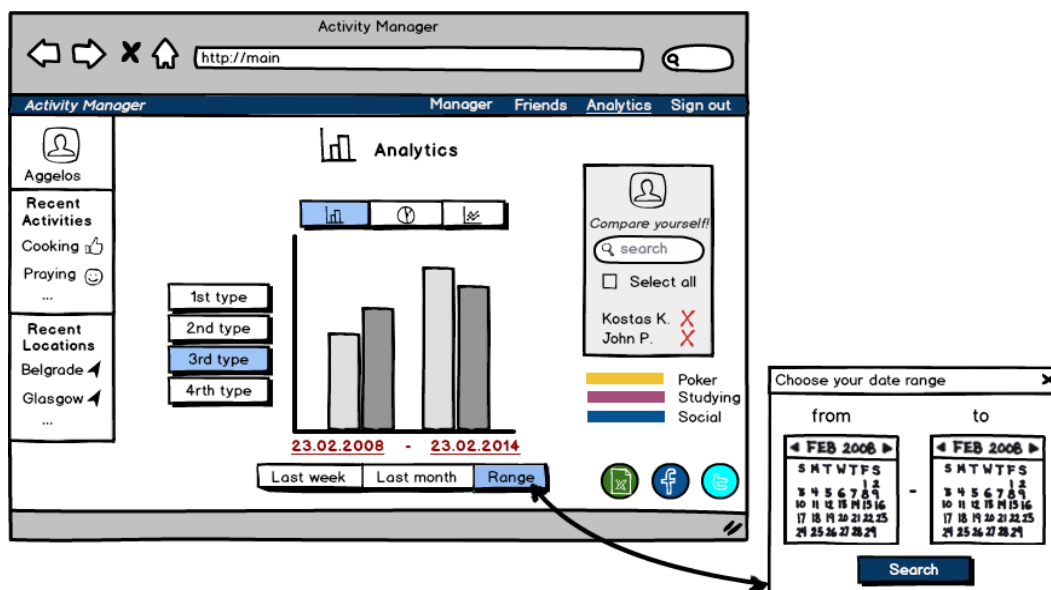
Εικόνα 5.1 Διεπιφάνεια Σύνδεσης και Εγγραφής Χρήστη στο Σύστημα



Εικόνα 5.2 Επισκόπηση Πλέγματος και Ημερολογίου



Εικόνα 5.3 Διεπαφή Βασικών Λειτουργιών των Δραστηριοτήτων



Εικόνα 5.4 Μοντέλο Παρουσίασης Αναλυτικών Στοιχείων και Διαγραμματικών Αναπαραστάσεων

Αξίζει να σημειωθεί ότι οι οθόνες αυτές αποτελούν μόνο ένα μέρος της συνολικής εφαρμογής, το οποίο κρίθηκε καθοριστικό να σχεδιαστεί ως Mock Up πριν υλοποιηθεί κανονικά. Συνεπώς, πολλές άλλες οθόνες και λειτουργίες δεν αποτυπώνονται στο υποκεφάλαιο αυτό. Οι παραπάνω εικόνες προβάλλουν μια πολύ συνοπτική εικόνα της βασικής σχεδίασης της εφαρμογής και δεν συνάδουν πλήρως με τις οθόνες της τελικής υλοποίησης.

5.2 Γλώσσες & Πλαίσια Προγραμματισμού

Προτού γίνει μετάβαση στην αρχιτεκτονική του συστήματος, είναι σημαντικό να αναφερθούν οι γλώσσες και τα πλαίσια (frameworks) που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής, ώστε να γίνει πιο κατανοητή η ανάλυση της σχεδίασης και αρχιτεκτονικής που ακολουθεί. Υλοποιώντας μοντέλο client-server, το υποκεφάλαιο χωρίζεται στις γλώσσες και πλαίσια που αφορούν την πλευρά πελάτη (front-end development) και σε αυτές που αφορούν τη πλευρά εξυπηρετητή (back-end development).

5.2.1 Ανάπτυξη Πλευράς Πελάτη (Front-End Development)

Το Activity Tracker, όντας διαδικτυακή εφαρμογή, χωρίζεται σε διαφορετικές οθόνες οι οποίες κατά συνήθως αναπαριστώνται μέσω διαφορετικών σελίδων στο διαδίκτυο. Κάθε σελίδα αποτελείται από στατικά μέρη (τμήματα που παραμένουν αμετάβλητα) και δυναμικά μέρη (μη στατικά τμήματα που αλλάζουν υπόσταση). Η υλοποίηση όλων των στατικών τμημάτων πραγματοποιήθηκε με χρήση απλή **HTML5**, ενώ των δυναμικών και όλων των επιπρόσθετων γραφικών τμημάτων (plugins) με **JavaScript**. Η γραφική διαμόρφωση πραγματοποιήθηκε και στα δυο (2) τμήματα με χρήση **CSS3**.

Για την ευκολότερη και καλύτερη υλοποίηση των διεπιφανειών έγινε χρήση της βιβλιοθήκης **jQuery 1.9.1** και του framework **Bootstrap 2.3.1**. Η πρώτη στηρίζεται στη JavaScript και προσφέρει έτοιμες υλοποιήσεις συναρτήσεων και μεθόδων για την αλληλεπίδραση με το χρήστη και το φυλλομετρητή (browser), ενώ το δεύτερο προσφέρει ένα έτοιμο σύνολο αντικειμένων υλοποιημένων με χρήση HTML, CSS και JavaScript, απαλλάσσοντας τον προγραμματιστή από την ευθύνη κατασκευής αναγκαίων front-end στοιχείων.

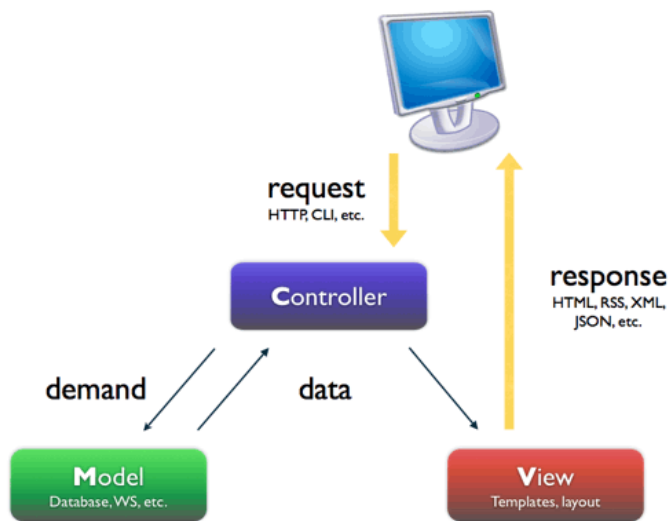
Η επικοινωνία της πλευράς client τόσο με το server του Activity Tracker, όσο και με τους server άλλων υπηρεσιών και εφαρμογών έγινε ασύγχρονα με JavaScript μέσω **A.J.A.X.**, μιας τεχνικής για την πραγματοποίηση ασύγχρονων HTTP αιτημάτων ανάμεσα σε client και server.

5.2.2 Ανάπτυξη Πλευράς Εξυπηρετητή (Back-End Development)

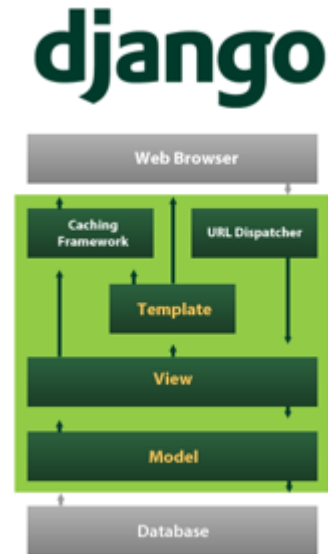
Η εφαρμογή έκανε χρήση του πλαισίου (framework) **Django 1.8** για τη σχεδίαση των λειτουργιών του server και της αλληλεπίδρασης του με τη βάση δεδομένων. Το framework αυτό ακολουθεί το πρότυπο MVC (Model-View-Controller) και είναι υλοποιημένο σε γλώσσα Python, προσφέροντας μεθόδους και τεχνικές εύκολης σχεδίασης και επικοινωνίας τόσο σε επίπεδο πελάτη – εξυπηρετητή (client – server), όσο και σε επίπεδο εξυπηρετητή – βάσης δεδομένων (server – database). Το Django εστιάζει στην κατασκευή εφαρμογών και η κατανόηση του τρόπου λειτουργίας του αποτελεί κλειδί στην κατανόηση του παρόντος κεφαλαίου. Το πρότυπο MVC αποτελείται από τρία (3) συστατικά μέρη: τα μοντέλα (models), τις όψεις (views) και του ελεγκτές (controllers). Ένα σύστημα που υλοποιεί το πρότυπο αυτό αποτελείται πάντα από τα παραπάνω μέρη, έχοντας ως τελικό στόχο την υλοποίηση της βέλτιστης αρχιτεκτονικής του συστήματος. Για την κατανόηση του προτύπου, θα θεωρηθεί ως παράδειγμα ένα πρόγραμμα επεξεργασίας κειμένου (όπως π.χ. το MS Word). Στο

σύστημα αυτό, οι ελεγκτές είναι υπεύθυνοι για την επεξεργασία ενός αιτήματος του χρήστη αποστέλλοντας εντολές τόσο στα μοντέλα για ανανέωση της κατάστασής τους (π.χ. επεξεργασία κειμένου), όσο και στις αντίστοιχες όψεις με σκοπό την αλλαγή της παρουσίας του προηγούμενου μοντέλου (π.χ. περιήγηση διαφορετικών σελίδων μέσω κύλισης). Τα μοντέλα αποθηκεύουν δεδομένα τα οποία επεξεργάζονται από τους ελεγκτές και παρουσιάζονται από τις όψεις. Κάθε στιγμή που υπάρχει μια αλλαγή στα δεδομένα, πραγματοποιείται ανανέωση από τους ελεγκτές. Οι όψεις αιτούνται πληροφοριών από τα μοντέλα με σκοπό την αναπαράστασή τους στο χρήστη. Το σχήμα (5.1) δείχνει την αλληλεπίδραση των συνιστωσών του προτύπου MVC.

Το framework Django παρουσιάζει μια αντίστοιχη συμπεριφορά. Συνοπτικά, κάθε εφαρμογή αποτελείται από models που είναι οι βασικές κλάσεις της εφαρμογής, οι οποίες καθορίζουν τα βασικά δεδομένα που μετατρέπονται σε πίνακες στη βάση δεδομένων, από templates τα οποία ορίζουν τις HTML σελίδες που βλέπει και αλληλεπιδρά ένας χρήστης, από urls τα οποία αντιστοιχίζουν μοναδικά τις ηλεκτρονικές διευθύνσεις σε συναρτήσεις χειρισμού και από views τα οποία αποτελούν τις συναρτήσεις χειρισμού, οι οποίες πραγματοποιούν επιθυμητές ενέργειες (αλληλεπιδρώντας ενδεχομένως με τη βάση δεδομένων) και επιστρέφουν συνήθως μια HTTP απάντηση στην πλευρά του client. Τα παραπάνω παρουσιάζονται στο σχήμα (5.2). Συγκρίνοντάς τα παραπάνω με το πρότυπο MVC, τα models του Django αντιστοιχίζονται στα models του MVC, τα urls και τα views του Django αντιστοιχίζονται στους controllers του MVC, ενώ τα templates του Django στα views του MVC. Στην απλούστερη εκδοχή, η φόρτωση μιας ιστοσελίδας απαιτεί την πληκτρολόγηση ενός συνδέσμου από το χρήστη, την άμεση αποστολή ενός HTTP αιτήματος στο server, την ενεργοποίηση (από το Django) της αντίστοιχης συνάρτησης χειρισμού (view) του server που αντιστοιχίζεται στον σύνδεσμο (url) αυτό και την επιστροφή ενός template ως απάντηση στο αίτημα HTTP του χρήστη. Με τον ίδιο τρόπο υλοποιούνται και τα ασύγχρονα αιτήματα. Μέσω A.J.A.X. αιτήματος αποστέλλονται δεδομένα σε ένα συγκεκριμένο σύνδεσμο (url), ενεργοποιείται η κατάλληλη συνάρτηση χειρισμού (η οποία μπορεί και λαμβάνει τα δεδομένα αυτά) και η συνάρτηση αυτή γυρνάει μια κωδικοποιημένη απάντηση στο A.J.A.X. αίτημα της πλευράς client. Οι συναρτήσεις χειρισμού των συνδέσμων (urls) του server υλοποιήθηκαν προγραμματιστικά σε **Python 2.7.9**, ενώ οποιαδήποτε δεδομένα αποστάλθηκαν από αυτές ως απάντηση σε ασύγχρονο αίτημα του client κωδικοποιήθηκαν σε **JSON**, μια κοινή κωδικοποίηση αταξινόμητης συστάδας αντικειμένων. Η βάση δεδομένων είναι SQLite, ωστόσο λόγω του framework Django, η επικοινωνία με τη βάση θα ήταν ίδια ανεξάρτητα του τύπου της. Τέλος, η αλληλεπίδραση του server του Activity Tracker με τους server άλλων υπηρεσιών (μέσω των APIs τους) πραγματοποιήθηκε με αποστολή HTTP ερωτημάτων ξανά σε Python 2.7.9.



Σχήμα 5.1 Διασύνδεση Προτύπου MVC



Σχήμα 5.15 Διασύνδεση Πλαισίου Django

Αξίζει να τονιστεί ξανά ότι η έννοια του view στο MVC αντιστοιχίζεται στην έννοια του template στο Django, ενώ η έννοια του view του Django στην έννοια του Controller στο MVC.

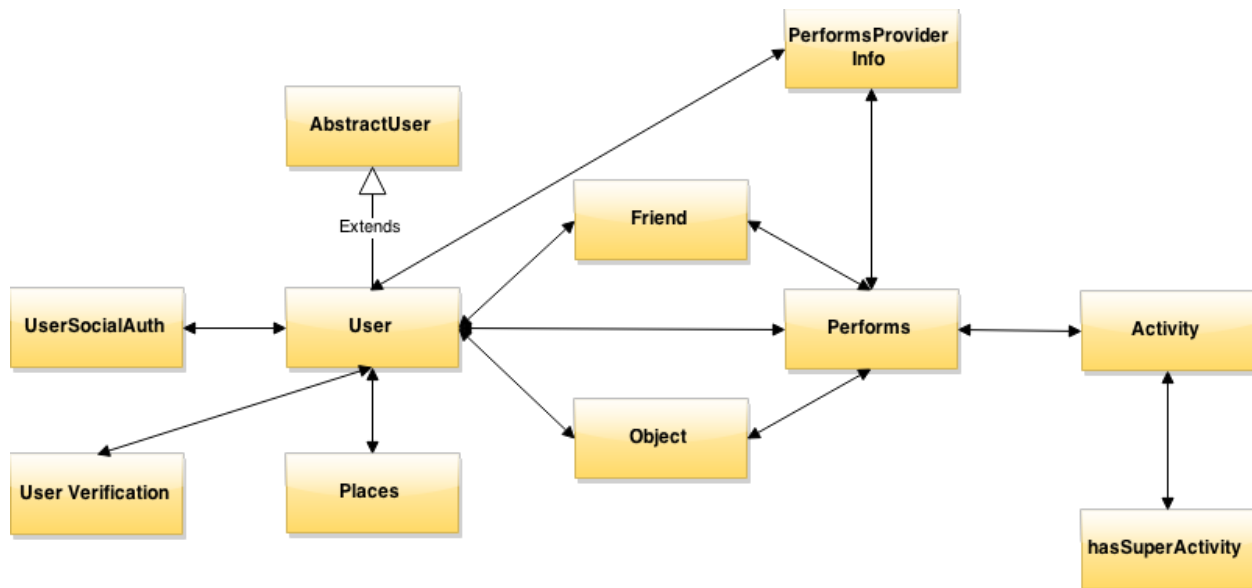
5.3 Αρχιτεκτονική Συστήματος σε Επίπεδο Κλάσεων

Έχοντας δει τόσο τις γλώσσες υλοποίησης της client πλευράς, όσο και το πλαίσιο (framework) του server, είναι πλέον δυνατή η μετάβαση στην αρχιτεκτονική του συστήματος. Σε αυτή την υποενότητα θα παρουσιαστούν συνοπτικά οι κλάσεις, οι οποίες δομούν τον κώδικα της εφαρμογής. Επειδή η διπλωματική – όπως αναφέρθηκε – βασίζεται σε Django framework, αυτό που ουσιαστικά θα περιγράψουμε είναι τα μοντέλα (models) τα οποία είναι ταυτοχρόνως κλάσεις της εφαρμογής και πίνακες στη βάση δεδομένων. Το Django δημιουργεί μια σχεσιακή βάση δεδομένων στηριζόμενη αποκλειστικά και μόνο στις κλάσεις που έχουν οριστεί μέσα στην εφαρμογή, ταυτίζοντας την έννοια αντικείμενου μιας κλάσης με τη «γραμμή» ενός πίνακα σχεσιακής βάσης δεδομένων. Συνεπώς η αρχιτεκτονική των κλάσεων ορίζει αυτόματα τη διασύνδεση και την οργάνωση των πινάκων στη βάση δεδομένων. Οι κλάσεις αυτές είναι οι ακόλουθες:

- **User:** Κλάση που περιγράφει και μοντελοποιεί τόσο τους χρήστες όσο και τους διαχειριστές του συστήματος.
- **Activity:** Κλάση που μοντελοποιεί τις δυνατές δραστηριότητες που υποστηρίζει το σύστημα και την κατηγορία στην οποία ανήκουν.
- **Performs:** Κλάση που μοντελοποιεί την ενέργεια ενός χρήστη. Συνδέει αντικείμενα των μοντέλων User και Activity και τα συνδυάζει με έξτρα χαρακτηριστικά για να περιγράψει πλήρως μια δραστηριότητα και το πλαίσió της.

- **Places:** Κλάση που περιγράφει τις προσωπικές τοποθεσίες ενός χρήστη (αντικείμενου του μοντέλου User).
- **UserVerification:** Κλάση που είναι υπεύθυνη για την επιβεβαίωση διεύθυνσης email ενός χρήστη και συνδέει τον ατομικό σύνδεσμο επιβεβαίωσης (verification url) με τον εκάστοτε χρήστη.
- **UserSocialAuth:** Η κλάση αυτή είναι απαραίτητη για τη σύνδεση, εγγραφή και ανταλλαγή δεδομένων μέσω social media. Κύριος λόγος ύπαρξής της είναι η αποθήκευση των κρυφών κλειδιών που επιστρέφει ο εκάστοτε πάροχος στο αίτημα που πραγματοποιήθηκε.
- **PerformsProviderInfo:** Όταν παρέχονται δεδομένα στο Activity Tracker από κάποιο άλλο πάροχο, τα μοναδικά αναγνωριστικά των δεδομένων – όπως αυτά περιγράφονται στον πάροχο – αποθηκεύονται με χρήση της κλάσης αυτής. Με άλλα λόγια, η κλάση αυτή συνδέει τις δραστηριότητες του Activity Tracker οι οποίες προέρχονται από κάποια υπηρεσία, με την αντίστοιχη υλοποίησή τους στην υπηρεσία αυτή.
- **Friend:** Κλάση που ορίζει το φίλο ενός χρήστη της εφαρμογής, ο οποίος συμμετέχει σε μια δραστηριότητα.
- **Object:** Κλάση που ορίζει το αντικείμενο που χρησιμοποιεί ένας χρήστης σε μια δραστηριότητα.
- **hasSuperActivity:** Η κλάση αυτή συνδέει τα αντικείμενα της κλάσης Activity, ορίζοντας σχέσεις υπερ- και υπο-δραστηριοτήτων με τελικό στόχο την κατασκευή ιεραρχιών ανάμεσα στις δραστηριότητες.

Οι παραπάνω κλάσεις συνδέονται μεταξύ τους σύμφωνα με το ακόλουθο διάγραμμα. Στο ίδιο σχήμα απεικονίζεται και η κληρονομικότητα των κλάσεων αυτών:



Σχήμα 5.2 Απλουστευμένο Block Diagram των Κλάσεων της Εφαρμογής

Στις ακόλουθες υποδιαίρεσεις του παρόντος υποκεφαλαίου ακολουθεί περαιτέρω ανάλυση των κλάσεων που παρουσιάστηκαν συνοπτικά παραπάνω.

5.3.1 Κλάση User

1. Η κλάση αυτή μοντελοποιεί τον χρήστη της εφαρμογής και μέσω των πεδίων της περιγράφει πληροφορίες και χαρακτηριστικά του.
2. Η ίδια κλάση μοντελοποιεί τόσο το διαχειριστή όσο και τον απλό χρήστη. Ένα κατάλληλο πεδίο τους διαφοροποιεί, υποδηλώνοντας τα δικαιώματα του καθενός.
3. Η κλάση αυτή αποτελεί μια επέκταση της κλάσης **AbstractUser** η οποία είναι προκαθορισμένη στη Django και ορίζει αυτόματα ένα σύνολο πεδίων και μεθόδων. Παραλείποντας εκείνα που η εφαρμογή δεν χρησιμοποιεί, τα πεδία που ορίζονται από την AbstractUser είναι τα εξής:

Πεδίο	Περιγραφή
id	Το μοναδικό αναγνωριστικό του χρήστη
username	Το ψευδώνυμο του χρήστη
password	Ο κωδικός του χρήστη
first_name	Το όνομα του χρήστη
last_name	Το επίθετο του χρήστη
email	Η ηλ. διεύθυνση του χρήστη
is_active	Πεδίο που υποδηλώνει αν ένας χρήστης είναι ενεργός ή όχι. Ένας Χρήστης θεωρείται ενεργός όταν αυτός έχει επαληθεύσει το email του
is_superuser	Πεδίο που δηλώνει αν ο χρήστης έχει δικαιώματα διαχειριστή στην εφαρμογή
date_joined	Η ημερομηνία που ο χρήστης εγγράφηκε στο σύστημα
last_login	Η τελευταία ημερομηνία που ο χρήστης συνδέθηκε στην εφαρμογή. Η τιμή αυτή είναι κενή αν ο χρήστης έχει εγγραφεί αλλά δεν έχει συνδεθεί

Πίνακας 5.1 Πεδία Κλάσης User κληρονομημένα από AbstractUser

Πέρα από τα προκαθορισμένα πεδία και μεθόδους της AbstractUser, το μοντέλο User επεκτάθηκε με τα ακόλουθα πεδία:

Πεδίο	Περιγραφή
gender	Υποδηλώνει το φύλο του χρήστη, λαμβάνοντας τις τιμές M για άνδρα και F για γυναίκα
date_of_birth	Η ημερομηνία γέννησης του χρήστη

Πίνακας 5.2 Επιπλέον Πεδία Κλάσης User ως Επέκταση AbstractUser

4. Παραλείποντας εκείνες που η εφαρμογή δεν χρησιμοποιεί, οι μέθοδοι που ορίζονται από την `AbstractUser` είναι οι εξής:

Μέθοδος	Περιγραφή
<code>get_username ()</code>	Επιστρέφει το Ψευδώνυμο του Χρήστη
<code>is_authenticated ()</code>	Ελέγχει αν ο χρήστης είναι εγγεγραμμένος
<code>check_password (given_password)</code>	Ελέγχει αν η παράμετρος <code>given_password</code> είναι ο κωδικός του χρήστη
<code>set_password (new_password)</code>	Τροποποιεί τον κωδικό του χρήστη και του αναθέτει την τιμή της παραμέτρου <code>new_password</code>
<code>email_user (subject, message, from_email)</code>	Αποστέλλει ηλ. μήνυμα στο χρήστη έχοντας ως τίτλο την τιμή <code>subject</code> , ως κύριο κείμενο την τιμή <code>message</code> και ως διεύθυνση email αποστολέα την τιμή <code>from_email</code> . Η διεύθυνση αποστολής ταυτίζεται με την τιμή του πεδίου <code>email</code> του χρήστη, το οποίο δεν είναι ποτέ κενό

Πίνακας 5.3 Μέθοδοι Κλάσης `User` κληρονομημένοι από `AbstractUser`

Πριν γίνει μετάβαση στην επόμενη κλάση, αξίζει να τονιστεί η σημασία μερικών πεδίων. Το πεδίο `date_joined` χρησιμοποιείται τόσο για έλεγχο όσο και για αρχικοποίηση διαστημάτων ημερομηνιών στην παρουσίαση αναλυτικών. Το πεδίο `last_login` βοηθάει στην ανακάλυψη των χρηστών που δεν έχουν συνδεθεί ξανά στην υπηρεσία. Αυτό με τη σειρά του καθορίζει την εμφάνιση ενός βοηθητικού tutorial για τον αρχάριο χρήστη. Η μέθοδος `email_user` χρησιμοποιείται τόσο για την λειτουργία «forget password», όσο και για την αποστολή μηνύματος επιβεβαίωσης ηλ. διεύθυνσης.

5.3.2 Κλάση `Activity`

1. Η κλάση αυτή ορίζει ένα σύνολο διαθέσιμων δραστηριοτήτων του συστήματος.
2. Οι δραστηριότητες δεν είναι επεκτάσιμες, συνεπώς ο χρήστης δεν μπορεί να ορίσει τις δικές του αλλά είναι αναγκασμένος να επιλέγει εκ των εννεήντα πέντε (95) υπάρχουσών. Αυτές αντικατοπτρίζουν τις συνήθειες δραστηριότητες των ανθρώπων με βάση τόσο την ιστοσελίδα [Wikipedia](https://en.wikipedia.org/?title=Activities_of_daily_living)¹⁵ και [Bloomberg](http://www.bloomberg.com/visual-data/best-and-worst//most-time-spent-daily-activities-1)¹⁶, όσο και έρευνες¹⁷ στις Ηνωμένες Πολιτείες Επιπλέον, στο σύστημα εισήχθησαν και άλλες μη-συνήθειες αθλητικές δραστηριότητες, έτσι ώστε να είναι δυνατή η υποστήριξη υπηρεσιών όπως

¹⁵ https://en.wikipedia.org/?title=Activities_of_daily_living

¹⁶ <http://www.bloomberg.com/visual-data/best-and-worst//most-time-spent-daily-activities-1>

¹⁷ <http://www.bls.gov/news.release/pdf/atus.pdf>

το RunKeeper για το συγχρονισμό των δραστηριοτήτων του χρήστη. Οι συνολικές δραστηριότητες είναι εμφανείς στον παρακάτω πίνακα:

Accounting	Clubbing	Freshing-up / Make-up	On the Road	Relaxing	Swimming
Arc Trainer	Cooking	Grocery Shopping	Other	Roadtripping	TV/Movie Watching
At Dinner / Dining	Core Strengthening	Grooming	Out for a Coffee	Rowing	Talking / Phoning
At School	Cross-country Skiing	Group Workout	Out for a Drink	Running	Teaching / Tutoring
At University	CrossFit	Hang Gliding	Out with Friends	Shopping	Toileting
At a Meeting	Cycling	Having Sex	Partying	Sick	Training
At the Beach	Dance	Hiking	Pilates	Skating	Travelling
At the Cinema	Dog Walking	Homework	Playing Basketball	Sleeping	Vacation / Holidays
At the Doctor	Downhill Skiing	Hunting	Playing Boardgames	Snowboarding	Visiting Relatives
Barre	Dressing	In Hospital	Playing Football	Social Media	Walking
Bathing/ Showering	Driving	Instrument Playing	Playing Poker	Spinning	Wheelchair
Bootcamp	Eating	Internet Surfing	Playing Videogames	Sports	Working
Boxing / MMA	Elliptical	Meditation	Praying	Stairmaster / Stepwell	Writing / Typing
Care of others	Financial Management	Mountain Biking	Procrastinating	Strength Training	Yoga
Circuit Training	Fishing	Music Listening	Public Transport	Studying	Zumba
Cleaning	Flying	Nordic Walking	Reading	Sunbathing	

Πίνακας 5.4 Υποστηριζόμενες Δραστηριότητες Συστήματος

- Κάθε δραστηριότητα ανήκει αυτόματα σε μια κατηγορία. Οι κατηγορίες είναι συνολικά έξι (6) και προέρχονται από τη βέλτιστη κατηγοριοποίηση των δραστηριοτήτων που ορίστηκαν προηγουμένως. Καθεμία είναι μια δυάδα (τούπλα) ενός χρώματος απεικόνισης και ενός ονόματος κατηγορίας και άρα κάθε

δραστηριότητα απεικονίζεται με τη χρήση του χρώματος της κατηγορίας στην οποία ανήκει. Οι συνδυασμοί είναι οι εξής:

Category	Color
Transportation	Purple
Communication / Socializing	Blue
Sports / Fitness	Light Green
Fun / Leisure / Hobbies	Orange
Responsibilities	Dark Red
Self-Care / Everyday Needs	Black

Πίνακας 5.5 Κατηγορίες Δραστηριοτήτων και Αντιστοίχιση Χρωμάτων

- Καθένα εκ των ενενήντα πέντε (95) αντικειμένων της κλάσης αντιστοιχίζεται σε ένα μοναδικό εικονίδιο το οποίο χρησιμοποιείται στην απεικόνιση πλέγματος που περιγράφηκε στο προηγούμενο κεφάλαιο.
- Τα πεδία που περιλαμβάνει η συγκεκριμένη κλάση είναι τα ακόλουθα:

Πεδίο	Περιγραφή
Activity_id	Το μοναδικό αναγνωριστικό της κάθε δραστηριότητας
Activity_name	Το όνομα της δραστηριότητας, όπως περιγράφεται στον παραπάνω πίνακα
Description	Σύντομη περιγραφή της δραστηριότητας, που βοηθάει το χρήστη να κατανοήσει την ενέργεια που περιγράφεται μέσω του τίτλου της δραστηριότητας
Icon_classname	Το όνομα του εικονιδίου που αντιστοιχίζεται στη δραστηριότητα αυτή
category	Τύπλα κατηγορίας και χρώματος, όπως περιγράφηκε παραπάνω

Πίνακας 5.6 Πεδία Κλάσης Activity

5.3.3 Κλάση Performs

- Η κύρια κλάση για την αναπαράσταση της εκτέλεσης μιας δραστηριότητας. Συνδέει το αντικείμενο User που εκτελεί μια ενέργεια με το αντικείμενο Activity που περιγράφει την ενέργεια αυτή, αποτυπώνοντας παράλληλα επιπλέον χαρακτηριστικά που περιγράφουν το πλαίσιο της ενέργειας αυτής.
- Τα χαρακτηριστικά της κλάσης που αναφέρονται στους συμμετέχοντες και στα αντικείμενα, είναι αυτά που καθορίζουν τα αντικείμενα και τους φίλους του εκάστοτε χρήστη.

- Τα αντικείμενα της κλάσης αυτής προέρχονται τόσο από την άμεση εισαγωγή δραστηριοτήτων στο Activity Tracker, όσο και από το συγχρονισμό τρίτων εφαρμογών/υπηρεσιών με αυτό.
- Τα πεδία που χαρακτηρίζουν την κλάση είναι τα εξής:

Πεδίο	Περιγραφή
performs_id	Το μοναδικό αναγνωριστικό κάθε αντικείμενου κλάσης Performs
user	Το αντικείμενο κλάσης User που εκτέλεσε τη δραστηριότητα
activity	Το όνομα της δραστηριότητας, όπως περιγράφηκε στον παραπάνω πίνακα
using	Τα αντικείμενα που χρησιμοποιήθηκαν. Συνδέει αντικείμενα της κλάσης Object με το συγκεκριμένο αντικείμενο της κλάσης Performs
result	Περιγράφει αλφαριθμητικά το αποτέλεσμα που επέφερε η εκτέλεση της δραστηριότητας
friends	Οι συμμετέχοντες της δραστηριότητας. Συνδέει αντικείμενα της κλάσης Friend με το συγκεκριμένο αντικείμενο της κλάσης Performs
start_date	Η ημερομηνία έναρξης της παρούσας δραστηριότητας
end_date	Η ημερομηνία λήξης της παρούσας δραστηριότητας
goal	Ο στόχος που συνδέεται με την παρούσα δραστηριότητα
goal_status	Η έκβαση του θεσπισμένου στόχου, ορισμένη με τη μορφή αποτυχίας, σε εξέλιξη ή επιτυχίας
location_address	Η αλφαριθμητική διεύθυνση στην οποία λαμβάνει χώρα η δραστηριότητα
location_lat	Το γεωγραφικό πλάτος που περιγράφει την τοποθεσία της δραστηριότητας
location_lng	Το γεωγραφικό μήκος που περιγράφει την τοποθεσία της δραστηριότητας

Πίνακας 5.7 Πεδία Κλάσης Performs

- Οι μέθοδοι που χαρακτηρίζουν την κλάση αυτή είναι οι ακόλουθες:

Μέθοδος	Περιγραφή
correct_time	Ελέγχει αν τα χρονικά όρια έναρξης και λήξης δραστηριότητας είναι λογικά σωστά
duration	Επιστρέφει σε μέρες, ώρες και λεπτά τη διάρκεια της δραστηριότητας
displayable_date	Επιστρέφει συμβολοσειρά που παρουσιάζει σε κατάλληλη μορφή τη χρονική διάρκεια της δραστηριότητας

Πίνακας 5.8 Μέθοδοι Κλάσης Performs

5.3.4 Κλάση Places

1. Η κλάση αυτή ορίζει τις προσωπικές τοποθεσίες κάθε χρήστη όπως κατοικία, χώρος εργασίας, κ.ά.
2. Χρησιμοποιείται αποκλειστικά για την παραγωγή καλύτερων αναλυτικών στοιχείων. Η βασική της χρήση είναι ο έλεγχος των δραστηριοτήτων που λαμβάνουν χώρα σε επιλεγμένη απόσταση (ακτίνα) από τις τοποθεσίες αυτές.
3. Χαρακτηρίζεται από τα ακόλουθα πεδία:

Πεδίο	Περιγραφή
user	Το αντικείμενο κλάσης User που συνδέεται με την τοποθεσία αυτή
place_id	Το μοναδικό αναγνωριστικό της τοποθεσίας
place_name	Το ψευδώνυμο της τοποθεσίας
place_address	Η αλφαριθμητική διεύθυνση που αντιστοιχεί στην τοποθεσία αυτή
place_lat	Το γεωγραφικό πλάτος που αντιστοιχεί στη συγκεκριμένη τοποθεσία
place_lng	Το γεωγραφικό μήκος που αντιστοιχεί στη συγκεκριμένη τοποθεσία

Πίνακας 5.9 Πεδία Κλάσης Places

5.3.5 Κλάση UserVerification

1. Η κλάση αυτή χρησιμοποιείται αποκλειστικά για την επαλήθευση της διεύθυνσης email ενός χρήστη.
2. Συνδέει ένα χρήστη με ένα μοναδικό σύνδεσμο (unique url) ο οποίος αποτελεί την προσωπική του διεύθυνση επαλήθευσης.
3. Όταν ο χρήστης επισκεφτεί τη διεύθυνση, διαγράφεται το αντίστοιχο αντικείμενο της κλάσης UserVerification, συνεπώς η κλάση αυτή έχει αντικείμενα μόνο όσο υπάρχουν μη-επικυρωμένοι χρήστης στην εφαρμογή.
4. Αποτελείται από τα ακόλουθα πεδία:

Πεδίο	Περιγραφή
verification_id	Το μοναδικό αναγνωριστικό του εκάστοτε αντικειμένου της κλάσης αυτής
User	Το αντικείμενο κλάσης User που συνδέεται με τον μοναδικό σύνδεσμο
Verification_code	Το μεταβλητό κομμάτι του συνολικού συνδέσμου επαλήθευσης που αντιστοιχεί στο χρήστη του πεδίου User

Πίνακας 5.10 Πεδία Κλάσης UserVerification

5.3.6 Κλάση UserSocialAuth

1. Η κλάση αυτή χρησιμοποιείται για τη σύνδεση ενός χρήστη με μια υπηρεσία.
2. Συνδέει τα αντικείμενα τύπου User με τα προκαθορισμένα ονόματα παρόχων/υπηρεσιών που υποστηρίζει το σύστημα.
3. Χρησιμοποιείται για να αποθηκεύει τις απαραίτητες πληροφορίες που απαιτούνται για τη σύνδεση όπως κρυφά κλειδιά, χρόνος παύσης ισχύος των κλειδιών κ.ά.
4. Στο συγχρονισμό δραστηριοτήτων χρησιμοποιείται για την αποθήκευση της τελευταίας στιγμής συγχρονισμού με την υπηρεσία αυτή.
5. Αποτελείται από τα ακόλουθα πεδία:

Πεδίο	Περιγραφή
user_social_id	Το μοναδικό αναγνωριστικό των αντικειμένων της κλάσης αυτής
user	Το αντικείμενο κλάσης User που συνδέεται με τον πάροχο/υπηρεσία
provider	Η υπηρεσία/πάροχος με την οποία συνδέεται ο χρήστης
uid	Το μοναδικό αναγνωριστικό που αντιστοιχεί στο χρήστη, όπως αυτό περιγράφεται στην κλάση User
extra_data	Πεδίο υπεύθυνο για την αποθήκευση πλειάδας δεδομένων σε κωδικοποίηση JSON. Λόγω της τελευταίας γίνεται εύκολη η προσθήκη ή διαγραφή πληροφορίας

Πίνακας 5.11 Πεδία Κλάσης UserSocialAuth

6. Οι μέθοδοι που χαρακτηρίζουν την κλάση αυτή είναι οι ακόλουθες:

Μέθοδος	Περιγραφή
get_social_auth (cls, provider, uid)	Επιστρέφει το αντικείμενο της παρούσας κλάσης (cls) με βάση τις τιμές provider και uid, όπως αυτές περιγράφηκαν παραπάνω
User_model (cls)	Επιστρέφει την κλάση/μοντέλο που περιγράφει το χρήστη. Στην παρούσα εφαρμογή είναι η User, ωστόσο σε κάποια επέκτασή της αυτή θα μπορούσε να διαφέρει. Χρησιμοποιείται από άλλες συναρτήσεις για τη δημιουργία αντικειμένου χρήστη, όταν αυτός εγγράφεται στο σύστημα με μια υπηρεσία
Username_max_length (cls)	Με βάση την κλάση που περιγράφει το χρήστη, επιστρέφεται το μέγιστο αλφαριθμητικό μήκος που υποστηρίζει. Η μέθοδος αυτή χρησιμοποιείται από άλλες συναρτήσεις για το σκοπό της δημιουργίας ψευδώνυμου χρήστη όταν αυτός εγγράφεται μέσω υπηρεσιών όπως Facebook ή Google

Πίνακας 5.12 Μέθοδοι Κλάσης UserSocialAuth

5.3.7 Κλάση PerformsProviderInfo

1. Η κλάση αυτή χρησιμοποιείται για τις δραστηριότητες που προέρχονται από άλλες υπηρεσίες και δεν έχουν εισαχθεί στην εφαρμογή άμεσα από το χρήστη.
2. Ο κύριος ρόλος της κλάσης είναι αφενός η αναγνώριση και αφετέρου η αντιστοίχιση των δραστηριοτήτων αυτών με τις αντίστοιχές τους στην υπηρεσία.
3. Τελικός στόχος είναι η δυνατότητα τροποποίησης μιας δραστηριότητας όταν η αντίστοιχή της στην υπηρεσία (από την οποία προήλθε) τροποποιηθεί από το χρήστη. Δίχως την κλάση αυτή, κάθε φορά που ο χρήστης θα τροποποιούσε μια παλαιά του δραστηριότητα στην υπηρεσία, το Activity Tracker δεν θα μπορούσε να γνωρίζει ποια δραστηριότητα στη βάση δεδομένων χρήζει αντίστοιχης τροποποίησης.
4. Βοηθάει στην απουσία εισαγωγής της ίδιας δραστηριότητας παραπάνω από μια (1) φορά.
5. Περιγράφεται από τα ακόλουθα πεδία:

Πεδίο	Περιγραφή
instance	Το αντικείμενο της κλάσης Performs το οποίο δημιουργήθηκε από συγχρονισμό με μια υπηρεσία
provider	Η υπηρεσία από την οποία προέρχεται η δραστηριότητα
provider_instance_id	Το μοναδικό αναγνωριστικό της δραστηριότητας, όπως αυτό ορίζεται στην υπηρεσία. Επειδή κάθε υπηρεσία χρησιμοποιεί διαφορετική υλοποίηση μοναδικών αναγνωριστικών, το πεδίο αυτό είναι αλφαριθμητικό (string) ώστε να είναι σε θέση να αποτυπώσει κάθε υλοποίηση

Πίνακας 5.13 Πεδία Κλάσης PerformsProviderInfo

5.3.8 Κλάση Friend

1. Η κλάση αυτή μοντελοποιεί τους φίλους του χρήστη.
2. Οι φίλοι αυτοί ορίζονται άμεσα από τους συμμετέχοντες στις δραστηριότητες.
3. Στα πλαίσια της διπλωματικής δεν μοντελοποιούνται ως άλλοι χρήστες της εφαρμογής, αλλά απλά ως «ονόματα».
4. Χρησιμοποιούνται στη λειτουργία αυτόματης συμπλήρωσης (συμμετεχόντων) κατά τη λειτουργία εισαγωγής και τροποποίησης δραστηριοτήτων.
5. Χρησιμοποιούνται στην ανάλυση του Activity Tracker προσφέροντας κατάλληλα διαγράμματα και στατιστικά πλήρους εποπτείας των ομαδικών του δραστηριοτήτων. Παραδείγματα είναι ο χρόνος που ο χρήστης δαπανά με φίλους, οι κυρίαρχες δραστηριότητες με φίλους κ.ά.
6. Τα πεδία της κλάσης αυτής είναι τα ακόλουθα:

Πεδίο	Περιγραφή
friends_id	Το μοναδικό αναγνωριστικό του εκάστοτε φίλου
friend_name	Το όνομα (μικρό ή πλήρες) που χαρακτηρίζει το φίλο, όπως αυτό εισήχθη από το χρήστη στη δραστηριότητα
friend_of_user	Το αντικείμενο κλάσης User που σχετίζεται φιλικά με το αντικείμενο κλάσης Friend

Πίνακας 5.14 Πεδία Κλάσης Friend

5.3.9 Κλάση Object

1. Η κλάση αυτή μοντελοποιεί τα αντικείμενα που μετέχουν σε μια δραστηριότητα και έχει αντίστοιχο ρόλο και χρήση με την κλάση Friend.
2. Όπως και στην κλάση Friend, τα αντικείμενα της κλάσης αυτής παράγονται άμεσα από τα αντικείμενα που εισάγει ο χρήστης στις δραστηριότητες.
3. Χρησιμοποιούν στην αυτόματη συμπλήρωση (αντικειμένων) κατά τη διάρκεια προσθήκης τροποποίησης μιας δραστηριότητας.
4. Χρησιμοποιούνται στην ανάλυση του Activity Tracker προσφέροντας στο χρήστη κατάλληλα διαγράμματα και στατιστικά, μέσω των οποίων έχει μια πλήρη εποπτεία των αντικειμένων και λοιπών υλικών οντοτήτων που χρησιμοποιεί στις δραστηριότητές του. Παραδείγματα ανάλυσης είναι ο οι φορές που ο χρήστης χρησιμοποιεί αντικείμενα, οι κυρίαρχες δραστηριότητες που πραγματοποιούνται με χρήση αντικειμένων, το αντικείμενο που χρησιμοποιεί συχνότερα στις δραστηριότητές του κ.ά.
5. Τα πεδία που χαρακτηρίζουν την κλάση αυτή είναι τα εξής:

Πεδίο	Περιγραφή
object_id	Το μοναδικό αναγνωριστικό κάθε αντικειμένου στην εφαρμογή
object_name	Το όνομα ή ψευδώνυμο ενός αντικειμένου όπως αυτό εισήχθη στο σύστημα μέσω της εισαγωγής ή τροποποίησης δραστηριότητας (δηλ. μέσω της δημιουργίας ή τροποποίησης ενός αντικειμένου κλάσης Performs)
object_of_user	Ο χρήστης, αντικείμενο της κλάσης User, που έκανε χρήση του αντικειμένου της κλάσης Object

Πίνακας 5.15 Πεδία Κλάσης Object

Όπως εύκολα παρατηρεί κανείς, οι κλάσεις **Friend** και **Object** έχουν ακριβώς αντίστοιχη συμπεριφορά. Η πρώτη αναφέρεται στις έμβιες οντότητες που μετέχουν στο πλαίσιο της εκάστοτε δραστηριότητας, ενώ η δεύτερη στις άβιες.

