



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

**Σχεδίαση και υλοποίηση εφαρμογής εθελοντή αιμοδότη
για κινητά τερματικά με χρήση τεχνολογιών αιχμής**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Χριστίνα-Αθηνά Λυκογιάννη-Μπεκάκου

Επιβλέπων: Παναγιώτης Τσανάκας
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2023



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

**Σχεδίαση και υλοποίηση εφαρμογής εθελοντή αιμοδότη
για κινητά τερματικά με χρήση τεχνολογιών αιχμής**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Χριστίνα-Αθηνά Λυκογιάννη-Μπεκάκου

Επιβλέπων: Παναγιώτης Τσανάκας
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 18^η Ιουλίου 2023.

.....
Παναγιώτης Τσανάκας
Καθηγητής Ε.Μ.Π.

.....
Δημήτριος Σούντρης
Καθηγητής Ε.Μ.Π.

.....
Άγγελος Αμδίτης
Ερευνητής ΕΠΙΣΕΥ

Αθήνα, Ιούλιος 2023

.....

Χριστίνα-Αθηνά Λυκογιάννη-Μπεκάκου

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Χριστίνα-Αθηνά Λυκογιάννη-Μπεκάκου

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

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.

Περίληψη

Στην παρούσα διπλωματική μελετήθηκε, σχεδιάστηκε και υλοποιήθηκε μία εφαρμογή, που σε συνεργασία με το «Εθνικό Μητρώο Αιμοδοτών», παρέχει χρήσιμες και εξατομικευμένες πληροφορίες στους αιμοδότες και λειτουργεί σε κινητό τερματικό (smartphone ή tablet).

Η εφαρμογή μπορεί να εκτελεσθεί σε Android λειτουργικό, ενώ για την υλοποίησή της διερευνήθηκαν οι σύγχρονες μεθοδολογίες ανάπτυξης και οι διαθέσιμες τεχνολογίες με σκοπό την ανάπτυξη μίας αποδοτικά λειτουργικής, εύχρηστης και ασφαλούς εφαρμογής. Με τη χρήση σύγχρονων τεχνολογιών, όπως είναι το React Native, επιδιώχθηκε η αποδοτική χρήση των πόρων του τερματικού.

Για τη διασύνδεση με το Εθνικό Μητρώο Αιμοδοτών αξιοποιήθηκε το διαθέσιμο API (application programming interface) της υπηρεσίας και επιπλέον, καταγράφηκαν προτάσεις τροποποίησής του, βασιζόμενες στις αρχές της REST αρχιτεκτονικής. Ακολουθώντας τις βέλτιστες πρακτικές προγραμματισμού και τεκμηρίωσης, η προτεινόμενη μορφή του API αποτυπώθηκε σύμφωνα με το OpenAPI specification.

Τέλος, πριν την υλοποίηση της εφαρμογής πραγματοποιήθηκε εκτενής μελέτη για το βέλτιστο σχεδιασμό της διεπαφής της και αποτύπωση όλων των προς υλοποίηση οθονών σε wireframes. Σημειώνεται ότι στα πλαίσια αυτής της εργασίας μελετήθηκαν και εφαρμόστηκαν οι βέλτιστες πρακτικές για ευχρηστία και οι Mobile Web Practices 1.0 του W3C για προσβασιμότητα ώστε η χρήση της εφαρμογής να είναι εφικτή από όλους τους εν δυνάμει χρήστες.

Λέξεις Κλειδιά

Εφαρμογή κινητού τερματικού, Android, React Native, JavaScript, REST API, OpenAPI, Swagger, Mobile Web Practices 1.0, wireframes, προσβασιμότητα

Abstract

This dissertation was focused on designing and implementing a mobile application for the «National Blood Donor Registry», that provides useful and personalized information to blood donors.

The application can operate on a mobile terminal (smartphone or tablet) with the Android operating system. For its implementation, modern development methodologies and available technologies were explored to develop an efficiently functional, user-friendly, and secure application. The aforementioned goals were achieved through the use of technologies such as React Native.

To connect with the National Blood Donor Registry, the available API (application programming interface) of the service was utilized, and furthermore, proposals for its modification were recorded based on the principles of REST architecture. Following the best programming and documentation practices, the proposed API was documented according to the OpenAPI specification.

Finally, before implementing the application, an extensive study was conducted for the optimal design of its interface. As part of this process, wireframes were created to visually represent all the screens that would be implemented. It's worth mentioning that within the scope of this work, best practices for usability and the Mobile Web Practices 1.0 of W3C for accessibility were studied and applied, ensuring that the application is accessible to all potential users.

Keywords

Mobile application, Android, React Native, JavaScript, REST architecture, API, OpenAPI, Swagger, Mobile Web Practices 1.0, wireframes, usability, accessibility, user experience

Ευχαριστίες

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω όλους όσους με υποστήριξαν και συνέβαλαν, άμεσα ή έμμεσα, στην εκπόνηση της παρούσας διπλωματικής εργασίας.

Καταρχάς, ευχαριστώ θερμά τον επιβλέποντα καθηγητή μου, κ. Παναγιώτη Τσανάκα, για την εμπιστοσύνη που μου έδειξε αναθέτοντάς μου το συγκεκριμένο θέμα διπλωματικής εργασίας καθώς και για την καθοδήγηση και συμπαράσταση που μου προσέφερε καθ' όλη τη διάρκεια της εκπόνησής της.

Ιδιαίτερες ευχαριστίες θα ήθελα να εκφράσω στο Δρ. Μηχ. Σολομώντα Ζάννο για τη συνεχή συμβολή του στην εξέλιξη της παρούσας εργασίας. Η συνεχής επικοινωνία, αρωγή, και ενθάρρυνση όπως και οι εύστοχες συμβουλές του συνέβαλαν καταλυτικά στην περάτωση της παρούσας εργασίας.

Επίσης, θα ήθελα να ευχαριστήσω το Εθνικό Κέντρο Αιμοδοσίας και το Εθνικό Δίκτυο Υποδομών Τεχνολογίας και Έρευνας και ιδιαίτερα το Δρ. Μηχ. Δημήτρη Νικητόπουλο, που μου έδωσαν την ευκαιρία να μελετήσω και να εργασθώ σε ένα σύστημα μεγάλης κλίμακας και κοινωνικής σημασίας, όπως είναι το Εθνικό Μητρώο Αιμοδοτών. Η εμπειρία αυτή υπήρξε καθοριστική για τον εμπλουτισμό των γνώσεων μου και την περαιτέρω ανάπτυξη των δεξιοτήτων μου.

Τέλος, θα ήθελα να ευχαριστήσω όλους τους δικούς μου ανθρώπους για τη στήριξη τους και την πολύπλευρη βοήθεια που μου προσέφεραν καθ' όλη τη διάρκεια της εργασίας μου.

Χριστίνα-Αθηνά Λυκογιάννη-Μπεκάκου

Ιούλιος 2023

Περιεχόμενα

Περίληψη	5
Abstract.....	7
Εισαγωγή	14
1 Το σύστημα της αιμοδοσίας στην Ελλάδα.....	15
2 Λόγοι αξιοποίησης τεχνολογιών αιχμής.....	18
3 Διαδικασία ανάπτυξης της εφαρμογής.....	21
4 Μελέτη της διαδικτυακής εφαρμογής	22
5 Σχεδιασμός διεπαφής χρήστη	30
5.1 Ευχρηστία συστήματος και εμπειρία χρήστη	30
5.1.1 Αρχές σχεδιασμού Android εφαρμογών (Android design principles).....	30
5.1.2 React Native αρχές.....	32
5.1.3 Συνοπτική αναφορά αρχών που εφαρμόστηκαν	32
5.2 Προσβασιμότητα (accessibility).....	33
5.3 Διεθνοποίηση και προσαρμογή λογισμικού σε τοπικές ιδιομορφίες	38
5.4 Wireframes – Προσχέδια εφαρμογής αιμοδότη	38
6 Σχεδιασμός διεπαφής προγραμματισμού εφαρμογών – API	48
6.1 Επιθυμητά χαρακτηριστικά του API.....	48
6.2 REST API.....	49
6.2.1 Ορισμός.....	49
6.2.2 Περιγραφή της αρχιτεκτονικής	50
6.3 Περιγραφή του προτεινόμενου API με OpenAPI Specification.....	54
6.3.1 OpenAPI Specification (OAS).....	54
6.3.2 Προτεινόμενο API.....	54
7 Υλοποίηση	91
7.1 Χαρακτηριστικά εφαρμογής.....	91
7.2 Διαθέσιμες τεχνολογίες.....	92
7.2.1 Java για την υλοποίηση Android εφαρμογών	92
7.2.2 Υβριδικές τεχνολογίες για εφαρμογές Android και iOS.....	92
7.2.3 React Native για εφαρμογές Android και iOS	93
7.3 Επιλεγμένες τεχνολογίες και εργαλεία	93
7.4 React και React Native	94
7.4.1 Αρχιτεκτονική	94
7.4.2 Πλεονεκτήματα και μειονεκτήματα React Native	96
8 Παρουσίαση υλοποιημένης εφαρμογής.....	100
9 Δυσκολίες που αντιμετωπίστηκαν και μελλοντικές επεκτάσεις.....	114
Βιβλιογραφία	116

Κατάλογος Εικόνων

Εικόνα 1-1 Δικτυακή εφαρμογή για τους αιμοδοτές (web client).....	17
Εικόνα 2-1 Ποσοστό χρήσης του διαδικτύου ανά ήπειρο το 2022.....	18
Εικόνα 2-2 Στατιστικά χρήσης του διαδικτύου παγκοσμίως από το 1995 έως και το 2022 [6]	19
Εικόνα 2-3 Χρήση ψηφιακών μέσων το 2019.....	20
Εικόνα 4-1 Οθόνη εισόδου δικτυακής εφαρμογής.....	23
Εικόνα 4-2 Στοιχεία προφίλ χρήστη.....	24
Εικόνα 4-3 Διαχείριση προφίλ χρήστη.....	24
Εικόνα 4-4 Επισκόπηση παλαιών καρτών.....	25
Εικόνα 4-5 Ιστορικό αιμοδοτήσεων.....	25
Εικόνα 4-6 Επισκόπηση ιστορικού καλύψεων.....	26
Εικόνα 4-7 Ενεργοποίηση υπενθύμισης επόμενης αιμοδότησης.....	27
Εικόνα 4-8 Λίστα αιτήσεων για έκδοση ταυτότητας αιμοδότη.....	27
Εικόνα 4-9 Ταυτότητα Εθελοντή Αιμοδότη (1 από 3).....	28
Εικόνα 4-10 Ταυτότητα Εθελοντή Αιμοδότη (2 από 3).....	28
Εικόνα 4-11 Ταυτότητα Εθελοντή Αιμοδότη (3 από 3).....	29
Εικόνα 5-1 Οθόνη σύνδεσης και μενού πλοήγησης μη συνδεδεμένου χρήστη.....	39
Εικόνα 5-2 Δημιουργία λογαριασμού.....	40
Εικόνα 5-3 Υπενθύμιση κωδικού πρόσβασης.....	41
Εικόνα 5-4 Εναλλακτικές προτάσεις μενού πλοήγησης συνδεδεμένου χρήστη.....	41
Εικόνα 5-5 Προβολή και τροποποίηση στοιχείων λογαριασμού σε διαφορετικές οθόνες.....	42
Εικόνα 5-6 Προβολή και τροποποίηση των στοιχείων προφίλ στην ίδια οθόνη.....	43
Εικόνα 5-7 Ιστορικό αιμοδοσιών.....	43
Εικόνα 5-8 Καλύψεις.....	44
Εικόνα 5-9 Ιστορικό αιμοδοσιών προγενέστερων του 2015 (καρτέλες νοσοκομείων).....	44
Εικόνα 5-10 Δυνατότητα επόμενης αιμοδοσίας και επιλογές ειδοποίησης.....	45
Εικόνα 5-11 Αιτήσεις έκδοσης Κάρτας Αιμοδότη.....	45
Εικόνα 5-12 Προβολή στοιχείων αίτησης Κάρτας Αιμοδότη.....	46
Εικόνα 5-13 Δημιουργία αίτησης για έκδοση Κάρτας Αιμοδότη.....	46
Εικόνα 5-14 Φόρμα επικοινωνίας.....	47
Εικόνα 6-1 Επικοινωνία διακομιστή-πελάτη μέσω διαδικτύου με χρήση REST.....	49
Εικόνα 6-2 Αρχιτεκτονική REST API.....	50
Εικόνα 6-3 Σύστημα με πολλαπλά επίπεδα.....	53
Εικόνα 6-4 Επισκόπηση της ιστοσελίδας.....	89

Εικόνα 6-5 Περιγραφή της δομής των requests και παραδείγματα εκτέλεσης τους με διάφορες γλώσσες και SDKs.....	89
Εικόνα 6-6 Περιγραφή και παραδείγματα των απαντήσεων των requests.....	90
Εικόνα 7-1 Τα τρία βασικά συστατικά του React Native.....	95
Εικόνα 7-2 Μηχανισμός εμφάνισης στοιχείων με χρήση WebViews σε αντιπαράθεση με τη χρήση React Native.....	97
Εικόνα 8-1 Οθόνη σύνδεσης χρήστη σε αρχική κατάσταση και αφού εισάγει ο χρήστης τα στοιχεία του	100
Εικόνα 8-2 Σφάλμα κατά την προσπάθεια σύνδεσης χρήστη	101
Εικόνα 8-3 Διαφορετικές επιλογές πλοήγησης ανάμεσα σε συνδεδεμένους και μη χρήστες	102
Εικόνα 8-4 Αρχική οθόνη	103
Εικόνα 8-5 Προφίλ χρήστη.....	104
Εικόνα 8-6 Προφίλ χρήστη.....	104
Εικόνα 8-7 Λίστα αιτήσεων έκδοσης ταυτότητας αιμοδότη	106
Εικόνα 8-8 Λεπτομέρειες αίτησης έκδοσης ταυτότητας αιμοδότη	107
Εικόνα 8-9 Λεπτομέρειες αίτησης έκδοσης ταυτότητας αιμοδότη	108
Εικόνα 8-10 Καρτέλες με τα στοιχεία των αιμοδοσιών χρήστη.....	109
Εικόνα 8-11 Αριστερά οι καλύψεις του χρήστη και δεξιά οι αιμοδοσίες του πριν τη λειτουργία του Εθνικού Μητρώου Αιμοδοτών.....	110
Εικόνα 8-12 Ημερολόγιο	111
Εικόνα 8-13 Επικοινωνία χρηστών με το γραφείο αρωγής της υπηρεσίας.....	112
Εικόνα 8-14 Πληροφορίες σχετικά με την εφαρμογή	113

Κατάλογος Πινάκων

Πίνακας 2-1 Χρήση διαδικτύου σε Ευρώπη και Ελλάδα το 2020.....	19
--	----

Εισαγωγή

Από το 2015 έχει υλοποιηθεί και λειτουργεί στην Ελλάδα το κεντρικό πληροφοριακό σύστημα «Εθνικό Μητρώο Αιμοδοτών», μέσω του οποίου υποστηρίζεται η ενιαία διαχείριση των διαδικασιών που αφορούν την αιμοδοσία στη χώρα.

Στα πλαίσια της παρούσας διπλωματικής εξετάστηκαν οι λειτουργικότητες που διατίθενται στους εθελοντές αιμοδότες από το Εθνικό Μητρώο Αιμοδοτών τόσο μέσω γραφικού περιβάλλοντος χρήστη (User Interface), όσο και μέσω της διεπαφής προγραμματισμού εφαρμογών (application programming interface ή εν συντομία API). Στη συνέχεια, χρησιμοποιώντας κατάλληλες τεχνολογίες αιχμής, αναπτύχθηκε μία εφαρμογή για τους αιμοδότες, η οποία λειτουργεί σε κινητά τερματικά.

Συγκεκριμένα, για τη διπλωματική αυτή τέθηκαν οι εξής στόχοι:

1. Μελέτη του API , που χρησιμοποιείται από τις εφαρμογές που εξυπηρετούν τις ανάγκες των εθελοντών αιμοδοτών και στη συνέχεια σχεδιασμός ενός νέου API, με τα εξής χαρακτηριστικά:
 - Κατανοητό και εύκολο στη χρήση
 - Αποδοτική μετάδοση της πληροφορίας μέσω του διαδικτύου (network performance)
2. Σχεδιασμός και υλοποίηση μίας εφαρμογής αιμοδότη, η οποία να λειτουργεί σε σύστημα Android. Ως ζητούμενα χαρακτηριστικά για αυτήν την εφαρμογή τέθηκαν:
 - Αποδοτική χρήση των πόρων της συσκευής στην οποία εκτελείται
 - Ανοιχτός κώδικας και διαφάνεια
 - Σύγχρονες τακτικές υλοποίησης και εξασφάλισης της ποιότητας έτσι ώστε να είναι δυνατή η συντήρηση του κώδικα σε βάθος χρόνου
 - Ευχρηστία
 - Διευκόλυνση της πρόσβασης ατόμων με δυσκολίες στην όραση
 - Σύγχρονο design

1 Το σύστημα της αιμοδοσίας στην Ελλάδα

Σύμφωνα με το Εθνικό Κέντρο Αιμοδοσίας (Ε.ΚΕ.Α.) [1], μέχρι το 2005 η οργάνωση της αιμοδοσίας περιελάμβανε την Κεντρική Υπηρεσία (Διεύθυνση Αιμοδοσίας) στο Υπουργείο Υγείας και τις Περιφερειακές Υπηρεσίες (Κέντρα και Σταθμούς Α και Β τάξεως) με έδρα στα νοσοκομεία της χώρας (Ν.1820/17-11-1988). Βάσει αυτού του οργανωτικού σχήματος στην χώρα μας προβλεπόταν η ύπαρξη:

- 14 περιφερειακών κέντρων
- 14 σταθμών Α τάξης με δυνατότητα συλλογής και ελέγχου του αίματος και
- 65 σταθμών Β τάξης με ή χωρίς δικαίωμα αιμοληψιών και ελέγχων του αίματος.