5.3.10 Κλάση `hasSuperActivity`

1. Η κλάση αυτή χρησιμοποιείται για τη συσχέτιση των προεπιλεγμένων δραστηριοτήτων του συστήματος, δηλαδή των αντικειμένων της κλάσης `Activity`.
2. Δημιουργεί μια σχέση ιεραρχίας δραστηριοτήτων με τη μορφή κατευθυνόμενου γράφου.
3. Ορίζει διαφορετικά επίπεδα – υπό μορφή βάθους – στα οποία αναλύεται μια δραστηριότητα, προσφέροντας τη δυνατότητα κατανόησης του βαθμού στον οποίο δυο δραστηριότητες συνδέονται ασθενώς ή άρρηκτα.
4. Αποτελεί μέρος της θεωρίας των δραστηριοτήτων (`Activity Theory`).
5. Απαρτίζεται από τα ακόλουθα πεδία:

Πεδίο	Περιγραφή
<code>has_superactivity_id</code>	Το μοναδικό αναγνωριστικό του εκάστοτε δεσμού, αντικειμένου της παρούσας κλάσης
<code>superactivity</code>	Το αντικείμενο της κλάσης <code>Activity</code> που αποτελεί την υπερδραστηριότητα
<code>subactivity</code>	Το αντικείμενο της κλάσης <code>Activity</code> που αποτελεί την υποδραστηριότητα.

Πίνακας 5.16 Πεδία Κλάσης `hasSuperActivity`

Η δραστηριότητα που περιγράφεται από το πεδίο `subactivity` βρίσκεται σε χαμηλότερη ιεραρχία, αποτελώντας υποδραστηριότητα για την αντίστοιχη που περιγράφεται μέσω του πεδίου `superactivity`. Στην παρούσα διπλωματική, η κλάση `hasSuperActivity` ορίστηκε αλλά δεν χρησιμοποιήθηκε στην ανάλυση των δεδομένων λόγω αυξημένης πολυπλοκότητας στην κατασκευή κατάλληλων αλγορίθμων αποσαφήνισης συσχετίσεων. Η ύπαρξή της, ωστόσο, είναι καθοριστική για την μελλοντική επέκταση – εξέλιξη της ανάλυσης και της ίδιας της εφαρμογής.

5.4 Αρχιτεκτονική Συστήματος Πλαισίου Django

Έχοντας ολοκληρώσει την ανάλυση κλάσεων, ακολουθεί η ανάλυση του βασικού σκελετού οργάνωσης του προγραμματιστικού κώδικα της διπλωματικής, με χρήση του πλαισίου (`framework`) Django. Συγκεκριμένα, η αρχιτεκτονική κλάσεων του κεφαλαίου (5.3) κατέστησε κατανοητή τη δομή της συνιστώσας των μοντέλων (`models`) του Django, συνεπώς στο κεφάλαιο αυτό θα πραγματοποιηθεί μια ανάλυση των όψεων (`views`), συνδέσμων (`urls`) και βασικών οθονών (`templates`) του ίδιου πλαισίου.

Η `Mock Up` σχεδίαση του κεφαλαίου (5.1) σε συνδυασμό με τις λειτουργικές απαιτήσεις (4.1) και τις αναγκαίες λειτουργίες του συστήματος (4.3), οδήγησε στη δημιουργία των οθονών

(templates) από τις οποίες θα αποτελείται το σύστημα. Τα ακόλουθα υποκεφάλαια αναλύουν τις σελίδες της εφαρμογής σε επίπεδο client και server. Είναι σημαντικό να τονιστεί ότι κάθε ασύγχρονη ανταλλαγή δεδομένων προϋποθέτει την ύπαρξη ζευγαριών όψης εξυπηρετητή – μονάδας πελάτη. Συνεπώς, πολλές φορές για κάθε όψη του συστήματος ενδέχεται να υπάρχει αντίστοιχη συνάρτηση ή μονάδα χειρισμού στον client που να συσχετίζεται με αυτή. Τέλος, τα templates δευτερευούσης σημασίας τα οποία περιέχονται εντός άλλων templates (π.χ. πίνακας με υποστηριζόμενες δραστηριότητες συστήματος) δεν αναλύονται ξεχωριστά, καθώς δεν αποτελούν βασική συνιστώσα της εφαρμογής. Οι οθόνες που θα αναλυθούν είναι οι ακόλουθες:

1. **Login** – Οθόνη εισόδου χρήστη στο σύστημα.
2. **Forgot Password** – Οθόνη απώλειας κωδικού πρόσβασης.
3. **Register** – Οθόνη χειρωνακτικής εγγραφής στο σύστημα.
4. **Index** – Βασική οθόνη εποπτείας και επεξεργασίας δραστηριοτήτων.
5. **Goals** – Οθόνη διαχείρισης θεσπισμένων στόχων.
6. **Timeline** – Οθόνη παρουσίασης χρονογραφήματος δραστηριοτήτων.
7. **Settings** – Οθόνη επεξεργασίας πληροφοριών χρήστη και συγχρονισμού δραστηριοτήτων.
8. **Friend Analytics** – Οθόνη παρουσίασης αναλυτικών κοινωνικού πλαισίου.
9. **Object Analytics** – Οθόνη παρουσίασης αναλυτικών με επίκεντρο τη χρήση αντικειμένων.
10. **Goal Analytics** – Οθόνη ανάλυσης θεσπισμένων στόχων.
11. **Places Analytics** – Οθόνη γεωγραφικής ανάλυσης δραστηριοτήτων.
12. **Activity Analytics** – Οθόνη γενικής ανάλυσης πορείας δραστηριοτήτων.

5.4.1 Οθόνη Login

1. Η σελίδα αυτή είναι υπεύθυνη για τη σύνδεση του χρήστη στο σύστημα.
2. Δίνει τη δυνατότητα τόσο χειρωνακτικής σύνδεσης μέσω πεδίων, όσο και σύνδεσης μέσω υπηρεσιών Facebook και Google.
3. Προσφέρει στο χρήστη πρόσβαση στη σελίδα χειρωνακτικής εγγραφής και στη σελίδα απώλειας κωδικού (forgot password)
4. Οι συναρτήσεις χειρισμού που συνδέονται με την οθόνη αυτή παρουσιάζονται παρακάτω. Αυτές έχουν υλοποιηθεί με χρήση Python και συνθέτουν τα συσχετιζόμενα με την οθόνη views.

Όνομα και Σύνδεσμος Όψης	Χρήση	Περιγραφή
login (activitytracker/login)	Υπεύθυνη για τη φόρτωση της οθόνης και τη σύνδεση του χρήστη στην εφαρμογή	Όταν ο χρήστης επισκεφτεί τη σελίδα αυτή τότε η όψη αυτή φορτώνει την οθόνη (επιστρέφει δηλαδή ένα template ως απάντηση). Αν ωστόσο πραγματοποιηθεί αποστολή δεδομένων τότε επιχειρείται σύνδεση και η όψη αυτή ελέγχει τα στοιχεία και την κατάσταση του χρήστη (επαληθευμένος ή όχι). Έπειτα είτε επιστρέφει μήνυμα σφάλματος είτε εισάγει το χρήστη στο σύστημα μεταφέροντάς τον στην κατάλληλη ιστοσελίδα.
social_login (activitytracker/social_login)	Υπεύθυνη για τη σύνδεση (και ενδεχομένως εγγραφή) του χρήστη στο σύστημα μέσω τρίτων υπηρεσιών	Η όψη αυτή, σε συνδυασμό με άλλες συναρτήσεις (και ενδεχομένως όψεις εξωτερικής βιβλιοθήκης ¹⁸ που συμμετέχει στη διαδικασία πραγματοποίησης σύνδεσης) είναι υπεύθυνη για τη σύνδεση και την ανακατεύθυνση του χρήστη ανάλογα με την περίπτωση σφάλματος ή επιτυχούς διαδικασίας

Πίνακας 5.17 Όψεις Συσχετιζόμενες με την Οθόνη Login

¹⁸ Για την πραγματοποίηση της σύνδεσης με υπηρεσία χρησιμοποιήθηκαν και οι εξωτερικές όψεις της βιβλιοθήκης Python-Social-Auth (<https://github.com/omab/python-social-auth>)

- Κατ' αντιστοιχία με τα παραπάνω views, στην πλευρά του client υπάρχουν αντίστοιχες μονάδες ή συναρτήσεις χειρισμού που παρουσιάζονται παρακάτω. Αυτές δεν είναι πάντοτε συναρτήσεις, αλλά ενδέχεται να είναι και απλοί jQuery ή JavaScript χειριστές (handlers) δίχως κάποια συγκεκριμένη ονομασία.

Όνομα Μονάδας/Συναρτησης	Χρήση	Περιγραφή
LoginHandler	Μεταφορά εισακτέων δεδομένων προς έλεγχο και παρουσίαση σφαλμάτων	Όταν ο χρήστης επιλέξει αποστολή πεδίων, η μονάδα αυτή αποστέλλει ασύγχρονα τα δεδομένα και αναμένει απάντηση από το server για να καθορίσει την παρουσίαση σφαλμάτων
SocialLoginHandler	Εκκίνηση διαδικασίας σύνδεσης με υπηρεσίες	Είναι υπεύθυνη για τη δημιουργία και κλείσιμο απαραίτητων παραθύρων για εξουσιοδότηση της εφαρμογής, όπως και για τις απαραίτητες ανακατευθύνσεις του χρήστη

Πίνακας 5.18 Μονάδες και Συναρτήσεις Χειρισμού Πελάτη Συσχετιζόμενες με την Οθόνη Login

5.4.2 Οθόνη Forgot Password

- Η σελίδα αυτή προσφέρει τη δυνατότητα σύνδεσης στο σύστημα όταν ο χρήστης έχει ξεχάσει τον κωδικό πρόσβασής του.
- Συμπληρώνοντας το όνομα χρήστη, το σύστημα αναθέτει και αποστέλλει έναν νέο τυχαίο κωδικό στο email που συνδέεται με τον αντίστοιχο λογαριασμό. Ο χρήστης ενημερώνεται για την επιτυχία ή αποτυχία έκβασης της λειτουργίας.
- Με την οθόνη αυτή συνδέεται μόνο η ακόλουθη όψη:

Όνομα και Σύνδεσμος Όψης	Χρήση	Περιγραφή
passwordForget (activitytracker/account/passforgot)	Φορτώνει την οθόνη, αναθέτει στο χρήστη νέο τυχαίο κωδικό και του τον αποστέλλει με email	Η όψη αυτή εξετάζει αν πραγματοποιήθηκε αποστολή δεδομένων. Αν όχι τότε φορτώνει απλά την οθόνη, διαφορετικά ελέγχει την εγκυρότητα ονόματος χρήστη που συμπληρώθηκε. Αν το όνομα είναι έγκυρο, παράγει τυχαίο κωδικό,

τροποποιεί τον κωδικό του χρήστη, αποστέλλει ηλεκτρονικό μήνυμα με τον νέο κωδικό και επιστρέφει μήνυμα ενημέρωσης έκβασης της λειτουργίας. Σε αντίθετη περίπτωση επιστρέφει μήνυμα αποτυχίας στο χρήστη

Πίνακας 5.19 Όψεις Συσχετιζόμενες με την Οθόνη Forgot Password

4. Σε αντιστοιχία με την όψη αυτή, στην πλευρά του client υπάρχει η ακόλουθη μονάδα χειρισμού:

Όνομα Μονάδας/Συναρτησης	Χρήση	Περιγραφή
PasswordForgetHandler	Αποστολή δεδομένων στην όψη και παρουσίαση μηνυμάτων	Η μονάδα αυτή στέλνει στην όψη, μέσω A.J.A.X., το συμπληρωμένο όνομα χρήστη και ενημερώνει το χρήστη τόσο για τυχόν σφάλματα, όσο και για την επιτυχή έκβαση λειτουργίας ανάθεσης νέου κωδικού

Πίνακας 5.20 Μονάδες και Συναρτήσεις Χειρισμού Πελάτη Συσχετιζόμενες με την Οθόνη Forgot Password

5.4.3 Οθόνη Χειρονακτικής Εγγραφής στο Σύστημα

1. Η οθόνη αυτή είναι υπεύθυνη για την εγγραφή του χρήστη στο σύστημα μέσω συμπλήρωσης πεδίων.
2. Περιλαμβάνει μια σειρά από υποχρεωτικά πεδία προς συμπλήρωση μέσω των οποίων ο χρήστης μπορεί να εισαχθεί στο σύστημα.
3. Η επιτυχής εγγραφή απαιτεί και την επαλήθευση του χρήστη, μέσω της επίσκεψης ενός συνδέσμου που αποστάλθηκε στην ηλ. διεύθυνση που συμπλήρωσε ο χρήστης κατά την εγγραφή του.
4. Η οθόνη συνδέεται με τις ακόλουθες όψεις (Django views):

Όνομα και Σύνδεσμος Όψης	Χρήση	Περιγραφή
register (activitytracker/account/register)	Υπεύθυνη για την φόρτωση της οθόνης, την εγγραφή του χρήστη και την αποστολή ηλεκτρονικού μηνύματος επαλήθευσης ταχυδρομείου	Η όψη αυτή ελέγχει αν αιτήθηκε μαζί με την αποστολή δεδομένων. Αν δεν έγινε αποστολή τότε φορτώνει το template της σελίδας, διαφορετικά πραγματοποιεί τους απαραίτητους ελέγχους και επιστρέφει μηνύματα σφάλματος ή επιτυχίας έκβασης λειτουργίας. Το μήνυμα επιτυχίας σημαίνει άμεσα την καταχώρηση των τιμών στη βάση δεδομένων, τη δημιουργία χρήστη, την παραγωγή τυχαίου συνδέσμου επαλήθευσης που αντιστοιχίζεται στο χρήστη και την αποστολή ηλ. μηνύματος με τον σύνδεσμο αυτό.
email_verification (activitytracker/account/Verification/<verification_code>)	Υπεύθυνη για την επαλήθευση του χρήστη στο σύστημα (μέσω επαλήθευσης της ηλ. διεύθυνσής του)	Ελέγχει αν ο σύνδεσμος συνδέεται με κάποιο εγγεγραμμένο χρήστη και είτε επαληθεύει το χρήστη στο σύστημα, είτε αποστέλλει μήνυμα σφάλματος στον client

Πίνακας 5.21 Όψεις Συσχετιζόμενες με την Οθόνη Register

- Από την πλευρά του χρήστη υπάρχει κατάλληλη μονάδα jQuery, που είναι υπεύθυνη για την αποστολή των δεδομένων και την ενημέρωση του χρήστη σε περίπτωση σφάλματος ή επιτυχίας. Αυτή είναι η ακόλουθη:

Όνομα Μονάδας/Συνάρτησης	Χρήση	Περιγραφή
RegisterHandler	Αποστολή συμπληρωμένων πεδίων και ενημέρωση για σφάλματα ή επιτυχή ολοκλήρωση διαδικασίας	Ομοίως με τις προηγούμενες μονάδες, όταν ο χρήστης ολοκληρώσει την εγγραφή του, η μονάδα αυτή αποστέλλει ασύγχρονα (A.J.A.X) τις τιμές των πεδίων στην αντίστοιχη όψη και παρουσιάζει τυχόν σφάλματα τιμών στο χρήστη. Διαφορετικά τον ενημερώνει για την επιτυχή εγγραφή και την αποστολή του email επαλήθευσης.

Πίνακας 5.22 Μονάδες και Συναρτήσεις Χειρισμού Πελάτη Συσχετιζόμενες με την Οθόνη Register

5.4.4 Βοηθητικές Οθόνες

1. Χρησιμοποιούνται αποκλειστικά για τις λειτουργίες επισκόπησης, τροποποίησης και διαγραφής δραστηριοτήτων.
2. Η οθόνες αυτές δεν αποτελούν επισκέψιμες (από το χρήστη) ιστοσελίδες, αλλά «καλούνται» μέσω άλλων οθονών, παρέχοντας μια σταθερή οθόνη αναπαράστασης και επεξεργασίας πληροφορίας.
3. Προσφέρουν μεταφερισιμότητα, καθώς οι βοηθητικές οθόνες είναι ανεξάρτητες της οθόνης από την οποία «κλήθηκαν». Συνεπώς, μιας και οι λειτουργίες επισκόπησης, διαγραφής και τροποποίησης πραγματοποιούνται σε παραπάνω από μια (1) σελίδες του συστήματος, διευκολύνεται η επέκτασή.
4. Φιλοξενούνται εντός ενός «καμβά», ο οποίος είναι τυπικά ένα μικρό επιπρόσθετο πλαίσιο στο προσκήνιο της εκάστοτε κύριας σελίδας (Modal).
5. Κάθε βοηθητική οθόνη είναι αποτέλεσμα της όψης Django με την οποία συνδέεται.
6. Οι βοηθητικές οθόνες είναι η `display-activity`, `display-group-activiy` και `edit-activity`. Αφορούν την επισκόπηση δραστηριότητας, επισκόπηση ομαδοποιημένης δραστηριότητας και τροποποίηση δραστηριότητας, και συνδέονται με τις Django όψεις (views) `showActivity`, `showGroupActivity` και `editActivity` αντίστοιχα. Αυτές παρουσιάζονται παρακάτω:

Όνομα και Σύνδεσμος Όψης	Χρήση	Περιγραφή
showActivity (activitytracker/activity/show-activity/<activity_id>)	Υπεύθυνη για την παροχή κατάλληλης διεπιφάνειας, η οποία παρουσιάζει αναλυτικά τις τιμές μιας δραστηριότητας στο χρήστη	Όταν ο χρήστης επιλέξει να δει αναλυτικά μια δραστηριότητα, η όψη αυτή επιστρέφει κατάλληλη οθόνη (template) με τις τιμές της δραστηριότητας
showGroupActivity (activitytracker/activity/show-group-activity/<group_id>)	Υπεύθυνη για την αναλυτική παρουσίαση των τιμών μιας ομαδοποιημένης ψευδοδραστηριότητας (πολλές δραστηριότητες ομαδοποιημένες ως μια), μέσω κατάλληλης διεπιφάνειας	Όταν ο χρήστης επιλέξει να δει αναλυτικά μια ομαδοποιημένη δραστηριότητα, η όψη αυτή επιστρέφει κατάλληλη οθόνη (template) με τις τιμές των δραστηριοτήτων που ανήκουν στην ομάδα αυτή
editActivity (activitytracker/activity/edit-activity/<activity_id>)	Υπεύθυνη για την παρουσίαση κατάλληλης διεπιφάνειας επεξεργασίας δραστηριότητας	Όταν ο χρήστης επιλέξει να τροποποιήσει μια δραστηριότητα, η όψη αυτή επιστρέφει κατάλληλη οθόνη (template) συμπλήρωσης πεδίων με αρχικοποιημένες τιμές σύμφωνα με την προς επεξεργασία δραστηριότητα

Πίνακας 5.23 Όψεις Συσχετιζόμενες με τις Βοηθητική Οθόνες

7. Στην πλευρά του πελάτη (client) υπάρχουν συναρτήσεις και μονάδες χειρισμού jQuery και JavaScript, υπεύθυνες για την αναπαράσταση των χαρτών, όπως και για τις γραφιστικές λεπτομέρειες. Αυτές παρουσιάζονται παρακάτω:

Όνομα Μονάδας/Συνάρτησης	Χρήση	Περιγραφή
ShowMapInitialize	Αρχικοποίηση χάρτη για την επισκόπηση της θέσης της δραστηριότητας	Όταν ο χρήστης αιτήσει την αναλυτική παρουσίαση μιας δραστηριότητας, η συνάρτηση αυτή κατασκευάζει ένα χάρτη με μια πινέζα (pin) στην τοποθεσία που η

		δραστηριότητα έλαβε χώρα, ρυθμίζοντας παράλληλα παραμέτρους εστίασης, οριοθέτησης κ.ά.
ShowGroupMapInitialize	Αρχικοποίηση χαρτών για την επισκόπηση της θέσης καθεμίας εκ των ομαδοποιημένων δραστηριοτήτων	Όταν ο χρήστης αιτήσει επισκόπηση μιας ομάδας δραστηριοτήτων, η συνάρτηση αυτή κατασκευάζει, για κάθε δραστηριότητα, ένα χάρτη με μια πινέζα (pin) στην τοποθεσία που αυτή έλαβε χώρα, ρυθμίζοντας παράλληλα παραμέτρους εστίασης, οριοθέτησης κ.ά.
EditMapInitialize	Αρχικοποίηση χάρτη για την παρουσίαση και εν δυνάμει τροποποίηση της θέσης της δραστηριότητας	Όταν ο χρήστης επιλέξει να τροποποιήσει μια δραστηριότητα, η συνάρτηση αυτή κατασκευάζει κατάλληλο χάρτη, του οποίου το pin μπορεί να μετακινηθεί είτε χειροκίνητα, είτε μέσω αναζήτησης νέας τοποθεσίας. Ταυτοχρόνως ρυθμίζει λεπτομέρειες όπως ο αριθμός αποτελεσμάτων αναζήτησης, ο βαθμός εστίασης, το κέντρο του χάρτη κ.ά.
GraphicsHandler	Σχεδίαση γραφικών λεπτομερειών οθόνης	Η συνάρτηση αυτή σχεδιάζει διακριτικά όρια με το χρώμα της κατηγορίας στην οποία η εκάστοτε δραστηριότητα ανήκει. Η συνάρτηση αυτή επιτελεί τον ίδιο ρόλο και στις τρεις (3) βοηθητικές οθόνες και συνεπώς αναφέρεται μια μόνο φορά

Πίνακας 5.24 Μονάδες και Συναρτήσεις Χειρισμού Πελάτη Συσχετιζόμενες με τις Βοηθητικές Οθόνες

5.4.5 Οθόνη Index

1. Η σελίδα αυτή είναι η κύρια σελίδα του συστήματος.
2. Είναι η μόνη οθόνη που επιτρέπει την εισαγωγή νέας δραστηριότητας στο σύστημα.
3. Υποστηρίζει τις λειτουργίες επισκόπησης, τροποποίησης και διαγραφής δραστηριοτήτων ενός χρήστη.
4. Προσφέρει πολλαπλή εποπτεία των δραστηριοτήτων και αποτελείται από τρία (3) βασικά μέρη:
 - Το ημερολόγιο, όπως αυτό περιγράφηκε στο κεφάλαιο (4.3.10). Μέσω αυτού εισάγονται και οπτικοποιούνται οι δραστηριότητες, ενώ ταυτόχρονα καθορίζει και τις δραστηριότητες που εμφανίζονται στην εποπτεία πλέγματος.
 - Την εποπτεία πλέγματος (Grid), όπως περιγράφηκε στο κεφάλαιο (4.3.10) με δυνατότητα αναδιάταξης και ομαδοποίησης των κοινών δραστηριοτήτων.
 - Ένα διάγραμμα μορφής πίτας στο οποίο παρουσιάζεται το μερίδιο ωρών κάθε κατηγορίας εκ των δραστηριοτήτων που εμφανίζονται στο πλέγμα και το ημερολόγιο.
5. Οι συναρτήσεις χειρισμού (υλοποιημένες σε Python) που συνδέονται με την οθόνη αυτή παρουσιάζονται παρακάτω. Πολλούς από τους παρακάτω συνδέσμους δεν τους επισκέπτεται ο χρήστης αλλά χρησιμοποιούνται για την ασύγχρονη ανταλλαγή δεδομένων ανάμεσα σε client και server:

Όνομα και Σύνδεσμος Όψης	Χρήση	Περιγραφή
<code>index</code> (activitytracker/index)	Υπεύθυνη για τη φόρτωση της ιστοσελίδας και την αρχικοποίηση τιμών	Ενεργοποιείται όταν ο χρήστης επισκέπτεται τον σύνδεσμο (url) της σελίδας αυτής
<code>addActivity</code> (activitytracker/activity/add)	Υπεύθυνο για την προσθήκη δραστηριότητας στο σύστημα	Η όψη αυτή δέχεται τις τιμές των πεδίων που συμπλήρωσε ο χρήστης και αφότου ελέγξει για τυχόν λάθη, εισάγει τη δραστηριότητα στη βάση δεδομένων
<code>showActivity</code> (activitytracker/activity/show-activity/<activity_id>)	Υπεύθυνη για την αναλυτική παρουσίαση των δεδομένων μιας δραστηριότητας στο χρήστη	Όταν ο χρήστης επιλέξει να δει αναλυτικά μια δραστηριότητα, η όψη αυτή επιστρέφει κατάλληλο template με τις τιμές της δραστηριότητας

<p><code>showGroupActivity</code> <i>(activitytracker/activity/show-group-activity/<group_id>)</i></p>	<p>Υπεύθυνη για την αναλυτική παρουσίαση των τιμών μιας ομαδοποιημένης ψευδοδραστηριότητας</p>	<p>Όταν ο χρήστης επιλέξει να δει αναλυτικά μια ομαδοποιημένη δραστηριότητα, η όψη αυτή επιστρέφει κατάλληλο <code>template</code> με τις τιμές των κοινών δραστηριοτήτων που ανήκουν στην ομάδα αυτή</p>
<p><code>editActivity</code> <i>(activitytracker/activity/edit-activity/<activity_id>)</i></p>	<p>Υπεύθυνη για την αρχικοποίηση τιμών της τροποποιούμενης δραστηριότητας</p>	<p>Όταν ο χρήστης επιλέξει να τροποποιήσει μια δραστηριότητα, η όψη αυτή επιστρέφει κατάλληλο <code>template</code> συμπλήρωσης πεδίων με αρχικοποιημένες τιμές</p>
<p><code>updateActivity</code> <i>(activitytracker/activity/update-activity)</i></p>	<p>Υπεύθυνη για την ανανέωση των τιμών της τροποποιημένης δραστηριότητας στη βάση δεδομένων</p>	<p>Όταν ο χρήστης επιλέξει την αποθήκευση των αλλαγών από το <code>template</code> του <code>editActivity</code>, τότε αποστέλλεται ασύγχρονο αίτημα από το <code>server</code> περιέχοντας τις αλλαγές που θα εισαχθούν από την όψη αυτή στη Β.Δ.</p>
<p><code>deleteActivity</code> <i>(activitytracker/activity/delete-activity)</i></p>	<p>Υπεύθυνη για την αφαίρεση επιλεγμένης δραστηριότητας από το σύστημα</p>	<p>Όταν ο χρήστης επιλέξει διαγραφή δραστηριότητας, η όψη αυτή δέχεται το μοναδικό αναγνωριστικό της προς διαγραφή δραστηριότητας και την αφαιρεί από τη Β.Δ.</p>
<p><code>listAllActivities</code> <i>(activitytracker/activity/listallactivities)</i></p>	<p>Υπεύθυνη για την παρουσίαση όλων των</p>	<p>Κατά την εισαγωγή μιας δραστηριότητας ο</p>

	ενεργειών και δραστηριοτήτων που υποστηρίζει το σύστημα, μαζί με την περιγραφή τους.	και χρήστης μπορεί να δει ένα πίνακα με όλες τις υποστηριζόμενες δραστηριότητες. Η όψη αυτή είναι υπεύθυνη για την παροχή των στοιχείων του πίνακα αυτού και την επιστροφή ενός <code>template</code> που τον παρουσιάζει
<code>eventsToJson</code> (activitytracker/index/eventstojson)	Επιστρέφει τις απαραίτητες πληροφορίες όλων των δραστηριοτήτων ενός χρήστη για την εισαγωγή στο ημερολόγιο	Κάθε φορά που υπάρχει μια αλλαγή στις δραστηριότητες, η όψη αυτή καλείται από τον <code>client</code> για να ανανεώσει τα συνολικά δεδομένα του ημερολογίου. Η όψη αυτή επιστρέφει μόνο τις απαραίτητες πληροφορίες που χρειάζεται η οντότητα του ημερολογίου
<code>displayPeriod</code> (activitytracker/index/displayperiod)	Επιστρέφει τις δραστηριότητες που ανήκουν στο διάστημα ημερολογίου επιλεχθηκε	Για την κατασκευή του πλέγματος (<code>Grid</code>), ο <code>client</code> πρέπει να γνωρίζει ποιές δραστηριότητες ανήκουν στο έπιλεγμένο ημερολογιακό διάστημα. Η όψη αυτή ενεργοποιείται κάθε φορά που αλλάζει το επιλεγμένο διάστημα στο ημερολόγιο και επιστρέφει τα αντικείμενα της κλάσης <code>Performs</code> που ανήκουν στο παραπάνω διάστημα
<code>chartDataJson</code> (activitytracker/index/chartdatajson)	Επιστρέφει τις απαραίτητες	Αθροίζει τις ώρες κάθε κατηγορίας των

	πληροφορίες κάθε δραστηριοτήτων της κατηγορίας για την όψης <code>displayPeriod</code> . σχεδίαση του αντίστοιχου Έπειτα παρέχει τα διαγράμματος αποδομένα στον <code>client</code> . καταμερισμού Προφανώς ενεργοποιείται όταν ενεργοποιείται και η <code>displayPeriod</code>
<code>getGroupedActivities</code> (activitytracker/index/getgroupedactivities)	Υπεύθυνο για την Όταν ο χρήστης ομαδοποίηση κοινών επιλέξει τη λειτουργία δραστηριοτήτων από την ομαδοποίηση η όψη απεικόνιση πλέγματος αθροίζει τις ώρες των κοινών δραστηριοτήτων και επιστρέφει ψευδοαντικείμενα της κλάσης <code>Performs</code>

Πίνακας 5.25 Όψεις Συσχετιζόμενες με την Οθόνη Index

6. Αντιστοίχως με τις όψεις στην πλευρά του server υπάρχουν συναρτήσεις και μονάδες χειρισμού στην πλευρά του client οι οποίες αλληλεπιδρούν με τις όψεις αυτές. Λόγω του μεγάλου αριθμού των πολλών δευτερευούσης σημασίας μονάδων, πολλές από αυτές συνοψίζονται σε μία (1) μονάδα χειρισμού. Αυτές είναι υλοποιημένες με χρήση `jQuery` και `JavaScript`, και αναλύονται συνοπτικά παρακάτω:

Όνομα Μονάδας/Συνάρτησης	Χρήση	Περιγραφή
<code>PlotDonutChart</code>	Υπεύθυνη για το σχεδιασμό του διαγράμματος τύπου πίτας	Η συνάρτηση αυτή καλείται από άλλες συναρτήσεις και μονάδες για τον επανασχεδιασμό του διαγράμματος, όποτε αυτό είναι απαραίτητο (όταν αλλάζουν οι δραστηριότητες)
<code>SubmitFields</code>	Υπεύθυνη για την ανάγνωση και κωδικοποίηση των εισακτέων τιμών	«Διαβάζει» και προετοιμάζει για αποστολή τις τιμές που συμπληρώνει ο χρήστης κατά τη δημιουργία ή τροποποίηση μιας δραστηριότητας

DrawActivity	Υπεύθυνη για το σχεδιασμό μιας δραστηριότητας του πλέγματος	Καλείται επαναληπτικά για όλες τις δραστηριότητες που ανήκουν στο πλέγμα και σχεδιάζει μια δραστηριότητα στην κατάλληλη θέση (γραμμή και στήλη), μαζί με την κατάλληλη απόχρωση και εικόνα που της αντιστοιχεί
DrawGroupUngroupSort	Υπεύθυνη για τον πλήρη σχεδιασμό πλέγματος ομαδοποιημένων ή/και ταξινομημένων δεδομένων	Χρησιμοποιεί την DrawActivity για το σχεδιασμό του πλέγματος όταν ο χρήστης επιλέξει ομαδοποίηση ή/και ταξινόμηση. Η συνάρτηση αυτή είναι υπεύθυνη για όλα τη διαδικασία σχεδίασης του πλέγματος. Τονίζεται, ότι το πλέγμα έχει προεπιλεγμένη αλφαβητική ταξινόμηση.
DrawGroupUngroupSort WithChart	Ομοίως με την DrawGroupUngroupSort, με τη διαφορά ότι επανασχεδιάζει και το διάγραμμα	Όταν ο χρήστης επιλέξει διαφορετική ταξινόμηση ή ομαδοποίηση, το διάγραμμα δεν χρήζει επανασχεδίασης. Ωστόσο, όταν αλλάξει η όψη του ημερολογίου τότε αλλάζουν οι δραστηριότητες και συνεπώς απαιτείται επανασχεδιασμός. Συνεπώς η συνάρτηση αυτή καλείται όταν αλλάζει η όψη του ημερολογίου, ενώ η προηγούμενή της όταν αλλάζει μόνο η ταξινόμηση και ομαδοποίηση στην ίδια όψη. Η συνάρτηση αυτή καλεί την PlotDonutChart
InitializePlugins	Υπεύθυνη για την αρχικοποίηση και ενεργοποίηση των εξωγενών βιβλιοθηκών (όπως ημερολόγιο,	Η συνάρτηση αυτή καλείται κατά τη φόρτωση της ιστοσελίδας index, αρχικοποιεί τις γραφικές οντότητες και καθορίζει τη

	αναλογικό ρολόι για επιλογή ώρας κ.λπ.)	συμπεριφορά των οντοτήτων αυτών (π.χ. κλήση όψης showActivity όταν επιλεχθεί μια δραστηριότητα από το ημερολόγιο κ.ά.)
CalendarDateRange	Υπεύθυνη για την ανάγνωση του εκάστοτε ενεργού ημερολογιακού διαστήματος	Χρησιμοποιείται για τον ορισμό των χρονικών ορίων που θα αποσταλούν στο server και θα χρησιμοποιηθούν για την εύρεση των δραστηριοτήτων που ανήκουν σε αυτό
RenderViewActivities	Υπεύθυνη για την ανανέωση του πλέγματος μέσω αλλαγής ημερολογιακής όψης.	Ενεργοποιείται κάθε φορά που αλλάζει όψη το ημερολόγιο. Αποστέλλει το διάστημα που καθορίστηκε από την CalendarDateRange, δέχεται ως απάντηση τις δραστηριότητες και τις σχεδιάζει με χρήση της DrawGroupUngroupSort WithChart
ListTheActivities	Υπεύθυνη για την δημιουργία παραθύρου με όλες τις υποστηριζόμενες δραστηριότητες	Δέχεται ως απάντηση τα δεδομένα της όψης listAllActivities και δημιουργεί το παράθυρο που απαιτείται για τη φιλοξενία του πίνακα δραστηριοτήτων
AddActivityHandler	Σύνολο μονάδων υπεύθυνο για τη λειτουργία προσθήκης δραστηριότητας	Οι βασικές αρμοδιότητες αφορούν την εμφάνιση διεπιφάνειας συμπλήρωσης πεδίων, την αποστολή δεδομένων, την κλήση της RenderViewActivities για ανανέωση δραστηριοτήτων της παρούσας όψης, την κλήση DrawGroupUngroupSort WithChart για ανανέωση πλέγματος και γραφίματος, την εξαφάνιση διεπιφάνειας

		συμπλήρωσης (τύπου Modal) και τον καθαρισμό των πεδίων και χαρτών
EditActivityHandler	Σύνολο μονάδων υπεύθυνο για τη λειτουργία τροποποίησης δραστηριότητας	Οι βασικές αρμοδιότητες αφορούν την κλήση όψης editActivity για την εμφάνιση template επεξεργασίας δεδομένων δραστηριότητας, την αποστολή δεδομένων στην όψη updateActivity, την κλήση RenderViewActivities για ανανέωση δραστηριοτήτων της παρούσας όψης, την κλήση DrawGroupUngroupSort WithChart για ανανέωση πλέγματος και γραφήματος, όπως και την εξαφάνιση της οθόνης τροποποίησης δραστηριότητας (τύπου Modal)
DisplayActivityHandler	Σύνολο μονάδων υπεύθυνο για τη λειτουργία επισκόπησης δραστηριότητας	Οι βασικές αρμοδιότητες αφορούν την κλήση της όψης showActivity για την επιστροφή του template με τα στοιχεία της δραστηριότητας και τη δυνατότητα πρόσβασης στις λειτουργίες τροποποίησης και διαγραφής δραστηριότητας
DisplayGroupActivity Handler	Σύνολο μονάδων υπεύθυνο για τη λειτουργία επισκόπησης ομαδικής ψευδοδραστηριότητας	Οι βασική αρμοδιότητα της μονάδας είναι η κλήση της όψης showGroupActivity για την επιστροφή του template με τα πλήρη στοιχεία των ομαδοποιημένων δραστηριοτήτων
DeleteActivityHandler	Μονάδα υπεύθυνη για τη διαγραφή μιας δραστηριότητας	Οι κύριες της αρμοδιότητες αφορούν την αποστολή της προς διαγραφή δραστηριότητας στην όψη

		deleteActivity, την κλήση RenderViewActivities για την ανανέωση των δραστηριοτήτων, την κλήση της συνάρτησης DrawGroupUngroupSort WithChart για την ανανέωση του πλέγματος και την εξαφάνιση της διεπιφάνειας μέσω της οποίας επιλέχθηκε η διαγραφή (τύπου Modal)
(GoogleMaps) Initialize	Συνάρτηση υπεύθυνη για την αρχικοποίηση και ορισμό συμπεριφοράς των χαρτών που αφορούν τη διεπιφάνεια προσθήκης δραστηριότητας	Αρχικοποιεί το χάρτη στο πλαίσιο που τον φιλοξενεί, ενώ ταυτόχρονα καθορίζει τον τρόπο εισαγωγής διευθύνσεων, τον αριθμό αποτελεσμάτων, τον τρόπο αναπαράστασης τοποθεσίας κ.ά.

Πίνακας 5.26 Συναρτήσεις και Μονάδες Χειρισμού Πελάτη Συσχετιζόμενες με την Οθόνη Index

5.4.6 Οθόνη Goals

1. Η οθόνη αυτή αφορά τους θεσπισμένους στόχους των δραστηριοτήτων και τους παρουσιάζει σε έναν πίνακα, προσφέροντας νόημα στην ανάθεση στόχων στις δραστηριότητες.
2. Προσφέρει μια πλήρη εποπτεία του εκάστοτε στόχου, της ημερομηνίας θέσπισής του, της δραστηριότητας με την οποία ο στόχος αυτός συνδέεται, όπως και της έκβασης/κατάστασής του (αποτυχημένος, σε εξέλιξη ή επιτυχημένος).
3. Προσφέρει δυνατότητα επεξεργασίας στόχου, αλλαγής/ενημέρωσης της κατάστασής του, καθώς και διαγραφής του στόχου χωρίς φυσικά να διαγράφεται και η δραστηριότητα με την οποία αυτός συνδέεται.
4. Απώτερος σκοπός είναι η υλοποίηση μιας σελίδας διαχείρισης, παρακολούθησης και ελέγχου των καθημερινά θεσπισμένων στόχων του χρήστη. Μέσω αυτής ο χρήστης είναι σε θέση να ελέγχει το βαθμό στον οποίο οι στόχοι του επιτυγχάνονται και ορίζονται ρεαλιστικά.
5. Οι όψεις Django που συνδέονται με την οθόνη αυτή είναι οι ακόλουθες:

Όνομα και Σύνδεσμος Όψης	Χρήση	Περιγραφή
goals (activitytracker/goals)	Υπεύθυνη για τη φόρτωση και αρχικοποίηση τιμών της ιστοσελίδας	Ενεργοποιείται όταν ο χρήστης επισκέπτεται τον σύνδεσμο (url) της σελίδας και επιστρέφει το αντίστοιχο template
goalsToJson (activitytracker/goals/goalstojson)	Επιστρέφει τις απαραίτητες πληροφορίες που θα εισαχθούν στον πίνακα στόχων	Κατά την αρχικοποίηση ή ανανέωση των στόχων, η όψη αυτή καλείται και επιστρέφει τις δραστηριότητες που χαρακτηρίζονται από κάποιο στόχο. Αυτές αποτελούν τα δεδομένα του πίνακα στόχων
goalHandler (activitytracker/goals/goalhandler)	Υπεύθυνη για κάθε ενέργεια ή τροποποίηση που πραγματοποιείται στον πίνακα της οθόνης	Κάθε φορά που τροποποιείται ή διαγράφεται ένας στόχος και κάθε φορά που αλλάζει η κατάσταση του στόχου αυτού, η όψη goalHandler είναι υπεύθυνη για την ενημέρωση των αλλαγών αυτών στη βάση δεδομένων
showActivity (activitytracker/activity/display-activity/<activity_id>)	Επιστρέφει template αναλυτικής παρουσίασης δραστηριότητας (όπως στην οθόνη Index)	Η όψη αυτή ενεργοποιείται όταν ο χρήστης επιλέξει να δει τη δραστηριότητα με την οποία ο στόχος συνδέεται, παρουσιάζοντας τις λεπτομέρειές αυτής στο γνωστό πλέον template

Πίνακας 5.27 Όψεις Συσχετιζόμενες με την Οθόνη Goals

- Αντίστοιχα, στην πλευρά του client οι συναρτήσεις και μονάδες χειρισμού (σε jQuery και javascript) είναι οι ακόλουθες:

Όνομα Μονάδας/Συνάρτησης	Χρήση	Περιγραφή
TableHandler	Βασική μονάδα κατασκευής του πίνακα και της συμπεριφοράς του	Εκκινεί την εξωτερική βιβλιοθήκη που υλοποιεί τον πίνακα, ορίζει τον σύνδεσμο τροφοδότησης δραστηριοτήτων (όψη goalsToJson), γεμίζει το περιεχόμενο του πίνακα και καθορίζει τη συμπεριφορά κάθε κελιού. Το 1 ^ο κελί επιτρέπει την επεξεργασία του στόχου, το 3 ^ο επιτρέπει την επισκόπηση της δραστηριότητας, το 4 ^ο την αλλαγή/ανανέωση της κατάστασης του στόχου, ενώ το 5 ^ο διαγράφει τον στόχο (όχι τη δραστηριότητα).
HandleGoalAction	Συνάρτηση υπεύθυνη για την πραγματοποίηση οποιασδήποτε τροποποίησης ή ενέργειας στα δεδομένα του πίνακα	Η συνάρτηση αυτή καλείται όταν ο χρήστης επιλέξει μια από τις ενέργειες που προσφέρει ο πίνακας. Είναι υπεύθυνη για την επικοινωνία με το server (όψη goalHandler), την αποστολή δεδομένων και την ανανέωση του περιεχόμενου του πίνακα
displayActivityHandler	Σύνολο μονάδων υπεύθυνο για τη λειτουργία επισκόπησης δραστηριότητας	Οι βασικές αρμοδιότητες αφορούν την κλήση της όψης showActivity για την επιστροφή του template με τα στοιχεία της δραστηριότητας. Διαφέρει με την αντίστοιχη της οθόνης Index, καθώς δεν προσφέρει πρόσβαση στις λειτουργίες επεξεργασίας και διαγραφής της δραστηριότητας

Πίνακας 5.28 Συναρτήσεις και Μονάδες Χειρισμού Πελάτη Συσχετιζόμενες με την Οθόνη Goals

5.4.7 Οθόνη Timeline

1. Η οθόνη αυτή προσφέρει ένα κάθετο χρονογράφημα με τις δραστηριότητες ταξινομημένες από την πιο πρόσφατη στην πιο παλαιά. Όσο πιο χαμηλά μεταβαίνει ο χρήστης στην ιστοσελίδα, τόσο πιο «πίσω στο χρόνο» θα πηγαίνει.
2. Προσφέρει γρήγορη και εύκολη εποπτεία των δραστηριοτήτων, των ημερομηνιών πραγματοποίησής τους και της διάρκειάς τους. Ακόμα, μέσω κατάλληλων εικονιδίων ο χρήστης ενημερώνεται για τις δραστηριότητες που συνοδεύονται από συμπληρωμένα πεδία αντικειμένων, συμμετεχόντων, τοποθεσίας, στόχου και αποτελέσματος.
3. Ο χρήστης έχει πρόσβαση στις λειτουργίες πλήρους επισκόπησης, τροποποίησης και διαγραφής δραστηριότητας, όπως και στην οθόνη Index.
4. Σκοπός της οθόνης αυτής είναι η εναλλακτική επισκόπηση δραστηριοτήτων (πέρα από το ημερολόγιο και το πλέγμα της σελίδας Index), προσφέροντας στο χρήστη δυνατότητα επιλογής προτιμώμενης επισκόπησης.
5. Οι Django όψεις του server που αλληλεπιδρούν με την οθόνη αυτή είναι οι ακόλουθες:

Όνομα και Σύνδεσμος Όψης	Χρήση	Περιγραφή
<code>timeline</code> (activitytracker/timeline)	Υπεύθυνη για τη φόρτωση του βασικού template της οθόνης	Ενεργοποιείται όταν ο χρήστης επισκέπτεται τη σελίδα Timeline, αρχικοποιεί τιμές και επιστρέφει το βασικό template της οθόνης (από το οποίο καλούνται και οι συναρτήσεις/μονάδες της πλευράς πελάτη)
<code>timeline_events_json</code> (activitytracker/timelinejson)	Υπεύθυνη για την ταξινομημένη επιστροφή των δραστηριοτήτων σε κωδικοποίηση JSON	Η όψη καλείται από τη συνάρτηση κατασκευής του χρονογράμματος και παρέχει σε φθίνουσα χρονική σειρά τα απαραίτητα δεδομένα για την κατασκευή του. Επειδή πολλές φορές τα δεδομένα είναι πολλά για να αποσταλούν μαζί, η όψη αυτή πραγματοποιεί σελιδοποίηση, χωρίζοντας τα συνολικά δεδομένα σε τμήματα που ονομάζονται

		«σελίδες», αποστέλλοντας μια σελίδα τη φορά.
showActivity (<i>activitytracker/activity/show-activity/<activity_id></i>)	Υπεύθυνη για την αναλυτική παρουσίαση των δεδομένων μιας δραστηριότητας στο χρήστη	Όταν ο χρήστης επιλέξει να δει αναλυτικά μια δραστηριότητα, η όψη αυτή επιστρέφει κατάλληλο template, παρουσιάζοντας τις τιμές της δραστηριότητας
editActivity (<i>activitytracker/activity/edit-activity/<activity_id></i>)	Υπεύθυνη για την αρχικοποίηση της τροποποιούμενης δραστηριότητας	Όταν ο χρήστης επιλέξει να τροποποιήσει μια δραστηριότητα, η όψη αυτή επιστρέφει κατάλληλο template συμπλήρωσης πεδίων με αρχικοποιημένες τιμές
updateActivity (<i>activitytracker/activity/update-activity</i>)	Υπεύθυνη για την ανανέωση των τιμών της τροποποιημένης δραστηριότητας στη βάση δεδομένων	Όταν ο χρήστης επιλέξει την αποθήκευση των αλλαγών από το template της editActivity, τότε αποστέλλεται ασύγχρονο αίτημα από το server περιέχοντας τις αλλαγές που θα εισαχθούν στο σύστημα από την όψη αυτή
deleteActivity (<i>activitytracker/activity/delete-activity</i>)	Υπεύθυνη για την αφαίρεση της επιλεγμένης δραστηριότητας από το σύστημα	Όταν ο χρήστης επιλέξει διαγραφή δραστηριότητας, η όψη αυτή δέχεται το μοναδικό αναγνωριστικό της επιλεγμένης δραστηριότητας και την αφαιρεί από τη βάση δεδομένων

Πίνακας 5.29 Όψεις Συσχετιζόμενες με την Οθόνη Timeline

6. Οι συναρτήσεις και μονάδες χειρισμού της client πλευράς είναι οι ακόλουθες:

Όνομα Μονάδας/Συνάρτησης	Χρήση	Περιγραφή
CreateTimelineInstance	Συνάρτηση κατασκευής μιας καταχώρησης του χρονογραφήματος	Δέχεται ως όρισμα τα στοιχεία μιας δραστηριότητας και την εισάγει στο χρονογράφημα. Καλείται επαναληπτικά για όλες τις δραστηριότητες
UpdateTimelineActivity	Συνάρτηση ανανέωσης δεδομένων μιας καταχώρησης μετά από τροποποίηση	Η συνάρτηση αυτή ελέγχει αν η τροποποίηση αφορούσε την ημερομηνία έναρξης της δραστηριότητας. Αν όχι τότε δεν απαιτείται επανασχεδιασμός όλου του γραφήματος αλλά μόνο της συγκεκριμένης καταχώρησης μέσω της συνάρτησης CreateTimelineInstance. Διαφορετικά, η αλλαγή της ημερομηνίας μετακινεί την καταχώρηση και συνεπώς το χρονογράφημα χρήζει επανασχεδιασμού μέσω της συνάρτησης TimelinePagination
DrawTimeline	Σχεδιάζει χρονογράφημα με τις δραστηριότητες που δέχτηκε ως απάντηση από το server	Αλληλεπιδρά με την όψη timeline_events_json και δέχεται τις δραστηριότητες μιας συγκεκριμένης «σελίδας δεδομένων». Έπειτα με χρήση της CreateTimelineInstance σχεδιάζει ένα χρονογράφημα με τα δεδομένα αυτά. Η συνάρτηση αυτή δέχεται ως όρισμα την επιθυμητή σελίδα δεδομένων προς σχεδίαση.
TimelinePagination	Κύρια συνάρτηση που καλείται για να δημιουργήσει,	Ορίζει τον τρόπο αλλαγής «σελίδας δεδομένων» και καλεί την DrawTimeline για

	αρχικοποιήσει και ορίσει τη συμπεριφορά του χρονογραφήματος (timeline)	την επιλεγμένη σελίδα. Συγκεκριμένα κάθε φορά που ο χρήστης μεταβαίνει στο 85% του ύψους της οθόνης Timeline, τότε αιτείται η επόμενη «σελίδα δεδομένων», η οποία περνάει ως όρισμα στην DrawTimeline
EditActivityHandler	Σύνολο μονάδων υπεύθυνο για τη λειτουργία τροποποίησης δραστηριότητας	Οι βασικές αρμοδιότητες αφορούν την κλήση της όψης editActivity για την εμφάνιση του template επεξεργασίας δεδομένων δραστηριότητας, την αποστολή δεδομένων στην όψη updateActivity για την αποθήκευση στη Β.Δ., την κλήση UpdateTimelineActivity για κατάλληλη ανανέωση του χρονογραφήματος και την εξαφάνιση της οθόνης τροποποίησης δραστηριότητας (τύπου Modal)
DisplayActivityHandler	Σύνολο μονάδων υπεύθυνο για τη λειτουργία επισκόπησης δραστηριότητας	Οι βασικές αρμοδιότητες αφορούν την κλήση της όψης showActivity και την επιστροφή του template με τα στοιχεία της δραστηριότητας και τη δυνατότητα πρόσβασης στις λειτουργίες τροποποίησης και διαγραφής δραστηριότητας
DeleteActivityHandler	Μονάδα υπεύθυνη για τη διαγραφή μιας δραστηριότητας	Οι κύριες της αρμοδιότητες αφορούν την αποστολή της προς διαγραφή δραστηριότητας στο server (στην όψη deleteActivity), την κλήση της συνάρτησης TimelinePagination για τον

		επανασχεδιασμό του χρονογραφήματος και την εξαφάνιση της διεπιφάνειας μέσω της οποίας επιλέχθηκε η διαγραφή (τύπου Modal)
--	--	---

Πίνακας 5.30 Συναρτήσεις και Μονάδες Χειρισμού Πελάτη Συσχετιζόμενες με την Οθόνη Timeline

5.4.8 Οθόνη Settings

1. Η οθόνη αυτή αποτελεί την προσωπική οθόνη του χρήστη, όπου μπορεί να επεξεργάζεται τα στοιχεία και τις πληροφορίες του στην εφαρμογή.
2. Προσφέρει δυνατότητα επισκόπησης και τροποποίησης στοιχείων (συμπεριλαμβανομένου και του κωδικού πρόσβασης), όπως επίσης και δυνατότητα πλήρους διαγραφής του χρήστη από το σύστημα.
3. Δίνει τη δυνατότητα προσθήκης, επισκόπησης, τροποποίησης και διαγραφής προσωπικών τοποθεσιών, οι οποίες θα χρησιμοποιηθούν στην αντίστοιχη ανάλυση. Συγκεκριμένα, μέσω αυτών, ο χρήστης είναι σε θέση να ελέγξει τις δραστηριότητες που λαμβάνουν χώρα γύρω από τις εισαγμένες τοποθεσίες και να εξετάσει το βαθμό στον οποίο οι τοποθεσίες αυτές επηρεάζουν την καθημερινότητά του.
4. Δίνει τη δυνατότητα στο χρήστη να συνδεθεί με την υπηρεσία RunKeeper (η οποία πλέον περιλαμβάνει και την υπηρεσία Sleepy), μέσω της οποίας θα μπορέσει να συγχρονίσει τις δραστηριότητές άθλησης και ύπνου του, όπως αυτές καταγράφηκαν από τις υπηρεσίες αυτές. Μελλοντικά ο χρήστης θα μπορεί να επιλέγει ανάμεσα σε μια πληθώρα υπηρεσιών, την υπηρεσία εκείνη με την οποία θα συνδέεται και θα συγχρονίζει δραστηριότητες.
5. Οι Django όψεις του server που αλληλεπιδρούν με την οθόνη αυτή περιλαμβάνουν όψεις για την επεξεργασία στοιχείων, την πραγματοποίηση συγχρονισμού δραστηριοτήτων και την επεξεργασία προσωπικών τοποθεσιών. Αυτές είναι οι ακόλουθες:

Όνομα και Σύνδεσμος	Χρήση	Περιγραφή
Όψης settings (activitytracker/settings)	Υπεύθυνη για τη φόρτωση του βασικού template της οθόνης settings, όπως και για την πραγματοποίηση όλων των λειτουργιών που σχετίζονται με το προφίλ ενός χρήστη στο σύστημα	Η όψη αυτή ενεργοποιείται είτε όταν ο χρήστης επισκέπτεται τη σελίδα Settings, είτε όταν πραγματοποιηθεί HTTP POST αίτημα σε αυτή. Στην πρώτη περίπτωση αυτή αρχικοποιεί τιμές και επιστρέφει το βασικό template της οθόνης. Στη

		<p>δεύτερη περίπτωση, η όψη δέχεται το είδος της ενέργειας που απαιτείται να πραγματοποιήσει και μαζί με τις απαραίτητες τιμές του client, αλληλεπιδρά με τη βάση δεδομένων. Οι υποστηριζόμενες ενέργειες αφορούν την τροποποίηση προσωπικών στοιχείων, αλλαγή κωδικού πρόσβασης και διαγραφή του χρήστη από το σύστημα.</p>
<p>places (<i>activitytracker/settings/places</i>)</p>	<p>Υπεύθυνη για όλα τα αιτήματα που σχετίζονται με την επεξεργασία μιας τοποθεσίας χρήστη</p>	<p>Η όψη αυτή αλληλεπιδρά με τη βάση δεδομένων και είναι υπεύθυνη για την προσθήκη, τροποποίηση και διαγραφή της τοποθεσίας ενός χρήστη. Τα δεδομένα καταφθάνουν μέσω ασύγχρονου HTTP αιτήματος από την πλευρά του πελάτη</p>
<p>placesToJson (<i>activitytracker/settings/placetjson</i>)</p>	<p>Υπεύθυνη για την παροχή των δεδομένων των τοποθεσιών για την απεικόνισή τους στον πίνακα</p>	<p>Η όψη αυτή κωδικοποιεί σε JSON τις προσωπικές τοποθεσίες του χρήστη και τις τροφοδοτεί στον πίνακα που είναι υπεύθυνος για την απεικόνισή τους στην πλευρά του πελάτη</p>
<p>syncProviderActivities (<i>activitytracker/settings/sync/<selected_provider></i>)</p>	<p>Υπεύθυνη για τον συγχρονισμό δραστηριοτήτων του χρήστη με την επιλεγμένη υπηρεσία</p>	<p>Η όψη αυτή καλεί συνάρτηση ελέγχου ύπαρξης εξουσιοδότησης του Activity Tracker στην υπηρεσία (ή στέλνει μήνυμα σφάλματος), έπειτα βρίσκει τους συνδέσμους (urls) και επικεφαλίδες (headers) που απαιτούνται για το αίτημα, και μέσω άλλης συνάρτησης πραγματοποιεί HTTP GET αίτημα στην υπηρεσία, λαμβάνει και εισάγει τις δραστηριότητες στη βάση δεδομένων.</p>

Πίνακας 5.31 Όψεις Συσχετιζόμενες με την Οθόνη Settings

Πριν παρουσιαστούν οι αντίστοιχες μονάδες στην πλευρά του πελάτη, αξίζει να αναλυθεί περαιτέρω η διαδικασία συγχρονισμού. Η όψη `syncProviderActivities` καλεί τη συνάρτηση `checkConnection`, η οποία ελέγχει αν ο χρήστης έχει εξουσιοδοτήσει το `Activity Tracker`, ελέγχοντας την ύπαρξη κρυφού κλειδιού στη βάση δεδομένων. Αν δεν υπάρχει, τότε ο χρήστης καλείται να συνδεθεί με την υπηρεσία, ακολουθώντας την ίδια διαδικασία όπως και με τη λειτουργία «σύνδεσης χρήστη μέσω τρίτων παρόχων». Η όψη `social_login` του κεφαλαίου (5.4.1) πραγματοποιεί ακριβώς αυτό. Αν ωστόσο ο χρήστης έχει ήδη εξουσιοδοτήσει την εφαρμογή, τότε μέσω μιας δομής λεξικού (`dictionary`) καλείται η κατάλληλη συνάρτηση συγχρονισμού (στην περίπτωσή μας, αυτή είναι η συνάρτηση `syncRunkeeperActivities`), η οποία βρίσκει τους απαραίτητους συνδέσμους και επικεφαλίδες και καλεί τις υποσυναρτήσεις `syncRunkeeperFitnessActivities` και `syncRunkeeperSleepActivities` για το συγχρονισμό των δραστηριοτήτων άθλησης και ύπνου αντίστοιχα. Κάθε μια από τις συναρτήσεις αυτές χωρίζεται σε δύο (2) υποσυναρτήσεις/τμήματα. Το 1^ο αφορά την πραγματοποίηση αιτήματος και τη λήψη των στοιχείων από την εφαρμογή, ενώ το 2^ο αφορά την σωστή εισαγωγή των πληροφοριών στη βάση δεδομένων. Οι συναρτήσεις αυτές είναι γραμμένες σε Python και είναι εμφανείς στο αρχείο `myfunctions.py`.

6. Η οθόνη αυτή συσχετίζεται με τις ακόλουθες συναρτήσεις και μονάδες χειρισμού `jQuery/javascript` στην πλευρά του πελάτη:

Όνομα Μονάδας/Συνάρτησης	Χρήση	Περιγραφή
DeleteAccountHandler	Μονάδα υπεύθυνη για τη λειτουργία διαγραφής λογαριασμού χρήστη	Η μονάδα αυτή παρουσιάζει κατάλληλη διεπιφάνεια έναρξης της λειτουργίας και αποστέλλει στην όψη <code>settings</code> τις απαραίτητες συμπληρωμένες τιμές, καθώς και μια μεταβλητή που υποδηλώνει την ενέργεια που εκτελείται (διαγραφή χρήστη). Σε αποτυχία παρουσιάζει κατάλληλο μήνυμα, ενώ σε επιτυχία ανακατευθύνει το χρήστη στη σελίδα σύνδεσης
EditInfoHandler	Μονάδα υπεύθυνη για την τροποποίηση των βασικών	Η μονάδα αυτή παρουσιάζει κατάλληλη διεπιφάνεια τροποποίησης πληροφοριών

	στοιχείων του λογαριασμού του χρήστη	και αποστέλλει στην όψη settings τις τροποποιημένες τιμές, καθώς και μια μεταβλητή που υποδηλώνει την ενέργεια που εκτελείται (τροποποίηση στοιχείων). Σε κάθε περίπτωση ενημερώνει το χρήστη για την έκβαση της διαδικασίας και τροποποιεί (ενδεχομένως) τις εμφανιζόμενες τιμές στην οθόνη
PassChangeHandler	Μονάδα υπεύθυνη για την αλλαγή του κωδικού πρόσβασης του χρήστη στο σύστημα	Εμφανίζει κατάλληλη διεπιφάνεια εισαγωγής παλαιού και νέου κωδικού, αλληλεπιδρά με την όψη settings αποστέλλοντας τα δεδομένα, εμφανίζει μήνυμα έκβασης αποτελέσματος διαδικασίας στο χρήστη και εξαφανίζει (σε περίπτωση επιτυχίας) τη διεπιφάνεια αυτή
SynchronizeProvider	Υπεύθυνη για την έναρξη της διαδικασίας συγχρονισμού δραστηριοτήτων	Σκοπός της είναι η αποστολή του ονόματος της επιλεγμένης υπηρεσίας συγχρονισμού δραστηριοτήτων στην όψη syncProviderActivities. Ακόμα ενημερώνει το χρήστη για την έκβαση της διαδικασίας συγχρονισμού
AppFunctionHandler	Υπεύθυνη για τη σύνδεση ή αποσύνδεση ενός χρήστη με μια υπηρεσία	Η συνάρτηση αυτή εκκινεί τη διαδικασία εξουσιοδότησης ή άρσης εξουσιοδότησης Activity Tracker απο μια υπηρεσία. Η αποσύνδεση πραγματοποιείται διαγράφοντας το μοναδικό κρυφό κλειδί του χρήστη

Initialize	Υπεύθυνη για την αρχικοποίηση και καθορισμό συμπεριφοράς των χαρτών της σελίδας	Η συνάρτηση αυτή αρχικοποιεί τους χάρτες, ορίζει τα επίπεδα εστίασης, τις μεθόδους αναζήτησης και παρουσίασης αποτελεσμάτων, όπως και τις δυνατότητες του χρήστη πάνω στους χάρτες. Οι χάρτες αυτοί χρησιμοποιούνται για την εισαγωγή ή/και τροποποίηση των τοποθεσιών του χρήστη
addPlaceHandler	Μονάδα υπεύθυνη για την προσθήκη τοποθεσίας χρήστη	Η μονάδα αυτή παρουσιάζει διεπιφάνεια εισαγωγής με χάρτη, αποστέλλει τις επιλογές στην όψη places και ενημερώνει το χρήστη για την έκβαση της διαδικασίας. Σε περίπτωση επιτυχούς εισαγωγής εξαφανίζει τη διεπιφάνεια
UpdatePlaceAndReload	Υπεύθυνη για την ανανέωση των στοιχείων μιας τοποθεσίας και την ανανέωση του πίνακα τοποθεσιών	Συνάρτηση που ενεργοποιείται όταν επιλεχθεί η λειτουργία τροποποίησης τοποθεσίας. Έχει την ευθύνη παρουσίασης της τοποθεσίας σε διεπιφάνεια, αποστολής των νέων δεδομένων στην όψη places και κλήσης κατάλληλης μεθόδου ανανέωσης των στοιχείων του πίνακα τοποθεσιών
DeletePlaceAndReload	Υπεύθυνη για τη διαγραφή μιας τοποθεσίας και την ανανέωση του πίνακα	Συνάρτηση που ενεργοποιείται όταν επιλεχθεί η λειτουργία διαγραφής μιας δραστηριότητας. Έχει την ευθύνη αποστολής του ονόματος της τοποθεσίας προς διαγραφή στην όψη

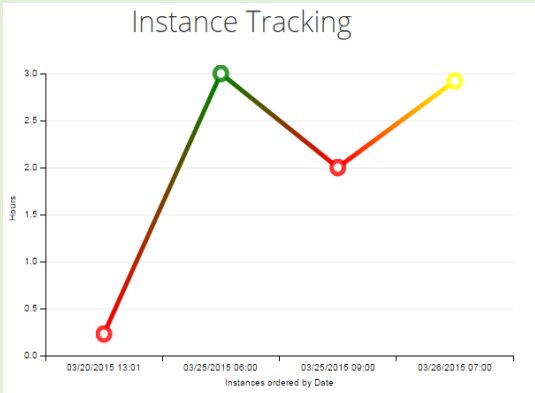
		places και κλήσης της κατάλληλης μεθόδου ανανέωσης δεδομένων του πίνακα τοποθεσιών
TableHandler	Υπεύθυνη για την κατασκευή του πίνακα τοποθεσιών	Η συνάρτηση αυτή αρχικοποιεί την εξωτερική βιβλιοθήκη που υλοποιεί τον πίνακα, ορίζει τις οντότητες που θα αποτελούν τα κελιά του, τις συμπεριφορές τους, καθώς και την όψη από την οποία ο πίνακας θα ανανεώνει τα δεδομένα του (placesToJson)

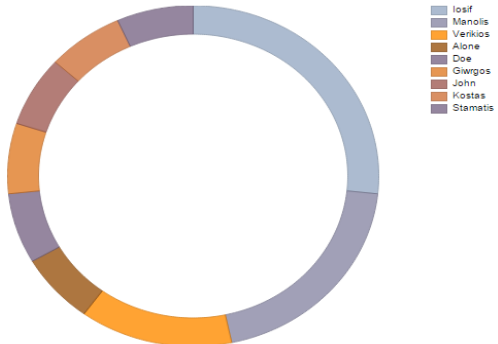
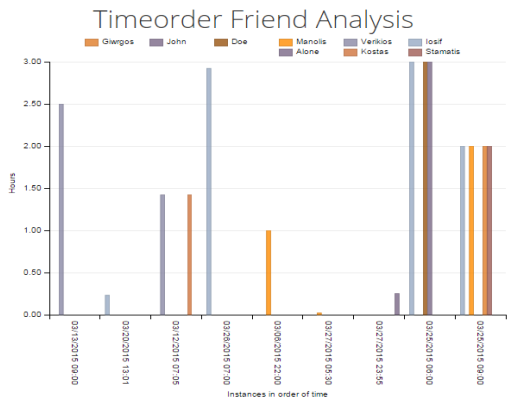
Πίνακας 5.32 Συναρτήσεις και Μονάδες Χειρισμού Πελάτη Συσχετιζόμενες με την Οθόνη Settings

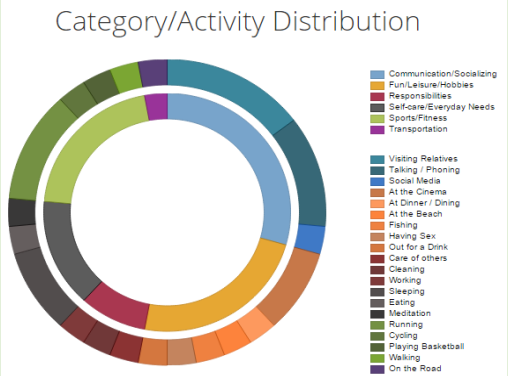
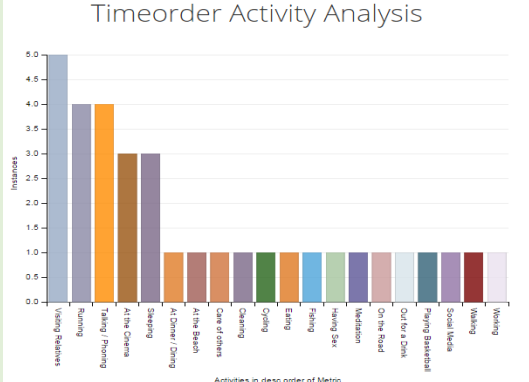
5.4.9 Οθόνη Friend Analytics

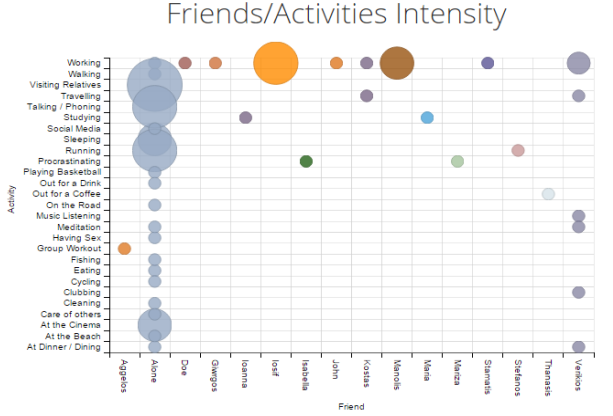
1. Η οθόνη αυτή είναι μια εκ των πέντε (5) σελίδων ανάλυσης, προσφέροντας πληροφορίες και διαγράμματα που χαρακτηρίζουν το κοινωνικό πλαίσιο των δραστηριοτήτων.
2. Πραγματοποιεί μια ανάλυση με επίκεντρο την έννοια των συμμετεχόντων/φίλων, συνδέοντάς τους με τις δραστηριότητες και παρέχοντας αναλυτικά στοιχεία για το βαθμό στον οποίο αυτοί επηρεάζουν την καθημερινότητα του χρήστη.
3. Η οθόνη αυτή αποτελείται από τρία (3) μέρη:
 - Το πλαίσιο φίλων, με τα οποία ο χρήστης καθορίζει τόσο το είδος πληροφορίας που παρουσιάζεται, όσο και τον τύπο των διαγραμμάτων. Η οθόνη αυτή παρέχει φίλτρο ημερομηνίας, φίλτρο δραστηριοτήτων, φίλτρο φίλων και φίλτρο μετρικών. Το 1^ο ορίζει το εξεταζόμενο ημερολογιακό διάστημα καθορίζοντας τα δεδομένα που θα επιλεγθούν. Το 2^ο καθορίζει την εξεταζόμενη δραστηριότητα (ή όλες), το 3^ο τον υπό εξέταση φίλο (ή όλους) και το 4^ο το είδος της μετρικής που μπορεί να είναι είτε αριθμός συμβάντων (Number of Instances) είτε αριθμός ωρών (Time Elapsed) . Η τελευταία ορίζει το μέγεθος το οποίο χρησιμοποιείται ως εξαρτημένη μεταβλητή στα διαγράμματα.
 - Το πλαίσιο στατιστικών, το οποίο παρουσιάζει σημαντικές τιμές και ποσοστά που παρουσιάζουν ενδιαφέρον. Αυτά αλλάζουν σύμφωνα με τα επιλεγμένα φίλτρα.

- Το πλαίσιο διαγραμμάτων, όπου παρουσιάζονται από ένα (1) έως δύο (2) διαγράμματα, σύμφωνα με τις επιλογές του χρήστη στο πλαίσιο φίλτρων. Τα είδη και τα περιεχόμενα των διαγραμμάτων αναλύονται παρακάτω.
4. Η διαφοροποίηση των παρουσιαζόμενων αναλυτικών οφείλεται στην επιλογή διαφορετικών τύπων φίλτρων. Συγκεκριμένα τα φίλτρα δραστηριότητας και φίλων καθορίζουν τα διαγράμματα και στοιχεία που απεικονίζονται. Οι δυνατοί συνδυασμοί φίλτρων, στατιστικών και διαγραμμάτων παρουσιάζονται στον ακόλουθο πίνακα:

Επιλεγμένη Δραστη/ητα	Επιλεγμένος Φίλος	Διαθέσιμες Μετρικές	Στατιστικά Στοιχεία	Διαθέσιμα Διαγράμματα															
Συγκεκριμένη	Συγκεκριμένος ή Κανένας (μη παρουσία φίλων)	<ul style="list-style-type: none"> Αριθμός Ωρών 	<ol style="list-style-type: none"> Συνολικές εμφανίσεις συγκεκριμένης δραστηριότητας. Εμφανίσεις συγκεκριμένης δραστηριότητας όντας με φίλους ή μόνος. Ποσοστιαία συμμετοχή φίλων στη συγκεκριμένη δραστηριότητα ή ποσοστό εμφανίσεων δραστηριότητας δίχως παρουσία φίλου αντίστοιχα. Συνολικός χρόνος συγκεκριμένης δραστηριότητας όντας με φίλους ή μόνος αντίστοιχα. 	<p>Το διάγραμμα παρουσιάζει την εξέλιξη της συγκεκριμένης δραστηριότητας με το συγκεκριμένο φίλο (ή όντας μόνος) μετρούμενη σε ώρες. Ταυτόχρονα, ενημερώνει για την επίτευξη στόχων, χρωματίζοντας τις κορυφές του διαγράμματος με κόκκινο (για αποτυχία), κίτρινο (για εξέλιξη), πράσινο (για επίτευξη) ή γκρι (για απουσία στόχου).</p>  <table border="1"> <caption>Data for Instance Tracking Chart</caption> <thead> <tr> <th>Date</th> <th>Hours</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>03/20/2015 13:01</td> <td>0.25</td> <td>Red (Failure)</td> </tr> <tr> <td>03/25/2015 06:00</td> <td>3.0</td> <td>Green (Achievement)</td> </tr> <tr> <td>03/25/2015 09:00</td> <td>2.0</td> <td>Red (Failure)</td> </tr> <tr> <td>03/28/2015 07:00</td> <td>3.0</td> <td>Yellow (Progress)</td> </tr> </tbody> </table>	Date	Hours	Status	03/20/2015 13:01	0.25	Red (Failure)	03/25/2015 06:00	3.0	Green (Achievement)	03/25/2015 09:00	2.0	Red (Failure)	03/28/2015 07:00	3.0	Yellow (Progress)
Date	Hours	Status																	
03/20/2015 13:01	0.25	Red (Failure)																	
03/25/2015 06:00	3.0	Green (Achievement)																	
03/25/2015 09:00	2.0	Red (Failure)																	
03/28/2015 07:00	3.0	Yellow (Progress)																	

Συγκεκριμένη	Όλοι	<ul style="list-style-type: none"> • Αριθμός Συμβάντων • Αριθμός Ωρών 	<ol style="list-style-type: none"> 1. Συνολικές εμφανίσεις συγκεκριμένης δραστηριότητας. 2. Εμφανίσεις συγκεκριμένης δραστηριότητας με φίλους. 3. Ποσοστιαία συμμετοχή φίλων στην συγκεκριμένη δραστηριότητα. 4. Συνολικός χρόνος με φίλους στη συγκεκριμένη δραστηριότητα. 	<p>Το 1^ο διάγραμμα δείχνει τον καταμερισμό των φίλων στις εμφανίσεις της συγκεκριμένης δραστηριότητας, είτε σε ώρες είτε σε αριθμό συμβάντων.</p> <p style="text-align: center;">Friend Distribution</p>  <p>Το 2^ο διάγραμμα κατατάσσει χρονολογικά τις εμφανίσεις της δραστηριότητας και παρουσιάζει τις συμμετοχές των φίλων σε αυτές.</p> <p style="text-align: center;">Timeorder Friend Analysis</p> 
--------------	------	---	---	---

Όλες	Συγκεκριμένος ή Κανένας (μη παρουσία φίλων)	<ul style="list-style-type: none"> • Αριθμός συμβάντων • Αριθμός ωρών 	<ol style="list-style-type: none"> 1. Αριθμός συνολικών δραστηριοτήτων. 2. Αριθμός δραστηριοτήτων με συγκεκριμένο όντας με φίλο ή μόνος. 3. Ποσοστιαία συμμετοχή συγκεκριμένου φίλου στις δραστηριότητες ή ποσοστό δραστηριοτήτων δίχως παρουσία φίλου αντίστοιχα. 4. Συνολικός χρόνος με όντας με το συγκεκριμένο φίλο ή μόνος αντίστοιχα. 	<p>Το 1^ο διάγραμμα είναι διπλό και παρουσιάζει τον καταμερισμό (σε ώρες ή συμβάντα) τόσο των κατηγοριών όσο και των δραστηριοτήτων που πραγματοποιήθηκαν με το συγκεκριμένο φίλο ή όντας μόνος.</p>  <p>Το 2^ο διάγραμμα παρουσιάζει σε αύξουσα σειρά είτε αριθμού εμφανίσεων είτε αριθμού ωρών, τις διαφορετικές δραστηριότητες που έγιναν με το συγκεκριμένο φίλο (ή που δεν συμμετείχε κανένας φίλος αντίστοιχα).</p> 
------	---	---	---	--

Όλες	Όλοι	<ul style="list-style-type: none"> • Αριθμός Συμβάντων • Αριθμός Ωρών 	<ol style="list-style-type: none"> 1. Αριθμός συνολικών δραστηριοτήτων. 2. Αριθμός δραστηριοτήτων με φίλους. 3. Ποσοστιαία συμμετοχή φίλων στις δραστηριότητες 4. Συνολικός χρόνος με φίλους. 	<p>Το μοναδικό διάγραμμα παρουσιάζει όλους τους συνδυασμούς φίλων και δραστηριοτήτων και αναθέτει μια φυσαλίδα σε κάθε υπαρκτό συνδυασμό φίλου – δραστηριότητας. Το μέγεθος της κάθε φυσαλίδας καθορίζει το βαθμό έντασης σε ώρες ή αριθμό συμβάντων αντίστοιχα, προσφέροντας γρήγορη εποπτεία επί του συνόλου της ανάλυσης.</p>  <p>The chart, titled "Friends/Activities Intensity", displays a grid of activities on the y-axis and friends on the x-axis. The y-axis activities include: Working, Walking, Visting Relatives, Travelling, Taking / Phoning, Studying, Social Media, Sleeping, Running, Procrastinating, Playing Basketball, Out for a Drink, Out for a Coffee, On the Road, Music Listening, Meditation, Having Sex, Group Workout, Fishing, Eating, Cycling, Clubbing, Cleaning, Care of others, At the Beach, and At Dinner / Dining. The x-axis friends are: Agapros, Alope, Oke, Olyros, Ioanna, Ioff, Isabella, John, Ioulis, Marolis, Kalia, Kaida, Stavris, Sirlenis, Thanasis, and Vasilis. Bubbles of varying sizes and colors (blue, orange, green, purple) are plotted at the intersections, representing the intensity of the relationship between a specific friend and activity.</p>
------	------	---	---	---

Πίνακας 5.33 Ανάλυση Οθόνης Friends Analytics

5. Έχοντας παρουσιάσει τα υποστηριζόμενα διαγράμματα, θα καθοριστούν συνοπτικά οι όψεις της παρούσας σελίδας. Κάθε διάγραμμα αντιστοιχίζεται μοναδικά σε μια όψη από την οποία αντλεί τα απαραίτητα – κωδικοποιημένα ως JSON - δεδομένα.