Όλοι αυτοί οι εμπλεκόμενοι φορείς, οι σταθμοί και τα κέντρα είχαν δικά τους, αυτόνομα συστήματα μηχανογράφησης. Σε πολλές περιπτώσεις τα αρχεία βασίζονταν σε χειρόγραφες καταγραφές. Υπήρχαν όμως και αρκετές περιπτώσεις όπου αναπτύχθηκαν ηλεκτρονικά συστήματα μηχανογράφησης, τα οποία στόχευαν στην κάλυψη των αναγκών των επιμέρους υπηρεσιών. Τα συστήματα αυτά ήταν πολλά σε αριθμό και δεν είχαν υλοποιηθεί με πρόβλεψη την επικοινωνία των στοιχείων μεταξύ τους.

Όλα τα παραπάνω επιβεβαίωσαν τις αρχικές εκτιμήσεις του Δρ. Ηλία Πολίτη [1], [2], [3], εμπνευστή και οραματιστή της αιμοδοσίας στην Ελλάδα, για την αναγκαιότητα συντονισμού όλου του έργου της αιμοδοσίας.

Το 2005 ψηφίστηκε νέος νόμος, ο οποίος αξιοποιώντας τη διεθνή εμπειρία, άλλαξε το σύνολο της οργανωτικής δομής και κατ' επέκταση τέθηκαν νέες απαιτήσεις για το πληροφοριακό σύστημα που θα την εξυπηρετούσε (Ν. 3402/17-10-2005). Η νέα δομή προέβλεπε, όπως όριζε και η Ευρωπαϊκή οδηγία 2002/98/ΕΚ και την οποία ουσιαστικά ενσωμάτωσε, τη δημιουργία:

- Εθνικού Κέντρου Αιμοδοσίας ως την αρμόδια αρχή για τον συντονισμό, τον έλεγχο, την αδειοδότηση και την εποπτεία των επιμέρους υπηρεσιών της αιμοδοσίας. Κύρια αποστολή του Εθνικού Κέντρου Αιμοδοσίας ορίστηκε ο συντονισμός της εξεύρεσης και συλλογής αίματος, ο έλεγχος της αποθεματοποίησης και κατεργασίας του καθώς και ο έλεγχος της διακίνησης.
- Κέντρων Αίματος, τα οποία ορίζονται ως εξωνοσοκομειακές υπηρεσίες αιμοδοσίας με κύριες αρμοδιότητες την διενέργεια αιμοληψιών και τη διακίνηση, την επεξεργασία και τον έλεγχο των μονάδων αίματος.
- Νοσοκομειακών υπηρεσιών αιμοδοσίας, οι οποίες ορίζονται ως νοσοκομειακές μονάδες που συνεργάζονται με τα Κέντρα αλλά έχουν σαν κύρια αρμοδιότητα τους την πραγματοποίηση αιμοληψιών, τη συντήρηση στις υποδομές τους αίματος και παραγώγων και τη διενέργεια όλων εκείνων των εξετάσεων προκειμένου να εξασφαλίζεται η ασφαλής και σωστή μετάγγιση του αίματος και των παραγώγων αυτού στον ασθενή.

Το Εθνικό Μητρώο Αιμοδοτών υλοποιήθηκε στο πλαίσιο της πράξης «Κεντρικές Ηλεκτρονικές Υπηρεσίες Διαχείρισης Εθνικού Μητρώου Εθελοντών Αιμοδοτών» και αποτελείται από ένα ενοποιημένο, κεντρικό και καινοτόμο πληροφοριακό σύστημα που επιτρέπει τη διαχείριση του μητρώου των αιμοδοτών και την καθιέρωση σύγχρονων και φιλικών διαδικασιών, οι οποίες διευκολύνουν τόσο τους αιμοδότες όσο και τις υπηρεσίες αιμοδοσίας της χώρας.

Σκοπός του Εθνικού Μητρώου Αιμοδοτών είναι:

- Να αυξήσει την ασφάλεια και την ποιότητα του αίματος όπως και των παραγώγων αυτού.
- Να ενοποιήσει και να αξιοποιήσει τα στατιστικά στοιχεία μέσα από τη δημιουργία καινοτόμων εργαλείων, τα οποία θα αναβαθμίσουν τον τρόπο λήψης αποφάσεων προς όφελος των πολιτών.
- Να αυξήσει την ευαισθητοποίηση των πολιτών και να ενθαρρύνει τον εθελοντισμό εμπνέοντας τους ήδη υπάρχοντες εθελοντές αιμοδότες και προσελκύοντας νέους.

Ειδικότερα, η υλοποίηση ενός καινοτόμου κεντρικού πληροφοριακού συστήματος για το Εθνικό Μητρώο Αιμοδοτών έδωσε τη δυνατότητα:

- Αντικατάστασης των υπάρχουσών κατακερματισμένων βάσεων δεδομένων που υπάρχουν σε τοπικό επίπεδο.
- Βελτίωσης και εξορθολογισμού της διαδικασίας καταχώρησης των μονάδων αίματος μέσω της λειτουργίας ενός διαδικτυακού συστήματος.
- Ενοποίησης των εγγραφών των αιμοδοτών και αύξησης της ασφάλειας και της ποιότητας του αίματος.
- Έκδοσης και παροχής Ταυτοτήτων Εθελοντών Αιμοδοτών στους εθελοντές δότες.
- Δημιουργίας αναφορών ανά αιμοδότη αντί ανά συλλεχθείσας μονάδας αίματος.
- Παρακολούθησης της συλλογής αίματος σε εθνικό επίπεδο.
- Ανάπτυξης ενός πλαισίου για τον καθορισμό της δίκαιης κατανομής των ασταθών παραγώγων του αίματος σε περιόδους σοβαρής έλλειψης.
- Καταγραφής και παρακολούθησης της δωρεάς αίματος.
- Σχεδιασμού και υλοποίησης στοχευμένων ενεργειών ευαισθητοποίησης για την προώθηση της δωρεάς αίματος και του «Δώρου Ζωής» στο ευρύ κοινό.
- Κάμψης της άρνησης δωρεάς αίματος εστιάζοντας σε στοχευμένες ενέργειες μάρκετινγκ στους δότες.
- Επανεξέτασης και αποτίμησης των μεθόδων μάρκετινγκ και του σχεδιασμού εξορμήσεων, προκειμένου να δημιουργηθεί ένα στρατηγικό σχέδιο μάρκετινγκ για την αιμοδοσία.
- Μεγιστοποίησης της μετατροπής των δυνητικών δοτών σε πραγματικούς δότες αίματος και της μετατροπής των δοτών αντικατάστασης σε εθελοντές μη αμειβόμενους αιμοδότες, βοηθώντας το Ε.ΚΕ.Α. να αναπτύξει και να εφαρμόσει μια ισχυρή και βιώσιμη υπηρεσία συντονισμού των αιμοδοτών. [4]

Το πληροφοριακό σύστημα που προαναφέρεται αποτέλεσε τη βάση για τη μελέτη που πραγματοποιήθηκε στο πλαίσιο της παρούσας διπλωματικής καθώς και για την υλοποίηση της εφαρμογής. Συγκεκριμένα, αναλύθηκε το ήδη υλοποιημένο API καθώς και ο web client (διαδικτυακή εφαρμογή-πελάτης) που χρησιμοποιείται από το σύνολο των αιμοδοτών της χώρας. Επιπλέον, αξιοποιήθηκαν στοιχεία που προέκυψαν όλα αυτά τα έτη λειτουργίας των ηλεκτρονικών υπηρεσιών αιμοδοσίας και από τη συνεχή επικοινωνία με τους χρήστες. Ο συνδυασμός των παραπάνω αξιοποιήθηκε για τη διαμόρφωση της πρότασης για ένα νέο API και μία εφαρμογή που να καλύπτει τους στόχους που προαναφέρθηκαν.

Εθνικό Μητρώο Αιμοδοτών

Ελληνικά / English

Έχετε συνδεθεί ως **admin** (Όροι Χρήσης) Αποσύνδεση

Επισκόπηση Ιστορικού Αιμοδοτήσεων

Παλιές Κάρτες

ΝΥΑ Αιμοληψίας

Πληκτρολογήστε την επιλογή σας, "" για όλα

Ημερομηνία Επίσκε...	Όνομα Νοσοκομείου	Προσφορά	Λόγος
09/05/2018	ΙΠΠΟΚΡΑΤΕΙΟ ΑΘΗΝΩΝ	Ολικού Αίματος	Για Σύλλογο ΕΑ
07/10/2016	401	Ολικού Αίματος	Για ασθενή
10/06/2016	401	Ολικού Αίματος	Για ασθενή

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Εικόνα 1-1 Δικτυακή εφαρμογή για τους αιμοδότες (web client)

Στο επόμενο κεφάλαιο περιγράφονται οι συνθήκες και οι ανάγκες που οδήγησαν στην ανάπτυξη μίας εφαρμογής για τους αιμοδότες της Ελλάδας που μπορεί να λειτουργήσει σε κινητά τερματικά.

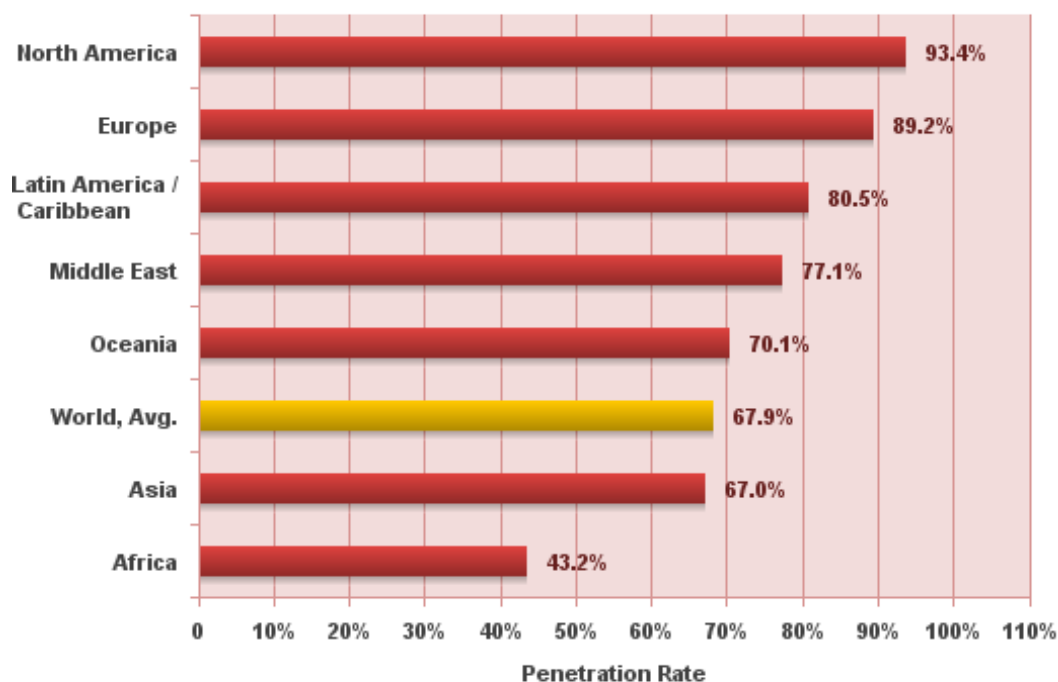
2 Λόγοι αξιοποίησης τεχνολογιών αιχμής

Η ποιότητα μετάδοσης πληροφοριών μεταξύ των ανθρώπων είναι κρίσιμος δείκτης της επικοινωνίας και της καλής συνεργασίας μεταξύ των μελών μιας ομάδας ή/και ολόκληρης της κοινωνίας. Η μετάδοση αυτή έχει αναντίρρητα εξελιχθεί ραγδαία, ιδιαίτερα τα τελευταία έτη. Στις μέρες μας, αξιοποιώντας τα τεχνολογικά εργαλεία που είναι διαθέσιμα, ο τρόπος μετάδοσης των πληροφοριών έχει ξεπεράσει προβλήματα που κάποτε θεωρούνταν ανυπέρβλητα, όπως η απόσταση, ο χρόνος ή η χαμηλή ποιότητα της μεταδιδόμενης πληροφορίας.

Μεγάλη επανάσταση στη διάδοση των πληροφοριών προκάλεσε το διαδίκτυο. Πλέον, διαθέσιμες ιδέες, γνώσεις, πληροφορίες και γεγονότα διαδίδονται εύκολα κι άμεσα σε μεγάλο μέρος του πληθυσμού. Επιπλέον, μέσω του διαδικτύου κατέστη δυνατόν να δημιουργηθούν και να διατεθούν λειτουργίες που απλοποιούν σημαντικά τη ζωή των ανθρώπων.

Μερικά χαρακτηριστικά παραδείγματα τέτοιων λειτουργιών είναι οι διαδικτυακές αγοραπωλησίες ή οι διαδικτυακές συναλλαγές των πολιτών με τη δημόσια διοίκηση. Είναι αδιαμφισβήτητο γεγονός ότι η λίστα όλων των τρόπων αξιοποίησης του διαδικτύου είναι ανεξάντλητη. Αυτό είναι εμφανές από την εξέταση των στατιστικών χρήσης του διαδικτύου στην πάροδο του χρόνου.

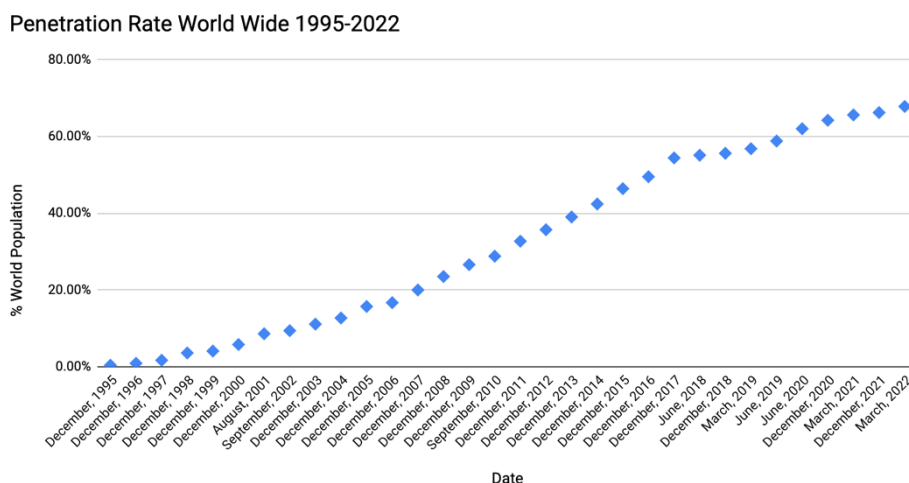
Internet World Penetration Rates by Geographic Regions - 2022



Source: Internet World Stats - www.internetworldstats.com/stats.htm
Penetration Rates are based on a world population of 7,932,791,734 and 5,385,798,406 estimated Internet users in June 30, 2022.
Copyright © 2022, Miniwatts Marketing Group

Εικόνα 2-1 Ποσοστό χρήσης του διαδικτύου ανά ήπειρο το 2022

Στο παραπάνω γράφημα παρουσιάζεται το ποσοστό των ανθρώπων που είναι χρήστες του διαδικτύου ανά ήπειρο αλλά και ο μέσος όρος αυτών, το 2022 [5]. Από την ίδια πηγή εκτιμάται ότι στις 30 Ιουνίου 2022 οι χρήστες του διαδικτύου παγκοσμίως ήταν 5.385.798.406, αριθμός που αντιστοιχεί στο 67,9% του παγκόσμιου πληθυσμού.



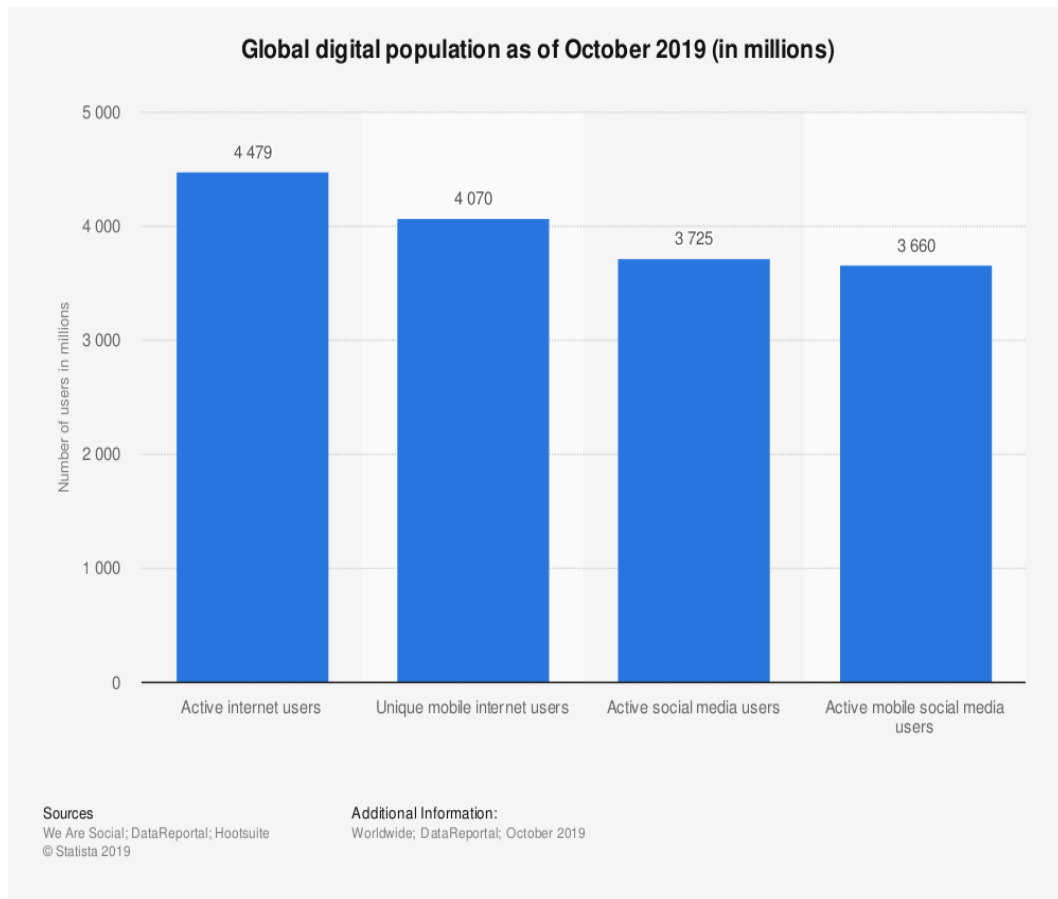
Εικόνα 2-2 Στατιστικά χρήσης του διαδικτύου παγκοσμίως από το 1995 έως και το 2022 [6]

Εξετάζοντας τα πιο πρόσφατα διαθέσιμα στοιχεία για την Ευρώπη και την Ελλάδα [7] βλέπουμε για τον Δεκέμβριο του 2020 τα εξής:

Γεωγραφική Περιοχή	Αριθμός Χρηστών	Ποσοστό
Ευρώπη	727,559,682	87,7 %
Ελλάδα	8.115.397	72,9 %

Πίνακας 2-1 Χρήση διαδικτύου σε Ευρώπη και Ελλάδα το 2020

Ένα άλλο σημαντικό σημείο που οδήγησε στην ραγδαία αύξηση της χρήσης του διαδικτύου παγκοσμίως είναι και η ανάπτυξη νέων μέσων πρόσβασης σε αυτό. Πέρα από τη χρήση των ηλεκτρονικών υπολογιστών, σημαντικό ποσοστό των ενεργειών στο διαδίκτυο στις μέρες μας γίνεται μέσω κινητών τερματικών, τηλεφώνων και tablets. Η πρόσβαση σε πληροφορίες και η διεκπεραίωση εργασιών είναι, πλέον, κυριολεκτικά στο χέρι του χρήστη από οπουδήποτε κι αν αυτός βρίσκεται.



Εικόνα 2-3 Χρήση ψηφιακών μέσων το 2019

Σύμφωνα με τα στοιχεία της Statista και του J.Clement [8] το 2019, περίπου 4,07 δισεκατομμύρια άνθρωποι χρησιμοποιούσαν το κινητό τους για να αποκτήσουν πρόσβαση στο διαδίκτυο. Ο αριθμός αυτός αποτελεί ένα σημαντικό ποσοστό του συνόλου των χρηστών του διαδικτύου.

Η μεγάλη χρήση κινητών μέσων προκαλεί και μία αντίστοιχη αύξηση στην ανάπτυξη υπηρεσιών που επιτρέπουν την εύκολη πρόσβαση σε αυτές με χρήση τέτοιων μέσων. Αυτές οι υπηρεσίες σχετίζονται με όλες τις εκφάνσεις της ζωής των ανθρώπων. Συνεπώς, μπορεί να διαθέτουν λειτουργίες για τη διευκόλυνση εμπορικών συναλλαγών ή την αλληλεπίδραση των πολιτών με τη δημόσια διοίκηση ή ακόμα και λειτουργίες σχετιζόμενες με το σύστημα υγείας. Επιπροσθέτως, η ανάπτυξη και η χρήση τέτοιων υπηρεσιών ωφελεί το κοινωνικό σύνολο όχι μόνο γιατί διευκολύνει την καθημερινότητα των μελών του, αλλά επειδή καθιστούν δυνατή και την αποτελεσματικότερη και με μεγαλύτερη διαφάνεια οργάνωση και διαχείριση των σχετικών πόρων και διαδικασιών.

Με βάση λοιπόν όλα τα παραπάνω δεδομένα, δηλαδή την υψηλή αλλά και αυξανόμενη χρήση του διαδικτύου και των κινητών συσκευών, τόσο σε παγκόσμια κλίμακα όσο και στην ίδια την Ελλάδα, είναι παραπάνω από προφανής η χρησιμότητα ανάπτυξης μίας εφαρμογής που θα λειτουργεί σε κινητά τερματικά και θα απευθύνεται στους εθελοντές αιμοδότες.

3 Διαδικασία ανάπτυξης της εφαρμογής

Στο παρόν κεφάλαιο περιγράφεται η διαδικασία, η μεθοδολογία και τα βήματα που ακολουθήθηκαν προκειμένου να επιτευχθούν οι στόχοι της διπλωματικής εργασίας. Η σειρά εργασιών προέκυψε από μελέτη αντίστοιχων διαδικασιών που ακολουθούνται στον ευρύτερο κλάδο της πληροφορικής.

Πρώτο βήμα για την ανάπτυξη της εφαρμογής αποτέλεσε η μελέτη του τρόπου χρήσης του web client από τους αιμοδότες. Μετά τον εντοπισμό και τη διερεύνηση του τρόπου χρήσης ηλεκτρονικών υπηρεσιών, εξετάστηκε και αναλύθηκε ο τρόπος διάταξης των στοιχείων, των πληροφοριών και των λειτουργιών που παρέχονται στους χρήστες (βλ. κεφ. 4).

Με βάση τα δύο παραπάνω και λαμβάνοντας υπ' όψιν τα χαρακτηριστικά των συσκευών που στοχεύει η προς σχεδιασμό εφαρμογή, εντοπίστηκαν πιθανές βελτιώσεις στην αρχιτεκτονική της πληροφορίας, στον τρόπο παρουσίασης των πληροφοριών και των ενεργειών που διατίθενται, καθώς και στα δεδομένα και στις ενέργειες που χρειάζεται να προσφέρει το API (βλ. κεφ. 5 και κεφ. 6).

Τα επόμενα βήματα αφορούσαν το σχεδιασμό της διεπαφής. Μελετήθηκαν πρακτικές για τον σχεδιασμό χρηστικών κι εύχρηστων διεπαφών όπως και προτεινόμενες πρακτικές που ακολουθούνται για εφαρμογές Android. Επιπλέον, δημιουργήθηκαν wireframes για την καλύτερη αποτύπωση της διάταξης των στοιχείων των οθονών (βλ. κεφ. 5).

Στη συνέχεια ακολούθησε λεπτομερής μελέτη του υπάρχοντος API καθώς και των σύγχρονων και ευρέως διαδεδομένων πρακτικών ανάπτυξης των APIs. Με βάση όλα τα παραπάνω προτάθηκε ένα νέο API, που στόχευε στην αποδοτική αξιοποίηση του διαδικτύου και στην βελτίωση της χρήσης των δυνατοτήτων που προσφέρονται στους clients (βλ. κεφ. 6).

Το επόμενο σημαντικό βήμα αφορούσε στην επιλογή των τεχνολογιών με τις οποίες θα υλοποιούνταν η εφαρμογή και οι δοκιμές ελέγχου (tests). Η διαδικασία που ακολουθήθηκε περιλάμβανε τον ορισμό των καθοριστικών κριτηρίων και τη διερεύνηση των σύγχρονων τακτικών για την ανάπτυξη εφαρμογών που εκτελούνται σε λειτουργικό σύστημα Android (βλ. κεφ 7).

Έπειτα έγινε η υλοποίηση των οθονών και των λειτουργιών τους καθώς και των αντίστοιχων tests. Κατά την διάρκεια της υλοποίησης επαναξιολογήθηκαν οι —έως εκείνη τη στιγμή— σχεδιαστικές επιλογές και εφαρμόστηκαν τροποποιήσεις όπου χρειάστηκε (βλ. κεφ. 7 και κεφ. 8).

Η παρούσα διπλωματική ολοκληρώνεται με την περιγραφή των κυριότερων δυσκολιών που αντιμετωπίστηκαν και με μία αναφορά σε ενδεχόμενες μελλοντικές επεκτάσεις και βελτιώσεις της εφαρμογής. (Βλ. κεφ. 9)

4 Μελέτη της διαδικτυακής εφαρμογής

Ο σαφής προσδιορισμός του ρόλου, των ιδιοτήτων και των σκοπών χρήσης της υπάρχουσας διαδικτυακής εφαρμογής υπήρξε σημαντικό σημείο μελέτης. Είναι θεμιτό μεγάλο μέρος των χρηστών να εξυπηρετούνται εύκολα και από την υπό σχεδίαση εφαρμογή για κινητά τερματικά. Προκείμενου να σχεδιαστεί εύστοχα η νέα εφαρμογή απαιτήθηκε η κατάλληλη συλλογή πληροφοριών για αυτούς και η σκιαγράφηση του προφίλ τους.

Οι εν λόγω χρήστες έχουν προσφέρει τουλάχιστον μία φορά αίμα στη ζωή τους. Ορισμένοι προσέφεραν αίμα και πριν τη λειτουργία του Εθνικού Μητρώου Αιμοδοτών. Επιπλέον, ένας αριθμός αιμοδοτών έχει διαθέσει αίμα για την κάλυψη των αναγκών συγκεκριμένων συνανθρώπων τους. Η συντριπτική πλειοψηφία των χρηστών είναι ελληνόφωνοι, κατοικούν στην Ελλάδα και έχουν στην κατοχή τους ελληνικά δημόσια έγγραφα για την ταυτοποίησή τους. Υπάρχουν και περιπτώσεις χρηστών που δεν έχουν γεννηθεί στην Ελλάδα ή/και χρησιμοποιούν έγγραφα άλλης χώρας για την ταυτοποίησή τους. Επίσης, υπάρχει ένας μικρός αριθμός χρηστών που τον δυσκολεύει η χρήση της εφαρμογής στην ελληνική γλώσσα. Τέλος, αξίζει να σημειωθεί ότι δεν είναι όλοι οι χρήστες εξοικειωμένοι με τη χρήση σύγχρονων τεχνολογιών και εργαλείων. Όλα αυτά τα στοιχεία έχουν ληφθεί υπ' όψιν κατά το σχεδιασμό της web εφαρμογής και στο σχεδιασμό της εφαρμογής κινητών τερματικών.

Μετά τον εντοπισμό των χαρακτηριστικών των χρηστών, εντοπίστηκαν οι στόχοι που θέλουν να επιτύχουν.

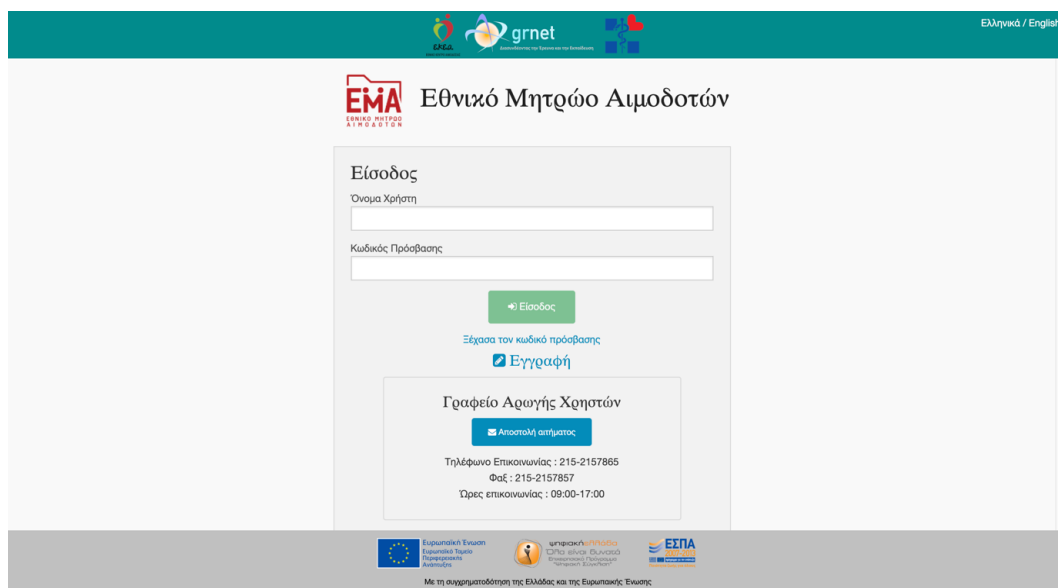
Οι εν λόγω στόχοι, που θέλουν να επιτύχουν οι αιμοδότες με τη χρήση της εφαρμογής είναι:

- Ο χρήστης θέλει να δει πότε μπορεί να προσφέρει αίμα ξανά.
- Ο χρήστης θέλει να δείξει στοιχεία της ταυτότητας ή του λογαριασμού του στη μονάδα αιμοδοσίας.
- Ο χρήστης θέλει να αλλάξει τα στοιχεία επικοινωνίας που έχει δηλώσει.
- Ο χρήστης θέλει να αλλάξει τις επιλογές υπενθύμισης.
- Ο χρήστης θέλει να αποκτήσει ταυτότητα αιμοδότη.
- Ο χρήστης θέλει να δει σε ποιο στάδιο βρίσκεται η αίτησή του για έκδοση ταυτότητας αιμοδότη.
- Ο χρήστης θέλει να δει τα στοιχεία των αιμοδοσιών ή/και των καλύψεων που έχει πραγματοποιήσει.

Από το συνδυασμό των παραπάνω στοιχείων με μία λεπτομερή ανάλυση της διαδικτυακής εφαρμογής προέκυψε η ακολουθία βημάτων που ακολουθούν οι χρήστες, στην παρούσα φάση, προκειμένου να επιτύχουν τους στόχους τους. Επιπλέον, μελετήθηκε η οργάνωση της πληροφορίας και η διάταξη των στοιχείων που περιέχονται σε κάθε οθόνη. Παρακάτω παρουσιάζονται μέσα από τις οθόνες της υπάρχουσας εφαρμογής όλες οι διαθέσιμες δυνατότητες και ο τρόπος παρουσίασής τους.

Η οθόνη που παρουσιάζεται στην εικόνα 4-1, είναι αυτή που συναντάει αρχικά κάθε χρήστης. Η κύρια λειτουργία της είναι η ταυτοποίηση του χρήστη προτού του δοθεί η δυνατότητα να δει και να επεξεργαστεί τα στοιχεία του λογαριασμού του. Εκτός αυτής της λειτουργίας, σε αυτό το σημείο δίνεται η δυνατότητα εγγραφής, υπενθύμισης κωδικού πρόσβασης καθώς και επικοινωνίας με το γραφείο αρωγής της υπηρεσίας. Επίσης, παρατηρήθηκε ότι η εφαρμογή είναι διαθέσιμη στα ελληνικά και στα αγγλικά. Επιπλέον, αναδεικνύεται το λογότυπο και η ονομασία της υπηρεσίας. Όλα αυτά τα στοιχεία και οι

λειτουργίες είναι καλό να είναι διαθέσιμες και στην εφαρμογή για κινητές συσκευές και μάλιστα, όσο είναι δυνατό, με αντίστοιχη διάταξη.



Εικόνα 4-1 Οθόνη εισόδου δικτυακής εφαρμογής

Η επόμενη οθόνη που παρουσιάζεται στον χρήστη περιλαμβάνει στη μεγαλύτερη έκτασή της τα στοιχεία του προφίλ του. Επίσης, στην πάνω πλευρά ξεχωρίζει μία λωρίδα, η οποία είναι σταθερή όποια οθόνη και αν επισκεφτεί ο χρήστης. Η λωρίδα, που βρίσκεται στην επάνω πλευρά της οθόνης, περιέχει το όνομα χρήστη, το όνομα της εφαρμογής και τους φορείς που την παρέχουν. Επιπλέον, υπάρχει σύνδεσμος για τους όρους χρήσης της υπηρεσίας και κουμπί για την αποσύνδεση και την αλλαγή γλώσσας. Ακριβώς κάτω από τη λωρίδα εμφανίζεται ο τίτλος της κάθε οθόνης. Το άλλο σταθερό σημείο της υπηρεσίας είναι το μενού πλοήγησης που βρίσκεται αριστερά. Εκεί αναφέρονται οι περισσότερες διαθέσιμες οθόνες που μπορεί να επισκεφθεί ο χρήστης. Μία αντίστοιχη διάταξη αξιοποιήθηκε για την Android εφαρμογή, η οποία περιλαμβάνει ένα πλευρικό μενού πλοήγησης και αντίστοιχα μία σταθερή λωρίδα στην επάνω πλευρά της οθόνης.

Επισκόπηση Στοιχείων Εθελοντή Αιμοδότη

Στοιχεία Αιμοδότη	
Αριθμός Μητρώου Αιμοδότη:	3000286634
ΑΜΚΑ:	5000286634
ΑΤΕΑ:	301602582936
Επώνυμο:	ΕΠΩΝΥΜΟ ΓΙΗΗΔΕ
Όνομα:	ΟΝΟΜΑ ΓΙΗΗΔΕ
Πατρώνυμο:	ΠΑΤΡΩΝΥΜΟ ΓΙΗΗΔΕ
Μητρώνυμο:	ΜΗΤΡΩΝΥΜΟ ΓΙΗΗΔΕ
Ημερομηνία Γέννησης:	05/02/1985
Φύλο:	Θήλυ
Χώρα γέννησης:	ΕΛΛΑΔΑ
Περιφέρεια γέννησης:	ΚΕΝΤΡΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
Μορφωτικό Επίπεδο:	Πτυχιούχος Πανεπιστημίου ή ΤΕΙ
Ομάδα Αίματος:	O
Rhesus:	(+)

Διεύθυνση	
Οδός:	ΟΔΟΣ ΓΙΗΗΔΕ
Αριθμός/ Τ.Θ.:	15-17
Δήμος:	ΚΑΛΑΜΑΡΙΑΣ (ΘΕΣΣΑΛΟΝΙΚΗΣ)
T.K.:	55133

Επικοινωνία	
E-mail:	blooddonsord8ss6634a@stg.bdr.grnet.gra
Σταθερό Τηλέφωνο:	2000186634
Κινητό Τηλέφωνο:	6923186634

Προσκομισθέντα Δημόσια Έγγραφα (2)	
Βιβλίο Υγείας:	05028501988 (Χώρα έκδοσης εγγράφου: ΕΛΛΑΔΑ)
Αστυνομική Ταυτότητα:	AK904036 (Χώρα έκδοσης εγγράφου: ΕΛΛΑΔΑ)

Εικόνα 4-2 Στοιχεία προφίλ χρήστη

Έπειτα εξετάζεται η οθόνη που έχει τίτλο «Διαχείριση Προφίλ». Εκεί παρατηρούμε ότι ο χρήστης βλέπει και πάλι τα στοιχεία του προφίλ του. Όμως, στο κάτω μέρος της σελίδας του δίνεται η δυνατότητα τροποποίησης των στοιχείων επικοινωνίας που έχει δηλώσει. Η τροποποίηση επιτυγχάνεται από μία φόρμα η οποία περιλαμβάνει ορισμένα στοιχεία με στόχο τη διευκόλυνση του χρήστη. Συγκεκριμένα, περιέχει μία λίστα με όλους τους δήμους και μία με τις περιφέρειες της Ελλάδος. Η τιμή του καθενός από αυτά τα πεδία επηρεάζει το άλλο. Επιπλέον, σε όποια πεδία είναι γνωστή η μορφή των στοιχείων, γίνεται κατά την πληκτρολόγηση έλεγχος και ενημέρωση του χρήστη σε περίπτωση που τα στοιχεία που εισάγει δεν είναι της αναμενόμενης μορφής.