Όνομα και Σύνδεσμος Όψης	Χρήση	Περιγραφή
analytics_friends (activitytracker/analytics/friends)	Υπεύθυνη για τη φόρτωση της σελίδας	Ενεργοποιείται όταν ο χρήστης επισκεφτεί τη σελίδα ανάλυσης φίλων. Η όψη αυτή επιστρέφει το template της οθόνης αυτής
updateFriendsBanner (activitytracker/analytics/friends/update/friends_chartbanner)	Υπεύθυνη για την παροχή των στατιστικών τιμών στο πλαίσιο παρουσίασής τους στη σελίδα	Ενεργοποιείται κάθε φορά που τροποποιούνται τα φίλτρα και είναι υπεύθυνη για την ανανέωση ή αλλαγή των στατιστικών τιμών
updateOneFriendManyActivitiesCharts (activitytracker/analytics/friends/update/activities_friend)	Υπεύθυνη για την παροχή δεδομένων στα διαγράμματα με συγκεκριμένο φίλο και όλες τις δραστηριότητες επιλεγμένες	Καλείται κάθε φορά που επιλέγονται τα συγκεκριμένα φίλτρα και παρέχει δεδομένα στα δύο (2) διαγράμματα του συνδυασμού αυτού
updateManyFriendsOneActivityCharts (activitytracker/analytics/friends/update/activity_friends)	Υπεύθυνη για την παροχή δεδομένων στα διαγράμματα με όλους τους φίλους επιλεγμένους, αναφορικά με μια συγκεκριμένη δραστηριότητα	Καλείται κάθε φορά που επιλέγονται τα συγκεκριμένα φίλτρα και παρέχει δεδομένα στα δύο (2) διαγράμματα του συνδυασμού αυτού
updateOneFriendOneActivityChart (activitytracker/analytics/friends/update/activity_friend)	Υπεύθυνη για την παροχή δεδομένων στο διάγραμμα που αναφέρεται σε συγκεκριμένο φίλο και συγκεκριμένη δραστηριότητα	Καλείται κάθε φορά που επιλέγονται τα συγκεκριμένα φίλτρα και παρέχει δεδομένα στο διάγραμμα του συνδυασμού αυτού
updateManyFriendsManyActivitiesChart (activitytracker/analytics/friends/update/activities_friends)	Υπεύθυνη για την παροχή δεδομένων στο διάγραμμα που αναφέρεται σε όλους τους δυνατούς συνδυασμούς φίλων και δραστηριοτήτων	Καλείται κάθε φορά που επιλέγονται τα συγκεκριμένα φίλτρα και παρέχει δεδομένα στο διάγραμμα του συνδυασμού

Πίνακας 5.34 Όψεις Συσχετιζόμενες με την Οθόνη Friend Analytics

6. Τέλος, οι παραπάνω όψεις συνδέονται άμεσα με μια σειρά συναρτήσεων και μονάδων χειρισμού στην πλευρά client. Αυτές επικοινωνούν με τις όψεις και είναι υπεύθυνες για την αλλαγή της αναπαράστασης, σύμφωνα με τις εκάστοτε επιλογές του χρήστη. Όπως προαναφέρθηκε, η αλλαγή του τύπου διαγραμματικής αναπαράστασης έγκειται στην αλλαγή των τιμών των φίλτρων φίλου και δραστηριότητας. Οι δυνατές περιπτώσεις είναι τέσσερις (4), ανάλογα με το αν το εκάστοτε φίλτρο λαμβάνει συγκεκριμένη ή μη τιμή.

Όνομα Μονάδας/Συνάρτησης	Χρήση	Περιγραφή
FilterChangedHandler	Μονάδα εντοπισμού αλλαγής φίλτρων, υπεύθυνη για την αλλαγή της αναπαράστασης	Κάθε φορά που ένα φίλτρο μεταβάλλεται, η μονάδα αυτή φροντίζει να ενεργοποιήσει την κατάλληλη απεικόνιση μέσω κλήσης της κατάλληλης συνάρτησης
GraphicEntitiesHandler	Μονάδα αρχικοποίησης και ενεργοποίησης εξωγενών γραφικών οντοτήτων	Η μονάδα αυτή είναι υπεύθυνη για την ενεργοποίηση ημερολογίων επιλογής χρονικού διαστήματος, λιστών με δυνατότητα αναζήτησης, όπως και άλλων γραφικών οντοτήτων που χρησιμοποιεί η σελίδα
UpdateFriendsBanner	Συνάρτηση υπεύθυνη για την ανανέωση των στατιστικών στοιχείων	Κάθε φορά που συγκεκριμένα φίλτρα μεταβάλλονται, η συνάρτηση αυτή αλληλεπιδρά με την όψη updateFriendsBanner και ανανεώνει τα στοιχεία στο πλαίσιο στατιστικών της σελίδας
OneFriendAllActivities	Συνάρτηση υπεύθυνη για τη σχεδίαση των γραφημάτων του συνδυασμού ενός φίλου και όλων των δραστηριοτήτων	Επικοινωνεί με την όψη updateOneFriendMany Activities και με τα δεδομένα αυτά, κατασκευάζει τα απαιτούμενα δύο (2) διαγράμματα
OneActivityAllFriends	Συνάρτηση υπεύθυνη για τη σχεδίαση των γραφημάτων του συνδυασμού μιας	Επικοινωνεί με την όψη updateManyFriendsOne Activity και με τα δεδομένα

	δραστηριότητας και όλων των φίλων	αυτά, κατασκευάζει τα απαιτούμενα δύο (2) διαγράμματα
OneActivityOneFriend	Συνάρτηση υπεύθυνη για τη σχεδίαση των γραφημάτων του συνδυασμού συγκεκριμένης τόσο δραστηριότητας, όσο και φίλου	Επικοινωνεί με την όψη updateOneFriendOneActivity και με τα δεδομένα αυτά, κατασκευάζει το απαιτούμενο διάγραμμα
AllActivitiesAllFriends	Συνάρτηση υπεύθυνη για τη σχεδίαση των γραφημάτων του συνδυασμού όλων των φίλων και όλων των δραστηριοτήτων	Επικοινωνεί με την όψη updateOneFriendOneActivity και με τα δεδομένα αυτά, κατασκευάζει το απαιτούμενο διάγραμμα

Πίνακας 5.35 Συναρτήσεις και Μονάδες Χειρισμού Πελάτη Συσχετιζόμενες με την Οθόνη Friend Analytics

5.4.10 Οθόνη Object Analytics

1. Η οθόνη αυτή είναι η δεύτερη εκ των πέντε (5) σελίδων ανάλυσης, προσφέροντας πληροφορίες και διαγράμματα που χαρακτηρίζουν το πλαίσιο υλικών οντοτήτων των δραστηριοτήτων.
2. Πραγματοποιεί μια ανάλυση με επίκεντρο έννοια των αντικειμένων, συνδέοντάς τα με τις δραστηριότητες και παρέχοντας αναλυτικά στοιχεία για το βαθμό στον οποίο η χρήση τους επηρεάζει και διαμορφώνει την καθημερινότητα του χρήστη.
3. Αποτελεί δυική της οθόνης Friend Analytics. Η συμπεριφορά όλων των οντοτήτων είναι ακριβώς αντίστοιχη, αντικαθιστώντας την έννοια του «φίλου» με την έννοια του «αντικειμένου».
4. Η οθόνη αυτή αποτελείται από τρία (3) μέρη:
 - Το πλαίσιο φίλτρων, με τα οποία ο χρήστης καθορίζει τόσο το είδος πληροφορίας που παρουσιάζεται, όσο και τον τύπο των διαγραμμάτων. Η οθόνη αυτή παρέχει φίλτρο ημερομηνίας, φίλτρο δραστηριοτήτων, φίλτρο αντικειμένων και φίλτρο μετρικών. Το 1^ο ορίζει το εξεταζόμενο ημερολογιακό διάστημα καθορίζοντας τα δεδομένα που θα επιλεχθούν. Το 2^ο καθορίζει την εξεταζόμενη δραστηριότητα (ή όλες), το 3^ο το υπό εξέταση αντικείμενο (ή όλα) και το 4^ο το είδος της μετρικής που μπορεί να είναι είτε αριθμός συμβάντων (Number of Instances) είτε αριθμός ωρών (Time Elapsed). Η τελευταία ορίζει το μέγεθος το οποίο χρησιμοποιείται ως εξαρτημένη μεταβλητή στα διαγράμματα.

- Το πλαίσιο στατιστικών, το οποίο παρουσιάζει σημαντικές τιμές και ποσοστά που παρουσιάζουν ενδιαφέρον. Αυτά αλλάζουν σύμφωνα με τα επιλεγμένα φίλτρα.
 - Το πλαίσιο διαγραμμάτων, όπου παρουσιάζονται από ένα (1) έως δύο (2) διαγράμματα, σύμφωνα με τις επιλογές του χρήστη στο πλαίσιο φίλτρων.
5. Οι δυνατοί συνδυασμοί στατιστικών και διαγραμμάτων που παρουσιάστηκαν στον πίνακα (5.33) για την ανάλυση φίλων, είναι ακριβώς ίδιοι και για την ανάλυση αντικειμένων. Για το λόγο αυτό, ο αναγνώστης καλείται να ανατρέξει σε αυτόν, αντικαθιστώντας την έννοια «φίλου» με την έννοια «αντικειμένου» και την έννοια «δίχως κανένα φίλο» με την έννοια «δίχως τη χρήση κανενός αντικειμένου».
6. Οι όψεις του πλαισίου Django της παρούσας οθόνης βρίσκονται σε πλήρη ευθυγράμμιση με αυτές της οθόνης ανάλυσης φίλων. Ωστόσο, για λόγους πληρότητας, θα παρουσιαστούν στον ακόλουθο πίνακα. Τονίζεται η JSON κωδικοποίηση των δεδομένων επιστροφής.

Όνομα και Σύνδεσμος Όψης	Χρήση	Περιγραφή
<code>analytics_objects</code> (activitytracker/analytics/objects)	Υπεύθυνη για τη φόρτωση της σελίδας	Ενεργοποιείται όταν ο χρήστης επισκεφτεί την ιστοσελίδα ανάλυσης αντικειμένων. Η όψη αυτή επιστρέφει το template της οθόνης αυτής
<code>updateObjectsBanner</code> (activitytracker/analytics/objects/update/objects_chartbanner)	Υπεύθυνη για την παροχή των στατιστικών τιμών στο πλαίσιο παρουσίασής τους στη σελίδα	Ενεργοποιείται κάθε φορά που τροποποιούνται τα φίλτρα και είναι υπεύθυνη για την ανανέωση ή αλλαγή των στατιστικών τιμών
<code>updateOneObjectManyActivitiesCharts</code> (activitytracker/analytics/objects/update/activities_object)	Υπεύθυνη για την παροχή δεδομένων στα διαγράμματα με συγκεκριμένο αντικείμενο και όλες τις δραστηριότητες επιλεγμένες	Καλείται κάθε φορά που επιλέγονται τα συγκεκριμένα φίλτρα και παρέχει δεδομένα στα δύο (2) διαγράμματα του συνδυασμού αυτού
<code>updateManyObjectsOneActivityCharts</code> (activitytracker/analytics/objects/update/activity_objects)	Υπεύθυνη για την παροχή δεδομένων στα διαγράμματα με όλα τα αντικείμενα επιλεγμένα, αναφορικά με μια συγκεκριμένη δραστηριότητα	Καλείται κάθε φορά που επιλέγονται τα συγκεκριμένα φίλτρα και παρέχει δεδομένα στα δύο (2) διαγράμματα του συνδυασμού αυτού

updateOneObjectOne ActivityChart (activitytracker/analytics/objects/update/activity_object)	Υπεύθυνη για την παροχή δεδομένων στο διάγραμμα που αναφέρεται σε συγκεκριμένο αντικείμενο και συγκεκριμένη δραστηριότητα	Καλείται κάθε φορά που επιλέγονται τα συγκεκριμένα φίλτρα και παρέχει δεδομένα στο διάγραμμα του συνδυασμού αυτού
updateManyObjectsMany ActivitiesChart (activitytracker/analytics/objects/update/activities_objects)	Υπεύθυνη για την παροχή δεδομένων στο διάγραμμα που αναφέρεται σε όλους τους δυνατούς συνδυασμούς αντικειμένων και δραστηριοτήτων	Καλείται κάθε φορά που επιλέγονται τα συγκεκριμένα φίλτρα και παρέχει δεδομένα στο διάγραμμα του συνδυασμού αυτού. Στο διάγραμμα εμπεριέχεται και η περίπτωση απουσίας χρήσης αντικειμένων

Πίνακας 5.36 Όψεις Συσχετιζόμενες με την Οθόνη Object Analytics

7. Οι jQuery/javascript συναρτήσεις και μονάδες χειρισμού στην πλευρά του πελάτη, βρίσκονται σε πλήρη ευθυγράμμιση με τις αντίστοιχες της σελίδας ανάλυσης φίλων. Για λόγους πληρότητας, αυτές παρουσιάζονται παρακάτω:

Όνομα Μονάδας/Συνάρτησης	Χρήση	Περιγραφή
FilterChangedHandler	Μονάδα εντοπισμού αλλαγής τιμών φίλτρων, υπεύθυνη για την αλλαγή της αναπαράστασης	Κάθε φορά που ένα φίλτρο μεταβάλλεται, η μονάδα αυτή φροντίζει να ενεργοποιήσει την κατάλληλη απεικόνιση μέσω κλήσης της κατάλληλης συνάρτησης διαγραμμάτων
GraphicEntitiesHandler	Μονάδα αρχικοποίησης και ενεργοποίησης εξωγενών γραφικών οντοτήτων	Η μονάδα αυτή είναι υπεύθυνη για την ενεργοποίηση ημερολογίων επιλογής χρονικού διαστήματος, λιστών με δυνατότητα αναζήτησης, όπως και άλλων γραφικών οντοτήτων που χρησιμοποιεί η σελίδα

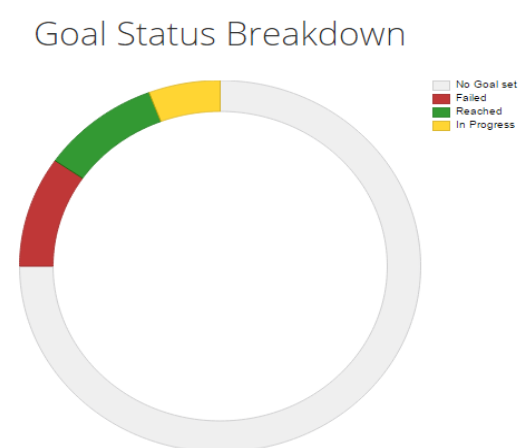
UpdateObjectsBanner	Συνάρτηση υπεύθυνη για την ανανέωση των στατιστικών στοιχείων	Κάθε φορά που συγκεκριμένα φίλτρα μεταβάλλονται, η συνάρτηση αυτή καλεί την όψη updateObjectsBanner και ανανεώνει ή αλλάζει τα στοιχεία στο πλαίσιο στατιστικών της σελίδας
OneObjectAllActivities	Συνάρτηση υπεύθυνη για τη σχεδίαση των γραφημάτων του συνδυασμού ενός αντικειμένου και όλων των δραστηριοτήτων	Επικοινωνεί με την όψη updateOneObjectMany Activities και με τα δεδομένα αυτά, κατασκευάζει τα απαιτούμενα δύο (2) διαγράμματα
OneActivityAllObjects	Συνάρτηση υπεύθυνη για τη σχεδίαση των γραφημάτων του συνδυασμού μιας δραστηριότητας και όλων των αντικειμένων	Επικοινωνεί με την όψη updateManyObjectsOne Activity και με τα δεδομένα αυτά, κατασκευάζει τα απαιτούμενα δύο (2) διαγράμματα
OneActivityOneObject	Συνάρτηση υπεύθυνη για τη σχεδίαση του γραφήματος συνδυασμού συγκεκριμένης δραστηριότητας, καθώς και αντικειμένου	Επικοινωνεί με την όψη updateOneObjectOne Activity και με τα δεδομένα αυτά, κατασκευάζει το απαιτούμενο διάγραμμα
AllActivitiesAllObjects	Συνάρτηση υπεύθυνη για τη σχεδίαση του γραφήματος συνδυασμού όλων των αντικειμένων και όλων των δραστηριοτήτων	Επικοινωνεί με την όψη updateOneObjectOne Activity και με τα δεδομένα αυτά, κατασκευάζει το απαιτούμενο διάγραμμα

Πίνακας 5.37 Συναρτήσεις και Μονάδες Χειρισμού Πελάτη Συσχετιζόμενες με την Οθόνη Friend Analytics

5.4.11 Οθόνη Goal Analytics

1. Η οθόνη αυτή είναι η τρίτη εκ των πέντε (5) σελίδων ανάλυσης, προσφέροντας πληροφορίες και διαγράμματα που σχετίζονται με τους θεσπισμένους στόχους των δραστηριοτήτων.
2. Πραγματοποιεί μια ανάλυση με επίκεντρο τους στόχους, συνδέοντάς τους με την έννοια των φίλων και των αντικειμένων και εξετάζοντας τους τρόπους με τους οποίους οι συμμετέχοντες και η χρήση συγκεκριμένων αντικειμένων μπορεί να επηρεάσει την έκβαση των δραστηριοτήτων.

3. Παρέχει αναλυτικά στοιχεία για το βαθμό στον οποίο η χρήση τους επηρεάζει και διαμορφώνει την καθημερινότητα του χρήστη.
4. Η οθόνη αυτή αποτελείται από τρία (3) μέρη:
 - Το πλαίσιο φίλτρων, με τα οποία ο χρήστης καθορίζει τόσο το είδος πληροφορίας που παρουσιάζεται, όσο και τον τύπο των διαγραμμάτων. Η οθόνη αυτή παρέχει φίλτρο ημερομηνίας, φίλτρο δραστηριοτήτων και φίλτρο ανάλυσης (για ευνόητους λόγους φίλτρο μετρικής δεν υπάρχει). Το 1^ο ορίζει το εξεταζόμενο ημερολογιακό διάστημα καθορίζοντας τα δεδομένα που θα επιλεχθούν. Το 2^ο καθορίζει την εξεταζόμενη δραστηριότητα (ή όλες), ενώ το 3^ο τον τύπο επιλεγμένο τύπο ανάλυσης. Υπάρχουν τρεις (3) διαφορετικές αναλύσεις και αυτές αφορούν τις οντότητες με τις οποίες θα συσχετιστούν οι δραστηριότητες ως προς την επίτευξη των στόχων. Οι δυνατές επιλογές είναι οι φίλοι, τα αντικείμενα και οι κατηγορίες. Επιλέγοντας τους φίλους (ή τα αντικείμενα) ο χρήστης μπορεί να ανακαλύψει ποιοι φίλοι (ή αντικείμενα) επηρεάζουν περισσότερο την επίτευξη των στόχων του, ενώ επιλέγοντας τις κατηγορίες ο χρήστης μπορεί να έχει μια πλήρη εποπτεία των στόχων τόσο σε επίπεδο δραστηριότητας όσο και σε επίπεδο κατηγορίας, αδιαφορώντας για τους φίλους και τα αντικείμενα.
 - Το πλαίσιο στατιστικών, το οποίο παρουσιάζει σημαντικές τιμές και ποσοστά που παρουσιάζουν ενδιαφέρον. Αυτά αλλάζουν σύμφωνα με τα επιλεγμένα φίλτρα.
 - Το πλαίσιο διαγραμμάτων, όπου παρουσιάζονται κατάλληλα διαγράμματα, ανάλογα με τις επιλογές του χρήστη στο πλαίσιο φίλτρων.
5. Στην οθόνη αυτή, τα φίλτρα δραστηριοτήτων και ανάλυσης είναι ανεξάρτητα μεταξύ τους. Το κάθε φίλτρο είναι υπεύθυνο για διαφορετικά διαγράμματα και οι διαφορετικοί συνδυασμοί μεταξύ των φίλτρων, δεν επηρεάζουν ούτε μεταβάλλουν τις διαγραμματικές απεικονίσεις. Οι συνδυασμοί τιμών φίλτρων, διαγραμμάτων και στατιστικών στοιχείων παρουσιάζονται στον παρακάτω πίνακα:

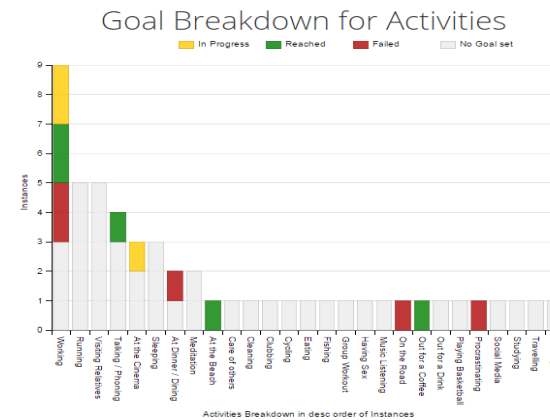
Επιλεγμένη Δραστη/τητα	Επιλεγμένη Ανάλυση	Στατιστικά Στοιχεία	Διαθέσιμα Διαγράμματα
Όλες ή Συγκεκριμένη	Ανεξάρτητο /Αδιάφορο	<ol style="list-style-type: none"> 1. Αριθμός συνολικών δραστηριοτήτων ή εμφανίσεων συγκεκριμένης δραστηριότητας αντίστοιχα. 2. Εξαρτώμενο από την επιλεγμένη ανάλυση. 3. Εξαρτώμενο από την επιλεγμένη ανάλυση. 	<p data-bbox="1031 337 1906 625">Το διάγραμμα παρουσιάζει μια ανάλυση στόχων πάνω στις δραστηριότητες που επιλέχθηκαν με βάση το ημερολογιακό διάστημα και το φίλτρο δραστηριοτήτων. Παρουσιάζει με γκρι τις εμφανίσεις δραστηριοτήτων που δεν συνοδεύονται από κάποιο στόχο, με κόκκινο αυτές των οποίων ο στόχος απέτυχε, με πορτοκαλί αυτές των οποίων ο στόχος βρίσκεται σε εξέλιξη και με πράσινο αυτές των οποίων ο στόχος επιτεύχθηκε.</p> <div data-bbox="1192 630 1724 1084">  <p data-bbox="1228 651 1606 683">Goal Status Breakdown</p> <ul data-bbox="1627 711 1717 764" style="list-style-type: none"> No Goal set Failed Reached In Progress </div>

Ανεξάρτητο
/Αδιάφορο

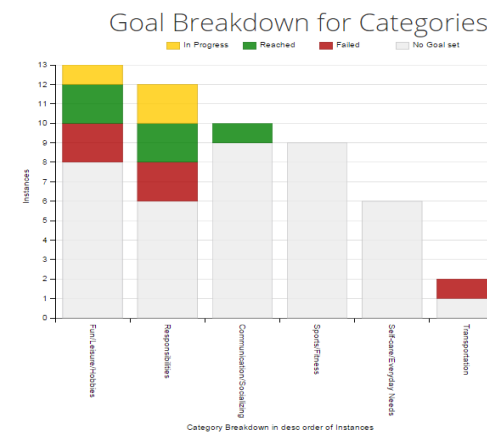
Δραστηριότητες
& Κατηγορίες

1. Αριθμός συνολικών δραστηριοτήτων ή εμφανίσεων συγκεκριμένης δραστηριότητας αντίστοιχα.
2. Αριθμός θεσπισμένων στόχων πάνω στις δραστηριότητες του στατιστικού στοιχείου (1).
3. Ποσοστό (%) στόχων που επιτεύχθηκαν εκ των παραπάνω δραστηριοτήτων.

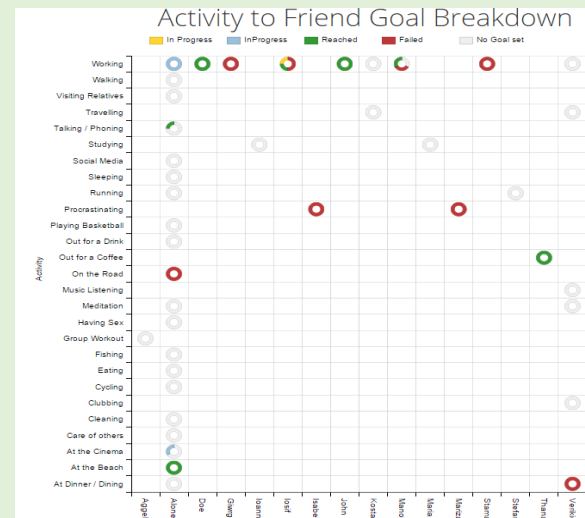
Το 1^ο διάγραμμα κατατάσσει το σύνολο των διαφορετικών δραστηριοτήτων σε φθίνουσα σειρά (σύμφωνα με το συνολικό αριθμό εμφανίσεών τους) και πραγματοποιεί μια ανάλυση στόχων για καθεμία, χρησιμοποιώντας τον ίδιο χρωματικό κώδικα με το προηγούμενο διάγραμμα.

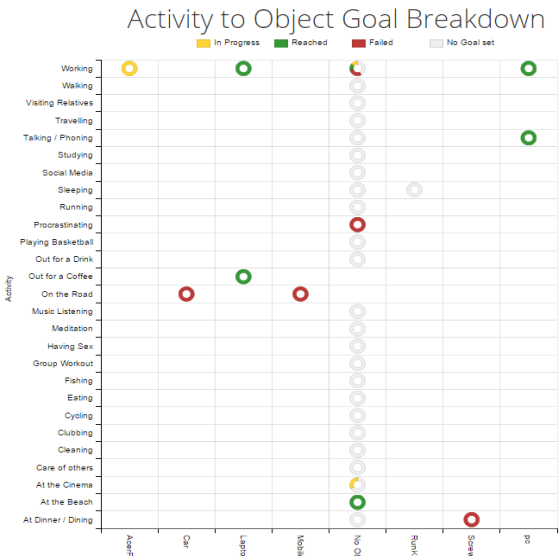


Το 2^ο διάγραμμα πραγματοποιεί μια ακριβώς ίδια ανάλυση εστιάζοντας αυτή τη φορά σύνολο των έξι (6) κατηγοριών, όπως αυτές διαμορφώνονται από τις δραστηριότητες.



<p>Ανεξάρτητο /Αδιάφορο</p>	<p>Δραστηριότητες & Φίλοι</p>	<ol style="list-style-type: none"> 1. Αριθμός συνολικών δραστηριοτήτων ή εμφανίσεων συγκεκριμένης δραστηριότητας αντίστοιχα. 2. Αριθμός θεσπισμένων στόχων επί των ομαδικών δραστηριοτήτων (δραστηριότητες που συμμετείχαν φίλοι). 5. Ποσοστό (%) στόχων που επιτεύχθηκαν πάνω στις ομαδικές δραστηριότητες αυτές. 	<p>Το διάγραμμα αυτό έχει παρεμφερή μορφή με το διάγραμμα που συσχέτιζε όλες τις δραστηριότητες με όλους τους φίλους. Συγκεκριμένα στον οριζόντιο άξονα βρίσκονται όλες οι δραστηριότητες, ενώ στον κάθετο όλοι οι φίλοι (συμπεριλαμβανομένης και της επιλογής κανενός φίλου). Έπειτα σε κάθε υπαρκτό συνδυασμό/ζευγάρι πραγματοποιείται μια ανάλυση στόχων μέσω μικρών διαγραμμάτων τύπου πίτας που αποτελούν μικρογραφίες του διαγράμματος του φίλτρου δραστηριοτήτων του παρόντος πίνακα.</p>
-----------------------------	-----------------------------------	---	--



<p>Ανεξάρτητο /Αδιάφορο</p>	<p>Δραστηριότητες & Αντικείμενα</p>	<ol style="list-style-type: none"> 1. Αριθμός συνολικών δραστηριοτήτων ή εμφανίσεων συγκεκριμένης δραστηριότητας αντίστοιχα. 2. Αριθμός θεσπισμένων στόχων επί των δραστηριοτήτων στις οποίες υπήρξε χρήση αντικειμένων 3. Ποσοστό (%) στόχων που επιτεύχθηκαν πάνω δραστηριότητες που χαρακτηρίζονται από χρήση αντικειμένων 	<p>Το διάγραμμα αυτό είναι ακριβώς ίδιο με το διάγραμμα της επιλογής «Δραστηριότητες & Φίλοι» με τη διαφορά ότι στη θέση των φίλων υπάρχουν τα χρησιμοποιημένα αντικείμενα. Ξανά σε εντός κάθε υπαρκτού συνδυασμού υπάρχει μικρογραφία διαγράμματος τύπου πίτας που παρουσιάζει μια πλήρη ανάλυση στόχων.</p>  <p>The chart 'Activity to Object Goal Breakdown' displays goal status for 30 activities across 8 objects. The legend indicates: Yellow circle for 'In Progress', Green circle for 'Reached', Red circle for 'Failed', and Grey circle for 'No Goal set'.</p> <table border="1"> <thead> <tr> <th>Activity</th> <th>Aspir</th> <th>Car</th> <th>Laptop</th> <th>Mobile</th> <th>No obj</th> <th>Print</th> <th>Screen</th> <th>SP</th> </tr> </thead> <tbody> <tr><td>Working</td><td>In Progress</td><td></td><td>Reached</td><td></td><td>No Goal set</td><td></td><td></td><td>Reached</td></tr> <tr><td>Walking</td><td></td><td></td><td></td><td></td><td>No Goal set</td><td></td><td></td><td></td></tr> <tr><td>Visiting Relatives</td><td></td><td></td><td></td><td></td><td>No Goal set</td><td></td><td></td><td></td></tr> <tr><td>Travelling</td><td></td><td></td><td></td><td></td><td>No Goal set</td><td></td><td></td><td></td></tr> <tr><td>Talking / Phoning</td><td></td><td></td><td></td><td></td><td>No Goal set</td><td></td><td></td><td>Reached</td></tr> <tr><td>Studying</td><td></td><td></td><td></td><td></td><td>No Goal set</td><td></td><td></td><td></td></tr> <tr><td>Social Media</td><td></td><td></td><td></td><td></td><td>No Goal set</td><td></td><td></td><td></td></tr> <tr><td>Sleeping</td><td></td><td></td><td></td><td></td><td>No Goal set</td><td>No Goal set</td><td></td><td></td></tr> <tr><td>Running</td><td></td><td></td><td></td><td></td><td>No Goal set</td><td></td><td></td><td></td></tr> <tr><td>Procrastinating</td><td></td><td></td><td></td><td></td><td>No Goal set</td><td></td><td></td><td></td></tr> <tr><td>Playing Basketball</td><td></td><td></td><td></td><td></td><td>No Goal set</td><td></td><td></td><td></td></tr> <tr><td>Out for a Drink</td><td></td><td></td><td></td><td></td><td>No Goal set</td><td></td><td></td><td></td></tr> <tr><td>Out for a Coffee</td><td></td><td></td><td>Reached</td><td></td><td>No Goal set</td><td></td><td></td><td></td></tr> <tr><td>On the Road</td><td></td><td>Failed</td><td></td><td>Failed</td><td>No Goal set</td><td></td><td></td><td></td></tr> <tr><td>Music Listening</td><td></td><td></td><td></td><td></td><td>No Goal set</td><td></td><td></td><td></td></tr> <tr><td>Meditation</td><td></td><td></td><td></td><td></td><td>No Goal set</td><td></td><td></td><td></td></tr> <tr><td>Having Sex</td><td></td><td></td><td></td><td></td><td>No Goal set</td><td></td><td></td><td></td></tr> <tr><td>Group Workout</td><td></td><td></td><td></td><td></td><td>No Goal set</td><td></td><td></td><td></td></tr> <tr><td>Fishing</td><td></td><td></td><td></td><td></td><td>No Goal set</td><td></td><td></td><td></td></tr> <tr><td>Eating</td><td></td><td></td><td></td><td></td><td>No Goal set</td><td></td><td></td><td></td></tr> <tr><td>Cycling</td><td></td><td></td><td></td><td></td><td>No Goal set</td><td></td><td></td><td></td></tr> <tr><td>Clubbing</td><td></td><td></td><td></td><td></td><td>No Goal set</td><td></td><td></td><td></td></tr> <tr><td>Cleaning</td><td></td><td></td><td></td><td></td><td>No Goal set</td><td></td><td></td><td></td></tr> <tr><td>Care of others</td><td></td><td></td><td></td><td></td><td>No Goal set</td><td></td><td></td><td></td></tr> <tr><td>At the Cinema</td><td></td><td></td><td></td><td></td><td>No Goal set</td><td></td><td></td><td></td></tr> <tr><td>At the Beach</td><td></td><td></td><td></td><td></td><td>No Goal set</td><td></td><td></td><td></td></tr> <tr><td>At Dinner / Dining</td><td></td><td></td><td></td><td></td><td>No Goal set</td><td></td><td></td><td></td></tr> </tbody> </table>	Activity	Aspir	Car	Laptop	Mobile	No obj	Print	Screen	SP	Working	In Progress		Reached		No Goal set			Reached	Walking					No Goal set				Visiting Relatives					No Goal set				Travelling					No Goal set				Talking / Phoning					No Goal set			Reached	Studying					No Goal set				Social Media					No Goal set				Sleeping					No Goal set	No Goal set			Running					No Goal set				Procrastinating					No Goal set				Playing Basketball					No Goal set				Out for a Drink					No Goal set				Out for a Coffee			Reached		No Goal set				On the Road		Failed		Failed	No Goal set				Music Listening					No Goal set				Meditation					No Goal set				Having Sex					No Goal set				Group Workout					No Goal set				Fishing					No Goal set				Eating					No Goal set				Cycling					No Goal set				Clubbing					No Goal set				Cleaning					No Goal set				Care of others					No Goal set				At the Cinema					No Goal set				At the Beach					No Goal set				At Dinner / Dining					No Goal set			
Activity	Aspir	Car	Laptop	Mobile	No obj	Print	Screen	SP																																																																																																																																																																																																																																																							
Working	In Progress		Reached		No Goal set			Reached																																																																																																																																																																																																																																																							
Walking					No Goal set																																																																																																																																																																																																																																																										
Visiting Relatives					No Goal set																																																																																																																																																																																																																																																										
Travelling					No Goal set																																																																																																																																																																																																																																																										
Talking / Phoning					No Goal set			Reached																																																																																																																																																																																																																																																							
Studying					No Goal set																																																																																																																																																																																																																																																										
Social Media					No Goal set																																																																																																																																																																																																																																																										
Sleeping					No Goal set	No Goal set																																																																																																																																																																																																																																																									
Running					No Goal set																																																																																																																																																																																																																																																										
Procrastinating					No Goal set																																																																																																																																																																																																																																																										
Playing Basketball					No Goal set																																																																																																																																																																																																																																																										
Out for a Drink					No Goal set																																																																																																																																																																																																																																																										
Out for a Coffee			Reached		No Goal set																																																																																																																																																																																																																																																										
On the Road		Failed		Failed	No Goal set																																																																																																																																																																																																																																																										
Music Listening					No Goal set																																																																																																																																																																																																																																																										
Meditation					No Goal set																																																																																																																																																																																																																																																										
Having Sex					No Goal set																																																																																																																																																																																																																																																										
Group Workout					No Goal set																																																																																																																																																																																																																																																										
Fishing					No Goal set																																																																																																																																																																																																																																																										
Eating					No Goal set																																																																																																																																																																																																																																																										
Cycling					No Goal set																																																																																																																																																																																																																																																										
Clubbing					No Goal set																																																																																																																																																																																																																																																										
Cleaning					No Goal set																																																																																																																																																																																																																																																										
Care of others					No Goal set																																																																																																																																																																																																																																																										
At the Cinema					No Goal set																																																																																																																																																																																																																																																										
At the Beach					No Goal set																																																																																																																																																																																																																																																										
At Dinner / Dining					No Goal set																																																																																																																																																																																																																																																										

Πίνακας 5.38 Ανάλυση Οθόνης Goal Analytics

6. Οι όψεις που συνδέονται με την οθόνη αυτή ακολουθούν την ίδια φιλοσοφία με τις όψεις των προηγούμενων οθονών ανάλυσης και παρουσιάζονται παρακάτω. Τα δεδομένα που επιστρέφουν βρίσκονται σε JSON κωδικοποίηση:

Όνομα και Σύνδεσμος Όψης	Χρήση	Περιγραφή
analytics_goals (activitytracker/analytics/goals)	Υπεύθυνη για τη φόρτωση της σελίδας	Ενεργοποιείται όταν ο χρήστης επισκεφτεί τη σελίδα ανάλυσης αντικειμένων. Η όψη αυτή επιστρέφει το template της οθόνης αυτής
updateGoalsBanner (activitytracker/analytics/goals/update/goals_chartbanner)	Υπεύθυνη για την παροχή των στατιστικών τιμών στο πλαίσιο παρουσίασής τους στη σελίδα	Ενεργοποιείται κάθε φορά που τροποποιούνται τα αντίστοιχα φίλτρα και είναι υπεύθυνη για την ανανέωση ή αλλαγή των στατιστικών τιμών
updateActivityDonutChart (activitytracker/analytics/goals/update/activity)	Υπεύθυνη για την παροχή δεδομένων στο διάγραμμα τύπου πίτας που πραγματοποιεί ανάλυση στόχων των δραστηριοτήτων	Καλείται κάθε φορά που τροποποιείται το φίλτρο της δραστηριότητας και επιστρέφει τα απαραίτητα δεδομένα για τη σχεδίαση του διαγράμματος πίτας
updateActivityAndCategoryBarChart (activitytracker/analytics/goals/update/categories_activities)	Υπεύθυνη για την παροχή δεδομένων στα δύο (2) ραβδογράμματα που πραγματοποιούν ανάλυση στόχων στις κατηγορίες και τις δραστηριότητες	Καλείται κάθε φορά που επιλέγεται η ανάλυση «Activities & Categories» και επιστρέφει (σε JSON) τα απαραίτητα δεδομένα για τη σχεδίαση των δύο (2) ραβδογραμμάτων
updateActivityAndFriendBubbleChart (activitytracker/analytics/goals/update/activities_friends)	Υπεύθυνη για την παροχή δεδομένων στο διάγραμμα που αναφέρεται στην ανάλυση στόχων των συνδυασμών δραστηριοτήτων και φίλων	Καλείται κάθε φορά που επιλέγεται η ανάλυση «Activities & Friends» και επιστρέφει τα απαραίτητα δεδομένα για τη σχεδίαση του διαγράμματος
updateActivityAndObjectBubbleChart (activitytracker/analytics/goals/update/activities_objects)	Υπεύθυνη για την παροχή δεδομένων στο διάγραμμα που αναφέρεται στην ανάλυση στόχων των συνδυασμών	Καλείται κάθε φορά που επιλέγεται η ανάλυση «Activities & Objects» και επιστρέφει τα απαραίτητα δεδομένα για τη σχεδίασή

Πίνακας 5.39 Όψεις Συσχετιζόμενες με την Οθόνη Goal Analytics

7. Οι συναρτήσεις και μονάδες χειρισμού στην πλευρά του server παρουσιάζονται παρακάτω:

Όνομα Μονάδας/Συνάρτησης	Χρήση	Περιγραφή
FilterChangedHandler	Μονάδα εντοπισμού αλλαγής φίλτρων, υπεύθυνη για την αλλαγή της αναπαράστασης	Κάθε φορά που ένα φίλτρο μεταβάλλεται, η μονάδα αυτή φροντίζει να ενεργοποιήσει την κατάλληλη απεικόνιση μέσω κλήσης της κατάλληλης συνάρτησης
GraphicEntitiesHandler	Μονάδα αρχικοποίησης και ενεργοποίησης εξωγενών γραφικών οντοτήτων	Η μονάδα αυτή είναι υπεύθυνη για την ενεργοποίηση ημερολογίων επιλογής χρονικού διαστήματος, λιστών με δυνατότητα αναζήτησης, όπως και άλλων γραφικών οντοτήτων της σελίδας
UpdateGoalsBanner	Συνάρτηση υπεύθυνη για την ανανέωση των στατιστικών στοιχείων της παρούσας σελίδας	Κάθε φορά που τα φίλτρα μεταβάλλονται, η συνάρτηση αυτή καλεί την όψη updateGoalsBanner και ανανεώνει τα στοιχεία στο πλαίσιο στατιστικών της σελίδας
ActivityGoalsDonutChart	Συνάρτηση υπεύθυνη για τη σχεδίαση του γραφήματος τύπου πίτας που παρουσιάζει μια ανάλυση στόχων επί των επιλεγμένων (από το φίλτρο) δραστηριοτήτων	Επικοινωνεί με την όψη updateActivityDonutChart και με τα δεδομένα αυτά, σχεδιάζει/κατασκευάζει το απαιτούμενο διάγραμμα ανάλυσης στόχων
ActivityAndCategoryGoals BarChart	Συνάρτηση υπεύθυνη για τη σχεδίαση των ραβδογραμμάτων ανάλυσης στόχων των	Επικοινωνεί με την όψη updateActivityAndCategory BarChart και με τα δεδομένα αυτά, κατασκευάζει τα

	δραστηριοτήτων και κατηγοριών	απαιτούμενα δύο (2) ραβδογράμματα. Το διάγραμμα δραστηριοτήτων αποτελεί μια διαφορετική οπτικοποίηση του διαγράμματος τύπου πίτας της συνάρτησης <code>ActivityGoalsDonutChart</code>
<code>ActivityAndFriendGoalsBarChart</code>	Συνάρτηση υπεύθυνη για τη σχεδίαση του διαγράμματος τύπου φυσαλίδας που πραγματοποιεί ανάλυση στόχων στους υπαρκτούς συνδυασμούς δραστηριοτήτων και φίλων	Επικοινωνεί με την όψη <code>updateActivityAndFriendBubbleChart</code> και με τα δεδομένα αυτά σχεδιάζει το απαιτούμενο διάγραμμα. Η συνάρτηση αυτή χρησιμοποιεί τη βοηθητική συνάρτηση <code>maxAxisCategoriesFriends</code> για να προσδιορίσει τη μέγιστη διάμετρο των μικρογραφιών πίτας που περικλείονται στα κελία του διαγράμματος αυτού
<code>ActivityAndObjectGoalsBarChart</code>	Συνάρτηση υπεύθυνη για τη σχεδίαση του διαγράμματος τύπου φυσαλίδας που πραγματοποιεί ανάλυση στόχων στους υπαρκτούς συνδυασμούς δραστηριοτήτων και αντικειμένων	Επικοινωνεί με την όψη <code>updateActivityAndObjectBubbleChart</code> και με τα δεδομένα αυτά σχεδιάζει το απαιτούμενο διάγραμμα. Η συνάρτηση αυτή χρησιμοποιεί τη βοηθητική συνάρτηση <code>maxAxisCategoriesObjects</code> για να προσδιορίσει τη μέγιστη διάμετρο των μικρογραφιών πίτας, ορίζοντας έτσι την απαραίτητη απόσταση μεταξύ των κελιών του πίνακα

Πίνακας 5.40 Συναρτήσεις και Μονάδες Χειρισμού Πελάτη Συσχετιζόμενες με την Οθόνη Goal Analytics

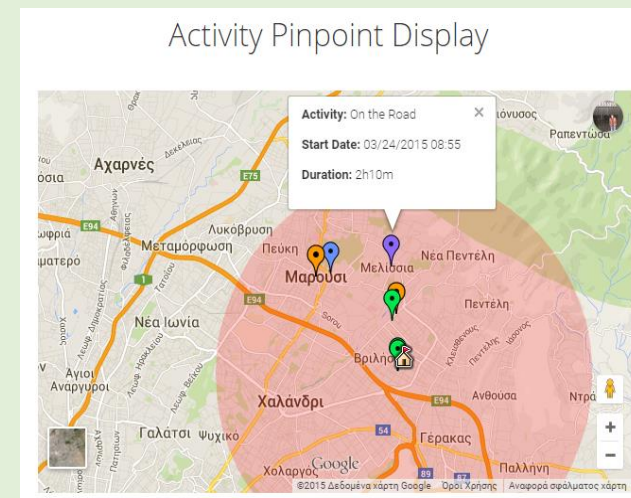
5.4.12 Οθόνη Place Analytics

1. Η οθόνη αυτή είναι η τέταρτη εκ των πέντε (5) σελίδων ανάλυσης, προσφέροντας πληροφορίες και διαγράμματα που σχετίζονται με τις τοποθεσίες στις οποίες λαμβάνουν χώρα οι δραστηριότητες.
2. Πραγματοποιεί μια ανάλυση με γεωγραφικό επίκεντρο, προσφέροντάς στο χρήστη τη δυνατότητα ανακάλυψης των κύριων τοποθεσιών δραστηριοποίησής του. Το τελευταίο με τη σειρά του, οδηγεί σε μια καλύτερη οργάνωση της καθημερινότητας.
3. Συσχετίζεται με τις τοποθεσίες του χρήστη που αποτελούν αντικείμενα της κλάσης Places του κεφαλαίου (5.3.4). Ο χρήστης συγκεκριμένα, έχει τη δυνατότητα να εξετάσει τις δραστηριότητες που πραγματοποιούνται γύρω από τις προσωπικές του τοποθεσίες, όπως και να ανακαλύψει αυτές που είναι πιο «ενεργές» από πλευράς δραστηριοτήτων.
4. Η οθόνη αυτή διαφέρει από τις υπόλοιπες στην πλευρά των αναλυτικών, καθώς εισάγει τη χρήση ενός χάρτη, πάνω στον οποίο απεικονίζεται μέρος της απαραίτητης πληροφορίας της παρούσας σελίδας ανάλυσης.
5. Η οθόνη αυτή αποτελείται από τρία (3) μέρη:
 - Το πλαίσιο φίλτρων, με τα οποία ο χρήστης καθορίζει το είδος πληροφορίας που του παρουσιάζεται. Η οθόνη αυτή παρέχει φίλτρο ημερομηνίας, φίλτρο προσωπικών τοποθεσιών, φίλτρο απόστασης και φίλτρο μετρικής. Το 1^ο ορίζει το εξεταζόμενο ημερολογιακό διάστημα, καθορίζοντας τα δεδομένα που θα επιλεχθούν. Το 2^ο καθορίζει την εξεταζόμενη προσωπική τοποθεσία (ή όλες) γύρω από την οποία θα εξεταστούν οι δραστηριότητες, το 3^ο την ακτίνα εξέτασης γύρω από την τοποθεσία (σε χιλιόμετρα, από 0.5 έως και 20) και το 4^ο το είδος της μετρικής που μπορεί να είναι είτε αριθμός συμβάντων (Number of Instances) είτε αριθμός ωρών (Time Elapsed). Η τελευταία ορίζει το μέγεθος το οποίο χρησιμοποιείται ως εξαρτημένη μεταβλητή στα διαγράμματα.
 - Το πλαίσιο στατιστικών, το οποίο παρουσιάζει σημαντικές τιμές και ποσοστά. Οι τιμές και οι μεταβλητές αλλάζουν σύμφωνα με τα επιλεγμένα φίλτρα.
 - Το πλαίσιο διαγραμμάτων και χάρτη, όπου παρουσιάζονται ένας συνδυασμός διαγραμμάτων και χάρτη ανάλογα με τις επιλογές του χρήστη στο πλαίσιο φίλτρων.
6. Στην οθόνη αυτή, τα φίλτρα δεν είναι αυτά που τροποποιούν τις απεικονίσεις. Η μοναδική εξαίρεση πραγματοποιείται όταν επιλεχθούν όλες οι προσωπικές τοποθεσίες προς εξέταση, όπου και εμφανίζεται ένα επιπλέον ραβδόγραμμα που πραγματοποιεί μια ανάλυση κατηγοριών δραστηριοτήτων για όλες τις προσωπικές τοποθεσίες. Οι συνδυασμοί τιμών φίλτρων, διαγραμμάτων και στατιστικών στοιχείων παρουσιάζονται στον παρακάτω πίνακα:

Επιλεγμένη Τοποθεσία	Διαθέσιμες Μετρικές	Στατιστικά Στοιχεία	Διαθέσιμα Διαγράμματα
Συγκεκριμένη ¹⁹ ή Όλες	<ul style="list-style-type: none"> • Αριθμός Συμβάντων • Αριθμός Ωρών 	<ol style="list-style-type: none"> 1. Αριθμός συνολικών δραστηριοτήτων ανεξάρτητα της τοποθεσίας τους 2. Συνολικές δραστηριότητες στην επιλεγμένη απόσταση από τη συγκεκριμένη ή όλες τις προσωπικές τοποθεσίες αντίστοιχα. 3. Ποσοστό επί των συνολικών δραστηριοτήτων, που λαμβάνουν χώρα στην επιλεγμένη απόσταση από τη συγκεκριμένη ή όλες τις προσωπικές τοποθεσίες αντίστοιχα. 4. Συνολικός χρόνος που ξοδεύτηκε στη συγκεκριμένη απόσταση από την επιλεγμένη ή όλες τις προσωπικές τοποθεσίες αντίστοιχα. 	<p>Το 1^ο διάγραμμα πραγματοποιεί μια ανάλυση των κατηγοριών και δραστηριοτήτων σύμφωνα με την επιλεγμένη μετρική. Εξετάζει τις δραστηριότητες που βρίσκονται εντός της εκάστοτε ακτίνας απόστασης από τη συγκεκριμένη ή όλες τις τοποθεσίες, και τις παρουσιάζει μέσω ενός διαγράμματος τύπου πίτας. Το εσωτερικό διάγραμμα παρουσιάζει τον καταμερισμό των κατηγοριών, ενώ το εξωτερικό των δραστηριοτήτων.</p> <div data-bbox="1192 688 1801 1140" style="text-align: center;"> <p>Category/Activity Distribution</p> <ul style="list-style-type: none"> Communication/Socializing Fun/Leisure/Hobbies Responsibilities Sports/Fitness Transportation Visiting Relatives Talking / Phoning At Dinner / Dining At the Cinema Out for a Coffee Studying Running Walking On the Road </div> <p>Η 2^η αναλυτική απεικόνιση περιλαμβάνει ένα χάρτη, πάνω στον οποίο απεικονίζεται η επιλεγμένη τοποθεσία (ή όλες), η επιλεγμένη ακτίνα απόστασης, καθώς και οι δραστηριότητες που ανήκουν στην ακτίνα αυτή (με</p>

¹⁹ Συμπεριλαμβάνεται και η επιλογή «παντού αλλού εκτός των προσωπικών τοποθεσιών»

εξαίρεση την επιλογή τοποθεσίας «παντού αλλού»). Οι τοποθεσίες παρουσιάζονται με το εικονίδιο του «σπιτιού», η επιλεγμένη απόσταση ως ένας ημιδιαφανής κόκκινος κύκλος γύρω από την εκάστοτε τοποθεσία, ενώ οι δραστηριότητες ως «πινέζες» (pins) ενός χάρτη, καθεμία από τις οποίες έχει το χρώμα της κατηγορίας στην οποία ανήκει. Όταν αυτές επιλεγθούν (πατηθεί κλικ), τότε εμφανίζονται συνοπτικές πληροφορίες τους.



Τέλος, όταν επιλεγθούν όλες οι τοποθεσίες προς εξέταση, τότε εμφανίζεται ένα 3^ο διάγραμμα το οποίο, βάσει της επιλεγμένης μετρικής, παρουσιάζει ένα καταμερισμό κατηγοριών των δραστηριοτήτων για κάθε τοποθεσία του χρήστη. Τελικός στόχος είναι η έμμεση κατηγοριοποίηση της ίδιας της τοποθεσίας.



Πίνακας 5.41 Ανάλυση Οθόνης Place Analytics

7. Οι όψεις του Django που συνδέονται με την οθόνη αυτή ακολουθούν ένα παρεμφερές μοτίβο με τις υπόλοιπες οθόνες ανάλυσης και παρουσιάζονται παρακάτω:

Όνομα και Σύνδεσμος Όψης	Χρήση	Περιγραφή
analytics_places (activitytracker/analytcs/places)	Υπεύθυνη για τη φόρτωση της σελίδας	Ενεργοποιείται όταν ο χρήστης επισκεφτεί τη σελίδα ανάλυσης τοποθεσιών. Η όψη αυτή επιστρέφει το template της οθόνης αυτής
updatePlacesBanner (activitytracker/analytcs/places/update/places_chartbanner)	Υπεύθυνη για την παροχή των στατιστικών τιμών στο πλαίσιο παρουσίασής τους στη σελίδα	Ενεργοποιείται κάθε φορά που τροποποιούνται τα φίλτρα και είναι υπεύθυνη για την ανανέωση ή αλλαγή των στατιστικών τιμών που παρουσιάζονται
updateActivitiesInPlace DonutChart (activitytracker/analytcs/places/update/categories_activities)	Υπεύθυνη για την παροχή δεδομένων στο διάγραμμα τύπου πίτας που παρουσιάζει τον καταμερισμό κατηγοριών και δραστηριοτήτων. Ακόμα παρέχει και τα απαραίτητα δεδομένα για την απεικόνιση του χάρτη	Καλείται κάθε φορά που τροποποιείται οποιοδήποτε φίλτρο της δραστηριότητας και είναι υπεύθυνο για να παρέχει τα απαραίτητα στοιχεία για τη σχεδίαση του διαγράμματος τύπου πίτας. Επειδή το παραπάνω διάγραμμα και η απεικόνιση χάρτη ανανεώνονται πάντα ταυτόχρονα, θεωρήθηκε αχρείαστη η ύπαρξη επιπλέον όψης. Συνεπώς, η ίδια αυτή όψη παρέχει τα δεδομένα και για τις δύο (2) γραφικές οντότητες
updateAllPlacesBarChart (activitytracker/analytcs/objects/update/places_all)	Υπεύθυνη για την παροχή δεδομένων στο επιπλέον ραβδόγραμμα που παρουσιάζεται μόνο όταν επιλεχθούν προς εξέταση όλες οι προσωπικές τοποθεσίες του χρήστη	Καλείται με την επιλογή όλων των τοποθεσιών και προσφέρει τα απαραίτητα δεδομένα για τη σχεδίαση του ραβδογράμματος αυτού, όπως αυτό παρουσιάστηκε στον παραπάνω πίνακα

Πίνακας 5.42 Όψεις Συσχετιζόμενες με την Οθόνη Place Analytics

8. Στην πλευρά του client, οι συναρτήσεις και μονάδες χειρισμού εμφανίζονται στον ακόλουθο πίνακα:

Όνομα Μονάδας/Συναρτησης	Χρήση	Περιγραφή
FilterChangedHandler	Μονάδα εντοπισμού αλλαγής φίλτρων, υπεύθυνη για την αλλαγή της αναπαράστασης	Κάθε φορά που ένα φίλτρο μεταβάλλεται, η μονάδα αυτή φροντίζει καλέσει τις όψεις που περιγράφηκαν παραπάνω, με στόχο την ανανέωση των δεδομένων που απεικονίζονται
GraphicEntitiesHandler	Μονάδα αρχικοποίησης και ενεργοποίησης εξωγενών γραφικών οντοτήτων	Η μονάδα αυτή είναι υπεύθυνη για την ενεργοποίηση ημερολογίων επιλογής χρονικού διαστήματος, λιστών με δυνατότητα αναζήτησης, κυλιόμενης μπάρας για την επιλογή ακτίνας απόστασης, καθώς και άλλων γραφικών οντοτήτων που χρησιμοποιεί η σελίδα
UpdatePlacesBanner	Συνάρτηση υπεύθυνη για την ανανέωση των στατιστικών στοιχείων της παρούσας σελίδας	Κάθε φορά που τα φίλτρα (πέρα από αυτό της μετρικής) μεταβάλλονται, η συνάρτηση αυτή καλεί την όψη updatePlacesBanner και ανανεώνει τα στοιχεία στο πλαίσιο στατιστικών της σελίδας Places Analytics
ActivityInPlaceDonutChart	Συνάρτηση υπεύθυνη για τη σχεδίαση του γραφήματος τύπου πίτας που παρουσιάζει τον καταμερισμό των κατηγοριών και δραστηριοτήτων σε επιλεγμένη απόσταση από την εκάστοτε τοποθεσία	Επικοινωνεί με την όψη updateActivitiesInPlaceDonutChart και με τα δεδομένα αυτά, σχεδιάζει/κατασκευάζει το απαιτούμενο διάγραμμα
PlacesCategoryDistributionBarChart	Συνάρτηση υπεύθυνη για τη σχεδίαση του ραβδογράμματος	Επικοινωνεί με την όψη updateAllPlacesBarChart και με τα δεδομένα αυτά,

	καταμερισμού των κατηγοριών των δραστηριοτήτων για όλες τις προσωπικές τοποθεσίες του χρήστη	κατασκευάζει το απαιτούμενο ραβδόγραμμα. Η συνάρτηση αυτή καλείται μόνο όταν επιλεγθούν όλες οι προσωπικές τοποθεσίες προς εξέταση και παρέχει μια γρήγορη εποπτεία του είδους δραστηριοτήτων που κυριαρχεί σε κάθε προσωπική τοποθεσία
InitializeAnalyticsPlacesMap	Συνάρτηση υπεύθυνη για την αρχικοποίηση και κατασκευή του χάρτη, πάνω στον οποίο παρουσιάζονται οι προσωπικές τοποθεσίες και οι δραστηριότητες του κάθε χρήστη	Καλείται κατά τη φόρτωση της ιστοσελίδας και ορίζει τη συμπεριφορά και τις διάφορες παραμέτρους του χάρτη της Google
UpdateMarkers	Συνάρτηση υπεύθυνη την παρουσίαση (και κατ' επέκταση ανανέωση) των δεδομένων που παρουσιάζει ο χάρτης.	Επικοινωνεί με την όψη updateActivitiesInPlace DonutChart και μέσω αυτής κατασκευάζει και ρυθμίζει τη συμπεριφορά των οντοτήτων του χάρτη. Συγκεκριμένα, κατασκευάζει τις «πινέζες» (pins) που αντιστοιχούν στις δραστηριότητες, τοποθετεί τις τοποθεσίες του χρήστη στο χάρτη και παρουσιάζει την επιλεγμένη ακτίνα απόστασης ως κύκλο

Πίνακας 5.43 Συναρτήσεις και Μονάδες Χειρισμού Πελάτη Συσχετιζόμενες με την Οθόνη Place Analytics

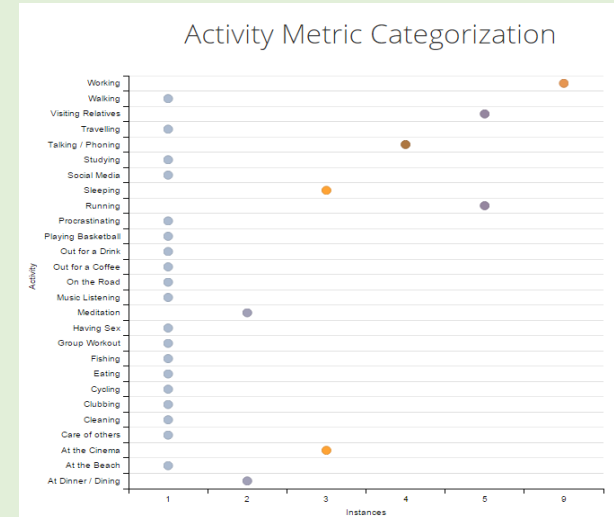
5.4.13 Οθόνη Activity Analytics

1. Η οθόνη αυτή είναι η τελευταία εκ των πέντε (5) σελίδων ανάλυσης, προσφέροντας πληροφορίες και διαγράμματα που σχετίζονται καθαρά και αποκλειστικά με τις ίδιες τις δραστηριότητες.
2. Η ανάλυση αυτή αντιμετωπίζει τις δραστηριότητες ως «μαύρα κουτιά», δίχως δηλαδή να πραγματοποιείται εστίαση στα πλαίσια που τις χαρακτηρίζουν.

3. Προσφέρει μια γενική αλλά βασική εποπτεία στην καθημερινότητα. Παρουσιάζει πολλαπλούς τύπους εποπτείας του καταμερισμού των καθημερινών δραστηριοτήτων, όπως επίσης και διαγράμματα παρακολούθησης της πορείας των δραστηριοτήτων τόσο από πλευράς απόδοσης όσο και σε σύγκριση με την κατηγορία στην οποία αυτές ανήκουν.
4. Η οθόνη αυτή αποτελείται από τρία (3) μέρη:
 - Το πλαίσιο φίλτρων, με τα οποία ο χρήστης καθορίζει το είδος πληροφορίας που του παρουσιάζεται. Η οθόνη αυτή παρέχει φίλτρο ημερομηνίας, φίλτρο δραστηριότητας και φίλτρο μετρικής. Το 1^ο ορίζει το εξεταζόμενο ημερολογιακό διάστημα καθορίζοντας τα δεδομένα που θα επιλεχθούν. Το 2^ο καθορίζει την υπό εξέταση δραστηριότητα (ή όλες) και το 3^ο το είδος της μετρικής που μπορεί να είναι είτε αριθμός συμβάντων (Number of Instances) είτε αριθμός ωρών (Time Elapsed). Αυτή ορίζει και το μέγεθος το οποίο χρησιμοποιείται ως εξαρτημένη μεταβλητή στα διαγράμματα.
 - Το πλαίσιο στατιστικών, το οποίο παρουσιάζει σημαντικές τιμές και ποσοστά που παρουσιάζουν ενδιαφέρον. Αυτά αλλάζουν τύπο ανάλογα με την επιλογή συγκεκριμένης ή όλων των δραστηριοτήτων στο αντίστοιχο φίλτρο, ενώ ανανεώνονται με την επιλογή διαφορετικού ημερολογιακού διαστήματος.
 - Το πλαίσιο διαγραμμάτων, όπου παρουσιάζονται κατάλληλα διαγράμματα ανάλογα με την επιλογή του χρήστη στο φίλτρο δραστηριοτήτων.
5. Στην οθόνη αυτή, μόνο το φίλτρο δραστηριοτήτων τροποποιεί τον τύπο των διαγραμμάτων και των στατιστικών στοιχείων. Οι συνδυασμοί τιμών φίλτρων, διαγραμμάτων και στατιστικών στοιχείων παρουσιάζονται στον παρακάτω πίνακα:

Επιλεγμένη Δραστ/τητα	Διαθέσιμες Μετρικές	Στατιστικά Στοιχεία	Διαθέσιμα Διαγράμματα
Όλες	<ul style="list-style-type: none"> • Αριθμός Συμβάντων • Αριθμός Ωρών 	<ol style="list-style-type: none"> 1. Αριθμός συνολικών δραστηριοτήτων επιλεγμένου χρονικού διαστήματος. 2. Ποσοστό μοναδικών δραστηριοτήτων εκ των συνολικών δραστηριοτήτων 3. Συνολικός χρόνος που έχει ξοδευτεί στο σύνολο των δραστηριοτήτων για το επιλεγμένο χρονικό διάστημα 	<p>Το 1^ο διάγραμμα παρουσιάζει τον – γνωστό πλέον – καταμερισμό καθημερινών δραστηριοτήτων και κατηγοριών, δίχως όμως αυτός να επηρεάζεται από τις τοποθεσίες, τους φίλους ή τα αντικείμενα. Το εσωτερικό διάγραμμα παρουσιάζει τις κατηγορίες, ενώ το εξωτερικό τις δραστηριότητες. Ανάλογα με την επιλεγμένη μετρική ο καταμερισμός πραγματοποιείται σε ώρες ή αριθμό συμβάντων.</p> <div data-bbox="1213 667 1843 1175" data-label="Figure"> <p>The figure is a donut chart titled "Category/Activity Distribution". It consists of two concentric rings. The inner ring represents categories, and the outer ring represents specific activities. A legend on the right lists 30 items, each with a corresponding color. The categories include Communication/Socializing, Fun/Leisure/Hobbies, Responsibilities, Self-care/Everyday Needs, Sports/Fitness, and Transportation. The activities include Visiting Relatives, Talking / Phoning, Social Media, At the Cinema, At Dinner / Dining, At the Beach, Clubbing, Fishing, Having Sex, Music Listening, Out for a Drink, Procrastinating, Working, Care of others, Cleaning, Studying, Sleeping, Meditation, Eating, Running, Cycling, Group Workout, Playing Basketball, Walking, On the Road, and Travelling.</p> </div> <p>Το 2^ο διάγραμμα παρουσιάζει ένα πίνακα που κατατάσσει τις διαφορετικές δραστηριότητες σε «ομάδες» ανάλογα με την επιλεγμένη μετρική. Συγκεκριμένα, ο κάθετος άξονας περιλαμβάνει τις δραστηριότητες του χρήστη, ενώ ο οριζόντιος είναι χωρισμένος σε διαστήματα της μετρικής,</p>

δηλαδή σε διαστήματα χρονικής διάρκειας ή αριθμού συμβάντων. Το διάγραμμα αυτό επιτρέπει στο χρήστη να ανακαλύψει εύκολα και γρήγορα τις δραστηριότητες αυτές που καταλαμβάνουν μεγάλο μέρος της καθημερινότητάς του.



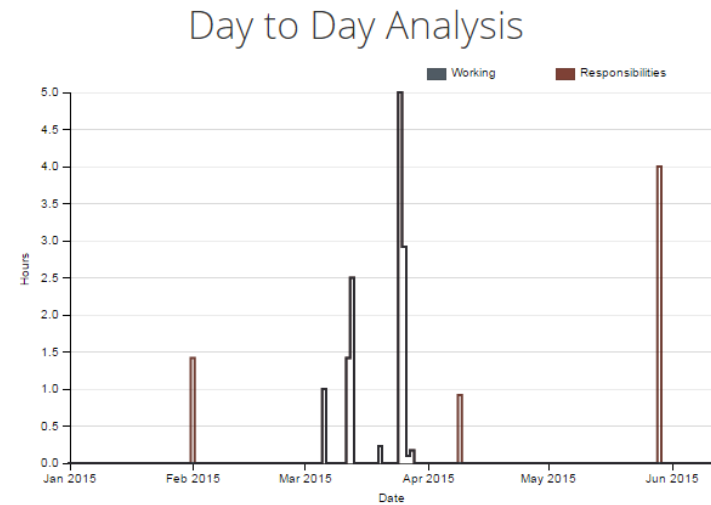
Συγκεκριμένη

- Αριθμός Συμβάντων (μόνο στο 1^ο διάγραμμα)
- Αριθμός Ωρών

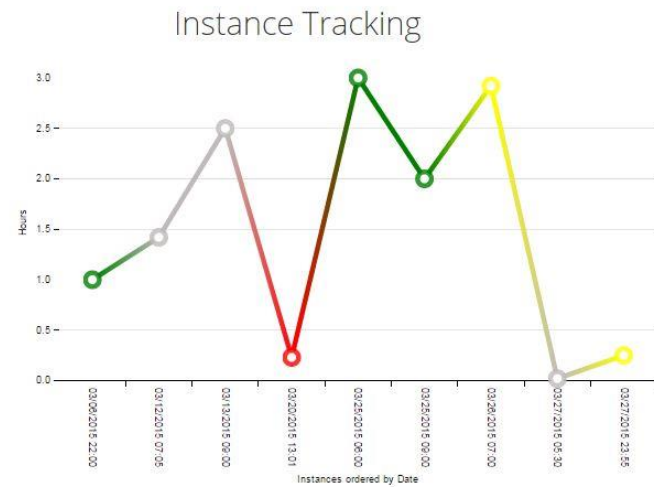
1. Αριθμός εμφανίσεων δραστηριότητας επιλεγμένου χρονικού διαστήματος.
2. Ποσοστό συμμετοχής συγκεκριμένης δραστηριότητας στην κατηγορία της.
3. Συνολικός χρόνος που έχει ξοδευτεί στη

Το 1^ο διάγραμμα παρουσιάζει την εξέλιξη και πορεία της συγκεκριμένης δραστηριότητας, όπως επίσης και της κατηγορίας της, στην πάροδο του επιλεγμένου χρονικού διαστήματος. Μέσω αυτού, ο χρήστης παρατηρεί τις εξάρσεις και υφέσεις της δραστηριότητας, οι οποίες μετριοούνται σε ώρες ή αριθμό συμβάντων.

συγκεκριμένη δραστηριότητα για το επιλεγμένο χρονικό διάστημα



Το 2^ο διάγραμμα παρουσιάζει μια διαφορετική αναπαράσταση της εξέλιξης της δραστηριότητας και ταυτίζεται με το διάγραμμα του συνδυασμού συγκεκριμένης δραστηριότητας και συγκεκριμένου φίλου/αντικειμένου των αντίστοιχων οθονών. Το διάγραμμα διαφέρει από το παραπάνω, καθώς αυτό περιλαμβάνει μόνο τις εμφανίσεις της δραστηριότητας, δίχως να αναπαριστά το «νεκρό» χρόνο ανάμεσα στις εμφανίσεις. Συνεπώς, προσφέρει μια οπτικοποίηση της πορείας της δραστηριότητας αμελώντας για την συχνότητα εμφάνισής της. Ταυτόχρονα, αυτό παρουσιάζει και την κατάσταση του στόχου της εκάστοτε εμφάνισης χρησιμοποιώντας το χρωματικό κώδικα που αναλύθηκε στο 1^ο διάγραμμα του πίνακα (5.33).



Πίνακας 5.44 Ανάλυση Οθόνης Activity Analytics

6. Οι όψεις που συνδέονται με την οθόνη αυτή είναι οι ακόλουθες:

Όνομα και Σύνδεσμος Όψης	Χρήση	Περιγραφή
analytics_activities (activitytracker/analytics/activities)	Υπεύθυνη για τη φόρτωση της σελίδας	Ενεργοποιείται όταν ο χρήστης επισκεφτεί τη σελίδα ανάλυσης δραστηριοτήτων. Η όψη αυτή επιστρέφει το template της οθόνης αυτής
updateActivitiesBanner (activitytracker/analytics/activities/update/activities_chartbanner)	Υπεύθυνη για την παροχή των στατιστικών τιμών στο πλαίσιο παρουσίασής τους στη σελίδα	Ενεργοποιείται κάθε φορά που τροποποιούνται τα φίλτρα δραστηριοτήτων και ημερολογιακού διαστήματος και είναι υπεύθυνη για την ανανέωση ή αλλαγή των στατιστικών τιμών που παρουσιάζονται
updateAllActivitiesCharts (activitytracker/analytics/activities/update/activities_all)	Υπεύθυνη για την παροχή δεδομένων στα δυο (2) διαγράμματα που αφορούν το σύνολο των δραστηριοτήτων	Καλείται κάθε φορά που επιλέγονται όλες οι δραστηριότητες στο φίλτρο δραστηριοτήτων και παρέχει τα απαραίτητα δεδομένα στο διάγραμμα καταμερισμού τύπου πίτας, όπως επίσης και στον πίνακα κατηγοριοποίησης διαφορετικών δραστηριοτήτων σύμφωνα με την επιλεγμένη μετρική
updateSingleActivityCharts (activitytracker/analytics/activities/update/activities_single)	Υπεύθυνη για την παροχή δεδομένων στα δύο (2) διαγράμματα που αφορούν την επιλογή συγκεκριμένης δραστηριότητας	Καλείται όταν επιλεχθεί συγκεκριμένη δραστηριότητα από το αντίστοιχο φίλτρο, και είναι υπεύθυνη για την παροχή των δεδομένων σχεδίασης των διαγραμμάτων εξέλιξης δραστηριότητας και αντίστοιχης κατηγορίας

Πίνακας 5.45 Όψεις Συσχετιζόμενες με την Οθόνη Activity Analytics

7. Οι μονάδες και συναρτήσεις χειρισμού στην πλευρά του πελάτη είναι υλοποιημένες με χρήση jQuery/javascript και παρουσιάζονται παρακάτω:

Όνομα Μονάδας/Συναρτησης	Χρήση	Περιγραφή
FilterChangedHandler	Μονάδα εντοπισμού αλλαγής φίλτρων, υπεύθυνη για την αλλαγή της αναπαράστασης	Κάθε φορά που τα φίλτρα δραστηριότητας και μετρικής μεταβάλλονται, η μονάδα αυτή φροντίζει καλέσει τις όψεις που περιγράφηκαν παραπάνω, με στόχο την ανανέωση των δεδομένων και γραφημάτων που απεικονίζονται
GraphicEntitiesHandler	Μονάδα αρχικοποίησης και ενεργοποίησης εξωγενών γραφικών οντοτήτων	Η μονάδα αυτή είναι υπεύθυνη για την ενεργοποίηση ημερολογίων επιλογής χρονικού διαστήματος, λιστών με δυνατότητα αναζήτησης, καθώς και άλλων γραφικών οντοτήτων που χρησιμοποιεί η σελίδα
UpdateActivitiesBanner	Συναρτηση υπεύθυνη για την ανανέωση των στατιστικών στοιχείων της παρούσας σελίδας	Κάθε φορά που τα φίλτρα (πέρα από αυτό της μετρικής) μεταβάλλονται, η συναρτηση αυτή καλεί την όψη updateActivitiesBanner και ανανεώνει τα στοιχεία στο πλαίσιο στατιστικών της σελίδας Activity Analytics
updateAllActivitiesCharts	Συναρτηση υπεύθυνη για τη σχεδίαση των δύο (2) διαγραμμάτων που αφορούν την επιλογή όλων των δραστηριοτήτων στο αντίστοιχο φίλτρο	Επικοινωνεί με την όψη updateAllActivitiesCharts και με τα δεδομένα αυτά, σχεδιάζει/κατασκευάζει τα απαιτούμενα διαγράμματα
updateSingleActivityCharts	Συναρτηση υπεύθυνη για τη σχεδίαση των δύο (2) διαγραμμάτων εξέλιξης της εκάστοτε συγκεκριμένης δραστηριότητας	Επικοινωνεί με την όψη updateSingleActivityCharts και με τα δεδομένα αυτά, κατασκευάζει τα απαιτούμενα διαγράμματα

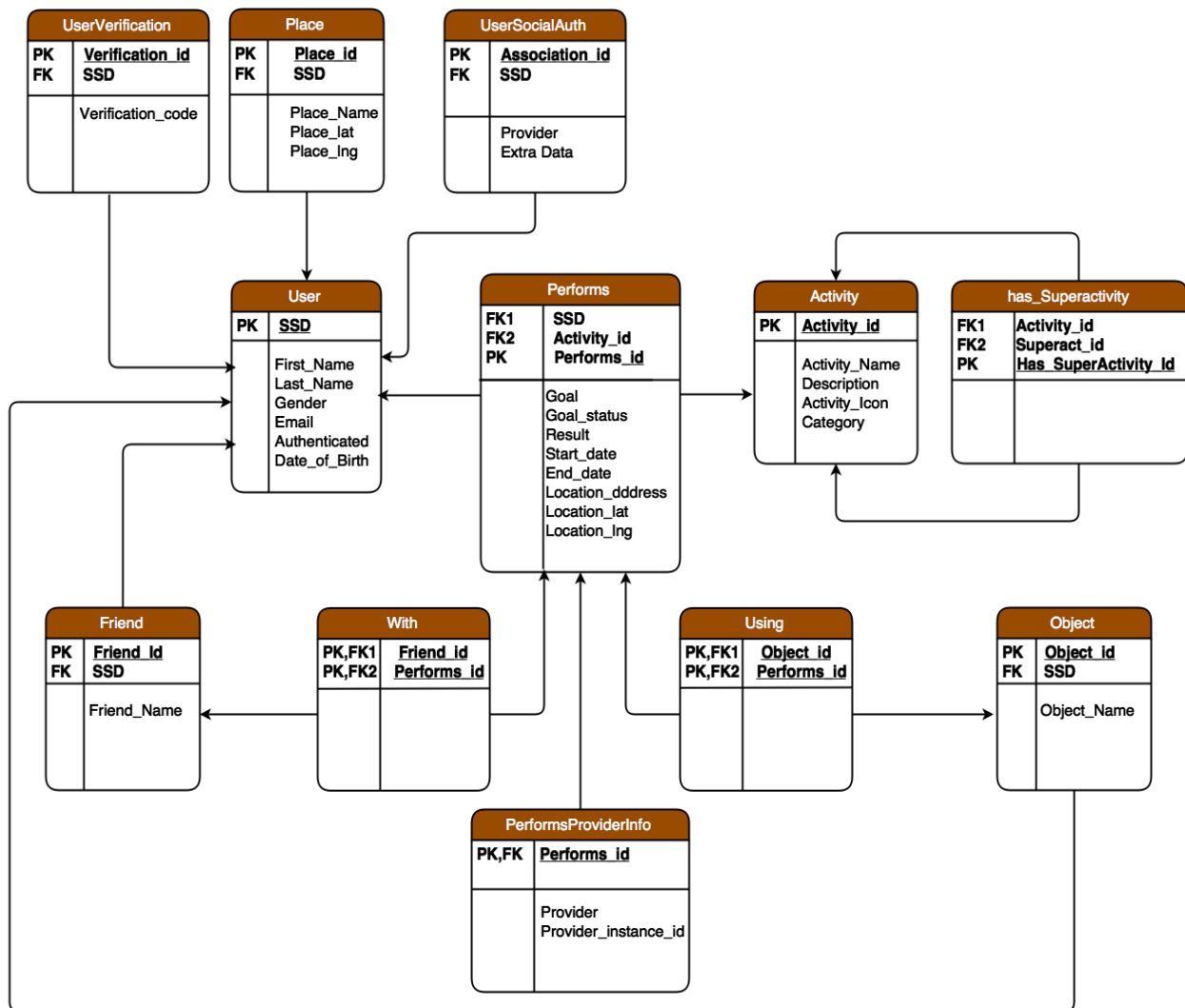
		γραμμής που παρουσιάζουν με διαφορετικό τρόπο την εξέλιξη της επιλεγμένης δραστηριότητας (και της κατηγορίας της) στο επιλεγμένο χρονικό διάστημα
--	--	---

Πίνακας 5.46 Συναρτήσεις και Μονάδες Χειρισμού Πελάτη Συσχετιζόμενες με την Οθόνη Activity Analytics

5.4 Βάση Δεδομένων

Η εφαρμογή μας κάνει χρήση της σχεσιακής βάσης δεδομένων που παρέχει αυτόματα το πλαίσιο του Django. Αυτή είναι μια ελαφριά (lightweight) βάση SQLite, η οποία ενδείκνυται για την εύκολη και γρήγορη αποθήκευση μικρού όγκου πληροφοριών. Το μειονέκτημά της είναι η αδυναμία ομαλής υποστήριξης πολλαπλής ταυτόχρονης πρόσβασης σε αυτή, με αποτέλεσμα την μη γρήγορη και κακή εμπειρία για τους χρήστες που χρησιμοποιούν ταυτοχρόνως την πλατφόρμα. Καθώς όμως η εφαρμογή υλοποιήθηκε στα πλαίσια διπλωματικής εργασίας, κρίθηκε περιττή η χρήση μεγαλύτερης ή δυνατότερης SQL βάσης δεδομένων.