Επισκόπηση Στοιχείων Εθελοντή Αιμοδότη

Στοιχεία Αιμοδότη	
Χώρα γέννησης:	ΕΛΛΑΔΑ
Περιφέρεια γέννησης:	ΚΕΝΤΡΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
Μορφωτικό Επίπεδο:	Πτυχιούχος Πανεπιστημίου ή ΤΕΙ
Ομάδα Αίματος:	O
Rhesus:	(+)

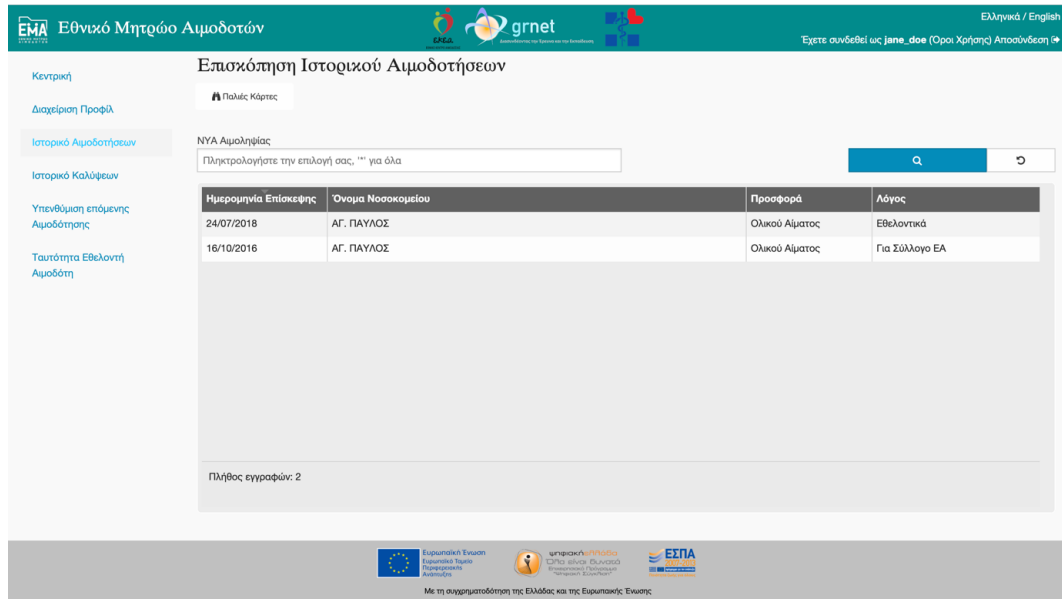
Διεύθυνση		
Οδός*	Αριθμός/ Τ.Θ.	T.K.
ΠΕΙΡΑΙΩΣ	15-17	11133
Δήμος*	Περιφέρεια*	
ΚΑΛΑΜΑΡΙΑΣ	ΚΕΝΤΡΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ	

Επικοινωνία		
E-mail	Σταθερό Τηλέφωνο	Κινητό Τηλέφωνο
blood@example.org	2000186634	6923186634

Εικόνα 4-3 Διαχείριση προφίλ χρήστη

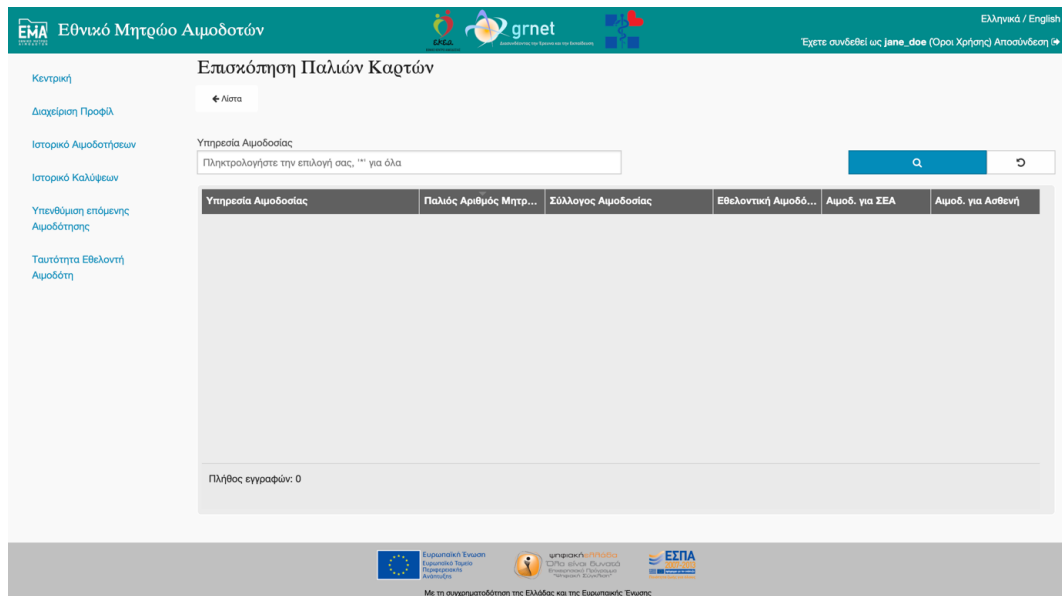
Η επόμενη προς ανάλυση οθόνη περιλαμβάνει τη λίστα με τα στοιχεία των αιμοδοσιών του χρήστη. Η οθόνη περιλαμβάνει τα σημαντικά στοιχεία της κάθε αιμοδοσίας. Από αυτήν την

οθόνη εξήχθησαν τα κύρια στοιχεία κάθε αιμοδοσίας για να παρουσιαστούν στην ανάλογη οθόνη της προς υλοποίηση εφαρμογής. Επειδή η νέα εφαρμογή πρόκειται να χρησιμοποιηθεί κατά κύριο λόγο από συσκευές που διαθέτουν μικρή σε μέγεθος οθόνη, η μορφή της λίστας που σχεδιάστηκε διαφέρει κατά πολύ από αυτή της διαδικτυακής εφαρμογής.



Εικόνα 4-4 Επισκόπηση παλαιών καρτών

Εάν ο χρήστης υπήρξε αιμοδότης και πριν τη λειτουργία του Εθνικού Μητρώου Αιμοδοτών μέσω του οποίου πλέον καταγράφονται αυτόματα όλες οι αιμοδοτήσεις, μπορεί να μεταβεί σε άλλη οθόνη προκειμένου να δει τα στοιχεία αυτών των αιμοδοτήσεων. Σε αυτήν την οθόνη μπορεί να μεταβεί από το κουμπί που βρίσκεται πάνω από τη λίστα των αιμοδοσιών, το οποίο φαίνεται στην εικόνα 4-4. Στη νέα εφαρμογή επιλέχθηκε να είναι πιο εμφανής ο τρόπος μετάβασης σε αυτά τα στοιχεία μέσω κατάλληλης ενότητας στο πλευρικό μενού.



Εικόνα 4-5 Ιστορικό αιμοδοτήσεων

Εξετάζοντας την οθόνη με τίτλο «Επισκόπηση Ιστορικού Καλύψεων», βλέπουμε ότι η διάταξη που χρησιμοποιείται είναι αντίστοιχη με αυτή που χρησιμοποιείται και για την παρουσίαση των στοιχείων των αιμοδοσιών του χρήστη. Ο ομοιόμορφος τρόπος παρουσίασης είναι ένα χρήσιμο χαρακτηριστικό που βελτιώνει την εμπειρία χρήσης της εφαρμογής, καθώς ο χρήστης «εκπαιδεύεται» ευκολότερα στη χρήση της. Για αυτό το λόγο θεωρήθηκε κρίσιμο να ακολουθηθεί ως τακτική σχεδιασμού και των οθονών της εφαρμογής για κινητά τερματικά.

Επισκόπηση Ιστορικού Καλύψεων

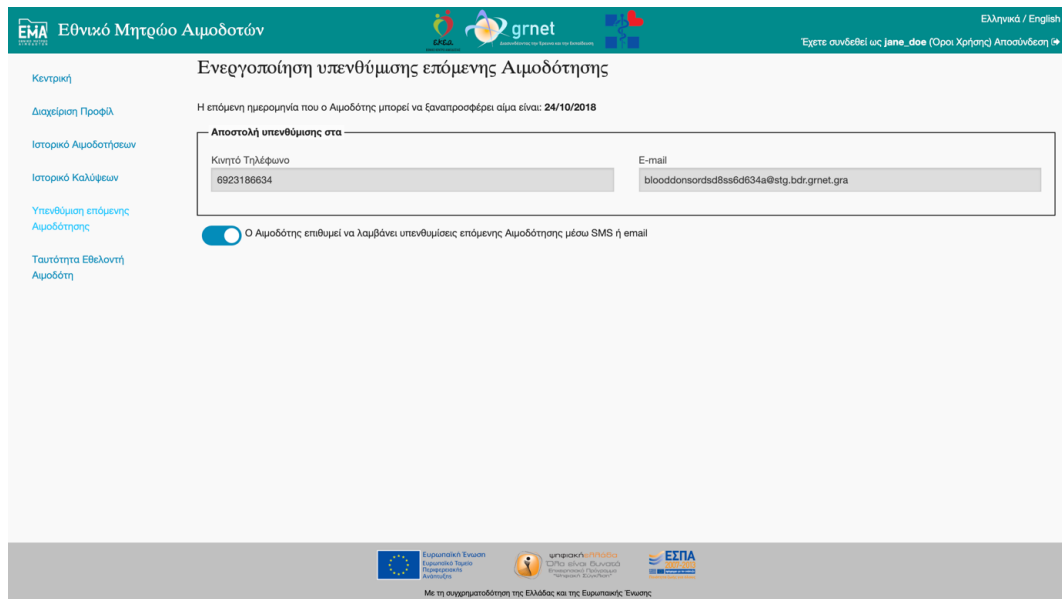
ΝΥΑ
 Πληκτρολογήστε την επιλογή σας, "" για όλα

Ημερομηνία Κάλυψης	Όνομα Νοσοκομείου	Αριθμός Μονάδων Κάλυψης
16/09/2018	ΑΓ. ΠΑΥΛΟΣ	1
30/03/2017	ΑΓ. ΠΑΥΛΟΣ	1

Πλήθος εγγραφών: 2

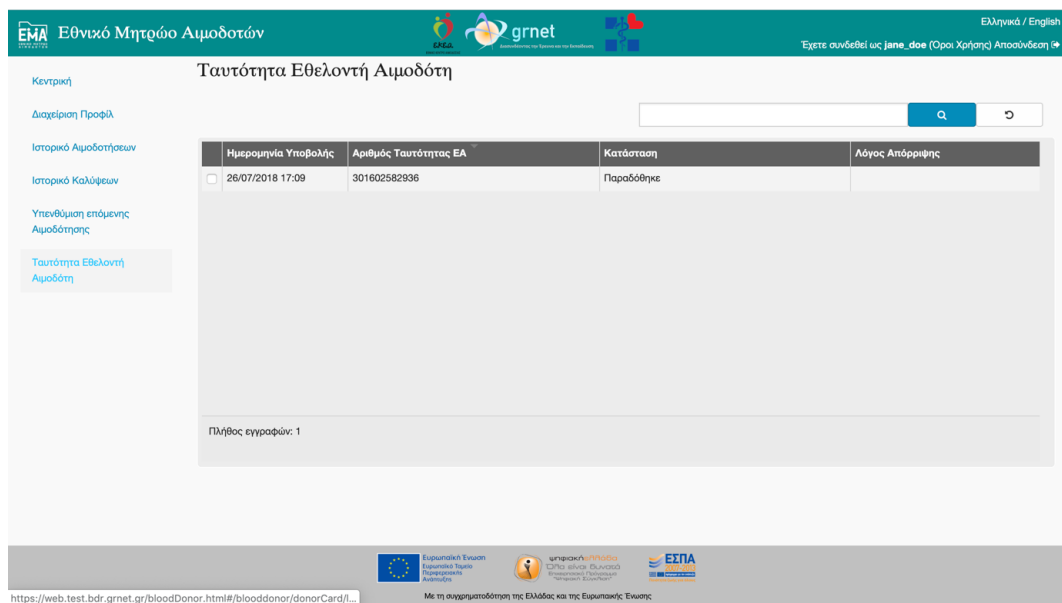
Εικόνα 4-6 Επισκόπηση ιστορικού καλύψεων

Η οθόνη με τίτλο «Ενεργοποίηση υπενθύμισης επόμενης Αιμοδότησης» ενημερώνει το χρήστη για την ημερομηνία μετά την οποία μπορεί να δώσει αίμα. Το σύστημα έχει τη δυνατότητα να ενημερώνει το χρήστη για την εν λόγω ημερομηνία και μέσω μηνύματος, που μπορεί να το λάβει ο χρήστης είτε ως e-mail είτε ως SMS. Σε αυτήν την οθόνη μπορεί να επιλέξει αν θέλει να λαμβάνει την ενημέρωση και σε ποιο κανάλι επικοινωνίας είναι δυνατό να τη λάβει.



Εικόνα 4-7 Ενεργοποίηση υπενθύμισης επόμενης αιμοδότησης

Η οθόνη που ακολουθεί περιλαμβάνει τις αιτήσεις που έχει υποβάλει ο χρήστης προκειμένου να αποκτήσει ταυτότητα εθελοντή αιμοδότη. Ο τίτλος της είναι «Ταυτότητα Εθελοντή Αιμοδότη» και βλέπουμε τον πίνακα που χρησιμοποιείται για τη λίστα των αιμοδοσιών και των καλύψεων να περιέχει τα κύρια στοιχεία κάθε αίτησης. Αν ο χρήστης θέλει να δει παραπάνω στοιχεία μίας αίτησης χρειάζεται να την επιλέξει και μετά να «πατήσει» το κουμπί που εμφανίζεται. Εάν θέλει να δημιουργήσει νέα αίτηση, όταν αυτό επιτρέπεται από το σύστημα, εμφανίζεται το αντίστοιχο κουμπί πάνω από τον πίνακα.



Εικόνα 4-8 Λίστα αιτήσεων για έκδοση ταυτότητας αιμοδότη

Όταν ο χρήστης μεταβεί στην οθόνη που περιλαμβάνει τα λεπτομερή στοιχεία της αίτησης, θα δει στην επάνω πλευρά τον τίτλο που προαναφέραμε και τη δυνατότητα να γυρίσει πίσω στη λίστα με τις αιτήσεις. Στη συνέχεια ακολουθεί η κατάσταση της αίτησης και τα πλήρη

στοιχεία της ταυτότητας. Αφού κυλίσει την οθόνη προς τα κάτω μπορεί να δει το σημείο από όπου μπορεί να παραλάβει την ταυτότητα του όταν είναι έτοιμη.

Εθνικό Μητρώο Αιμοδοτών Ελληνικά / English Έχετε συνδεθεί ως jane_doe (Όροι Χρήσης) Αποσύνδεση

Κεντρική
Διαχείριση Προφίλ
Ιστορικό Αιμοδοτήσεων
Ιστορικό Καλύψεων
Υπενθύμιση επόμενης Αιμοδότησης
Ταυτότητα Εθελοντή Αιμοδότη

Ταυτότητα Εθελοντή Αιμοδότη

← Λίστα

Κατάσταση: Παραδόθηκε

Φωτογραφία*

Η φωτογραφία του κατόχου πρέπει να πληροί τα παρακάτω κριτήρια:

- Οι διαστάσεις πρέπει να είναι τουλάχιστον **240 x 240 px**
- Ο τύπος της φωτογραφίας πρέπει να είναι : **jpg**
- Το μέγεθος της φωτογραφίας δεν πρέπει να ξεπερνά τα **3 MB**
- Η φωτογραφία πρέπει να είναι **πρόσφατη, έγχρωμη, σε ουδέτερο φόντο** και να απεικονίζει ευκρινώς **μόνο το πρόσωπο** του δικαιούχου

Φωτογραφίες στις οποίες ο εικονιζόμενος φορά γυαλιά ήλθου δεν θα γίνονται δεκτές. Στις περιπτώσεις όπου υπάρχουν πέλμα, μαντίλια ή καπέλα θα πρέπει να αφαιρεθούν. Σε περιπτώσεις που αυτά δεν μπορούν να αφαιρεθούν για θρησκευτικούς λόγους, θα πρέπει να εξασφαλιστεί ότι τα καλύμματα δεν κρύβουν οποιαδήποτε χαρακτηριστικά του προσώπου.

Επίσημο: ΕΠΩΝΥΜΟ ΓΗΗΔΕ
Επίσημο (Λατινικοί Χαρακτήρες): ΕΡΩΝΥΜΟ ΓΙΗΙΔΕ
Όνομα: ΟΝΟΜΑ ΓΗΗΔΕ
Όνομα (Λατινικοί Χαρακτήρες): ΟΝΟΜΑ ΓΙΗΙΔΕ
Πατρώνυμο: ΠΑΤΡΩΝΥΜΟ ΓΗΗΔΕ
Πατρώνυμο (Λατινικοί Χαρακτήρες): ΡΑΤΡΩΝΥΜΟ ΓΙΗΙΔΕ
Μητρώνυμο: ΜΗΤΡΩΝΥΜΟ ΓΗΗΔΕ
Μητρώνυμο (Λατινικοί Χαρακτήρες): ΜΙΤΡΩΝΥΜΟ ΓΙΗΙΔΕ
Αριθμός Ταυτότητας ΕΑ: 301602582936
E-mail*: donorCard286634@stg.bdr.grnet.gr
Κινητό Τηλέφωνο*: 6900286634

Ευρωπαϊκή Ένωση Ευρωπαϊκό Ταμείο Περιφερειακής Ανάπτυξης
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ "ΥΓΕΙΑ ΚΑΙ ΑΝΑΠΤΥΞΗ ΑΝΘΡΩΠΙΝΩΝ ΠΟΡΩΝ"
ΕΣΠΑ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Εικόνα 4-9 Ταυτότητα Εθελοντή Αιμοδότη (1 από 3)

Εθνικό Μητρώο Αιμοδοτών Ελληνικά / English Έχετε συνδεθεί ως jane_doe (Όροι Χρήσης) Αποσύνδεση

Κεντρική
Διαχείριση Προφίλ
Ιστορικό Αιμοδοτήσεων
Ιστορικό Καλύψεων
Υπενθύμιση επόμενης Αιμοδότησης
Ταυτότητα Εθελοντή Αιμοδότη

Η φωτογραφία του κατόχου πρέπει να πληροί τα παρακάτω κριτήρια:

Επίσημο: ΕΠΩΝΥΜΟ ΓΗΗΔΕ
Επίσημο (Λατινικοί Χαρακτήρες): ΕΡΩΝΥΜΟ ΓΙΗΙΔΕ
Όνομα: ΟΝΟΜΑ ΓΗΗΔΕ
Όνομα (Λατινικοί Χαρακτήρες): ΟΝΟΜΑ ΓΙΗΙΔΕ
Πατρώνυμο: ΠΑΤΡΩΝΥΜΟ ΓΗΗΔΕ
Πατρώνυμο (Λατινικοί Χαρακτήρες): ΡΑΤΡΩΝΥΜΟ ΓΙΗΙΔΕ
Μητρώνυμο: ΜΗΤΡΩΝΥΜΟ ΓΗΗΔΕ
Μητρώνυμο (Λατινικοί Χαρακτήρες): ΜΙΤΡΩΝΥΜΟ ΓΙΗΙΔΕ
Αριθμός Ταυτότητας ΕΑ: 301602582936
E-mail*: donorCard286634@stg.bdr.grnet.gr
Κινητό Τηλέφωνο*: 6900286634

Ομάδα Αίματος*: O
Rhesus*: (+)

Ημερομηνία Υποβολής: 26/07/2018 17:09
Αίτησας:
 Ο αιμοδότης δήλωσε ότι επιθυμεί την έκδοση Ταυτότητας Εθελοντή Αιμοδότη

Σημείο Παράδοσης Ταυτότητας:
Διεύθυνση : ΕΘΝ.ΑΝΤΙΣΤΑΣΕΩΣ 2, Τ.Κ. 55133
Δήμος : ΚΑΛΑΜΑΡΙΑΣ
Ωράριο Λειτουργίας :
Δευτέρα, Τετάρτη, Σάββατο: 09:00-15:00,
Τρίτη, Πέμπτη, Παρασκευή: 09:00-14:30 / 17:30-21:00
Σταθερό Τηλέφωνο : 2310478415
Email : v321@vshop.gr

Ευρωπαϊκή Ένωση Ευρωπαϊκό Ταμείο Περιφερειακής Ανάπτυξης
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ "ΥΓΕΙΑ ΚΑΙ ΑΝΑΠΤΥΞΗ ΑΝΘΡΩΠΙΝΩΝ ΠΟΡΩΝ"
ΕΣΠΑ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Εικόνα 4-10 Ταυτότητα Εθελοντή Αιμοδότη (2 από 3)

Εάν ο χρήστης υποβάλει νέα αίτηση για ταυτότητα θα δει στην οθόνη δημιουργίας τα στοιχεία της ταυτότητας προσυμπληρωμένα με τα στοιχεία που είναι καταχωρισμένα στο λογαριασμό του. Έπειτα, αυτό που χρειάζεται να κάνει είναι να καταχωρίσει μία φωτογραφία του, με τα χαρακτηριστικά που ζητούνται καθώς και να επιλέξει το σημείο διανομής, από το οποίο θα την παραλάβει.

Εθνικό Μητρώο Αιμοδοτών Ελληνικά / English
 Έχετε συνδεθεί ως **jane_doe** (Όροι Χρήσης) Αποσύνδεση

Ταυτότητα Εθελοντή Αιμοδότη

← Αίτηση → Επιλογή Σημείου Παράδοσης

Φωτογραφία*

Η φωτογραφία του κατόχου πρέπει να πληροί τα παρακάτω κριτήρια:

- Οι διαστάσεις πρέπει να είναι τουλάχιστον **240 x 240 px**
- Ο τύπος της φωτογραφίας πρέπει να είναι: **jpg**
- Το μέγεθος της φωτογραφίας δεν πρέπει να ξεπερνά τα **3 MB**
- Η φωτογραφία πρέπει να είναι **πρόσωπο**, **έγχρωμη**, σε **ουδέτερο** φόντο και να απεικονίζει ευκαρινώς **μόνο το πρόσωπο** του αιμοδότη.

Φωτογραφίες στις οποίες ο αιμοδότης φορά γυαλιά ή/και δεν θα γίνονται δεκτές. Στις περιπτώσεις όπου υπάρχουν πλάι, μαντίλια ή κορίτσια θα πρέπει να αφαιρεθούν. Σε περιπτώσεις που αυτά δεν μαρκούν να αφαιρεθούν για λειτουργικούς λόγους, θα πρέπει να ερμηνευτεί ότι τα καλύμματα δεν κρύβουν οποιαδήποτε χαρακτηριστικά του προσώπου.

Προσθήκη νέας φωτογραφίας

Επίσημο: ΕΠΩΝΥΜΟ ΔΑΔΑΗΓ* ΕΠΩΝΥΜΟ ΔΑΤΗΑΓ* ΟΝΟΜΑ ΔΑΔΑΗΓ* ΟΝΟΜΑ ΔΑΤΗΑΓ*

Πατρώνυμο: ΠΑΤΡΩΝΥΜΟ ΔΑΔΑΗΓ* ΠΑΤΡΩΝΥΜΟ ΔΑΤΗΑΓ* Μητρώνυμο: ΜΗΤΡΩΝΥΜΟ ΔΑΔΑΗΓ* ΜΗΤΡΩΝΥΜΟ ΔΑΤΗΑΓ*

Αριθμός Ταυτότητας ΕΑ: 303000307062 E-mail: blooddonor307062@stg.bdr.grnet.gr Κινητό Τηλέφωνο*: 6900307062

Ομάδα Αιματος*: A Ρhesus*: (i)

Ο αιμοδότης δήλωσε ότι επιθυμεί την έκδοση Ταυτότητας Εθελοντή Αιμοδότη

Σημείο Παράδοσης Ταυτότητας: -

Ευρωπαϊκή Ένωση Ευρωπαϊκό Ταμείο Περιφερειακής Ανάπτυξης Υπουργείο Υγείας ΕΣΠΑ
 Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Εικόνα 4-11 Ταυτότητα Εθελοντή Αιμοδότη (3 από 3)

5 Σχεδιασμός διεπαφής χρήστη

Μετά τη μελέτη που περιγράφεται στο προηγούμενο κεφάλαιο αναζητήθηκαν μεθοδολογίες που χρησιμοποιούνται για τη βελτίωση της εμπειρίας του χρήστη και την εξασφάλιση της ευχρηστίας του συστήματος. Βασισμένοι σε αυτές τις μεθοδολογίες και στα στοιχεία που συλλέξαμε για την προς υλοποίηση εφαρμογή σχεδιάστηκαν προσχέδια της εφαρμογής (wireframes), τα όποια έδωσαν την πρώτη εικόνα για τη μορφή της.

Σε αυτό το κεφάλαιο ορίζονται και αναλύονται οι κύριες κατευθυντήριες που ακολουθήθηκαν για να εξασφαλισθεί η ευχρηστία και η καλή εμπειρία χρήσης. Προκειμένου ο σχεδιασμός της διεπαφής της εφαρμογής να είναι άρτιος χρειάστηκε να μελετηθούν και τεχνικές που την καθιστούν προσβάσιμη από όλους τους εν δυνάμει χρήστες (accessibility standards). Για το λόγο αυτό, στη συνέχεια αυτού του κεφαλαίου παρατίθενται και πληροφορίες που αφορούν την προσβασιμότητα και τη διεθνοποίηση της εφαρμογής. Στο τέλος του κεφαλαίου παρουσιάζονται τα wireframes της εφαρμογής που συνθέτουν τον αρχικό σχεδιασμό της διεπαφής.

5.1 Ευχρηστία συστήματος και εμπειρία χρήστη

Σύμφωνα με το πρότυπο ISO 9241-11 [9], ως ευχρηστία ορίζεται η δυνατότητα ενός προϊόντος, συστήματος ή υπηρεσίας που χρησιμοποιείται από καθορισμένους χρήστες, με καθορισμένους στόχους, υπό καθορισμένες συνθήκες χρήσης, να παρέχει αποτελεσματικότητα, αποδοτικότητα και υποκειμενική ικανοποίηση στους χρήστες του.

Λόγω του παραπάνω ορισμού, διενεργήθηκε η ανάλυση που αναφέρεται στο κεφάλαιο 4 και περιλαμβάνει την περιγραφή των χρηστών, των συνθηκών χρήσης και των κύριων στόχων που θα έχουν κατά τη χρήση της εφαρμογής. Στόχος του συνολικού σχεδιασμού είναι η ανάπτυξη μίας αποτελεσματικής και αποδοτικής εφαρμογής που επιπλέον είναι «φιλική προς τους χρήστες της».

Προκειμένου να επιτύχουμε τους παραπάνω στόχους ανατρέξαμε στις προτεινόμενες πρακτικές για την υλοποίηση εύχρηστων διεπαφών.

5.1.1 Αρχές σχεδιασμού Android εφαρμογών (Android design principles)

Μία πρώτη πηγή κατεύθυνσης και πληροφοριών υπήρξαν οι αρχές που διατίθενται ως προτεινόμενες από το Android [10]. Συγκεκριμένα, το Android έχει ορίσει συγκεκριμένες ομάδες/κατηγορίες αρχών. Ακολουθεί μία σύντομη περιγραφή κάθε κατηγορίας καθώς και οι κύριες αρχές που περιλαμβάνει.

Κατηγορία: ευχαρίστηση χρήστη

Η πρώτη κατηγορία σχετίζεται με την ευχαρίστηση που προσφέρει η χρήση μίας εφαρμογής στους χρήστες της. Περιγράφονται τέσσερις τρόποι που μπορούν να αυξήσουν το συναίσθημα αυτό:

Συγκεκριμένα αναφέρεται ότι μία όμορφη επιφάνεια ή μία κίνηση (animation) ή ακόμα και ένας ήχος, αν έχει τοποθετηθεί προσεκτικά και διακριτικά, μπορεί να δώσει μία ευχάριστη αίσθηση χρήσης.

Επιπλέον, προτείνεται να διατίθεται στους χρήστες η δυνατότητα αλληλεπίδρασης με εικονικά αντικείμενα αντί να αλληλεπιδρούν με κουμπιά και μενού. Με αυτό τον τρόπο

μειώνεται η νοητική προσπάθεια που απαιτείται για την ολοκλήρωση της ενέργειας που επιθυμεί ο χρήστης και αυξάνεται το αίσθημα ικανοποίησης.

Μία άλλη πρόταση είναι να δίνεται η δυνατότητα τροποποίησης της διεπαφής σύμφωνα με τις αισθητικές προτιμήσεις που έχει ο κάθε χρήστης. Έτσι, μπορεί κάθε χρήστης να τροποποιήσει την εφαρμογή ώστε να την αισθάνεται πιο οικεία. Αυτό βέβαια δε σημαίνει ότι θα πρέπει να αμεληθεί η αρχική εμφάνιση της διεπαφής. Σημαίνει όμως, ότι θα πρέπει να δοθεί προσοχή ώστε να μην εμποδιστούν οι κύριες λειτουργίες της εφαρμογής, ανεξαρτήτως εμφάνισης.

Η τελευταία κατευθυντήρια αυτής της κατηγορίας αφορά τη «μνήμη» της εφαρμογής. Προτείνεται η εφαρμογή να μην ζητάει από το χρήστη να καταχωρίσει τις ίδιες επιλογές πολλές φορές και όσες επιλογές είναι προφανείς, να είναι και εύκολα επιλέξιμες.

Η επόμενη κατηγορία περιγράφει τακτικές που κάνουν πιο εύκολη τη χρήση μίας εφαρμογής.

Κατηγορία: εύκολη χρήση εφαρμογής

Αρχικά, είναι καλό να χρησιμοποιούνται μικρές προτάσεις με απλές λέξεις και όπου είναι δυνατό να χρησιμοποιούνται εικόνες για τη μετάδοση πληροφοριών. Μία εικόνα μπορεί να τραβήξει πιο εύκολα την προσοχή του χρήστη και να του μεταδώσει πιο αποδοτικά πληροφορίες συγκριτικά με ένα κείμενο.

Έπειτα προτείνεται η εφαρμογή να παίρνει αποφάσεις και να πραγματοποιεί ενέργειες αντί του χρήστη, χωρίς να τον ρωτήσει. Όμως, απαραίτητως θα πρέπει να δίνει τη δυνατότητα στο χρήστη να τις αναίρεσει εύκολα.

Στη συνέχεια γίνεται αναφορά στο πλήθος πληροφοριών που εμφανίζονται σε μία εφαρμογή. Τονίζεται ότι πρέπει να αποφεύγεται ο κατακλυσμός του χρήστη με πληροφορίες. Οι πληροφορίες θα πρέπει να εμφανίζονται κατά τμήματα στο χρήστη. Λεπτομέρειες και περιττές πληροφορίες είναι καλό να αποφεύγονται και η εκπαίδευση των χρηστών στη λειτουργία της εφαρμογής πρέπει να πραγματοποιείται σταδιακά.

Σε κάθε οθόνη ή στάδιο λειτουργίας είναι πολύ σημαντικό ο χρήστης να ξέρει πού βρίσκεται και να αντιλαμβάνεται πως μπορεί να διαχειριστεί τη διεπαφή ανά πάσα στιγμή. Για αυτό είναι καλό να υπάρχουν σαφώς διαχωρισμένες περιοχές σε μία διεπαφή και προφανείς συνδέσεις μεταξύ των διαθέσιμων οθονών.

Μία άλλη πολύ σημαντική αρχή αναφέρεται στη σχέση εμφάνισης και λειτουργικότητας. Οι χρήστες θα πρέπει να μπορούν εύκολα να διαχωρίσουν από τον τρόπο εμφάνισης στοιχεία της διεπαφής που συμβάλλουν σε διαφορετικές λειτουργίες. Παρόμοια στοιχεία θα πρέπει να αφορούν παρόμοιες λειτουργίες.

Τέλος, προκειμένου ο χρήστης να έχει την αίσθηση της απλότητας κατά τη χρήση της εφαρμογής προτείνεται ο όγκος ειδοποιήσεων που τον διακόπτουν να περιορίζεται στο ελάχιστο.

Κατηγορία: τρόπος σχεδίασης εφαρμογής με στόχο την ικανοποίηση των χρηστών

Οι άνθρωποι αισθάνονται ικανοποίηση και χαρά όταν ανακαλύπτουν μόνοι τους τι συμβαίνει στον κόσμο γύρω τους. Για αυτό προτείνεται ο σχεδιασμός της διεπαφής να αξιοποιεί ίδια οπτικά μοτίβα που χρησιμοποιούνται από άλλες Android εφαρμογές. Με αυτόν τον τρόπο οι χρήστες διευκολύνονται σημαντικά κατά τη χρήση, καθώς χρησιμοποιούν τη μυϊκή μνήμη που έχουν αποκτήσει από τη χρήση όλων αυτών των εφαρμογών.

Ένα άλλο σημείο που καλούνται οι σχεδιαστές να δώσουν προσοχή είναι ο τρόπος που ενημερώνονται οι χρήστες για λάθη που έχουν πραγματοποιήσει. Σε καμία περίπτωση δεν θα πρέπει οι χρήστες να αισθανθούν ανόητοι ή ανεπαρκείς για τη χρήση της εφαρμογής. Αν κάτι πάει στραβά, η εφαρμογή με ευγενικό και κατανοητό τρόπο μπορεί να περιγράψει επαρκώς το σφάλμα που προέκυψε και τον τρόπο που μπορεί ο χρήστης να την επαναφέρει στην ορθή λειτουργία της. Καλύτερα ακόμα, εάν η εφαρμογή μπορεί να διορθώσει το σφάλμα χωρίς να το αντιληφθεί ο χρήστης.

Επίσης, είναι καλό οι περίπλοκες διαδικασίες να έχουν διαμοιραστεί σε πιο απλά και μικρά βήματα.

Ένα άλλο στοιχείο που παίζει σημαντικό ρόλο και στη σχεδίαση της διεπαφής αλλά και στην υλοποίηση της εφαρμογής αφορά την ταχύτητα των ενεργειών. Δεν χρειάζεται όλες οι ενέργειες ή διαδικασίες να εκτελούνται πολύ γρήγορα αλλά είναι καθοριστικής σημασίας οι σημαντικές λειτουργίες να μπορούν να ολοκληρωθούν στο λιγότερο δυνατό χρόνο.

5.1.2 React Native αρχές

Οι παραπάνω αρχές αφορούν την ευρεία συμπεριφορά της εφαρμογής και τη λογική που είναι επιθυμητό να έχει ο σχεδιασμός της. Πέρα από αυτές μελετήθηκαν και προτάσεις για τα επιμέρους στοιχεία που αποτελούν μία διεπαφή. Το React Native, που όπως περιγράφεται στο κεφ. 7 χρησιμοποιήθηκε για την υλοποίηση της εφαρμογής, δίνει ορισμένες τέτοιες συμβουλές [11], όπως:

Κουμπιά

Κουμπιά είναι τα στοιχεία που καλείται ο χρήστης να «πιέσει» προκειμένου να εκτελέσει μία ενέργεια. Είναι καλό να είναι μεγάλα σε μέγεθος και τοποθετημένα σε απόσταση από τα άλλα στοιχεία της διεπαφής ώστε να είναι εύκολος ο εντοπισμός τους και η αλληλεπίδραση μαζί τους. Επιπλέον, προτείνεται να εμφανίζεται ένας κυματισμός όταν ο χρήστης «πατάει» κάποιο κουμπί, όπως συνηθίζεται και στις άλλες Android εφαρμογές.

Στοιχεία εισαγωγής κειμένου

Αν υπάρχει στοιχείο εισαγωγής κειμένου στην οθόνη που βλέπει ο χρήστης, το οποίο είναι πολύ πιθανό ο χρήστης να το χρησιμοποιήσει, είναι καλό να είναι ο κέρσορας πάνω του εξ' αρχής. Έτσι, ο χρήστης θα μπορεί να πληκτρολογήσει χωρίς να προηγηθεί κάποια κίνησή του. Επίσης, διευκολύνει τους χρήστες να υπάρχουν παραδείγματα ορθής συμπλήρωσης των στοιχείων εισαγωγής κειμένου, όπως επίσης να εμφανίζεται το κατάλληλο είδος πληκτρολογίου ανάλογα με το είδος του κειμένου που καλείται ο χρήστης να συμπληρώσει (αριθμητικό, πληκτρολόγιο για e-mail κ.λπ.). Μία λειτουργία που ενισχύει την ευχρηστία είναι το πλήκτρο «enter» να οδηγεί τον κέρσορα στο επόμενο στοιχείο εισαγωγής κειμένου ή να αποθηκεύει τη φόρμα.