Το σχήμα της βάσης δεδομένων, όπως προκύπτει από το διάγραμμα οντοτήτων – συσχετίσεων του σχήματος (4.14) παρουσιάζεται παρακάτω. Αξίζει να τονιστεί ότι, καθώς οι κλάσεις της εφαρμογής αποτελούν μοντέλα του Django τα οποία με τη σειρά του αποτελούν πίνακες στη σχεσιακή βάση δεδομένων, το διάγραμμα κλάσεων του σχήματος (5.3) σε συνδυασμό με την ανάλυση αρχιτεκτονικής κλάσεων του κεφαλαίου (5.4) είναι σε θέση να παρουσιάσει ήδη μια εικόνα των πινάκων της βάσης δεδομένων. Το ακόλουθο σχεσιακό διάγραμμα διαφέρει στις ονομασίες ορισμένων πεδίων σε σχέση με το σχήμα (4.14)



Σχήμα 5.3 Σχεσιακό Διάγραμμα Βάσης Δεδομένων Εφαρμογής

Κλείνοντας, αξίζει να αναφερθεί ότι το Django τροποποιεί αυθαίρετα τα ονόματα στη βάση δεδομένων, όπως και τα ονόματα των μοναδικών κλειδιών. Η συμπεριφορά αυτή δεν παρακάμφθηκε, συνεπώς η άμεση πρόσβαση στη βάση δεδομένων SQLite δίχως τη χρήση Django εντολών απαιτεί ενδεχομένως τη χρήση διαφορετικών ονομασιών.

6

Υλοποίηση Συστήματος

Έχοντας ολοκληρώσει το σχεδιασμό και την ανάλυση αρχιτεκτονικής του συστήματος, το παρόν κεφάλαιο θα εστιάσει στην ανάλυση ζητημάτων που αφορούν την υλοποίηση της εφαρμογής. Συγκεκριμένα, θα γίνει αναφορά στις πλατφόρμες, εργαλεία και βιβλιοθήκες που χρησιμοποιήθηκαν για την υλοποίησή της, θα αναλυθούν οι σημαντικές συναρτήσεις και αλγόριθμοι που χρησιμοποιήθηκαν, θα παρουσιαστούν στιγμιότυπα της εφαρμογής και θα αναφερθούν οι τρόποι με τους οποίους πραγματοποιήθηκε έλεγχος του τελικού συστήματος.

6.1 Πλατφόρμες, Βιβλιοθήκες & Προγραμματιστικά Εργαλεία

Για την ανάπτυξη της εφαρμογής Activity Tracker έγινε χρήση του ολοκληρωμένου περιβάλλοντος ανάπτυξης **PyCharm 4.5 (Professional Edition)**, στο οποίο υλοποιήθηκαν όλες οι συνιστώσες της εφαρμογής, τόσο σε επίπεδο εξυπηρετητή όσο και πελάτη. Για την εποπτεία αυτής, έγινε χρήση της τελευταίας έκδοσης των φυλλομετρητών **Google Chrome** και **Mozilla Firefox**. Τα επιπλέον προγραμματιστικά εργαλεία που χρησιμοποιήθηκαν σε κάθε πλευρά είναι τα ακόλουθα:

- **Πλευρά Εξυπηρετητή**

Για την ανάπτυξη της εφαρμογής στην πλευρά του server χρησιμοποιήθηκε το πλαίσιο (framework) **Django 1.8** που είναι υλοποιημένο σε γλώσσα προγραμματισμού Python, ενώ το σύνολο του προγραμματιστικού κώδικα στην πλευρά αυτή είναι υλοποιημένο και αυτό σε **Python 2.7.9**. Το Django έχει «ενσωματωμένη» μια βάση δεδομένων **SQLite** και ένα server **Apache** για χρήση εντός του διαστήματος ανάπτυξης του συστήματος. Τέλος, στην πλευρά του εξυπηρετητή έγινε χρήση των ακόλουθων εξωτερικών βιβλιοθηκών και εργαλείων:

1. **PythonSocialAuth**: Εξωτερική επιπρόσθετη μονάδα, υπεύθυνη για τη σύνδεση ενός χρήστη με υπηρεσίες όπως Facebook, Google, Runkeeper, όπως επίσης και άλλων υπηρεσιών που χρησιμοποιούν πλαίσια εξουσιοδότησης (authorization frameworks) OAuth και OAuth2. Έπαιξε καθοριστικό ρόλο στην δημιουργία HTTP αιτημάτων για εξουσιοδότηση, αποθηκεύοντας τα κλειδιά, δημιουργώντας τα

απαραίτητα μοντέλα ή τροποποιώντας τα ήδη υπάρχοντα, και συνδέοντας το χρήστη στην εφαρμογή. Η βιβλιοθήκη αυτή είναι υλοποιημένη σε Python και δεν υποστηρίζει αποκλειστικά και μόνο το πλαίσιο Django.

2. **Requests:** Βιβλιοθήκη υπεύθυνη για την εύκολη κατασκευή HTTP αιτημάτων σε Python, προσφέροντας μια εύκολη μέθοδο για την πραγματοποίηση αιτημάτων εξουσιοδότησης και ανάγνωσης δραστηριοτήτων από εξωτερική υπηρεσία. Αναφορικά, η βιβλιοθήκη PythonSocialAuth κάνει χρήση της βιβλιοθήκης αυτής και για τα δικά της αιτήματα.
3. **Pygeocoder:** Η βιβλιοθήκη αυτή είναι υπεύθυνη για την πραγματοποίηση αντίστροφης εύρεσης τοποθεσίας συντεταγμένων σε Python (πλευρά server) με βάση τους χάρτες της Google. Συγκεκριμένα, μέσω αυτής πραγματοποιήθηκε «μετάφραση» των γεωγραφικών συντεταγμένων των δραστηριοτήτων που προήλθαν από την υπηρεσία Runkeeper σε αλφαριθμητικές διευθύνσεις. Η βιβλιοθήκη αυτή αλληλεπιδρά με τη διαπροσωπεία (API) της υπηρεσίας Google πραγματοποιώντας HTTP αιτήματα και επιστρέφει στο server τα ζητούμενα δεδομένα.
4. **Python-oauthlib και Python-oauth2:** Η βιβλιοθήκες αυτές χρησιμοποιούνται από την PythonSocialAuth και επιτρέπουν την αλληλεπίδραση με τις διαπροσωπείες (APIs) των υπηρεσιών σύνδεσης που υλοποιούν πλαίσια εξουσιοδότησης OAuth ή OAuth2 για την πρόσβαση τρίτων εφαρμογών στις υπηρεσίες.

- **Πλευρά Πελάτη**

Για την υλοποίηση όλων των γραφιστικών και γραφικών οντοτήτων έγινε χρήση των προγραμματιστικών γλωσσών **HTML5**, **CSS3** και **JavaScript**. Ωστόσο, κυρίαρχο ρόλο παίζουν οι εξωτερικές βιβλιοθήκες και οι επιπρόσθετες οντότητες που παρουσιάζονται παρακάτω:

1. **Bootstrap 2.3.1:** Προσφέρει ένα έτοιμο σύνολο αντικειμένων υλοποιημένων με χρήση HTML, CSS και JavaScript, απαλλάσσοντας τον προγραμματιστή από την ευθύνη κατασκευής αναγκαίων στοιχείων της πλευράς αυτής. Μεταξύ άλλων, προσφέρει έτοιμες υλοποιήσεις επιπρόσθετων πλαισίων (modals), οργάνωσης οθονών, λιστών, μπαρών ολοκλήρωσης διαδικασιών, ειδοποιήσεων κ.ά.
2. **jQuery-1.9.1:** Βιβλιοθήκη JavaScript που προσφέρει μεθόδους και συναρτήσεις αλληλεπίδρασης με server και χρήστη, διευκολύνοντας κατά πολύ τον προγραμματιστή. Παραδείγματα αυτών είναι οι μονάδες χειρισμού ενέργειας του χρήστη στην ιστοσελίδα, η εύκολη αποστολή δεδομένων με A.J.A.X, η πραγματοποίηση HTTP αιτημάτων κ.ά. Χρησιμοποιείται για την υλοποίηση επιπρόσθετων προγραμματιστικών οντοτήτων.
3. **jQuery-ui-1.10.0:** Γραφική βιβλιοθήκη που χρησιμοποιείται για την απεικόνιση πολλών οντοτήτων που είναι υλοποιημένες με χρήση της βιβλιοθήκης jQuery.
4. **FullCalendar v1.6:** Επιπρόσθετη μονάδα (plugin) υλοποιημένη σε JavaScript και CSS, υπεύθυνη για την υλοποίηση του ημερολογίου, προσφέροντας μεθόδους ρύθμισης τόσο της απεικόνισης όσο και της συμπεριφοράς του.

5. **jQuery-dataTables v1.10.5:** Εξωτερική μονάδα (plugin) υπεύθυνη για την υλοποίηση των πινάκων. Η χρήση της έγκειται στη δυνατότητα ταξινόμησης, ασύγχρονης αναζήτησης, ρύθμισης συμπεριφοράς, καθώς και ρύθμισης συνδέσμου (συνεπώς όψης Django) τροφοδότησης δεδομένων του πίνακα.
6. **jQuery flot, flot-pie και flot-resize v0.7:** Βιβλιοθήκες υπεύθυνες για την γραφική κατασκευή και δημιουργία διαγραμμάτων πίτας, υλοποιημένες με χρήση JavaScript, jQuery και CSS. Χρησιμοποιούνται για την απεικόνιση του διαγράμματος καταμερισμού κατηγοριών των δραστηριοτήτων στη σελίδα Index.
7. **jQuery-chosen v0.9.8:** Εξωτερική επιπρόσθετη μονάδα (plugin) υλοποιημένη με χρήση jQuery και CSS, που προσφέρει υλοποίηση λιστών (dropdown lists) με δυνατότητα φιλτραρίσματος επιλογής, κατηγοριοποίησης και αναζήτησης στοιχείων εντός αυτών. Η κύρια της χρήση γίνεται στα φίλτρα των οθονών ανάλυσης.
8. **jQuery-tokenize v2.3:** Μονάδα υλοποιημένη σε jQuery και CSS, υπεύθυνη για τη δημιουργία πεδίων εισόδου με δυνατότητα φιλτραρίσματος και πολλαπλής επιλογής τιμών. Χρησιμοποιείται κατά την επιλογή φίλων και αντικειμένων στη λειτουργία προσθήκης και τροποποίησης δραστηριότητας.
9. **Bootstrap-clockpicker v0.0.7:** Μονάδα (plugin) υλοποιημένη σε jQuery και CSS, υπεύθυνη για την γραφική κατασκευή ενός αναλογικού ρολογιού για την επιλογή της χρονικής στιγμή έναρξης και λήξης μιας δραστηριότητας. Χρησιμοποιείται στις λειτουργίες προσθήκης και τροποποίησης δραστηριότητας.
10. **Moment v2.9.0:** Βιβλιοθήκη υλοποιημένη σε JavaScript που προσφέρει μεθόδους επεξεργασίας του χρόνου. Παραδείγματα αυτών αποτελούν η εύρεση της πρώτης ημέρας τυχαίου μήνα, η πραγματοποίηση πράξεων μεταξύ χρονικών στιγμών κ.λπ.
11. **Daterangepicker v1.3.19:** Μονάδα υλοποιημένη σε jQuery και CSS, που διευκολύνει την επιλογή χρονικών διαστημάτων μέσω της απεικόνισης δύο (2) μικρών ημερολογίων. Χρησιμοποιείται στο φίλτρο χρονικού διαστήματος σε όλες τις οθόνες ανάλυσης.
12. **Google Maps API v3:** Εξωτερική βιβλιοθήκη που προσφέρει ένα σύνολο μεθόδων, ικανό για τη δημιουργία των χαρτών της Google. Περιλαμβάνει όλα τα απαραίτητα εργαλεία για τη δημιουργία και τη ρύθμιση συμπεριφοράς ενός χάρτη.
13. **Bootstrap-slider v2.0.0:** Εξωτερική επιπρόσθετη μονάδα που υλοποιεί μια μονάδα κύλισης για επιλογή τιμής (slider). Χρησιμοποιείται στη σελίδα Places Analytics για την επιλογή ακτίνας απόστασης σε χιλιόμετρα.
14. **D3 v3.5.5:** Βιβλιοθήκη υλοποιημένη σε JavaScript που προσφέρει συναρτήσεις και μεθόδους για τη σχεδίαση απλών αλλά και σύνθετων διαγραμμάτων.
15. **Dimple v2.1.2:** Βιβλιοθήκη που προσφέρει μια προγραμματιστική διαπροσωπεία πάνω στη βιβλιοθήκη D3. Προσφέρει εύκολες εντολές, μεθόδους και συναρτήσεις καθιστώντας εύκολη την υλοποίηση διαγραμμάτων μέσω της παραπάνω χαοτικής βιβλιοθήκης. Η βιβλιοθήκη αυτή είναι υπεύθυνη για την κατασκευή όλων των διαγραμμάτων των σελίδων/οθονών ανάλυσης.

Κλείνοντας, αναφέρουμε συνοπτικά και άλλες βιβλιοθήκες που βοήθησαν στην ομαλή λειτουργία του συνόλου της εφαρμογής και αποτέλεσαν προαπαιτούμενο για διάφορες οντότητες. Αυτές είναι οι **jQuery.ui.touch.punch**, **jQuery-migrate-1.0.0** και **jQuery.iphone.toggle**. Το σύνολο των CSS και JavaScript αρχείων είναι διαθέσιμο στους αντίστοιχους υποφακέλους της εφαρμογής ([activitytracker/static/activitytracker/css](#) και [activitytracker/static/activitytracker/js](#)).

6.2 Λεπτομέρειες Υλοποίησης

Στο κεφάλαιο αυτό θα αναλυθούν οι συναρτήσεις οι οποίες έπαιξαν σημαντικό ρόλο στην εφαρμογή και θα παρουσιαστούν ορισμένοι εκ των αλγορίθμων που χρησιμοποιήθηκαν. Η εφαρμογή Activity Tracker δεν εστίασε στην απόδοση, συνεπώς δεν έκανε χρήση πολλών αλγορίθμων και μεθόδων βελτιστοποίησης, μιας και αυτό δεν απασχολούσε το αντικείμενο της παρούσας διπλωματικής. Τα σημεία που αξίζει να εστιάσουμε είναι:

1. Ο αλγόριθμος υπολογισμού απόστασης
2. Η μονάδα αυτόματου συγχρονισμού δεδομένων
3. Δομές Δεδομένων & Βελτιστοποιήσεις

6.2.1 Αλγόριθμος Υπολογισμού Απόστασης

Ο υπολογισμός της ευκλείδειας απόστασης είναι εύκολος όταν το σχήμα αναφοράς είναι το επίπεδο. Ωστόσο ο πλανήτης Γη δεν είναι επίπεδος αλλά σφαιρικός. Συνεπώς κατά τον υπολογισμό της απόστασης μεταξύ δύο (2) σημείων πρέπει να ληφθεί υπόψιν η καμπυλότητα της γης, η οποία αυξάνει την απόσταση, δεδομένου του ότι η έννοια της ευθείας ταυτίζεται πάντα με τη μικρότερη δυνατή απόσταση. Για το λόγο αυτό, το Activity Tracker κάνει χρήση του τύπου Haversine για τον υπολογισμό της επικαμπύλιας απόστασης πάνω στη Γη. Ο τύπος αυτός εφαρμόζει τον κλασικό τύπο της απόστασης, συνυπολογίζοντας την επιπλέον απόσταση που παράγεται από την καμπυλότητα. Ο αλγόριθμος παρουσιάζεται σε ψευδοκώδικα παρακάτω:

$$LatDist \leftarrow Lat2 - Lat1$$

$$LngDist \leftarrow Lng2 - Lng1$$

$$DistInKm \leftarrow 2R \arcsin \sqrt{\sin^2\left(\frac{LatDist}{2}\right) + \cos(Lat1) \cos(Lat2) \sin^2\left(\frac{LngDist}{2}\right)}$$

Οι τιμές $Lat1$ και $Lat2$ αποτελούν τις τιμές του γεωγραφικού πλάτους, ενώ οι τιμές $Lng1$ και $Lng2$ του γεωγραφικού μήκους των δύο (2) σημείων αντίστοιχα. Οι πρώτες μεταβάλλονται μεταξύ των πραγματικών τιμών -90 και 90, ενώ οι δεύτερες μεταξύ των πραγματικών τιμών -180 και 180. Η μεταβλητή $DistInKm$ παρουσιάζει την επικαμπύλια σημειακή απόσταση υπολογισμένη σε χιλιόμετρα. Ενδιαφέρον παρουσιάζει η άγνωστη μεταβλητή R που αντιστοιχίζεται στην ακτίνα της Γης, η οποία δεν είναι σταθερή αλλά εξαρτάται από την τοποθεσία. Συγκεκριμένα, η τιμή της ποικίλει από 6356.752 χιλιόμετρα στους πόλους έως

6378.137 χιλιόμετρα κοντά στον Ισημερινό. Για την εφαρμογή, η R ορίστηκε ως 6367, αποτελώντας έτσι τη μέση τιμή των δύο (2) ακραίων τιμών ακτίνας της Γης.

6.2.2 Μονάδα Αυτόματου Συγχρονισμού Δεδομένων

Η υποενότητα αυτή αναφέρεται στην λειτουργία συγχρονισμού δεδομένων των χρηστών που πραγματοποιείται σε συγκεκριμένες χρονικές στιγμές κατά τη διάρκεια της ημέρας, δίχως ο χρήστης να επιλέξει χειρωνακτικά τον συγχρονισμό. Η λειτουργία αυτή διευκολύνει το χρήστη (καθώς δεν απαιτείται να πραγματοποιεί ο ίδιος συγχρονισμό κάθε φορά που συνδέεται στο σύστημα), ενώ παράλληλα συμβάλλει στην ρεαλιστική απεικόνιση γραφημάτων και στατιστικών στοιχείων μέσω της ανανέωσης των δραστηριοτήτων σε τακτά χρονικά διαστήματα. Για την υλοποίηση της μονάδας αυτής, κατασκευάστηκε κατάλληλη συνάρτηση σε Python η οποία μετατράπηκε σε εντολή Django, προσφέροντας πρόσβαση μέσω της εντολής `manage.py autosync`. Η συνάρτηση αυτή κάνει χρήση του προγραμματιστή (scheduler) που προσφέρει η ίδια η Python 2.7.9 και επιτρέπει σε τμήματα κώδικα να «κοιμούνται» και να «ξυπνάνε» σε ορισμένα χρονικά διαστήματα. Λόγω της υποστήριξης μιας μόνο υπηρεσίας συγχρονισμού, κρίθηκε περιττός ο συγχρονισμός δεδομένων παραπάνω από μια (1) φορά την ημέρα. Έτσι, κάθε μέρα στις 6:00 (GMT +2) ενεργοποιείται η συνάρτηση, εκτελεί τη λειτουργία της, ορίζει την επόμενη στιγμή που θα «ξυπνήσει» ξανά και θέτει τον εαυτό της σε κατάσταση ύπνου. Η συνάρτηση συγχρονισμού διατρέχει όλους τους χρήστες που έχουν εξουσιοδοτήσει τις υποστηριζόμενες υπηρεσίες συγχρονισμού δραστηριοτήτων του Activity Tracker (στα πλαίσια της διπλωματικής μόνο τη RunKeeper) και εκτελεί την ίδια ακριβώς διαδικασία που πραγματοποιείται εάν καθένας εξ' αυτών επέλεγε χειρωνακτικά συγχρονισμό δραστηριοτήτων. Η μονάδα αυτή παρουσιάζεται παρακάτω σε ψευδοκώδικα:

```
Function AutoSyncActivities {
```

```
    Function SynchronizeActivities {
```

```
        for Provider in SupportedProviders
```

```
            UserSet ← Users that have connected with Provider
```

```
            for User in UserSet:
```

```
                ClientSecretKey ← getSecretKey(User)
```

```
                if CheckAuthentication(User,ClientSecretKey):
```

```
                    sync(Provider, User)
```

```
            NextWakeUp ← CreateWakeUpInstance(today + 1 day, 6:00:00)
```

```
            Scheduler ← Insert(NextWakeUp, SynchronizeActivities)
```

```

    }
    Scheduler ← CreateNewSchedulerInstance(wake, sleep)
    FirstWakeUp ← CreateWakeUpInstance(today, 6:00:00)
    Scheduler ← Insert(FirstWakeUp, SynchronizeActivities)
    Scheduler.run()
}

```

Η συνάρτηση *AytoSyncActivities* καλείται μέσω της εντολής Django *manage.py autosync*, όταν η συνάρτηση αυτή περικλεισθεί εντός της κλάσης *Command(BaseCommand)* της Django. Το πρώτο πράγμα που εκτελείται – στον παραπάνω κώδικα – είναι η δημιουργία ενός χρονοπρογραμματιστή (*scheduler*), ο οποίος «ξυπνάει» και «κοιμίζει» τον εαυτό του σε ορισμένους χρόνους. Έπειτα κατασκευάζεται το αντικείμενο *FirstWakeUp* της κλάσης χρόνου, που ορίζει την πρώτη στιγμή που θα «ξυπνήσει» ο χρονοπρογραμματιστής. Αυτή εισάγεται στη λίστα χρονικών στιγμών, με παράμετρο τη συνάρτηση *SynchronizeActivities* που θα εκτελέσει αυτός όταν ξυπνήσει. Η όλη διαδικασία ξεκινάει με την εντολή *Scheduler.run()*. Η συνάρτηση *SynchronizeActivities* βρίσκει τους χρήστες που έχουν εξουσιοδοτήσει τους λογαριασμούς τους με τους υποστηριζόμενους παρόχους δραστηριοτήτων (στην παρούσα στιγμή μόνο το *RunKeeper*), και για καθένα εξ' αυτών εξετάζει την ισχύ του κρυφού κλειδιού και καλεί τη συνάρτηση συγχρονισμού του ζευγαριού χρήστη – υπηρεσίας. Τέλος, όταν συγχρονιστούν οι δραστηριότητες όλων των χρηστών δημιουργεί νέο αντικείμενο χρόνου, το εισάγει στη λίστα χρονικών στιγμών που ο προγραμματιστής θα «ξυπνήσει» και ολοκληρώνεται η διαδικασία. Τονίζεται, ότι το τμήμα κώδικα εκτός της συνάρτησης *SynchronizeActivities* εκτελείται μόνο μια (1) φορά, καθώς όλες τις υπόλοιπες εκτελείται μόνο η συνάρτηση *SynchronizeActivities* μέχρι η διαδικασία να σταματήσει με χειρωνακτικό τρόπο από το διαχειριστή του συστήματος.

6.2.3 Δομές Δεδομένων & Βελτιστοποιήσεις

Στο υποκεφάλαιο αυτό θα κάνουμε μια συνοπτική ανάλυση στις κύριες δομές δεδομένων που χρησιμοποιήθηκαν, καθώς και σε ποικίλες μικρές βελτιστοποιήσεις στην αναζήτηση, ομαδοποίηση και άθροιση δεδομένων. Για την αποθήκευση και αποστολή δεδομένων η κύρια δομή που χρησιμοποιήθηκε ήταν αυτή του λεξικού (*dictionary*) που ορίζει η Python. Η δομή αυτή επιτρέπει εύκολη «1-1» αντιστοίχιση ενός «κλειδιού» με μια «τιμή» δημιουργώντας ζευγάρια συνδυασμών κλειδιού – τιμής. Η δομή είναι παρεμφερής με αυτή του πίνακα, ωστόσο διαφέρει από αυτόν, καθώς ο δείκτης προσπέλασης (κλειδί) μπορεί να είναι μια αυθαίρετη λέξη, ασυσχέτιστη και ανεξάρτητη από τις υπόλοιπες λέξεις κλειδιά. Ακόμα, η δομή αυτή επεκτείνεται έτσι ώστε να ένα κλειδί να μην αντιστοιχίζεται σε μια τιμή αλλά και σε μια άλλη δομή, όπως μια ολόκληρη συνάρτηση. Το τελευταίο χρησιμοποιείται για την επιλογή κατάλληλης συνάρτησης συγχρονισμού δεδομένων, σύμφωνα με την επιλεγμένη υπηρεσία συγχρονισμού.

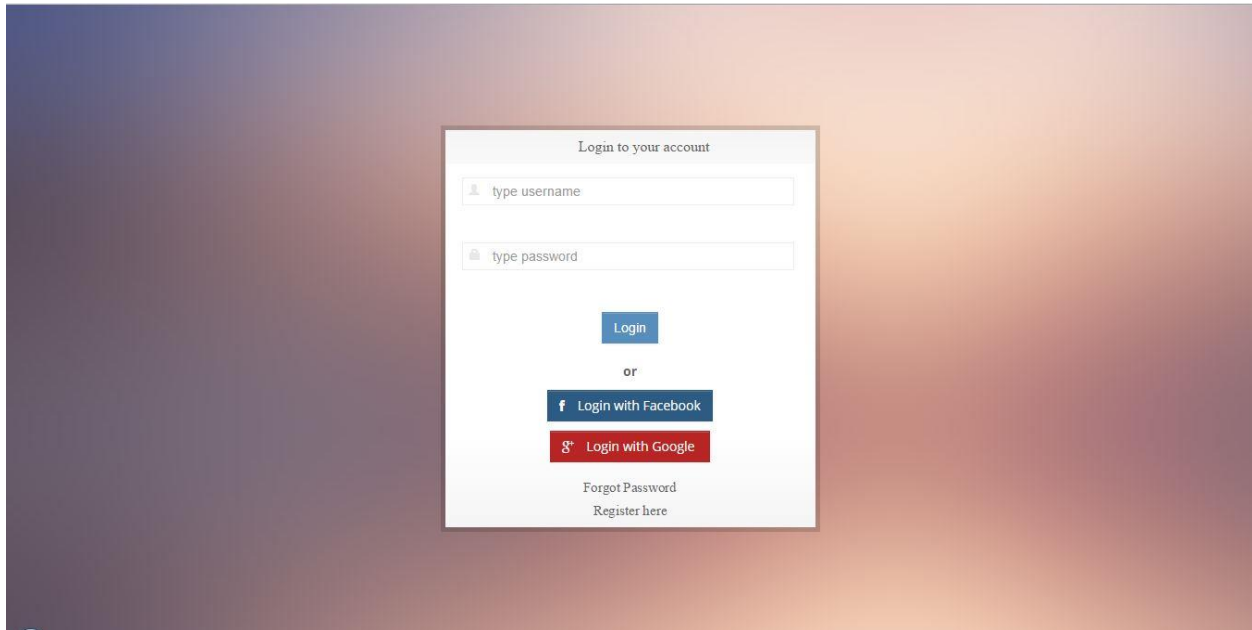
Η δομή του λεξικού χρησιμοποιήθηκε επίσης και για τη βελτιστοποίηση διαφόρων λειτουργιών όπως η διαγραφή διπλών στοιχείων από μια λίστα, η οποία χρειάστηκε μόνο $O(n)$ χρόνο εκτέλεσης. Ακόμα, η δυνατότητα του κλειδιού να μην αντιστοιχίζεται σε μια τιμή αλλά και σε μια ολόκληρη δομή δεδομένων, κατέστησε δυνατή την υλοποίηση ενός πολυεπίπεδου λεξικού (λεξικό που αντιστοιχίζεται σε λεξικό κ.ο.κ.), γεγονός που διευκόλυνε την πραγματοποίηση πολυεπίπεδων μερικών αθροισμάτων για τα διαγράμματα των σελίδων ανάλυσης της εφαρμογής.

Η τελευταία βελτιστοποίηση αφορά την έννοια της σελιδοποίησης που χρησιμοποιείται στην σελίδα Timeline. Επειδή οι δραστηριότητες του κάθε χρήστη μπορεί αυξηθούν σημαντικά σε αριθμό στην πάροδο του χρόνου, η άμεση αποστολή όλων από τον εξυπηρετητή στον πελάτη δύναται να δημιουργήσει πολλή «κίνηση» με αποτέλεσμα τόσο την αργή αποστολή τους, όσο και τη δυσαρέσκεια του χρήστη της εφαρμογής. Ο χρήστης δεν πρόκειται ποτέ να έχει ταυτόχρονη εποπτεία όλων των δραστηριοτήτων του στο χρονογράφημα (timeline) μιας και αυτές δεν θα είναι σε θέση να «χωρέσουν» σε μια μόνο οθόνη του φυλλομετρητή. Συνεπώς, αυτός χρειάζεται να μεταβεί χαμηλότερα στην ιστοσελίδα (scroll down) για να δει τις υπόλοιπες δραστηριότητες, γεγονός αίρει την ανάγκη ταυτόχρονης αποστολής των δεδομένων στην πλευρά του client. Έτσι τα δεδομένα «σελιδοποιήθηκαν», μια διαδικασία που περιλαμβάνει την ανάκτηση δεδομένων από τη βάση στο server, τον χωρισμό τους σε ίσου μεγέθους κομμάτια που ονομάζονται σελίδες και την αποστολή μιας σελίδας δεδομένων σε κάθε αίτημα του client. Στην αρχή, ο client αιτείται της 1^{ης} σελίδας δεδομένων και κάθε φορά που ο χρήστης μεταβαίνει (πραγματοποιεί scroll down) στο 85% του τέλους της οθόνης, ο client αιτείται της επόμενης σελίδας δεδομένων. Αυτή την σχεδιάζει, η οθόνη αυξάνεται σε ύψος και η διαδικασία επαναλαμβάνεται όταν ο χρήστης μεταβεί ξανά στο 85% του νέου ύψους των δεδομένων της οθόνης.

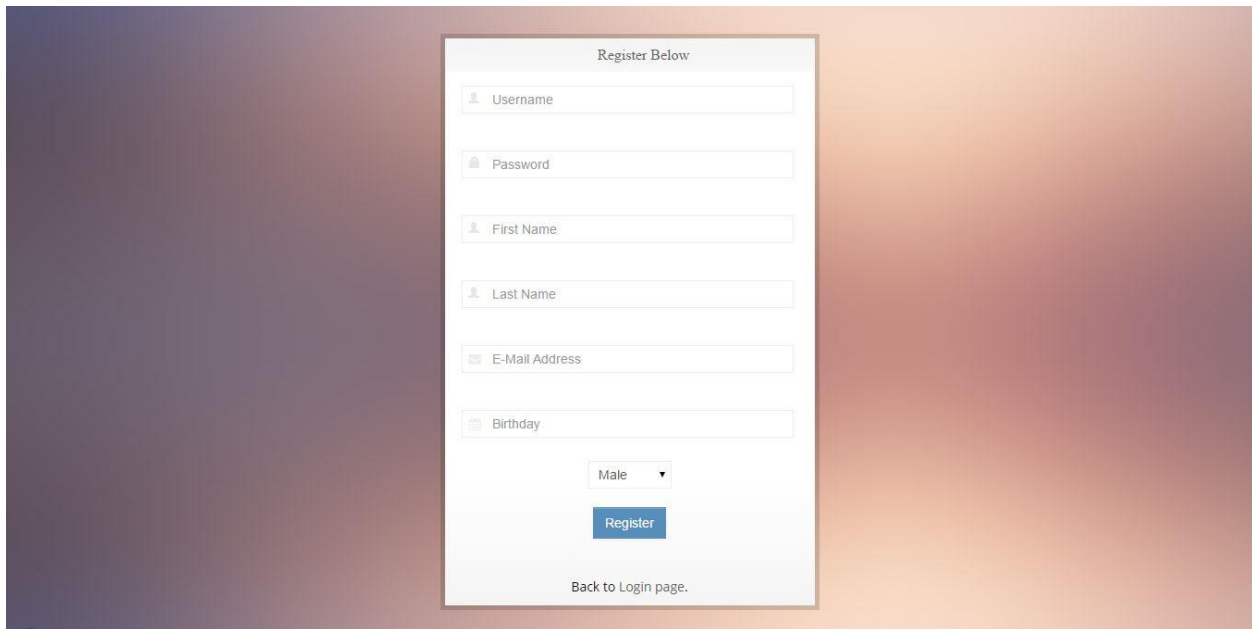
6.3 Στιγμιότυπα & Καθοδήγηση

Έχοντας ολοκληρώσει την περιγραφή, τις λειτουργίες, την αρχιτεκτονική και την υλοποίηση, θα παρουσιαστούν στιγμιότυπα (screenshots) απευθείας από την εφαρμογή, η οποία κατά το χρόνο συγγραφής του κειμένου είναι διαθέσιμη στο κοινό μέσω του συνδέσμου <http://snf-561492.vm.oceanos.grnet.gr:8083/activitytracker>. Το κεφάλαιο (5.1) παρουσίασε προσχέδια (Mock Ups), τα οποία έδρασαν ως γνώμονα για την τελική σχεδίαση που παρουσιάζεται παρακάτω. Η παρουσίασή τους θα πραγματοποιηθεί σε ζεύγη στιγμιότυπου – κειμένου, βάσει της σειράς ανάλυσης των λειτουργιών και οθονών στα κεφάλαια (4) και (5) αντίστοιχα. Στις ακόλουθες εικόνες χρησιμοποιήθηκε ο τυχαίος χρήστης με ψευδώνυμο «Envy».

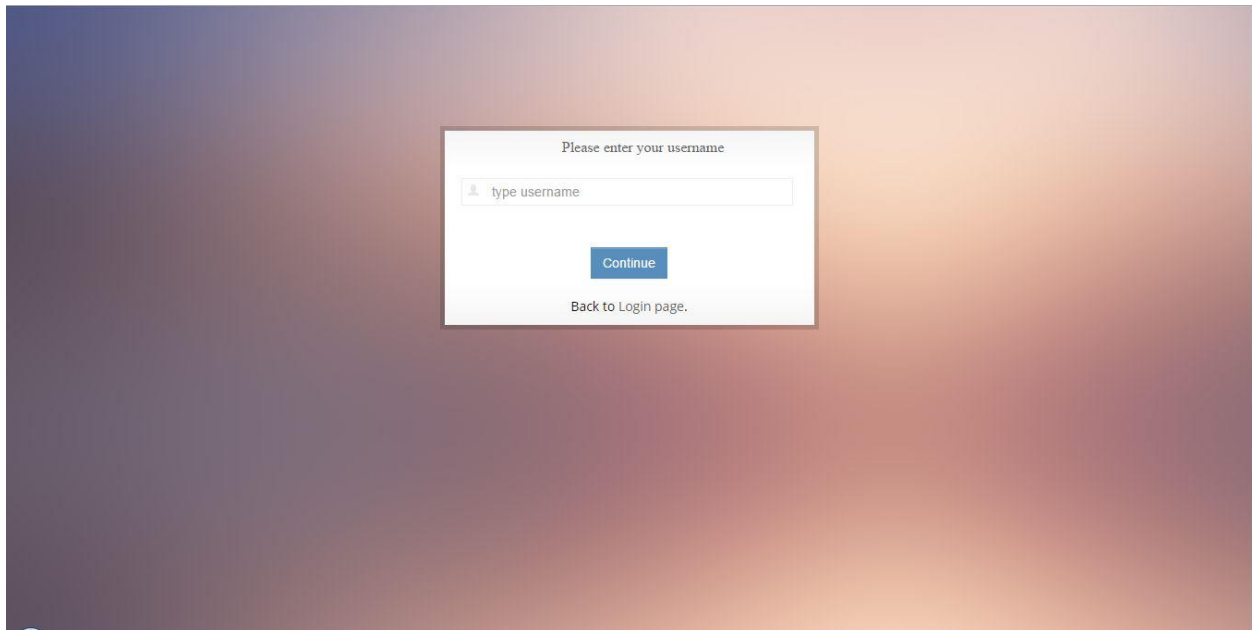
- Στιγμιότυπα Οθονών Εισόδου, Εγγραφής και Απώλειας Κωδικού



Εικόνα 6.1 Στιγμιότυπο Οθόνης Login



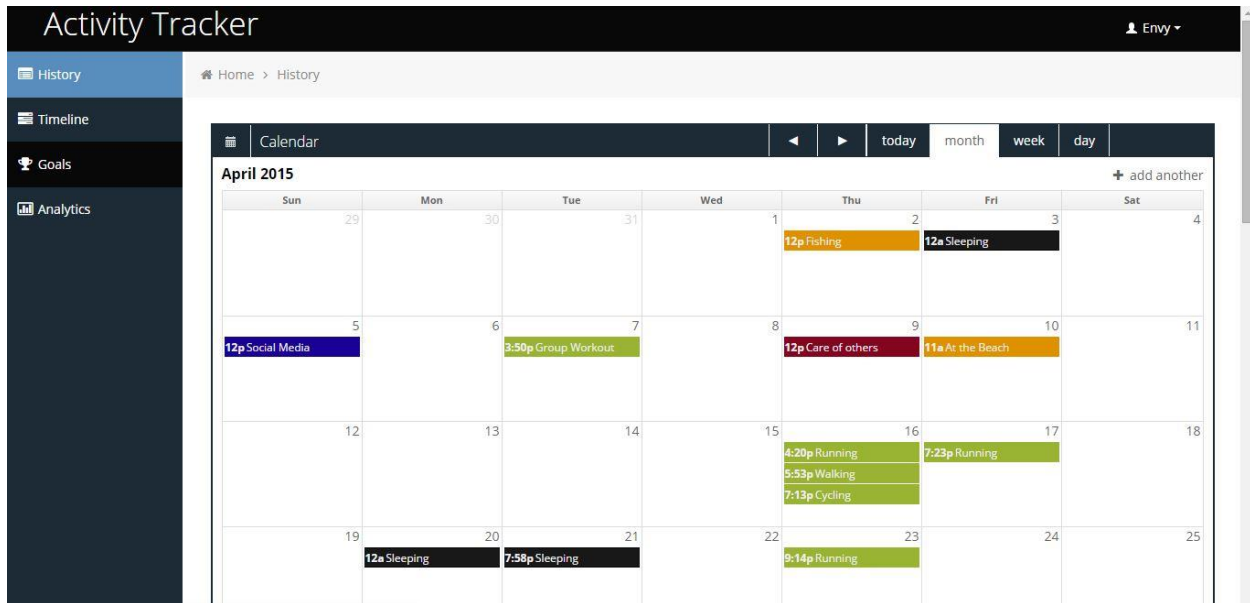
Εικόνα 6.2 Στιγμιότυπο Οθόνης Register



Εικόνα 6.3 Στιγμιότυπο Οθόνης Forgot Password

Στην οθόνη Login μπορεί κανείς εύκολα να παρατηρήσει τη δυνατότητα σύνδεσης με την υπηρεσία Facebook ή Google. Κάθε σφάλμα παρουσιάζεται κάτω από το πεδίο στο οποίο αναφέρεται, συνεπώς η χρήση ήδη χρησιμοποιούμενου email κατά τη λειτουργία χειρωνακτικής εγγραφής χρήστη στο σύστημα, θα οδηγήσει σε σφάλμα το οποίο θα παρουσιαστεί κάτω από το πεδίο εισαγωγής email. Οι λειτουργίες που επιτελούν οι οθόνες αυτές έχουν ήδη παρουσιαστεί σε προηγούμενα κεφάλαια, συνεπώς δεν χρήζουν περαιτέρω ανάλυσης.