5.1.3 Συνοπτική αναφορά αρχών που εφαρμόστηκαν

Συνοψίζοντας, οι κύριες αρχές που ακολουθήθηκαν μπορούν να περιγραφούν με την παρακάτω λίστα. Η λίστα περιλαμβάνει, χωρίς να εξαντλεί, τις επιλογές που ακολουθήθηκαν για να επιτευχθεί εύκολη και ευχάριστη χρήση της εφαρμογής:

- Προσεκτικός επιμερισμός των πληροφοριών.
- Χρήση οπτικών μοτίβων.

- Μεγαλύτερο μέγεθος σε στοιχεία που ο χρήστης χρειάζεται να αλληλοεπιδράσει όπως είναι τα κουμπιά, τα στοιχεία εισαγωγής κειμένου και οι υπερσυνδέσμοι. Επιπλέον, τους έχει δοθεί επαρκής απόσταση από τα άλλα στοιχεία της διεπαφής για να είναι εύκολος ο εντοπισμός και η χρήση τους.
- Τίτλο σε όλες τις οθόνες ώστε να ξέρει πάντα ο χρήστης που βρίσκεται.
- Απλό, σαφή και διαδομένο τρόπο πλοήγησης στην εφαρμογή.
- Απλό, καθημερινό και φιλικό λεξιλόγιο.
- Υιοθέτηση χρωματικού κώδικα που συμβαδίζει με τα επιστημονικά στοιχεία για την ανάλυση των χρωμάτων από τον ανθρώπινο εγκέφαλο. Για παράδειγμα, χρησιμοποιείται το πράσινο για θετικά αποτελέσματα ή ενέργειες.
- Συνεπής χρωματικός κώδικας σε όλες τις οθόνες της εφαρμογής.
- Τα στοιχεία που είναι πιο πιθανό να χρησιμοποιήσει ο χρήστης έχουν τοποθετηθεί έτσι ώστε να μπορεί να τα εντοπίσει εύκολα και γρήγορα.
- Περιγραφικά μηνύματα λάθους που επεξηγούν με απλά λόγια στο χρήστη τι συνέβη και πως μπορεί να προχωρήσει μετά το σφάλμα που προέκυψε.
- Χρήση εικονιδίων, σε όσα σημεία κρίθηκε χρήσιμο, συνοδευόμενα από κείμενο για πιο εύκολη αναγνώριση και εξοικείωση του χρήστη με το περιεχόμενο της εφαρμογής.

5.2 Προσβασιμότητα (accessibility)

Το άλλο σημείο, για το οποίο διενεργήθηκε εκτενής μελέτη, αφορά την προσβασιμότητα της εφαρμογής. Συγκεκριμένα, η παρούσα εφαρμογή αφορά το σύνολο των πολιτών που επιθυμεί και είναι σε θέση να προσφέρει στον συνάνθρωπό του αίμα. Συνεπώς, έχει μεγάλη σημασία η εφαρμογή αυτή να είναι προσβάσιμη από όλους τους ανθρώπους, ακόμα και αν οι ίδιοι αντιμετωπίζουν κάποια δυσκολία στην όραση ή στην κίνηση τους. Είναι καθήκον μας να εξασφαλίσουμε ότι η διαδικασία της αιμοδοσίας θα είναι το ίδιο εύκολη για όλους τους ανθρώπους, με όποιον τρόπο μπορούμε.

Για την ανάπτυξη της εφαρμογής αιμοδοτών χρησιμοποιήθηκαν ως οδηγός, κατά κύριο λόγο, οι πρακτικές που προτείνονται από το W3C για δικτυακές εφαρμογές που χρησιμοποιούνται μέσω κινητής συσκευής (Mobile Web Practices 1.0 [12]).

Οι πρακτικές που περιγράφονται στο Mobile Web Practices 1.0 [12] ως βέλτιστες είναι οι παρακάτω:

1. Θεματική συνέπεια

2. Δυνατότητες συσκευών

Αξιοποιήστε τις δυνατότητες των συσκευών ώστε να βελτιώσετε την εμπειρία του χρήστη.

3. Ελλείψεις

Ακολουθήστε λογικά βήματα ώστε να αποφύγετε ελλειψείς ή ελλαττωματικές υλοποιήσεις.

4. Testing

Δοκιμάστε και ελέγξτε τις εφαρμογές σας σε προσομοιωτές αλλά και σε πραγματικές συσκευές.

5. URIs

Διατηρήστε τα URIs μικρά σε μήκος.

6. NavBar

Χρησιμοποιήστε λίγες και βασικές επιλογές πλοήγησης στην επάνω πλευρά της οθόνης και φροντίστε ώστε να μη γεμίζουν την οθόνη και ο χρήστης χρειάζεται να κάνει κύλιση της οθόνης για να δει περιεχόμενο.

7. Ισορροπία

Ισορροπήστε τον αριθμό των υπερσυνδέσμων που τοποθετείτε σε μία οθόνη με τον αριθμό των υπερσυνδέσμων που χρειάζεται ο χρήστης να επιλέξει προκειμένου να ολοκληρώσει το στόχο του.

8. Πλοήγηση

Προσφέρετε στο χρήστη συνεπείς μηχανισμούς πλοήγησης.

9. Πλήκτρα πρόσβασης

Χρησιμοποιήστε πλήκτρα πρόσβασης για επιλογές πλοήγησης και λειτουργικότητα που χρησιμοποιείται συχνά από τους χρήστες.

10. Ταυτοποίηση προορισμού υπερσυνδέσμου

Ταυτοποιήστε ξεκάθαρα τον προορισμό κάθε υπερσυνδέσμου που χρησιμοποιείτε.

11. Ταυτοποίηση μορφής υπερσυνδέσμου

Λόγω του ότι στις κινητές συσκευές επηρεάζεται η εμπειρία χρήσης σε μεγάλο βαθμό από το μέγεθος των δεδομένων όπως και τη μορφή τους, προτείνεται να είναι ξεκάθαρο το είδος και μέγεθος τους πριν από τη λήψη τους.

12. Χάρτες εικόνων

Μην χρησιμοποιείτε χάρτες εικόνων, εάν δεν εξασφαλίσετε ότι η αποδοτική χρήση τους είναι εφικτή από όλες τις ζητούμενες συσκευές.

13. Αναδυόμενα παράθυρα

Αποφύγετε την ανάδυση παραθύρων ή την αλλαγή της οθόνης που βλέπει ο χρήστης χωρίς να τον έχετε ενημερώσει.

14. Αυτόματη ανανέωση

Αποφύγετε την αυτόματη, περιοδική ανανέωση των δεδομένων και των οθονών, αν δεν έχετε ενημερώσει το χρήστη. Επίσης, είναι απαραίτητο να δίνετε στο χρήστη δυνατότητα να σταματήσει την ανανέωση, εφ' όσον το επιθυμεί.

15. Ανακατεύθυνση

Επιλέξτε οι ανακατευθύνσεις να γίνονται αυτόματα από το διακομιστή και όχι από την εφαρμογή που τρέχει στη συσκευή του χρήστη ή αφού αυτή έχει φορτώσει δεδομένα.

16. Εξωτερικοί πόροι

Χρησιμοποιήστε, όσο το δυνατόν λιγότερους υπερσυνδέσμους σε εξωτερικούς πόρους.

17. Καταλληλότητα

Φροντίστε ώστε το περιεχόμενο να είναι κατάλληλο για κινητές συσκευές.

18. Σαφήνεια

Χρησιμοποιήστε καθαρή και απλή γλώσσα.

19. Περιορισμένο περιεχόμενο

Περιορίστε το περιεχόμενο, ώστε να δώσετε στο χρήστη μόνο ότι έχει ζητήσει.

20. Χρηστικό μέγεθος οθόνης

Χωρίστε τις οθόνες σε χρηστικά αλλά και περιορισμένου μεγέθους κομμάτια.

21. Περιορισμένο μέγεθος οθόνης

Εξασφαλίστε ότι το συνολικό μέγεθος κάθε οθόνης δεν υπερβαίνει περιορισμούς μνήμης που μπορεί να έχουν οι επιθυμητές οθόνες των συσκευών.

22. Κύλιση

Φροντίστε η κύλιση να είναι δυνατή σε μία κατεύθυνση. Δυνατότητα κύλισης και οριζόντια και κάθετα να είναι δυνατή, μόνο όπου δεν είναι δυνατό να αποφευχθεί.

23. Κεντρικό νόημα

Φροντίστε το περιεχόμενο που σχετίζεται με το κύριο νόημα να προηγείται του δευτερεύοντος περιεχομένου.

24. Γραφικά για αποστάσεις και τοποθέτηση στοιχείων

Αποφύγετε να χρησιμοποιήσετε τεχνάσματα που χρησιμοποιούν γραφικά στοιχεία για την τοποθέτηση στοιχείων.

25. Μεγάλα γραφικά στοιχεία

Μην χρησιμοποιείτε εικόνες που δεν μπορούν να προβληθούν στις ζητούμενες οθόνες. Επίσης, χρησιμοποιήστε εικόνες υψηλής ανάλυσης ή μεγάλου μεγέθους μόνο εάν είναι απολύτως απαραίτητο

26. Χρήση χρωμάτων

Φροντίστε πληροφορίες που διατίθενται με χρήση χρωμάτων να είναι διαθέσιμες και χωρίς χρώματα.

27. Αντίθεση χρωμάτων

Φροντίστε τα χρώματα του φόντου και των στοιχείων που βρίσκονται πάνω από αυτά να έχουν επαρκή αντίθεση.

28. Αναγνώσιμο περιεχόμενο πάνω από φόντο με εικόνα

Όταν χρησιμοποιείτε εικόνες για φόντο, προσέξτε να είναι ευανάγνωστο το περιεχόμενο που τοποθετείτε από πάνω του.

29. Τίτλος οθόνης

Επιλέξτε ένα σύντομο και αντιπροσωπευτικό τίτλο για κάθε οθόνη.

30. Frames

Μην χρησιμοποιείτε frames (iframes κ.λπ.) καθώς υπάρχουν συσκευές που δεν τα υποστηρίζουν.

- 31. Δομή**
Αξιοποιήστε δομικά στοιχεία στον κώδικά σας που εκφράζουν τη λογική δομή της πληροφορίας.
- 32. Πίνακες**
Αποφύγετε τους πίνακες γιατί σε συσκευές με περιορισμένο μέγεθος οθόνης ο χρήστης δυσκολεύεται κατά τη χρήση τους.
- 33. Εμφωλευμένοι πίνακες**
Μην χρησιμοποιείται εμφωλευμένους πίνακες.
- 34. Διάταξη οθόνης με πίνακες**
Μην χρησιμοποιείτε πίνακες για να φτιάξετε τη διάταξη των στοιχείων που περιλαμβάνονται σε μία οθόνη.
- 35. Εναλλακτικές πινάκων**
Όπου είναι δυνατό, να μην χρησιμοποιείτε πίνακες αλλά ακολουθείστε άλλες λύσεις, όπως είναι η τοποθέτηση στοιχείων σε καρτέλες.
- 36. Εναλλακτικές των μη λεκτικών στοιχείων**
Να παράσχετε λεκτικές εναλλακτικές για κάθε πληροφορία που παρουσιάζεται χωρίς τη χρήση κειμένου.
- 37. Αντικείμενα ή κώδικας**
Μη βασίζεστε σε αντικείμενα ή κώδικα.
- 38. Καθορισμένο μέγεθος εικόνων**
Προσδιορίστε το μέγεθος των εικόνων στον κώδικά σας.
- 39. Αλλαγή μεγέθους εικόνων**
Προτιμήστε να αλλάζετε το μέγεθος των εικόνων στο διακομιστή.
- 40. Έγκυρη γλώσσα σήμανσης**
Όταν χρησιμοποιείτε HTML έγγραφα να ελέγχετε ότι είναι έγκυρα σύμφωνα με τις επίσημες γραμματικές.
- 41. Μονάδες μέτρησης**
Μην χρησιμοποιείτε ως μονάδα μέτρησης το pixel, ούτε απόλυτες τιμές.
- 42. Χρήση stylesheet**
Να διαμορφώνετε τη διάταξη των οθονών με χρήση stylesheets, όπου είναι εφικτό.
- 43. Υποστήριξη stylesheet**
Φροντίστε οι οθόνες σας να είναι διαμορφωμένες έτσι ώστε να είναι κατανοητές ακόμη και αν δεν εμφανίζονται οι κανόνες που ορίζονται στα stylesheets.
- 44. Μέγεθος stylesheets**
Διατηρήστε το μέγεθος των stylesheets περιορισμένο.
- 45. Ελαχιστοποίηση**
Χρησιμοποιήστε λιτή και αποδοτική γλώσσα σήμανσης.

- 46. Υποστήριξη της μορφής του περιεχομένου**
Στείλτε περιεχόμενο σε μορφή που γνωρίζετε ότι υποστηρίζεται από τις κινητές συσκευές.
- 47. Προτιμώμενη μορφή περιεχομένου**
Όπου είναι δυνατό, στείλτε το περιεχόμενο στην προτιμώμενη μορφή.
- 48. Υποστήριξη κωδικοποίησης χαρακτήρων**
Φροντίστε οι χαρακτήρες του περιεχομένου να είναι κωδικοποιημένοι, σε μορφή που υποστηρίζεται από τις ζητούμενες συσκευές.
- 49. Χρήση κωδικοποίησης χαρακτήρων**
Να προσδιορίζετε στις απαντήσεις που στέλνονται από τους διακομιστές η κωδικοποίηση που χρησιμοποιείται.
- 50. Μηνύματα σφάλματος**
Να παρέχετε ενημερωτικά μηνύματα σφαλμάτων και τρόπους ώστε ο χρήστης να μπορεί να μεταφερθεί από το σημείο που προέκυψε το σφάλμα σε σημείο χρήσιμο για αυτόν.
- 51. Cookies**
Μην βασίζετε τη λειτουργία της εφαρμογής στα cookies, καθώς μπορεί να μην είναι ενεργοποιημένη λειτουργία στην κινητή συσκευή.
- 52. Προσωρινή μνήμη**
Να περιλαμβάνετε στοιχεία σχετικά με τον μηχανισμό προσωρινής μνήμης που χρησιμοποιείτε στα http responses.
- 53. Γραμματοσειρές**
Μη βασίζεστε στη μορφή που παρουσιάζονται οι γραμματοσειρές που χρησιμοποιείτε. Είναι πιθανό να υπάρχουν διαφορές από συσκευή σε συσκευή.
- 54. Ελαχιστοποιήστε την πληκτρολόγηση**
- 55. Αποφύγετε την δυνατότητα εισαγωγής ελεύθερου κειμένου**
Όπου είναι δυνατό, αποφύγετε την προσθήκη ελεύθερου κειμένου από τους χρήστες.
- 56. Χρησιμοποιήστε προεπιλεγμένες τιμές στα πεδία εισαγωγής κειμένου**
Όπου είναι εφικτό, χρησιμοποιήστε προεπιλεγμένες τιμές.
- 57. Default input mode**
Προσδιορίστε το είδος του κειμένου που ζητείται από το χρήστη να εισάγει ή/και τη γλώσσα που αναμένεται να χρησιμοποιήσει, εάν οι ζητούμενες συσκευές είναι γνωστό ότι υποστηρίζουν τέτοιου είδους λειτουργίες.
- 58. Σειρά με την οποία εστιάζονται τα στοιχεία με χρήση του πλήκτρου tab**
Τοποθετήστε με λογική σειρά τους υπερσυνδέσμους, τα επιμέρους στοιχεία των φορμών και τα αντικείμενα.
- 59. Τίτλοι σε στοιχεία φόρμας**
Τοποθετήστε τίτλους σε όλα τα στοιχεία κάθε φόρμας και συνδέστε ξεκάθαρα το κάθε στοιχείο με συγκεκριμένο τίτλο.

60. Έλεγχος τοποθέτησης

Τοποθετήστε τις ετικέτες με τέτοιο τρόπο ώστε να είναι σαφές σε ποιο στοιχείο αναφέρονται.

Στόχος του εγγράφου που περιλαμβάνει τις παραπάνω πρακτικές είναι να δοθούν κατευθυντήριες γραμμές για την ανάπτυξη εφαρμογών που δίνουν στους χρήστες κινητών συσκευών πρόσβαση σε περιεχόμενο που ανακτάται μέσω διαδικτύου. Για αυτόν το λόγο ορισμένες προτάσεις, όπως οι υπ' αριθμόν 37 ή 40, απευθύνονται σε δημιουργούς, συντηρητές και διαχειριστές ιστοσελίδων.

5.3 Διεθνοποίηση και προσαρμογή λογισμικού σε τοπικές ιδιομορφίες

Η διεθνοποίηση ή όπως αναφέρεται ευρέως στον κλάδο της πληροφορικής «internationalization», χρησιμοποιείται για να εκφράσει την σχεδιαστική απόφαση ένα λογισμικό να υποστηρίζει ή να μπορεί με ευκολία να υποστηρίξει τις ανάγκες των χρηστών που ομιλούν διαφορετικές γλώσσες και που έχουν διαφορετική πολιτιστική προέλευση. Λαμβάνοντας την απόφαση ένα λογισμικό να υποστηρίξει τέτοιου είδους διαφορές μεταξύ των χρηστών του, προάγει και την επικοινωνία μεταξύ των ανθρώπων σε παγκόσμια κλίμακα. [13], [14], [15]

Ο όρος «localization» αναφέρεται στην ουσιαστική προσαρμογή ενός λογισμικού στην γλώσσα, τις αντιλήψεις, τη νοοτροπία και τις συνθήκες οι οποίες αναπτύσσονται στις επιμέρους τοποθεσίες των χρηστών. Τέτοιες προσαρμογές μπορούν να πραγματοποιηθούν στην χρησιμοποιούμενη γλώσσα ή στο νόμισμα, ακόμα και στο είδος μονάδων μέτρησης που παρουσιάζονται στο χρήστη. [14], [15]

Στα πλαίσια της εφαρμογής που αναπτύχθηκε, εφαρμόστηκαν κυρίως τακτικές για τη διεθνοποίηση της. Δημιουργήθηκε μηχανισμός ώστε να μπορεί η εφαρμογή να υποστηρίξει την αλλαγή και την προσαρμογή της γλώσσας των κειμένων, στη γλώσσα του χρήστη. Αυτό επιτυγχάνεται με εντοπισμό της γλώσσας του κινητού. Μετά τον εντοπισμό η εφαρμογή ελέγχει αν έχει τα λεκτικά στη συγκεκριμένη γλώσσα. Αν τα έχει, τα παρουσιάζει σε αυτή τη γλώσσα. Αν δεν υπάρχουν σε αυτήν τη γλώσσα, έχει οριστεί μία «fallback» γλώσσα και τα παρουσιάζει σε αυτή.

Για την παρουσίαση του μηχανισμού χρησιμοποιήθηκαν οι γλώσσες:

- Ελληνικά
- Αγγλικά

Η αγγλική γλώσσα ορίστηκε και ως «fallback» γλώσσα.

Για την υλοποίηση του μηχανισμού χρησιμοποιήθηκε η open source βιβλιοθήκη «react-native-i18n».

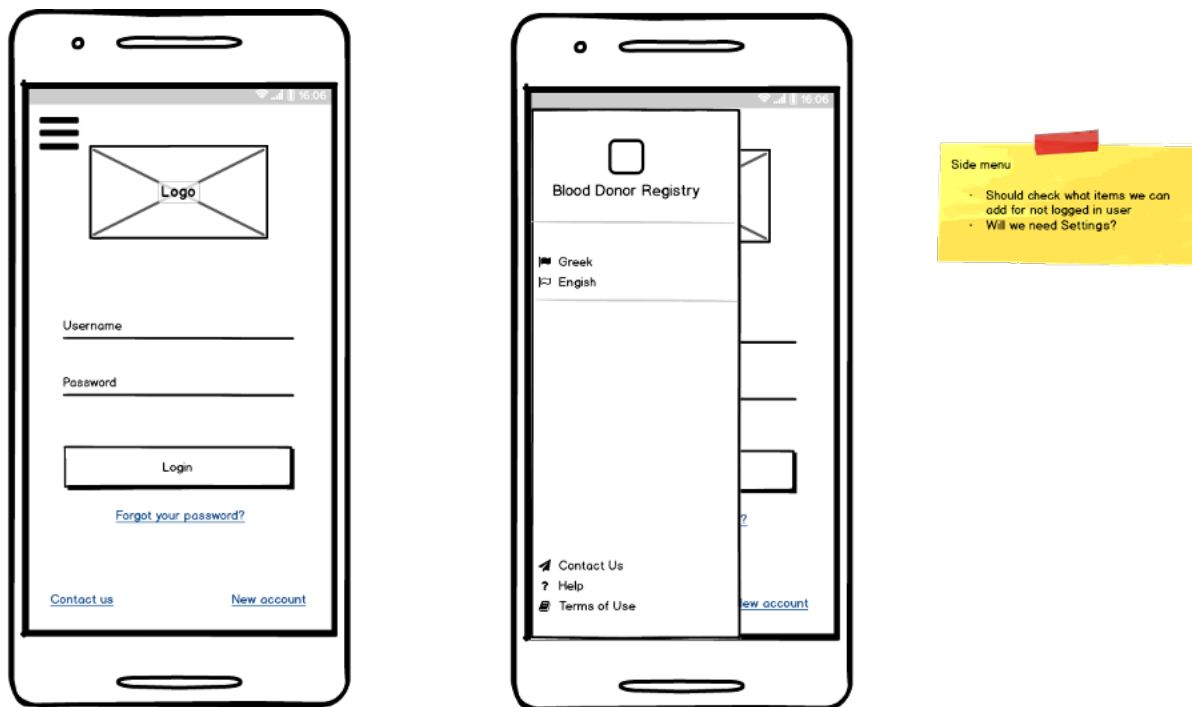
5.4 Wireframes – Προσχέδια εφαρμογής αιμοδότη

Με βάση όλα όσα έχουν αναφερθεί στα προηγούμενα κεφάλαια δημιουργήθηκαν τα παρακάτω προσχέδια/wireframes. Αυτά τα σχέδια περιέχουν πληροφορίες για τη διάταξη των στοιχείων που περιέχονται σε κάθε οθόνη. Επιπλέον, παρουσιάζουν τις επιλογές και τον τρόπο που τις διαθέτουν στο χρήστη. Τα σχέδια αυτά δεν πρέπει να επιμείνουν στη λεπτομέρεια παρουσίασης των στοιχείων ή στην εμφάνιση των οθονών όσο στα χαρακτηριστικά που θα

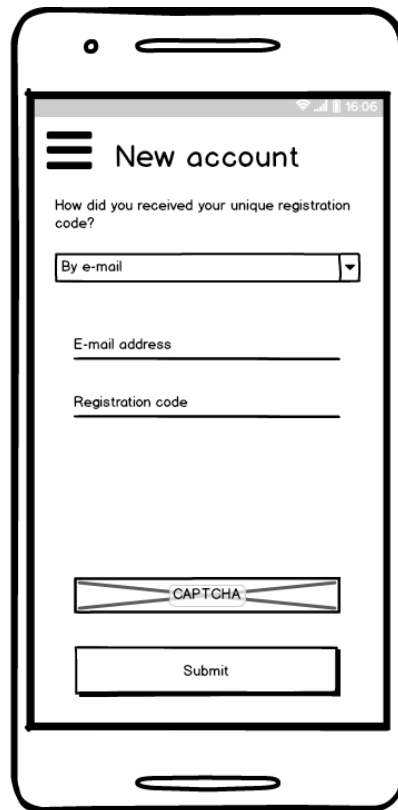
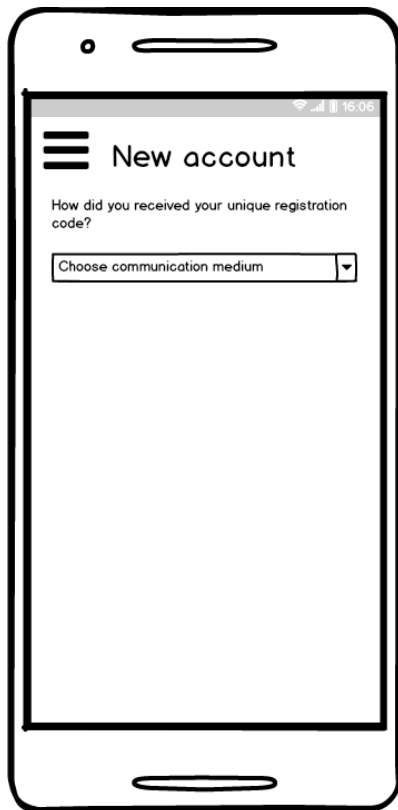
έχουν αυτά τα στοιχεία και οι οθόνες ώστε να διευκολύνουν το χρήστη να επιτύχει το στόχο του κατά τη χρήση της εφαρμογής.

Για την ανάπτυξη της εφαρμογής αιμοδότη αρχικά σχεδιάστηκαν wireframes, με χρήση του λογισμικού Balsamiq, τα οποία υπήρξαν οδηγός για την μετέπειτα υλοποίηση. Να σημειωθεί όμως ότι κατά τη διάρκεια της υλοποίησης υπήρξαν στάδια επανεξέτασης των διεπαφών με αποτέλεσμα σε ορισμένες περιπτώσεις να επιλεγθούν διαφορετικές σχεδιαστικές επιλογές για τις διεπαφές.

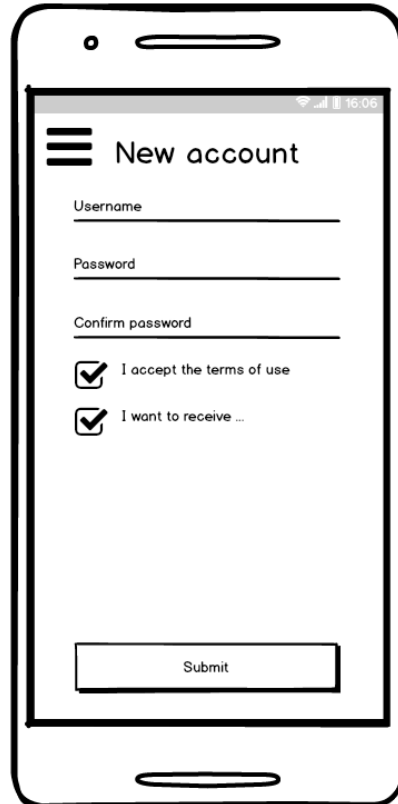
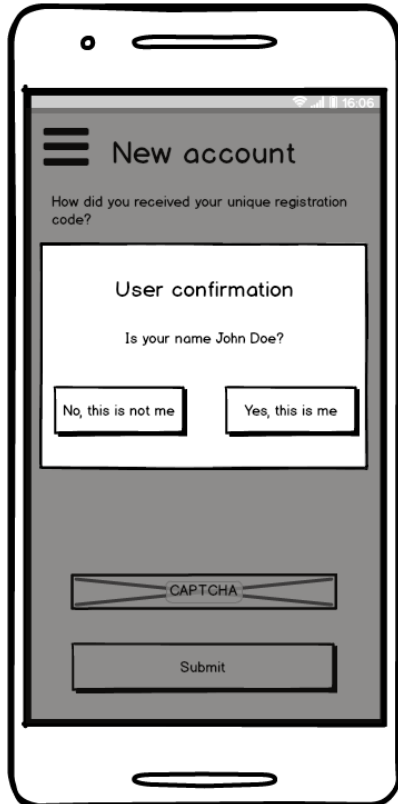
Παρακάτω ακολουθούν τα wireframes που σχεδιάστηκαν στο πρώτο στάδιο σχεδιασμού της διεπαφής της εφαρμογής.



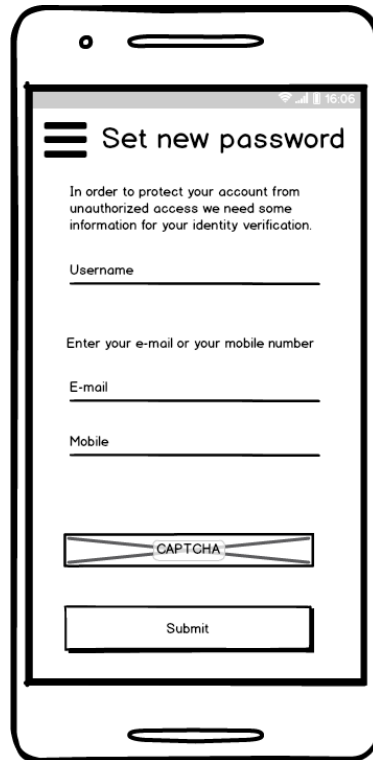
Εικόνα 5-1 Οθόνη σύνδεσης και μενού πλοήγησης μη συνδεδεμένου χρήστη



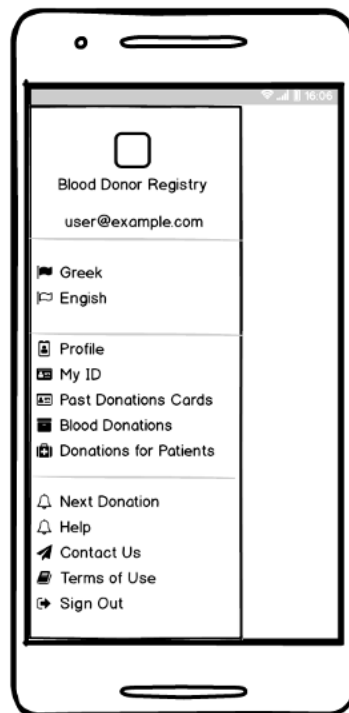
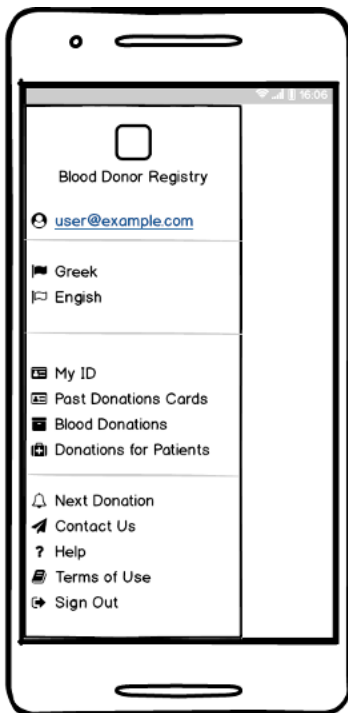
Validation for registration code length



Εικόνα 5-2 Δημιουργία λογαριασμού



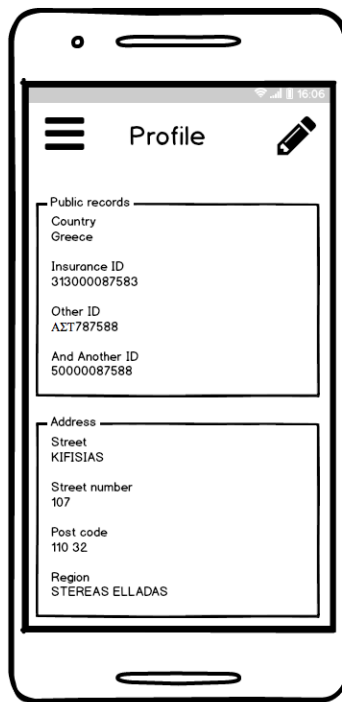
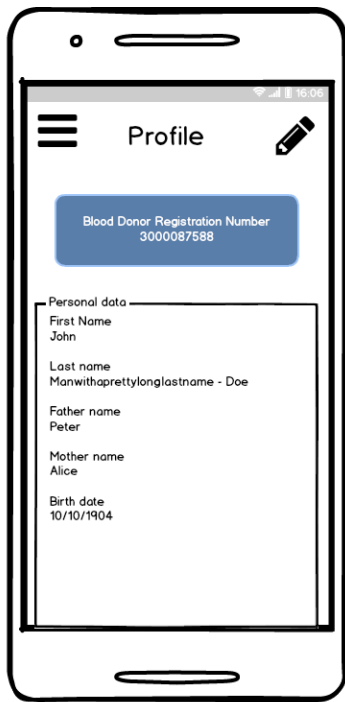
Εικόνα 5-3 Υπενθύμιση κωδικού πρόσβασης



Navigation menu

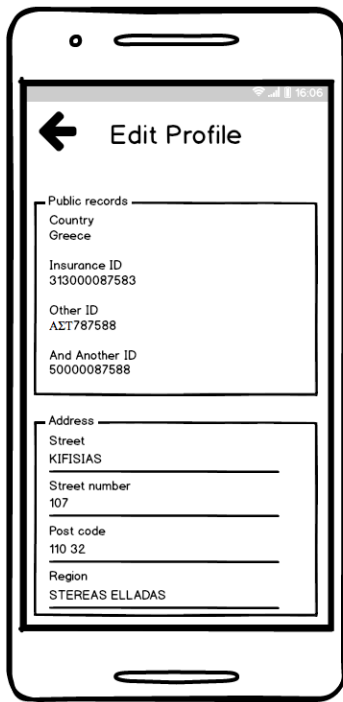
- Proportions between list item and total height and width are not realistic
- Go to profile by e-mail or by item profile
- Should check if we are ok with the items (Do we need Settings or ...)

Εικόνα 5-4 Εναλλακτικές προτάσεις μενού πλοήγησης συνδεδεμένου χρήστη

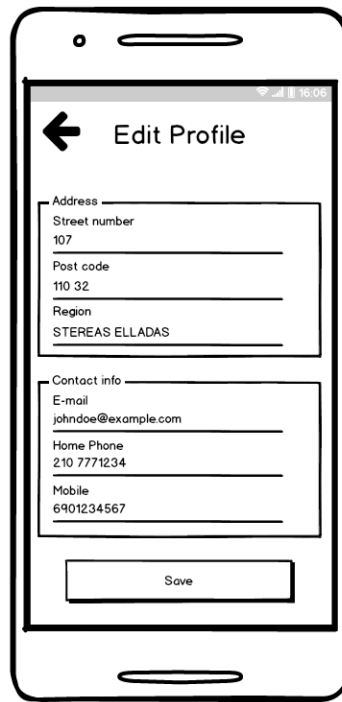


- Different view for details and edit
- Title: in the same height with the icon and aligned left