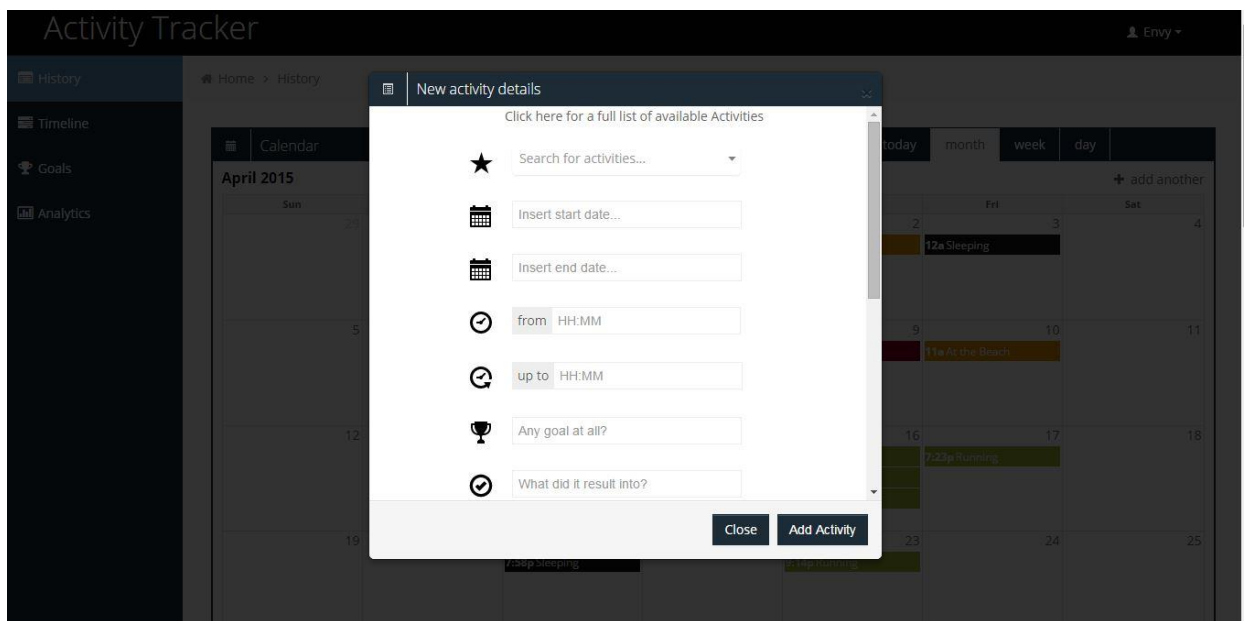
- Οθόνη Index & Διεπιφάνειες Προσθήκης, Επισκόπησης, Τροποποίησης και Διαγραφής Δραστηριότητας



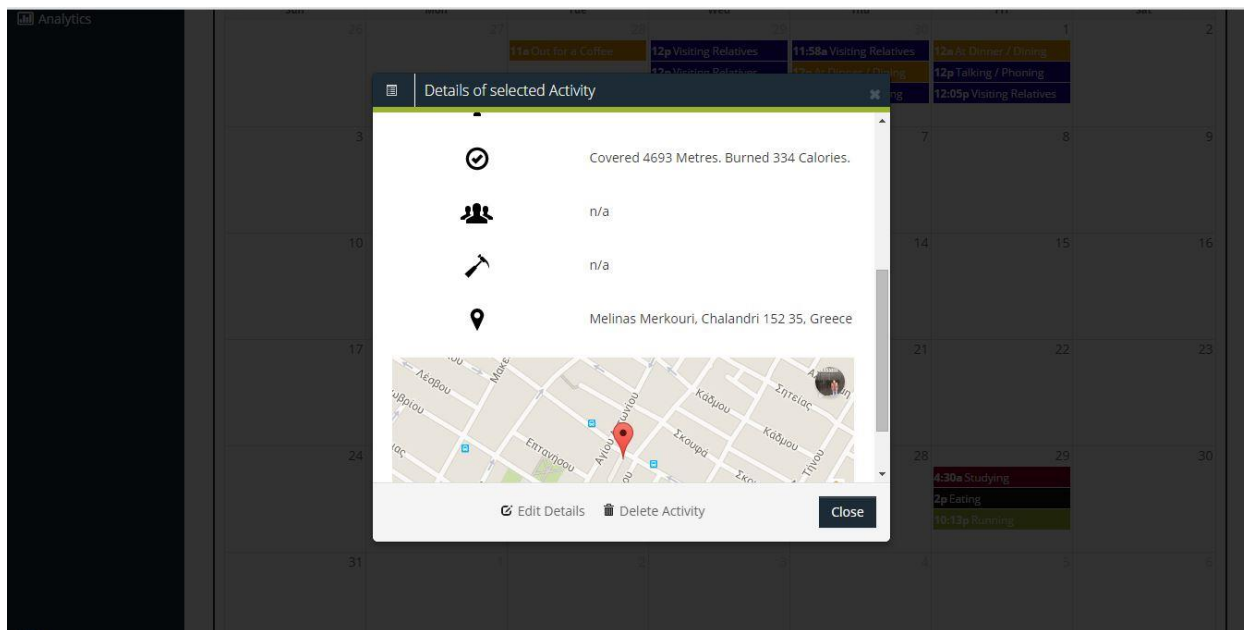
Εικόνα 6.4 Στιγμιότυπο Οθόνης Index (πάνω τμήμα)



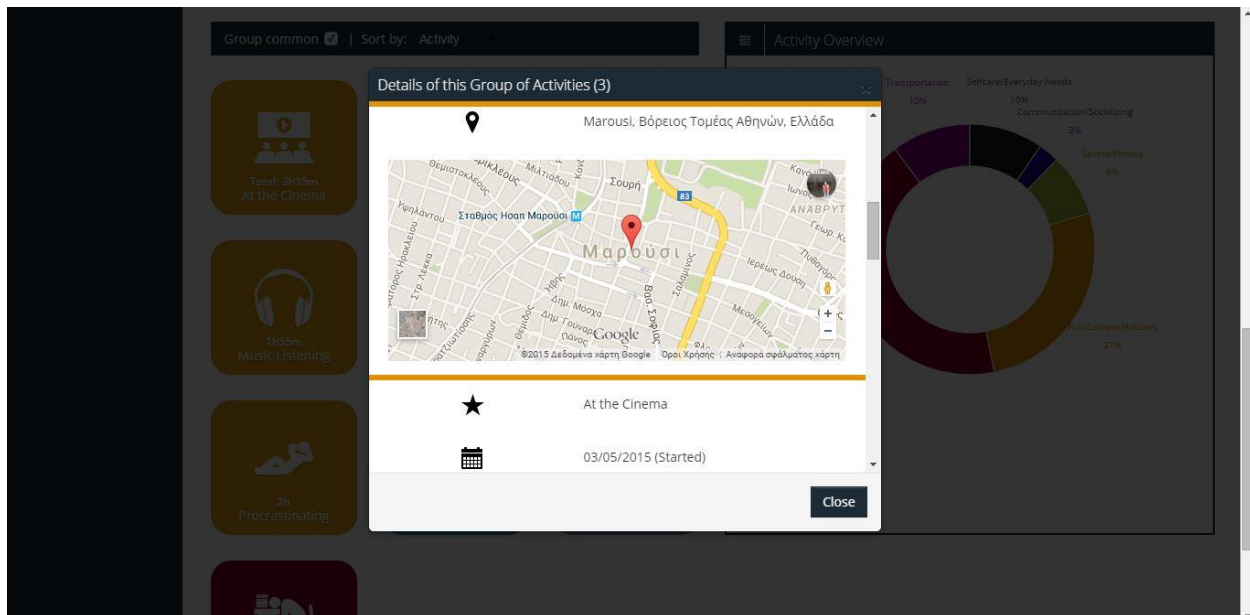
Εικόνα 6.5 Στιγμιότυπο Οθόνης Index (κάτω μέρος)



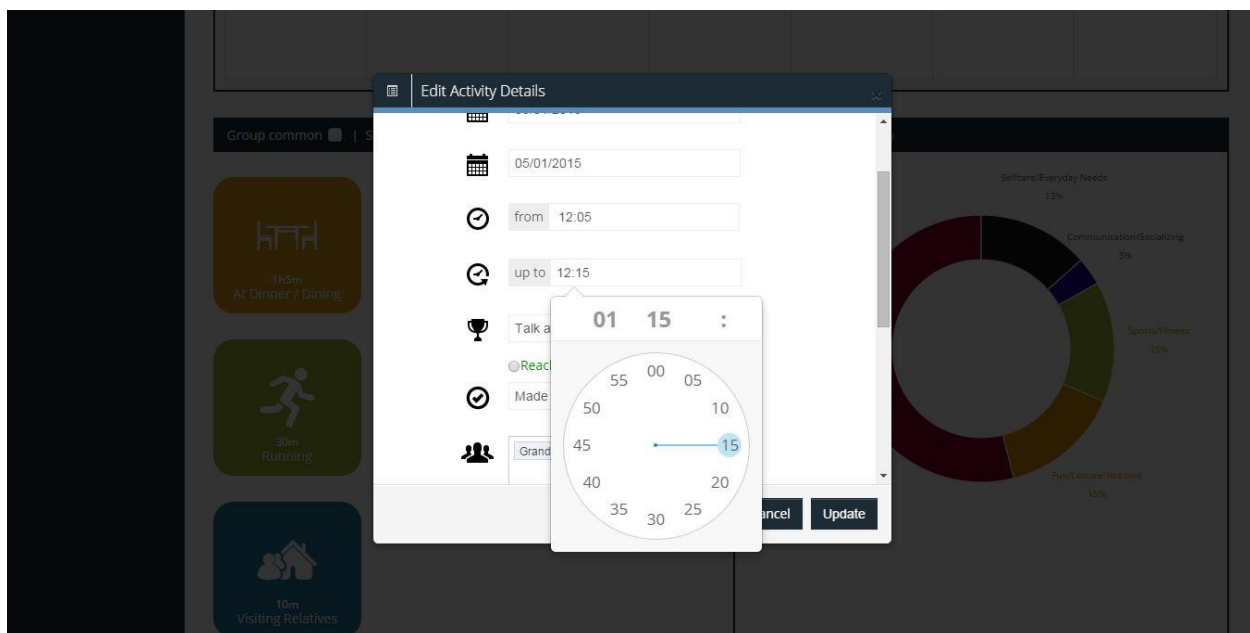
Εικόνα 6.6 Στιγμιότυπο Οθόνης Index κατά την Προσθήκη Δραστηριότητας



Εικόνα 6.7 Στιγμιότυπο Βοηθητικής Οθόνης showActivity



Εικόνα 6.8 Στιγμιότυπο Βοηθητικής Οθόνης showGroupActivity



Εικόνα 6.9 Στιγμιότυπο Βοηθητικής Οθόνης editActivity

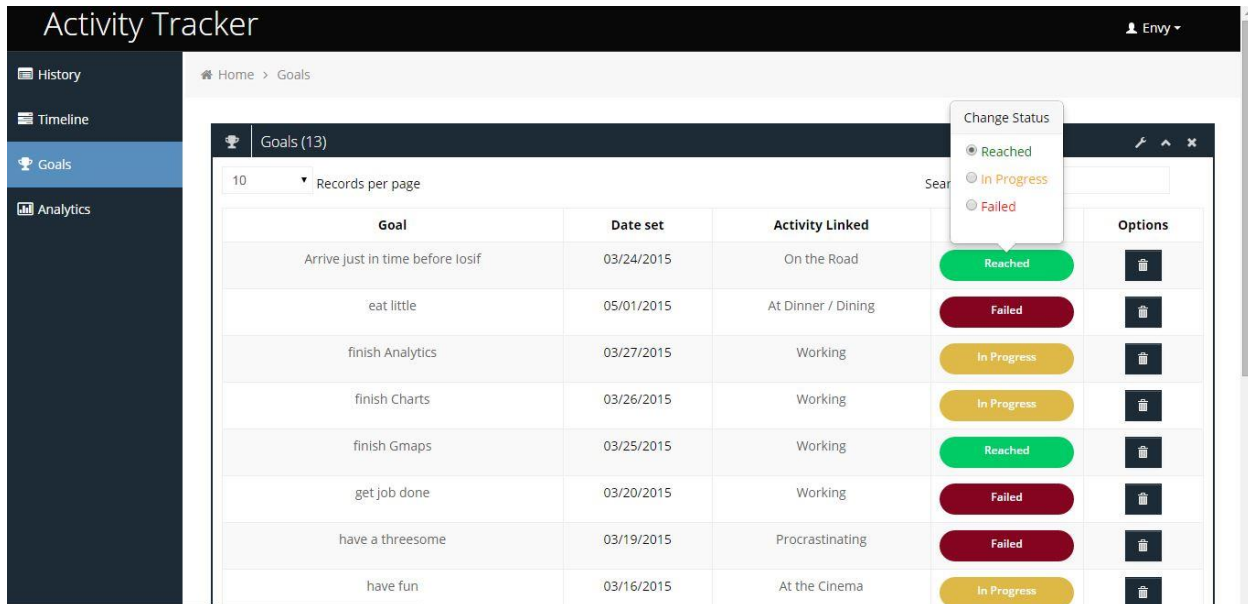
Στο στιγμιότυπο (6.4) παρατηρούμε την ύπαρξη της λειτουργίας του ημερολογίου. Ο χρήστης μπορεί να επιλέξει ανάμεσα στις ημερολογιακές όψεις ημέρας, εβδομάδας ή μηνός, τροποποιώντας τις παρουσιαζόμενες δραστηριότητες. Ο ίδιος, μπορεί να προσθέσει μια δραστηριότητα είτε επιλέγοντας το πλήκτρο « + add another » που βρίσκεται στην πάνω δεξιά γωνία του ημερολογίου, είτε πατώντας πάνω σε μια ημέρα του ημερολογίου. Η διεπιφάνεια προσθήκης δραστηριότητας που εμφανίζεται παρουσιάζεται στην εικόνα (6.6) και επιτρέπει την προσθήκη κάθε πληροφορίας που σχετίζεται με τη δραστηριότητα, όπως αυτές περιγράφηκαν κατά τη μοντελοποίηση του κεφαλαίου (2). Ο χρήστης μπορεί επίσης να

παρατηρήσει τις πλήρεις λεπτομέρειες μιας δραστηριότητας επιλέγοντάς την στο ημερολόγιο, όπου στην περίπτωση αυτή ενεργοποιείται η διεπιφάνεια showActivity της εικόνας (6.7) που προσφέρει πρόσβαση στις λειτουργίες τροποποίησης (εικόνα 6.9) και διαγραφής δραστηριότητας.

Στο κάτω μέρος της οθόνης Index (εικόνα 6.5) παρουσιάζεται η επισκόπηση πλέγματος σε συνδυασμό με το συνοπτικό διάγραμμα καταμερισμού κατηγοριών δραστηριοτήτων. Οι δραστηριότητες του πλέγματος (οι οποίες κάνουν χρήση του εικονιδίου που αντιστοιχεί σε καθεμία εξ' αυτών) ταυτίζονται με τις εκείνες που παρουσιάζονται την ίδια στιγμή στο ημερολόγιο και έχουν ως στόχο τη διαφορετική μορφή απεικόνισης της ίδιας πληροφορίας. Εμφανείς στην ίδια εικόνα είναι οι δυνατότητες ομαδοποίησης και ταξινόμησης των δραστηριοτήτων που προσφέρουν μια ποικιλία οργανώσεων του πλέγματος. Κάθε φορά που ο χρήστης επιλέγει μια δραστηριότητα στο πλέγμα ενεργοποιείται η διεπιφάνεια της εικόνας (6.7), ενώ κάθε φορά που επιλέγει μια ομαδοποιημένη δραστηριότητα ενεργοποιείται η διεπιφάνεια επισκόπησης ομαδοποιημένης δραστηριότητας που παρουσιάζεται στην εικόνα (6.8), με τις δραστηριότητες της ομάδας να τοποθετούνται η μία κάτω από την άλλη.

Τέλος, στην εικόνα (6.5) μπορεί να παρατηρήσει κανείς το διάγραμμα τύπου πίτας στο οποίο παρουσιάζεται ένα καταμερισμός των δραστηριοτήτων του ημερολογίου και πλέγματος σε επίπεδο κατηγοριών. Το διάγραμμα αυτό προσφέρει μια γενική εικόνα της καθημερινότητας του χρήστη στο επιλεγμένο (από το ημερολόγιο) χρονικό διάστημα.

- **Οθόνη Goals**



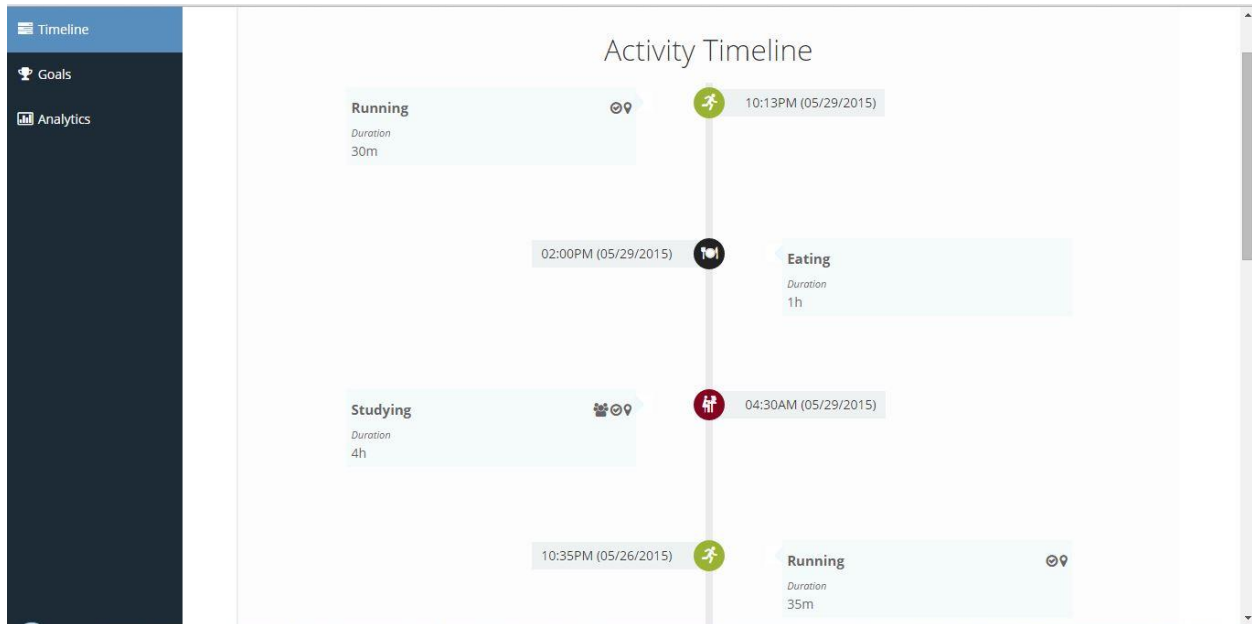
Goal	Date set	Activity Linked	Status	Options
Arrive just in time before losif	03/24/2015	On the Road	Reached	🗑️
eat little	05/01/2015	At Dinner / Dining	Failed	🗑️
finish Analytics	03/27/2015	Working	In Progress	🗑️
finish Charts	03/26/2015	Working	In Progress	🗑️
finish Gmaps	03/25/2015	Working	Reached	🗑️
get job done	03/20/2015	Working	Failed	🗑️
have a threesome	03/19/2015	Procrastinating	Failed	🗑️
have fun	03/16/2015	At the Cinema	In Progress	🗑️

Εικόνα 6.10 Στιγμιότυπο Οθόνης Goals

Η εικόνα (6.10) παρουσιάζει τον πίνακα στόχων όπως αυτός περιγράφηκε στο κεφάλαιο (5.4.5). Ο χρήστης επιλέγοντας τον τίτλο του στόχου έχει τη δυνατότητα να τον

τροποποιήσει, ενώ επιλέγοντας το όνομα της δραστηριότητας με την οποία αυτός συνδέεται, ενεργοποιεί τη διεπιφάνεια `showActivity` όπως αυτή παρουσιάζεται στην εικόνα (6.7). Πατώντας πάνω στην κατάσταση (status) του στόχου, ο χρήστης μπορεί να τον ανανεώσει (όπως είναι εμφανές στην παραπάνω εικόνα), ενώ επιλέγοντας το κουμπί της τελευταίας στήλης μπορεί να τον διαγράψει δίχως να διαγράψει τη δραστηριότητα με την οποία αυτός συνδέεται.

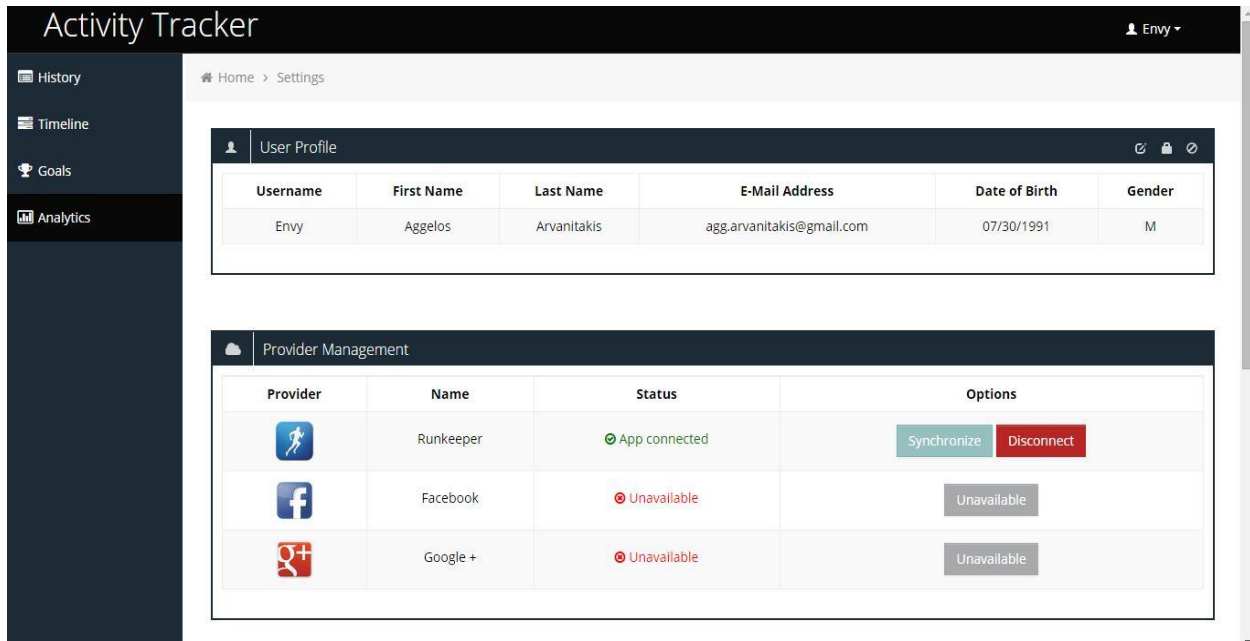
- **Οθόνη Timeline**



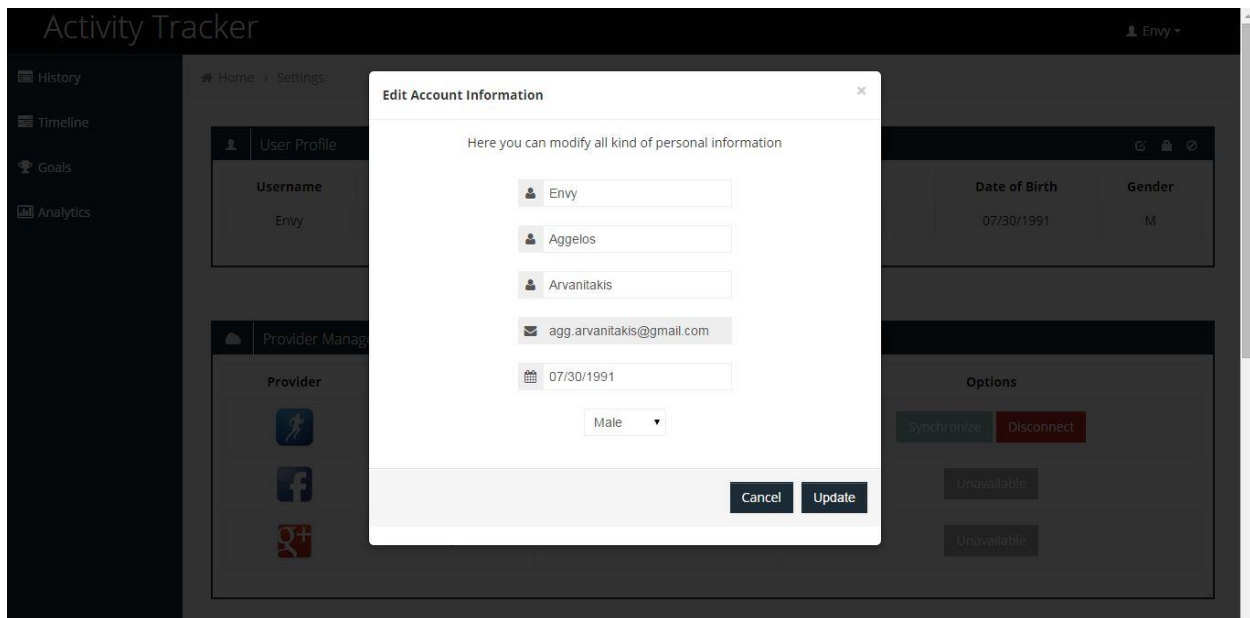
Εικόνα 6.11 Στιγμιότυπο Οθόνης Timeline

Η παραπάνω εικόνα παρουσιάζει το χρονογράφημα (Timeline) με τον άξονα του χρόνου να παρουσιάζεται ως κάθετη γραμμή στο κέντρο της οθόνης. Κάθε δραστηριότητα συνοδεύεται από το εικονίδιο και τον χρόνο έναρξής της, ενώ ο χρήστης μπορεί να έχει πρόσβαση στις λειτουργίες επισκόπησης, τροποποίησης και διαγραφής πατώντας πάνω στη δραστηριότητα (γαλάζιο ορθογώνιο). Οι δραστηριότητες διατάσσονται σε φθίνουσα σειρά, είναι σελιδοποιημένες (κεφάλαιο 6.2.3) και προσφέρουν μια γρήγορη ματιά στο πλαίσιο που τις συνοδεύει μέσω των μικρών μαύρων εικονιδίων στο πάνω δεξιά τμήμα κάθε δραστηριότητας. Συγκεκριμένα, τα εικονίδια αυτά συμβολίζουν την ύπαρξη ή μη συμπλήρωσης των πεδίων στα οποία αυτά αντιστοιχίζονται. Τα υποστηριζόμενα εικονίδια αφορούν τα πεδία στόχου, αποτελέσματος, αντικειμένων, συμμετεχόντων και τοποθεσίας. Έτσι για παράδειγμα η δραστηριότητα «Eating» έχει συμπληρωμένα μόνο τα απαραίτητα πεδία χρόνου έναρξης και λήξης, ενώ η δραστηριότητα «Studying» έχει συμπληρωμένα επιπλέον τα πεδία των συμμετεχόντων, αποτελέσματος και τοποθεσίας.

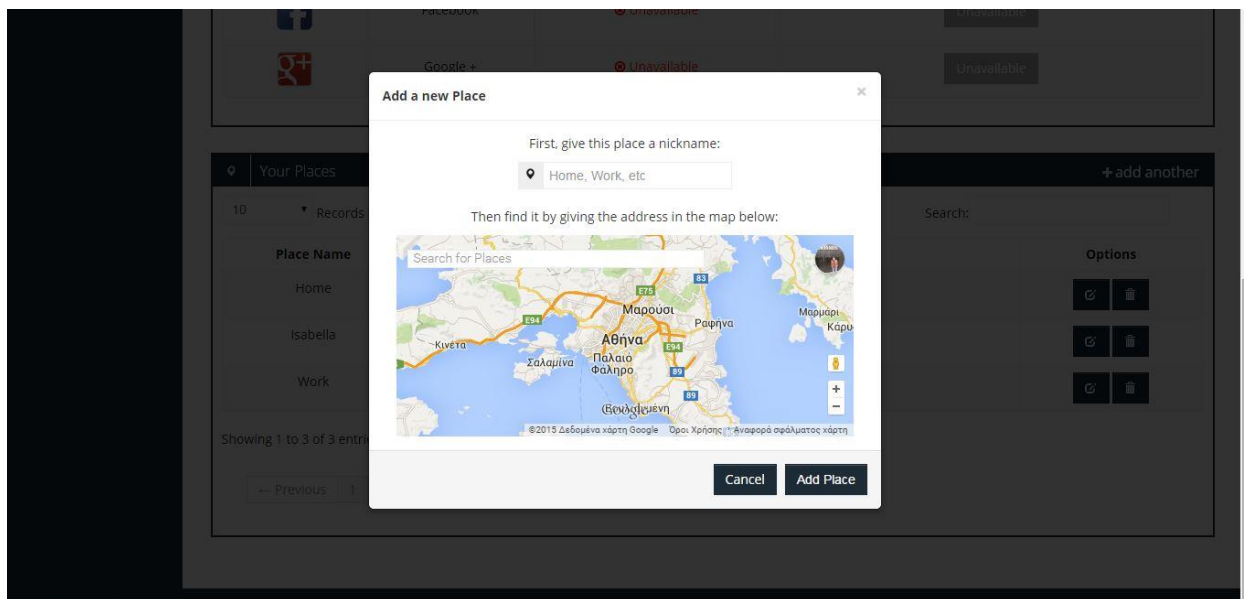
- Οθόνη Settings



Εικόνα 6.12 Στιγμιότυπο Οθόνης Settings



Εικόνα 6.13 Στιγμιότυπο Οθόνης Settings κατά την Επεξεργασία Προσωπικών Στοιχείων

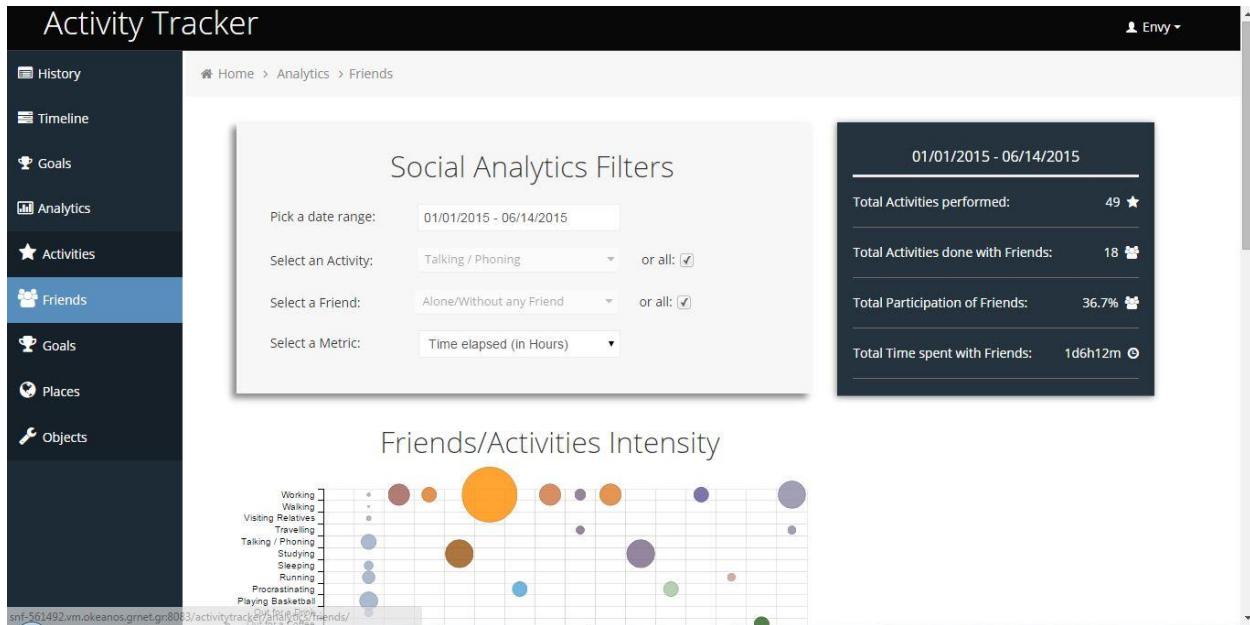


Εικόνα 6.14 Στιγμιότυπο Οθόνης Settings κατά την Προσθήκη Τοποθεσίας

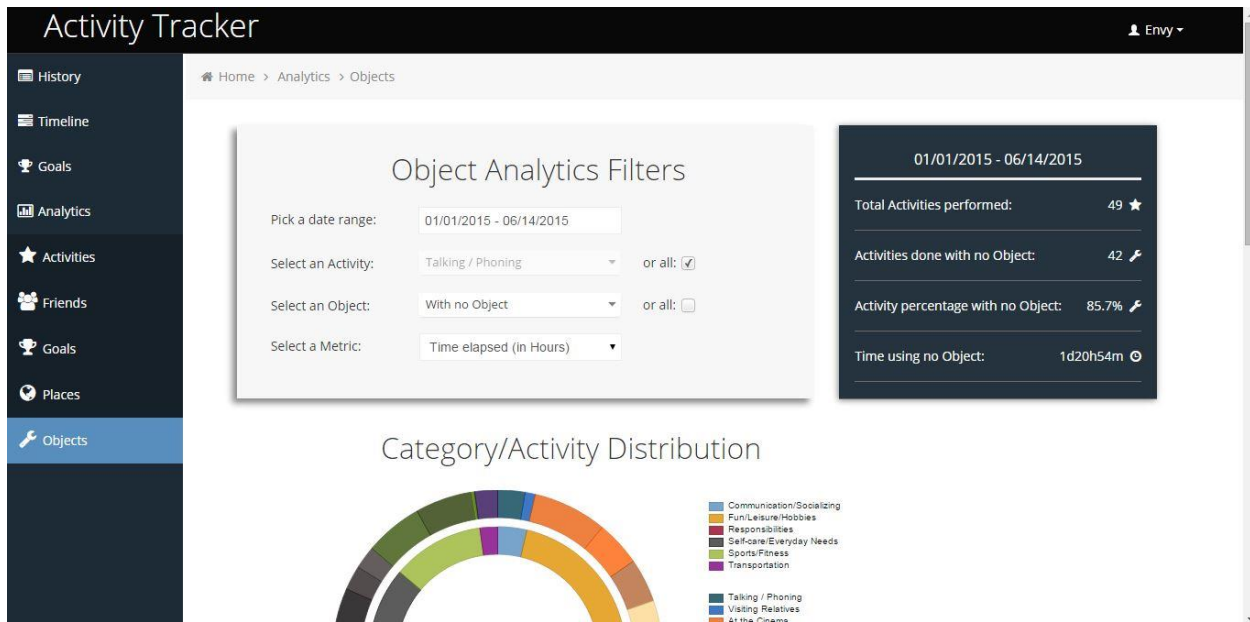
Η εικόνα (6.12) παρουσιάζει τους πίνακες στοιχείων χρήστη και συγχρονισμού δραστηριοτήτων αντίστοιχα. Στον 1^ο παρουσιάζονται όλες οι πληροφορίες που έχει συμπληρώσει ο χρήστης, προσφέροντας τη δυνατότητα τροποποίησης στοιχείων, αλλαγή κωδικού πρόσβασης και διαγραφή λογαριασμού κατ' αντιστοιχία των εικονιδίων που απεικονίζονται στην πάνω δεξιά γωνία του πίνακα. Συγκεκριμένα, η λειτουργία τροποποίησης στοιχείων γίνεται εμφανής στην εικόνα (6.13). Στον 2^ο πίνακα παρουσιάζονται οι υποστηριζόμενες υπηρεσίες συγχρονισμού δραστηριοτήτων, όπου κατά τη συγγραφή της διπλωματικής μόνο η υπηρεσία RunKeeper είναι ενεργή. Επιλέγοντας το κουμπί «Synchronize» ο χρήστης συγχρονίζει τις δραστηριότητές του με αυτές της υπηρεσίας, ενώ επιλέγοντας το κουμπί «Disconnect» το σύστημα διαγράφει το κρυφό κλειδί του χρήστη και δεν έχει πια πρόσβαση στις δραστηριότητές του. Αξίζει να τονιστεί ότι εάν ο χρήστης δεν έχει εξουσιοδοτήσει το Activity Tracker, τότε τα δύο (2) παραπάνω κουμπιά αντικαθίστανται από το κουμπί «Connect App», το οποίο εκκινεί τη διαδικασία εξουσιοδότησης σε εξωτερικό παράθυρο φυλλομετρητή.

Η εικόνα (6.14) παρουσιάζει το χαμηλότερο τμήμα της σελίδας Settings στο οποίο απεικονίζεται ο πίνακας προσωπικών τοποθεσιών του χρήστη κατά τη διαδικασία προσθήκης νέας τοποθεσίας. Το σύστημα χρησιμοποιεί τις τοποθεσίες αυτές στη σελίδα Place Analytics και ο χρήστης έχει τη δυνατότητα επεξεργασίας και διαγραφής τους μέσω των επιλογών της τελευταίας στήλης του πίνακα.

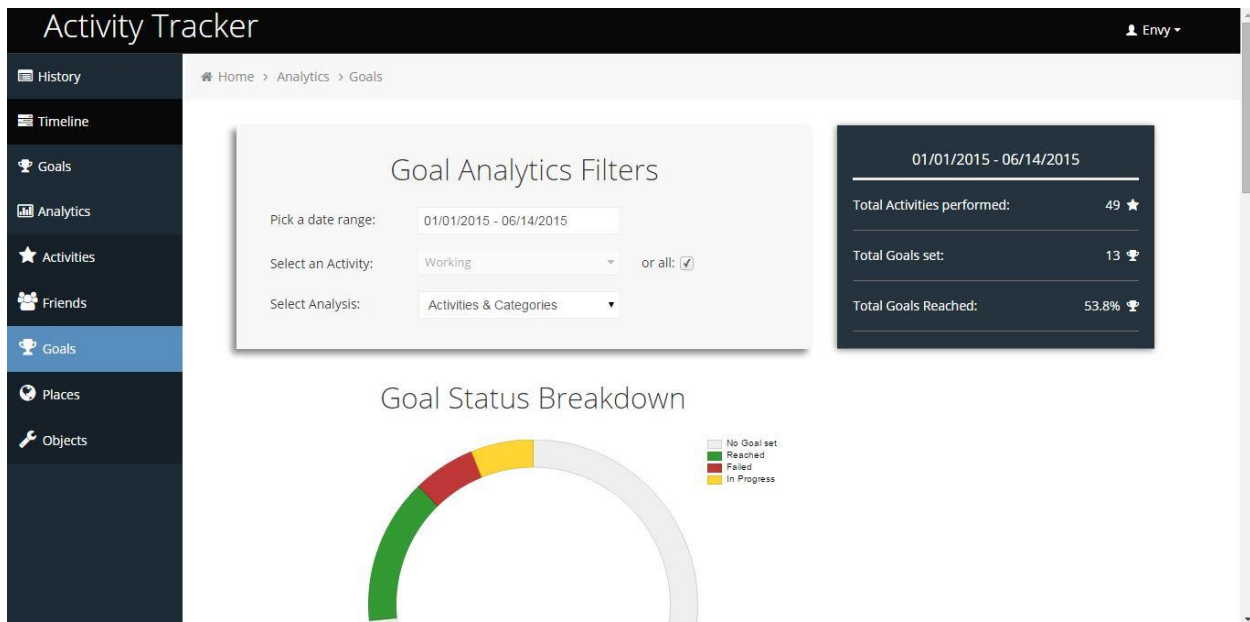
- **Οθόνες Ανάλυσης (Analytics)**



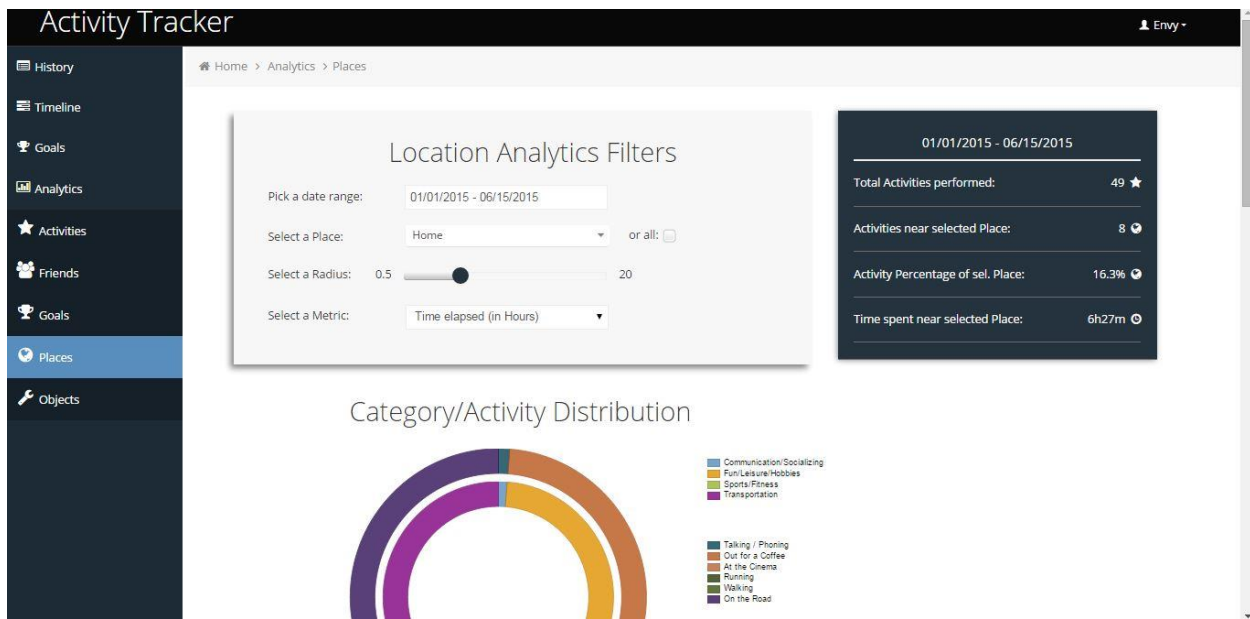
Εικόνα 6.15 Στιγμιότυπο Οθόνης Friend Analytics



Εικόνα 6.16 Στιγμιότυπο Οθόνης Object Analytics

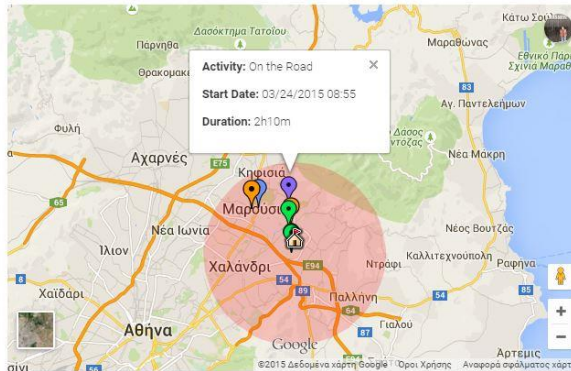


Εικόνα 6.17 Στιγμιότυπο Οθόνης Goal Analytics

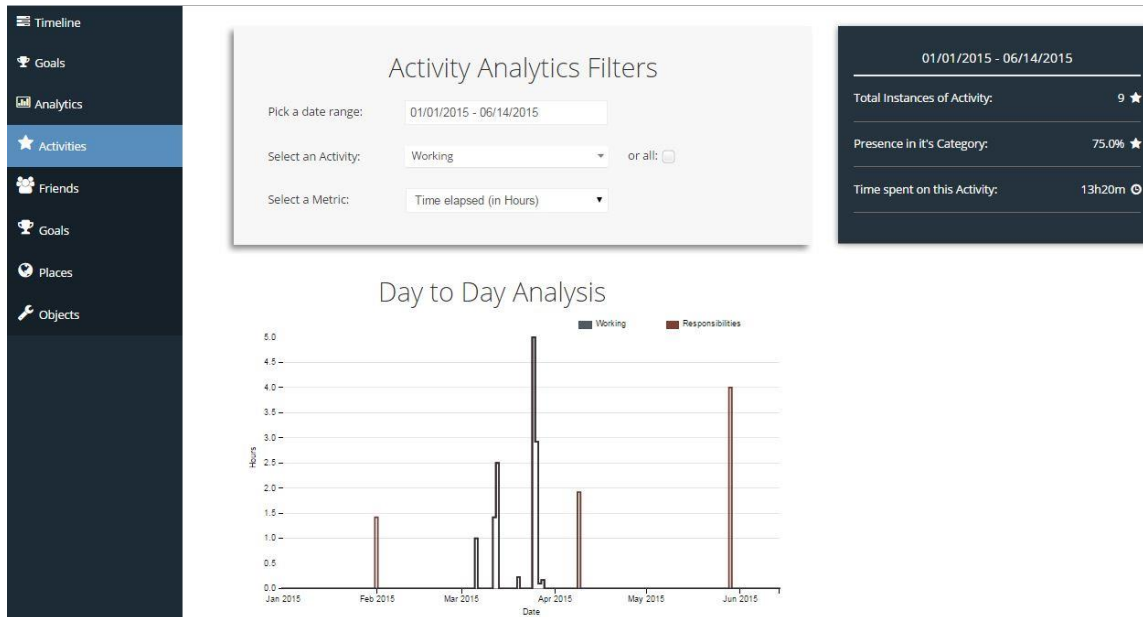


Εικόνα 6.18 Στιγμιότυπο Οθόνης Location Analytics

Activity Pinpoint Display



Εικόνα 6.19 Στιγμιότυπο Χάρτη Οθόνης Location Analytics



Εικόνα 6.20 Στιγμιότυπο Οθόνης Activity Analytics

Τα παραπάνω στιγμιότυπα παρουσιάζουν τις πέντε (5) διαφορετικές οθόνες ανάλυσης, στις οποίες ο χρήστης έχει πρόσβαση επιλέγοντας την καρτέλα «Analytics» στην αριστερή πλευρά κάθε σελίδας. Κάθε στιγμιότυπο παρουσιάζει τα φίλτρα της κάθε οθόνης, τα στατιστικά στοιχεία και ένα τυχαίο διάγραμμα. Με κάθε τροποποίηση ενός φίλτρου άμεσα τροποποιούνται και όλες οι αναπαραστάσεις της εκάστοτε σελίδας ανάλυσης. Η πλήρης

ανάλυση των διαγραμμάτων πραγματοποιήθηκε ήδη στα υποκεφάλαια (5.4.8) – (5.4.12) και κρίθηκε περιττή στο παρόν κεφάλαιο.

6.4 Μεθοδολογία & Αναλυτική Παρουσίαση Ελέγχου

Η παρούσα εφαρμογή φιλοξενήθηκε σε εξυπηρετητή του εργαστηρίου Συστημάτων Αποφάσεων & Διοίκησης με σκοπό να είναι προσβάσιμη από τον καθένα. Ο έλεγχος διαιρέθηκε σε δύο (2) στάδια, το **α' στάδιο ελέγχου** (alpha testing) και το **β' στάδιο ελέγχου** (beta testing).

Κατά το α' στάδιο, ο προγραμματιστής της εφαρμογής, πραγματοποίησε ποικίλα σενάρια χρήσης, ελέγχοντας όλες τις διαφορετικές εκβάσεις κάθε λειτουργίας. Συγκεκριμένα, ερευνήθηκαν όλες οι δυνατές εκδοχές έκβασης λειτουργιών σύνδεσης, εγγραφής, εξουσιοδότησης και αποσύνδεσης υπηρεσίας, επεξεργασίας δραστηριοτήτων, ακύρωσης λειτουργιών κατά τη διάρκεια αλληλεπίδρασης του συστήματος με τον εξυπηρετητή, εσφαλμένης και κενής εισόδου, ανακατεύθυνσης, συγχρονισμού δραστηριοτήτων και άμεσης ακύρωσής του, αναπαράστασης καμίας ή εξαιρετικά πολλών δραστηριοτήτων στις οθόνες ανάλυσης, εκμετάλλευσης λογαριασμών μέσω ηλ. ταχυδρομείου, πρόσβασης δεδομένων και οθονών δίχως σύνδεση, καθώς και πολλών άλλων εξαιρετικά εξειδικευμένων περιπτώσεων. Όταν δεν βρέθηκαν άλλα σφάλματα, έγινε μετάβαση στο β' στάδιο ελέγχου. Στο στάδιο αυτό ο έλεγχος της εφαρμογής πραγματοποιήθηκε από άτομα που δεν συμμετείχαν στην ανάπτυξη της εφαρμογής. Τα άτομα αυτά απασχολούνταν στο εργαστήριο Συστημάτων Αποφάσεων & Διοίκησης και παίζανε το ρόλο των τελικών χρηστών της εφαρμογής Activity Tracker. Πιο συγκεκριμένα, επιλέχθηκαν έξι (6) άτομα του εργαστηρίου τα οποία ήταν τελείως αμέτοχοι στη διαδικασία ανάπτυξης, δίχως να γνωρίζουν το σκοπό – ρόλο της εφαρμογής, και τους παρουσιάστηκε συνοπτικά η πλατφόρμα. Έπειτα, απαιτήθηκε από αυτούς η εγγραφή και η χρήση της εφαρμογής δίχως οδηγίες, τονίζοντας τυχόν συγχίσεις που θα βίωναν, όσο και ανωμαλίες ή σφάλματα που θα παρατηρούσαν στην πορεία. Στη συνέχεια, ακολούθησε η εκτέλεση ποικίλων σεναρίων χρήσης, κατά την οποία εφαρμόστηκε μια πρακτική εξωτερίκευσης των σκέψεων τους σε κάθε βήμα της εκάστοτε εκτελούμενης λειτουργίας, έχοντας ως στόχο την κατανόηση του τρόπου σκέψης των μη-εξοικειωμένων χρηστών της πλατφόρμας και τη βελτίωση της εμπειρίας τους. Μέσω του ελέγχου αυτού διορθώθηκαν επιπλέον λάθη στον προγραμματιστικό κώδικα της εφαρμογής, τροποποιήθηκε η θέση και η απεικόνιση διαφόρων χρηστικών οντοτήτων σελίδων, εξερευνήθηκε η προστιθέμενη αξία της εφαρμογής και βελτιστοποιήθηκε η αλληλεπίδραση της εφαρμογής με τον χρήστη μέσω προτάσεων και υποδείξεων των ελεγκτών της β' φάσης αυτής. Η τελευταία ολοκληρώθηκε μετά την πάροδο καθορισμένου χρονικού διαστήματος στο οποίο δεν παρατηρήθηκαν άλλες ανωμαλίες στη συμπεριφορά της εφαρμογής και κρίθηκε μη επιτακτική η πραγματοποίηση άλλων αλλαγών. Στο σημείο αυτό ολοκληρώθηκε η διαδικασία ελέγχου και η εφαρμογή ήταν πλέον διαθέσιμη σε ένα ευρύ κοινό χρηστών εκτός εργαστηρίου και των κλειστών κύκλων.

7

Επίλογος

Το κεφάλαιο θα αποτελέσει μια σύνοψη της συνολικής εργασίας, όπου θα παρουσιαστούν τα συμπεράσματα στα οποία αυτή κατέληξε και οι πιθανές προεκτάσεις που προέκυψαν κατά το στάδιο ανάπτυξης της εφαρμογής.

7.1 Σύνοψη & Συμπεράσματα

Μετά από μελέτη των ήδη υπάρχουσών εφαρμογών του χώρου των δραστηριοτήτων παρατηρήθηκε μια έλλειψη στον τομέα που αφορά τις καθημερινές δραστηριότητες, οι οποίες δεν ανήκουν αποκλειστικά και μόνο σε μια κατηγορία. Η έλλειψη θεσπισμένων οντολογιών και μοντέλων για την περιγραφή των ενεργειών και δραστηριοτήτων οδήγησε σε μια έρευνα πάνω στις θεωρητικές πτυχές του χώρου και μετά από ανάγνωση διαφόρων εργασιών του σχετικού τομέα, καθορίστηκε ένα μοντέλο ικανό να περιγράψει σε ικανοποιητικό βαθμό τη γενική και αόριστη ανθρώπινη δραστηριότητα. Με άξονα αυτό, ξεκίνησε η σχεδίαση μιας εφαρμογής η οποία θα ήταν σε θέση να υποστηρίζει κάθε δραστηριότητα, με τελικό στόχο την παραγωγή προστιθέμενες αξίας για το χρήστη από την παρουσίαση αναλυτικών διαγραμμάτων και στατιστικών στοιχείων στην πάροδο του χρόνου. Καθορίστηκαν έτσι οι απαιτήσεις του συστήματος και οι υποστηριζόμενες λειτουργίες του, σχεδιάστηκε η βασική αρχιτεκτονική του, επιλέχθηκαν τα προγραμματιστικά εργαλεία και υλοποιήθηκε η διαδικτυακή εφαρμογή.

Εξετάζοντας τα αποτελέσματα στα οποία αυτή κατέληξε, τα διαγράμματα και τα στατιστικά στοιχεία ήταν σε θέση να προσφέρουν σημαντικότερη πληροφορία έναντι οποιουδήποτε άλλου απλού προγραμματιστή καθημερινών δραστηριοτήτων. Συνεπώς, η χρήση του Activity Tracker για την καταγραφή, οργάνωση και παρακολούθηση της καθημερινότητας οδηγεί σε μεγαλύτερα οφέλη από την απλή σημείωσή τους σε κάποιο ημερολόγιο. Αξίζει να τονιστεί ακόμα ότι η συσσώρευση των δραστηριοτήτων εξωγενών υπηρεσιών σε μια μόνο εφαρμογή, όντως οδήγησε σε μεγάλα οφέλη για τους χρήστες από τη δυνατότητα συσχέτισης και σύγκρισης διαφορετικών τύπων δραστηριοτήτων. Για πρώτη φορά οι χρήστες ήταν σε θέση να συγκρίνουν την αθλητική τους απόδοση (δραστηριότητες άθλησης RunKeeper) σε σχέση με την πορεία του ύπνου τους (δραστηριότητα ύπνου Sleepy

– RunKeeper) και να ανακαλύψουν τον βαθμό στον οποίο η μία επηρεάζει την άλλη. Μειονέκτημα επ' αυτού θεωρήθηκε η απώλεια μέρους πληροφορίας από τη μοντελοποίηση, καθώς αυτή δεν είναι σε θέση να περιγράψει κάθε δυνατή λεπτομέρεια της εκάστοτε δραστηριότητας. Αυτό είναι κάτι το αναμενόμενο μιας και δεν υπάρχει, ούτε και μπορεί να υπάρξει μια ιδανική μοντελοποίηση που να μπορεί να επαφίεται σε κάθε τύπο ή μορφή δραστηριότητας. Το τελευταίο αποτελεί βασικό αρνητικό και συνθέτει τον κύριο λόγο για τον οποίο δεν υπάρχει πληθώρα εφαρμογών τύπου Activity Tracker διαθέσιμη στους χρήστες. Παρόλα αυτά, η εφαρμογή ανταποκρίθηκε στις προσδοκίες των χρηστών, κατάφερε να τους δώσει κίνητρο για την καθημερινή καταγραφή των ενεργειών τους, τους παρουσίασε ενδιαφέροντα στοιχεία που δεν είχαν παρατηρήσει και εν τέλει, κρίνοντας από την ανατροφοδότησή τους, βοήθησε στη βελτίωση και καλύτερη οργάνωση της καθημερινότητάς τους.

Από την πλευρά της ανάπτυξης, η ολοκλήρωση της εφαρμογής οδήγησε στην εξαγωγή ποικίλων συμπερασμάτων. Το σημαντικότερο εξ' αυτών είναι η αδυναμία κατασκευής μιας βέλτιστης μοντελοποίησης για όλες τις διαθέσιμες ανθρώπινες δραστηριότητες. Κάθε μοντελοποίηση, όσο καλά δομημένη και αν είναι, θα έχει απώλειες πληροφορίας στην περίπτωση χειροκίνητης εισαγωγής πληροφορίας και ιδιαίτερα στην περίπτωση συγχρονισμού μέσω άλλων υπηρεσιών. Στην τελευταία, άθλο αποτελεί η εύρεση της κατάλληλης αντιστοίχισης μεταξύ των δύο (2) διαφορετικών μοντέλων περιγραφής δραστηριότητας, η οποία μάλιστα πολλές φορές ενδέχεται να είναι και ανύπαρκτη. Στο Activity Tracker, η μοντελοποίηση ήταν αυτή που καθόρισε το διαχωρισμό και το περιεχόμενο των σελίδων ανάλυσης, άρα μια διαφορετική μοντελοποίηση δεν επηρεάζει μόνο την πληροφορία που αποθηκεύεται αλλά και εκείνη που παρουσιάζεται στον χρήστη. Πρέπει να αναφερθεί ωστόσο, ότι η επιλογή ενός εκτενούς μοντέλου δεν είναι πάντα κάτι το βέλτιστο, μιας και αυτή οδηγεί στην ανάγκη εισαγωγής πολλής πληροφορίας, με αποτέλεσμα την απώθηση των χρηστών από τη συμπλήρωση των πολλών πεδίων δεδομένων. Το τελευταίο είναι ένα από τα μεγάλα μειονεκτήματα της εφαρμογής και είναι κάτι που χρήζει βελτίωσης, μιας και ανακαλύφθηκε ότι η χειρωνακτική εισαγωγή πολλών δραστηριοτήτων (μαζί με το πλαίσιο το οποίο τις συνοδεύει) είναι μια εξαιρετικά κουραστική και χρονοβόρα διαδικασία για τους περισσότερους χρήστες. Αυτό καθιστά άμεσα αναγκαία την υποστήριξη συγχρονισμού με περισσότερες από μια υπηρεσίες, ώστε οι χρήστες να μπορούν να αυτοματοποιούν στο μέγιστο την εισαγωγή των καθημερινών τους ενεργειών στο σύστημα.

Μεταβαίνοντας στο κομμάτι της σχεδίασης, τονίζεται η αξία μιας σωστής αρχιτεκτονικής με βάση τις καθορισμένες λειτουργικές απαιτήσεις του συστήματος. Στην παρούσα εργασία η αρχιτεκτονική δεν ήταν βέλτιστα σχεδιασμένη εξ' αρχής και συνεπώς μικρές επεκτάσεις, τροποποιήσεις και προσθήκες αποδείχτηκαν δύσκολο να υλοποιηθούν δίχως τη χρήση ανορθόδοξων πρακτικών. Κλείνοντας, αξίζει να σημειωθεί η ανάγκη σχεδίασης κατάλληλων διαπροσωπειών χρήστη. Η ανάπτυξη με γνώμονα τη σχεδίαση και η χρήση κατανοητών διαγραμμάτων, λιτών διαπροσωπειών, ξεκάθαρων συμβόλων/γραφικών και

πολλαπλών δυνατοτήτων επιλογής, δεν πρέπει να αποτελούν δευτερεύουσα προτεραιότητα. Μέσω της διαδικασίας ελέγχου τροποποιήθηκαν αρκετά σημεία της εφαρμογής με σκοπό την αύξηση της φιλικότητάς του συστήματος προς τους χρήστες, κάνοντας κατανοητή την ανάγκη τοποθέτησής των τελευταίων στο επίκεντρο της ανάπτυξης.

7.2 Μελλοντικές Επεκτάσεις

Κλείνοντας την διπλωματική εργασία, θα γίνει αναφορά σε ενδεχόμενες μελλοντικές επεκτάσεις της εφαρμογής.

Η πιο ενδιαφέρουσα, η οποία συμφωνεί και με τις απόψεις των τελικών χρηστών, είναι η ανάγκη υλοποίησης αντίστοιχης εφαρμογής για λογισμικό Android ή iOS, έτσι ώστε να υποστηρίζεται από συσκευές τύπου smartphone. Οι χρήστες που πραγματοποιούν συγχρονισμό δραστηριοτήτων με RunKeeper, κάνουν χρήση του κινητού τους τηλεφώνου για την καταγραφή των δραστηριοτήτων άθλησης και ύπνου τους. Κατά συνέπεια, θα τους ήταν ιδιαίτερα βολικό να τροφοδοτούν δραστηριότητες και να παρατηρούν τις αναλύσεις δίχως την ανάγκη αλλαγής συσκευής. Φυσικά, θα μπορούσαν να μεταβαίνουν στην ιστοσελίδα του Activity Tracker από το κινητό, ωστόσο αυτό μπορεί να αποδειχθεί επίπονο ειδικά εν έτη 2015 που οι αναλύσεις στις οθόνες των κινητών διαφοροποιούνται σε τεράστιο βαθμό. Με την υλοποίηση αυτής της εφαρμογής, το Activity Tracker μπορεί να γίνει ακόμα πιο εύκολο και απλό στη χρήση του, δίχως να χάνει στο ελάχιστο τα πλεονεκτήματα και την προστιθέμενη αξία που προσφέρει στο χρήστη. Ένα ακόμα πλεονέκτημα της υλοποίησης της εφαρμογής για κινητά είναι η απουσία ανάγκης διαδικτυακής σύνδεσης για όλες τις λειτουργίες (εξαιρουμένων της σύνδεσης με υπηρεσίες και του συγχρονισμού δραστηριοτήτων). Οι δραστηριότητες έχουν τη δυνατότητα να αποθηκεύονται τοπικά και με ασφάλεια σε κάθε συσκευή, γεγονός που σημαίνει ότι οι βασικές λειτουργίες και η παρουσίαση αναλυτικών δεν θα απαιτεί τη χρήση διαδικτύου.

Μια διαφορετική επέκταση αφορά την αποκόλληση όλης της πλευράς του πελάτη από την εφαρμογή και τη χρήση του Activity Tracker ως ένα ενδιάμεσο σύστημα (middleware) που θα πραγματοποιεί μια κοινή μοντελοποίηση δραστηριοτήτων, ανεξάρτητα της υπηρεσίας από την οποία η δραστηριότητα αυτή προήλθε. Συνεπώς, θα μπορούσε να αποτελέσει ένα στάδιο «μετάφρασης» της μοντελοποίησης της εκάστοτε υπηρεσίας, με τελικό στόχο το χειρισμό των δεδομένων με σταθερό τρόπο. Αυτό θα μπορούσε επεκταθεί περαιτέρω μέσω της σύνδεσης με εξωτερικές οντολογίες και άλλες σημασιολογικές οντότητες για την εξαγωγή ανώνυμων ή μη δεδομένων προς άλλες υπηρεσίες. Το τελευταίο θα είχε ως αποτέλεσμα την παραγωγή αναλυτικών δεδομένων που θα υπερέβαινε τα όρια του Activity Tracker και θα επέτρεπε την αλληλεπίδραση με άλλα συστήματα, αυτή τη φορά σε επίπεδο παροχής δεδομένων.

Τέλος, θα μπορούσε να διατηρηθεί αμετάβλητη η εφαρμογή και να επεκταθούν οι λειτουργίες και οι δυνατότητές της. Συγκεκριμένα, θα ήταν εξαιρετικά ενδιαφέρουσες οι ακόλουθες επεκτάσεις:

1. Δυνατότητα προσθήκης προσωπικών δραστηριοτήτων από το χρήστη, ώστε το Activity Tracker να προσαρμόζεται στην καθημερινότητα και τις ενέργειες του καθενός.
2. Δυνατότητα τροφοδότησης δεδομένων μέσω ημερολογίου δραστηριοτήτων τρίτων υπηρεσιών (π.χ. Google Calendar), όπου ο χρήστης θα μπορούσε να ορίζει λέξεις κλειδιά (keywords), τα οποία όταν αναγνωρίζονται στον τίτλο μιας δραστηριότητας θα μπορούν να αντιστοιχίζονται σε δραστηριότητες του συστήματος. Αν αυτό συνδυαστεί με το προηγούμενο, ο χρήστης θα μπορούσε να δημιουργήσει μια προσωπική δραστηριότητα στο Activity Tracker (π.χ. «Συνάντηση με Πελάτες»), να ορίσει μία ή παραπάνω λέξεις κλειδιά που θα αντιστοιχίζονται στην προηγούμενη δραστηριότητα (π.χ. «Συνάντηση»), να χρησιμοποιεί τις λέξεις αυτές στους τίτλους των δραστηριοτήτων του στο Google Calendar (π.χ. «Συνάντηση με Υπεύθυνους της Εταιρίας Χ») και όταν πατάει συγχρονισμό δραστηριοτήτων να δημιουργείται αυτόματα η αντιστοιχισμένη (σύμφωνα με τις λέξεις του τίτλου της) δραστηριότητα στο Activity Tracker.
3. Δυνατότητα υποστήριξης παραπάνω υπηρεσιών συγχρονισμού δεδομένων (π.χ. Facebook status).
4. Παροχή παραπάνω μορφών επισκόπησης και οπτικοποίησης δεδομένων, πέρα από ημερολόγιο, πλέγμα και χρονογράφημα (π.χ. οπτικοποίηση φυσαλίδας) .
5. Παρουσίαση διαφορετικών διαγραμματικών αναπαραστάσεων και στατιστικών στοιχείων όπως π.χ. διαγράμματα λέξεων.
6. Επέκταση μοντελοποίησης για την καταγραφή παραπάνω χαρακτηριστικών π.χ. συναίσθημα χρήστη κατά τη διάρκεια της δραστηριότητας. Αυτό οδηγεί αυτόματα σε νέα διαγράμματα και στοιχεία π.χ. Αίσθηση χαράς σε 60% των εμφανίσεων της δραστηριότητας «Working».
7. Δυνατότητα προσθήκης «ζωντανής» δραστηριότητας λαμβάνοντας αυτόματα το χρόνο έναρξης και την τοποθεσία του χρήστη τη στιγμή της προσθήκης μέσω κινητού τηλεφώνου τύπου smartphone.
8. Δυνατότητα υποστήριξης φωνητικής λειτουργίας για γρήγορη και εύκολη προσθήκη του πλαισίου που συνοδεύει μια δραστηριότητα. Παράδειγμα αυτού αποτελεί η φράση «Για Καφέ με Γιώργο στο Σύνταγμα», από την οποία το σύστημα θα αναγνωρίζει λέξεις εκείνες (Καφέ, Γιώργο, Σύνταγμα), που έπονται συγκεκριμένων συνδυασμών λέξεων κλειδίων (για, με, στο) και θα δημιουργεί αυτόματα νέα δραστηριότητα αντιστοιχίζοντας τις πρώτες με τα κατάλληλα πεδία της μοντελοποίησης σύμφωνα με τις συνδυαστικές λέξεις κλειδιά.

9. Δυνατότητα ανακάλυψης προτύπων (patterns) ανάμεσα στις δραστηριότητες (π.χ. μετά από κάθε δραστηριότητα τρεξίματος ακολουθεί η δραστηριότητα καθαρισμού σώματος).
10. Υλοποίηση ιεραρχίας και υποδιαίρεσης των δραστηριοτήτων στα πλαίσια της «θεωρίας των δραστηριοτήτων» (Activity Theory), η οποία ορίστηκε αλλά δεν υλοποιήθηκε στην παρούσα διπλωματική. Η τελευταία σε συνδυασμό με την προηγούμενη επέκταση θα είναι σε θέση να παράγει ενδιαφέροντα συμπεράσματα τα οποία θα διαφοροποιούνται από άτομο σε άτομο, εξατομικεύοντας έτσι την ανάλυση.

8

Βιβλιογραφία

Georgios Meditskos, Stamatia Dasiopoulou, Vasiliki Efstathiou, and Ioannis Kompatsiaris. (2013). *Ontology Patterns for Complex Activity Modelling*.

Engestrom, Yrjo, Miettinen, Reijo, Punamaki, Raija-Leena. (1999). *Perspectives on Activity Theory*.

Engestrom, Yrjo. (1987). *Learning by Expanding: An Activity-theoretical Approach to Developmental Research*.

Liming Chen, Chris Nugent. (2011). *Ontology-based Activity Recognition in Intelligent Pervasive Environments*.

Longbing Cao. (2010). *In-depth behavior understanding and use: The behavior informatics approach*.

Roy T. Fielding, James Gettys, Jeffrey C. Mogul, Henrik Frystyk Nielsen, Larry Masister, Paul J. Leach, Tim Berners-Lee. (1999). *Hypertext Transfer Protocol – HTTP/1.1*.

Stephan Poslad. (2010). *Ubiquitous Computing Smart Devices, Smart Environments and Smart Interaction*

T.R. Gruber. (1993). *Toward principles for the design of ontologies used for knowledge sharing*.

V.Kaptelinin, K. Kuutti, L. Bannon. (1997). *Activity Theory: Basic Concepts and Applications*.

V. Kaptelinin, B. Nardi and C. Macaulay. (1999). *The Activity Checklist: A Tool for Representing the “Space” of Context*.

Xiao Hang Wang, Da Qing Zhang, Tao Gu, Hung Keng Pung. (2004). Ontology Based Context Modeling and Reasoning using OWL.

9

Παραρτήματα

Ακολουθούν τα παραρτήματα της διπλωματικής, αναλύοντας ζητήματα που δεν έχουν άμεση σχέση με το αντικείμενο της διπλωματικής εργασίας, αλλά συνδέονται με αυτή.

9.1 Παράρτημα I: Οδηγίες εγκατάστασης

Στο υποκεφάλαιο αυτό θα παρουσιαστούν οι οδηγίες για την εγκατάσταση της εφαρμογής Activity Tracker σε τοπικό υπολογιστή. Η διαδικασία χωρίζεται σε δύο (2) μέρη. Το 1^ο περιλαμβάνει τη λήψη των απαραίτητων προγραμματιστικών εργαλείων και πηγαίου κώδικα που απαιτούνται για την ομαλή λειτουργία της εφαρμογής, ενώ το 2^ο περιλαμβάνει τη σύνδεση της εφαρμογής με τα εργαλεία, την πραγματοποίηση αλλαγών και την εκτέλεση εντολών αρχικοποίησης.

- **Λήψη Προγραμματιστικών Πλατφορμών, Εργαλείων & Πηγαίου Κώδικα**

Η διαδικασία λήψης και εγκατάστασης των απαραίτητων μονάδων θα περιγραφεί μέσω των ακόλουθων βημάτων. Αυτή απευθύνεται σε χρήστες λειτουργικού συστήματος Windows, αλλά η διαδικασία είναι παρεμφερής και για λειτουργικά συστήματα Mac OS, Linux, Ubuntu κ.λπ.:

1. Λήψη και εγκατάσταση του φυλλομετρητή **Google Chrome** ([Λήψη](#)) ή **Mozilla Firefox** ([Λήψη](#)), με τους οποίους έχει επαληθευθεί η ορθή λειτουργία της εφαρμογής. Για τους χρήστες λειτουργικού Mac OS ενδείκνυται ο Google Chrome, ωστόσο ακόμα και ο προεγκατεστημένος **Safari** είναι σε θέση να υποστηρίξει το Activity Tracker με μικρές σχεδιαστικές διαφορές.
2. Λήψη και εγκατάσταση της **Python 2.7.9** ([Λήψη](#)) ή νεότερης έκδοσης της Python 2. Μετά το τέλος της εγκατάστασης, απαιτείται τροποποίηση της μεταβλητής συστήματος **Path**, έτσι ώστε να περιλαμβάνει το φάκελο εγκατάστασης της Python, όπως επίσης και τον υποφάκελο scripts. Σε περιβάλλον windows, αν αυτή είχε εγκατασταθεί στην τοποθεσία **C:/Python27**, τότε θα έπρεπε να προστεθεί το κείμενο **;C:\Python27;C:\Python27\Scripts** στη μεταβλητή Path.

3. Λήψη και εγκατάσταση του framework **Django 1.8** ή νεότερης έκδοσης αυτού. Έχοντας εγκαταστήσει την γλώσσα Python και έχοντας ρυθμίσει τη μεταβλητή Path, αυτό γίνεται εύκολα ανοίγοντας τη γραμμή εντολών (command line) και πληκτρολογώντας την εντολή **pip install Django==1.8.2** από οπουδήποτε στο σύστημα. Αν υπάρχει νεότερη έκδοση διαθέσιμη, τότε ο αριθμός 1.8.2 πρέπει να τροποποιηθεί ώστε να αντιστοιχεί στην έκδοση αυτή.
4. Λήψη και εγκατάσταση του **Pycharm Professional Edition 4.5.x** ([Λήψη](#)) ή ενός άλλου IDE κατάλληλου για την ανάπτυξη εφαρμογών Django. Το βήμα αυτό είναι προαιρετικό αν ο προγραμματιστής εργάζεται σε περιβάλλον γραμμής εντολών.
5. Λήψη του πηγαίου κώδικα της εφαρμογής από το εργαστήριο Συστημάτων και Αποφάσεων Διοίκησης.
6. Εάν ο κώδικας λήφθηκε υπό μορφή .zip, αποσυμπίεση σε μια γνωστή τοποθεσία, εκκίνηση γραμμής εντολών εντός της τοποθεσίας αυτής και εκτέλεση της εντολής **pip install -r requirements.txt**, όπου requirements είναι το όνομα του αρχείου κειμένου που περιλαμβάνει τις απαραίτητες βιβλιοθήκες και εξωτερικές επιπρόσθετες οντότητες που απαιτεί το σύστημα. Το αρχείο αυτό είναι μέρος του κώδικα και ενδέχεται να περιέχει παλαιότερες εκδόσεις των βιβλιοθηκών.

- **Σύνδεση, Πραγματοποίηση Αλλαγών και Αρχικοποίηση Συστήματος**

Στο σημείο αυτό έχουν εγκατασταθεί όλες οι απαραίτητες μονάδες και εργαλεία. Ακολουθώντας τα παρακάτω βήματα αρχικοποιείται το σύστημα και είναι σε θέση να εκτελείται σε περιβάλλον τοπικού εξυπηρετητή.

1. Μέσα από το περιβάλλον του Pycharm IDE, επιλογή ανοίγματος (open) ενός υπάρχοντος έργου (project), μετάβαση στην τοποθεσία όπου αποθηκεύτηκε ο πηγαίος κώδικας και επιλογή του φακέλου με όνομα **Activitytracker_Project**.
2. Επιλογή **File** → **Settings** → **Project: Activitytracker_Project** → **Project Interpreter** → Επιλογή της εγκατεστημένης έκδοσης της Python μέσω του φακέλου **Python27** → **Ok** και αναμονή για τη δεικτοδότηση.
3. Πληκτρολόγηση **Ctrl + Alt + R** για εκκίνηση των εντολών manage.py. Εκτέλεση της εντολής **migrate** για τη δημιουργία πινάκων στη βάση δεδομένων (αν εμφανιστεί επιλογή διαγραφής μοντέλου επιλέξτε **no**). Εναλλακτικά, εκκίνηση γραμμής εντολών, μετάβαση στο φάκελο **Activitytracker_Project** και εκτέλεση της εντολής **python manage.py migrate**.
4. Τροποποίηση της διεύθυνσης του εξυπηρετητή στα αρχεία. Συγκεκριμένα απαιτείται μετάβαση στο αρχείο **Activitytracker_Project/activitytracker/views.py** και τροποποίηση της γραμμής 90«http://SERVER_URL/activitytracker/account/verification/» Η τιμή **SERVER_URL** πρέπει να αντιστοιχίζεται στο server, ο οποίος για τοπική εκτέλεση του κώδικα λαμβάνει την τιμή 127.0.0.1:8000. Συνεπώς, η παραπάνω γραμμή θα πρέπει να τροποποιηθεί ως εξής:

«<http://127.0.0.1:8000/activitytracker/account/verification/>». Ομοίως, αντίστοιχες τροποποιήσεις πρέπει να εφαρμοστούν και στο αρχείο `Activitytracker_Project/activitytracker/static/activitytracker/js/custom.js`. Οι δύο (2) πρώτες γραμμές του αρχείου αυτού αφορούν τις μεταβλητές `BASE_URL` και `SERVER_URL`. Η μεταβλητή `BASE_URL` πρέπει να λάβει την τιμή «<http://127.0.0.1:8000/activitytracker/>», ενώ η μεταβλητή `SERVER_URL` την τιμή «http://127.0.0.1:8000».

5. Πληκτρολόγηση **Ctrl + Alt + R** και εκτέλεση της εντολής **runserver** ή εκκίνηση γραμμής εντολών, μετάβαση στο φάκελο **Activitytracker_Project** και εκτέλεση της εντολής **python manage.py runserver**. Αν όλα τα βήματα έχουν πραγματοποιηθεί σωστά, τότε στην διεύθυνση <http://127.0.0.1:8000/activitytracker> πρέπει να εμφανίζεται η εφαρμογή. Ακόμα στη διεύθυνση <http://127.0.0.1:8000/admin> μπορεί κανείς να έχει πλήρη εποπτεία των μοντέλων – πινάκων της βάσης δεδομένων. Αξίζει να τονιστεί ότι οι λειτουργίες σύνδεσης μέσω Facebook και Google δεν θα λειτουργούν, καθώς είναι ρυθμισμένες έτσι ώστε να ανταποκρίνονται στη διεύθυνση εξυπηρετητή του εργαστηρίου που φιλοξενείται η ιστοσελίδα του επίσημου Activity Tracker. Για την υποστήριξη τους τοπικά απαιτείται η δήλωση νέας εφαρμογής στις υπηρεσίες, η λήψη κλειδιών εξουσιοδότησης από τις υπηρεσίες και η τροποποίησή τους στο αρχείο `Activitytracker_Project/config.py`

Πλέον η εφαρμογή έχει εγκατασταθεί και είναι πλήρως επεξεργάσιμη. Ο αναγνώστης καλείται να διαβάσει αναλυτικά τα κεφάλαια (5) και (6) της παρούσας εργασίας, για την κατανόηση της αρχιτεκτονικής, της οργάνωσης και του τρόπου λειτουργίας της εφαρμογής.