After scroll

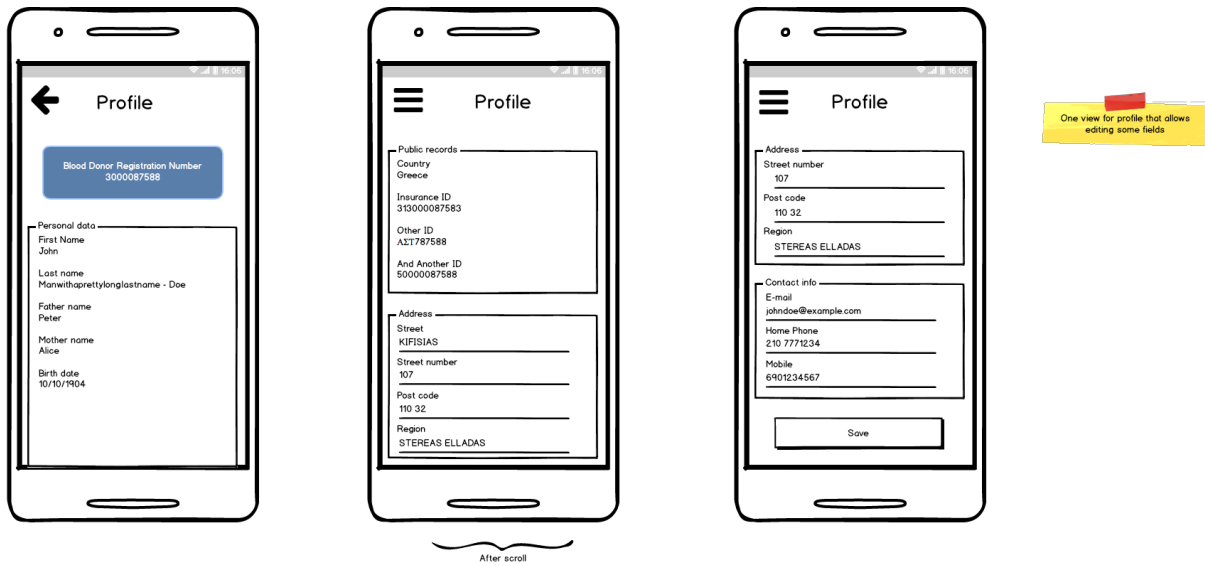


Edit mode, after scroll

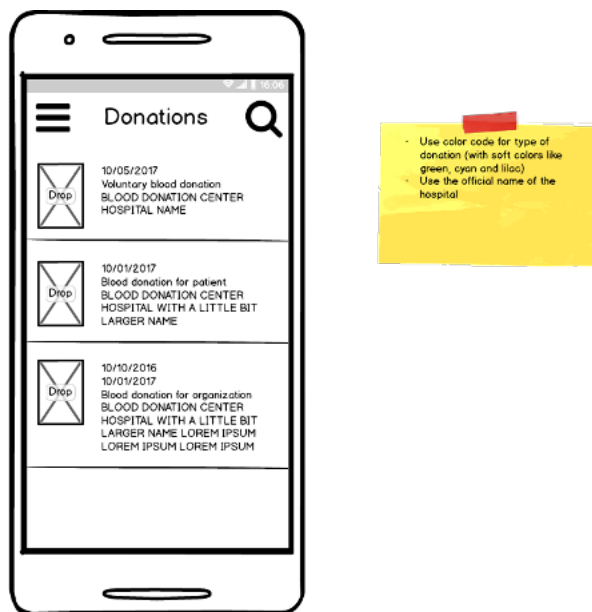


After scroll

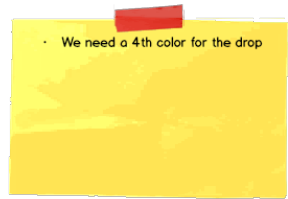
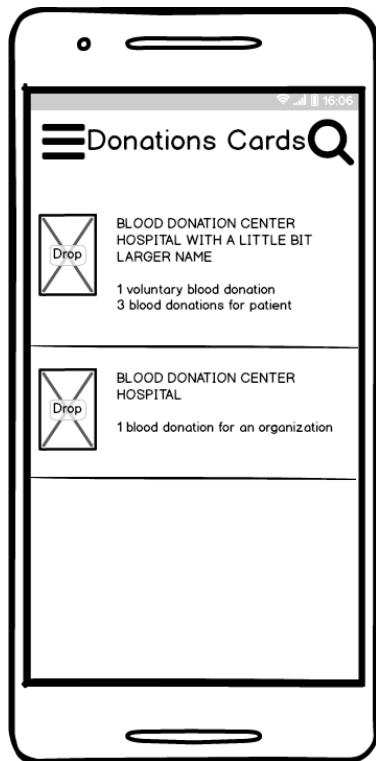
Εικόνα 5-5 Προβολή και τροποποίηση στοιχείων λογαριασμού σε διαφορετικές οθόνες



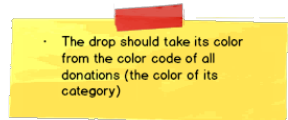
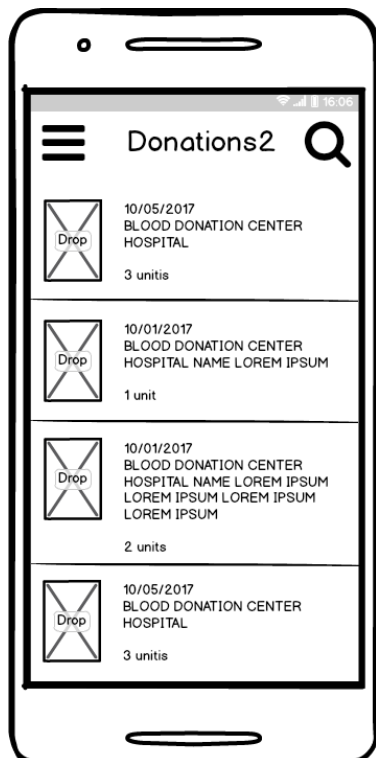
Εικόνα 5-6 Προβολή και τροποποίηση των στοιχείων προφίλ στην ίδια οθόνη



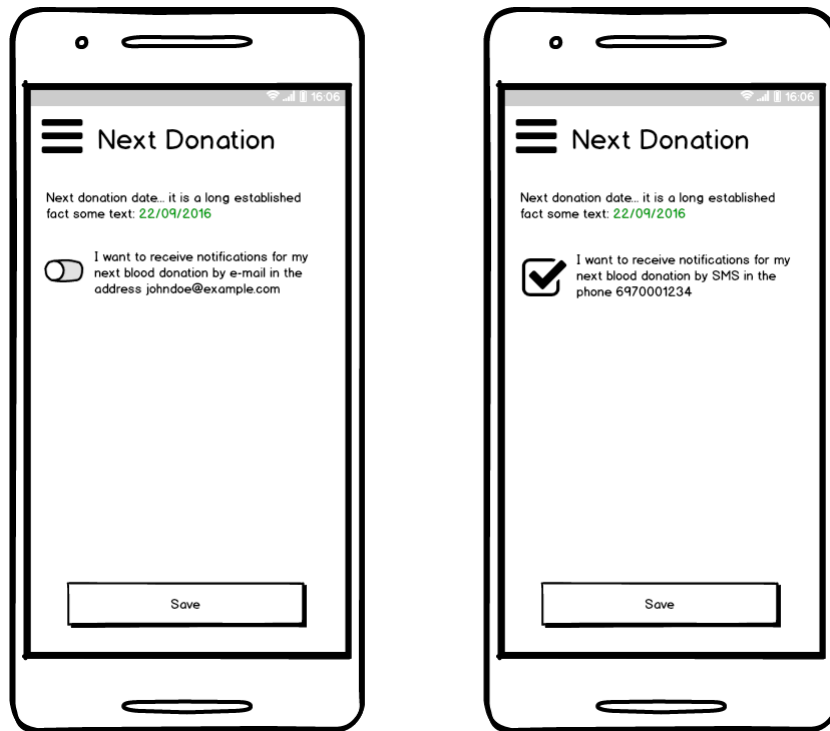
Εικόνα 5-7 Ιστορικό αιμοδοσιών



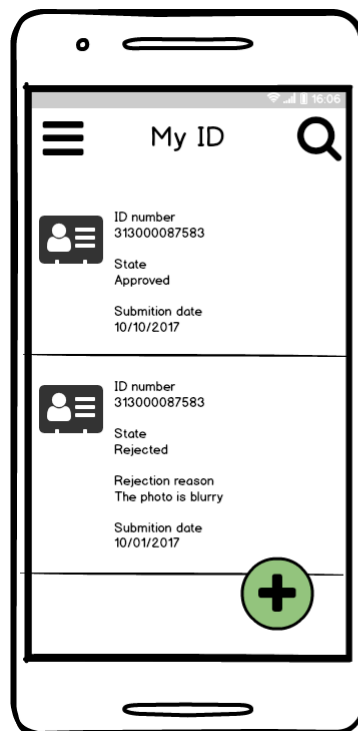
Εικόνα 5-8 Καλύψεις



Εικόνα 5-9 Ιστορικό αιμοδοσιών προγενέστερων του 2015 (καρτέλες νοσοκομείων)



Εικόνα 5-10 Δυνατότητα επόμενης αιμοδοσίας και επιλογές ειδοποίησης



States:

- pending
- submitted
- approved
- sent
- ready
- delivered
- lost
- rejected
- cancelled

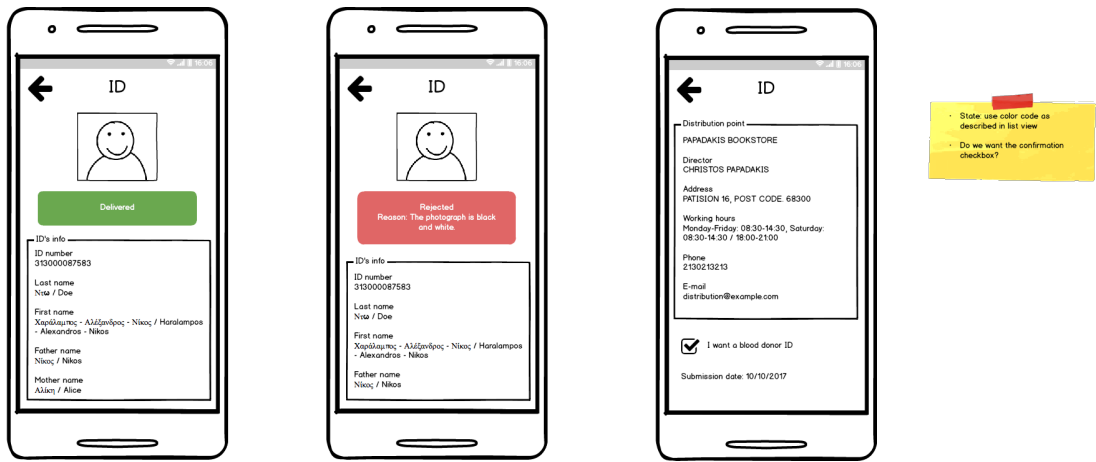
We could use color code for the card icon:

- yellow: pending
- blue/cyan: submitted, approved, sent, ready
- green: delivered
- red: lost, rejected, cancelled

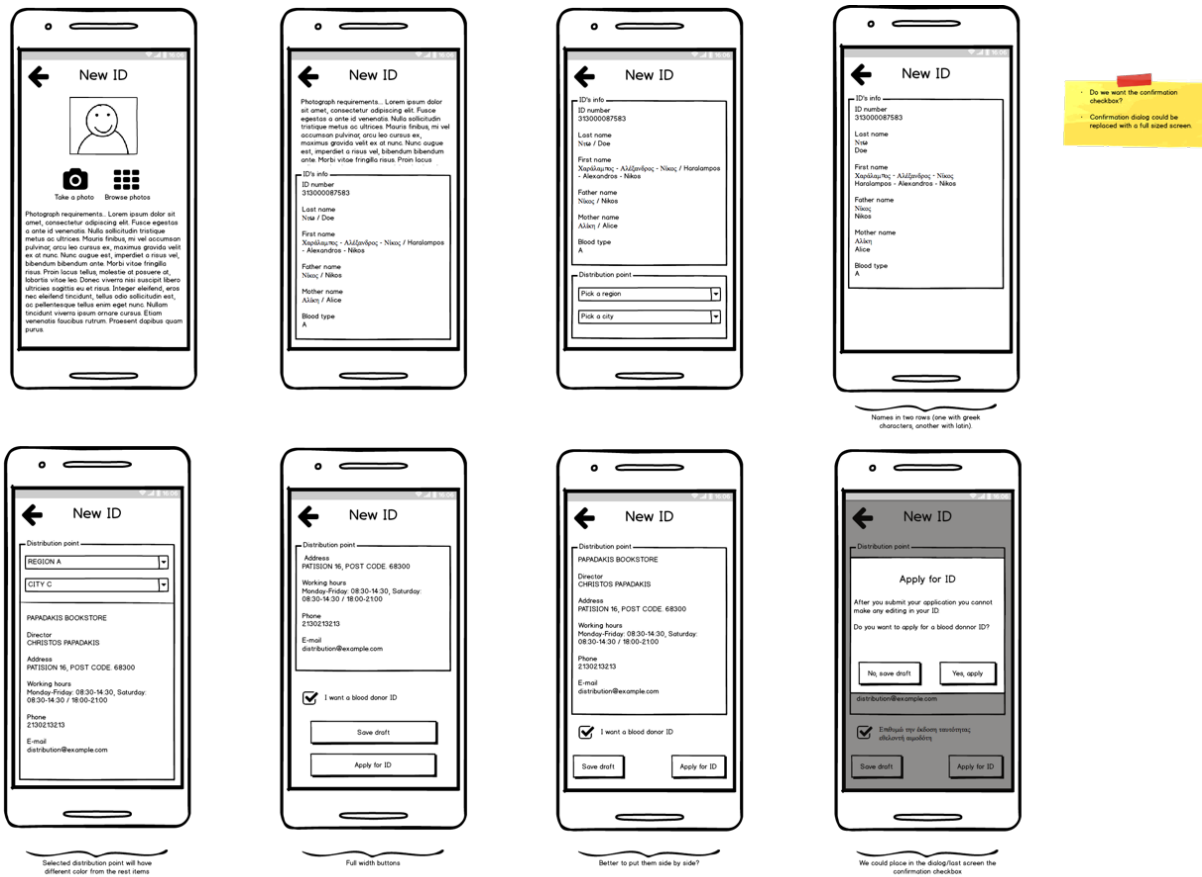
BUT Images are not necessary

Another option is to use the photograph of each

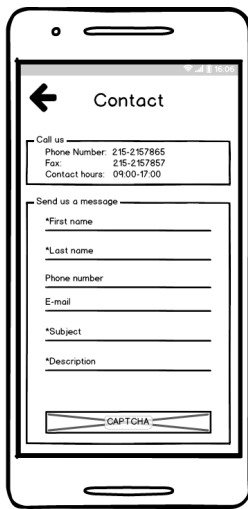
Εικόνα 5-11 Αιτήσεις έκδοσης Κάρτας Αιμοδότη



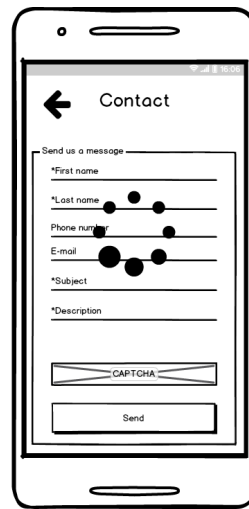
Εικόνα 5-12 Προβολή στοιχείων αίτησης Κάρτας Αιμοδότη



Εικόνα 5-13 Δημιουργία αίτησης για έκδοση Κάρτας Αιμοδότη



After scroll



After send

In list views, we will have the hamburger
 In details, we will have a left arrow
 Title in the height with the icon and
 aligned left

Εικόνα 5-14 Φόρμα επικοινωνίας

6 Σχεδιασμός διεπαφής προγραμματισμού εφαρμογών – API

Ένα άλλο σημείο στο οποίο δόθηκε ιδιαίτερο βάρος ήταν ο σχεδιασμός κατάλληλης διεπαφής προγραμματισμού εφαρμογών (API – Application Programming Interface) που θα εξυπηρετήσει την Android εφαρμογή που αναπτύχθηκε αλλά και οποιονδήποτε άλλον client εξυπηρετεί τις ανάγκες του χρήστη-αιμοδότη. Σκοπός ήταν να σχεδιαστεί ένα API με τέτοιο τρόπο ώστε οι καταναλωτές του να μπορούν να το κατανοήσουν εύκολα και να αξιοποιήσουν τις δυνατότητες του με τον πιο αποδοτικό τρόπο. Για αυτό κρίθηκε ως πολύ σημαντικό το API να συνοδεύεται από τεκμηρίωση που περιλαμβάνει τη δομή και όλες τις χρήσιμες πληροφορίες για τους πόρους που προσφέρει, όπως αυτά προέκυψαν κατά τη φάση του σχεδιασμού.

Να σημειωθεί ότι κατά την ανάπτυξη οποιουδήποτε πληροφοριακού συστήματος είναι σημαντικό να γίνει ο απαραίτητος σχεδιασμός και τεκμηρίωση για τη μορφή και τις δυνατότητες του API. Μερικά από τα οφέλη που προσφέρουν ο σχεδιασμός και η τεκμηρίωση είναι [16]:

- Καλύτερη υλοποίηση του API
- Διευκόλυνση της σταδιακής υλοποίησης σε βάθος χρόνου
- Καλύτερη τεκμηρίωση του πληροφοριακού συστήματος
- Βελτίωση της εμπειρίας του προγραμματιστή

6.1 Επιθυμητά χαρακτηριστικά του API

Ένα API, όπως περιγράφεται παραπάνω, χρειάζεται να χαρακτηρίζεται από τις εξής ιδιότητες [17]:

- Εύκολο στην ανάγνωση και εύκολη η αλληλεπίδραση με αυτό:

Ένα καλοσχεδιασμένο API μπορεί να χρησιμοποιηθεί με ευκολία. Επιπλέον οι πόροι και οι ενέργειες που διαθέτει στους καταναλωτές του πρέπει να μπορούν να απομνημονευθούν γρήγορα.

- Δύσκολο να χρησιμοποιηθεί με λανθασμένο τρόπο:

Η υλοποίηση που βασίζεται σε ένα καλοσχεδιασμένο API θα πρέπει να είναι μία απλή διαδικασία, με μειωμένες πιθανότητες ανάπτυξης λανθασμένου κώδικα.

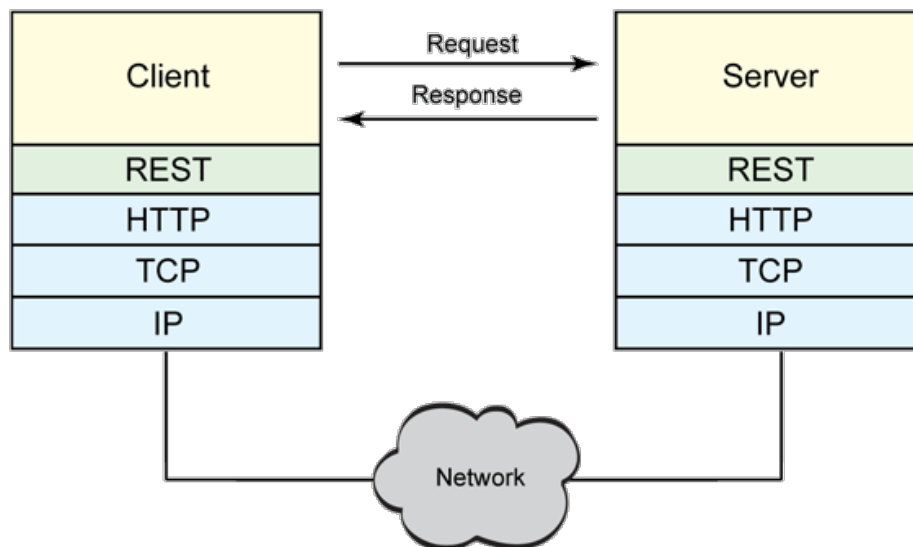
- Πλήρες και συνοπτικό:

Ένα πλήρες API θα δώσει τη δυνατότητα σε άλλους προγραμματιστές να υλοποιήσουν με τη σειρά τους, πλήρεις εφαρμογές που βασίζονται στα δεδομένα και τις λειτουργικότητες που προσφέρει το API. Θα πρέπει βέβαια να σημειωθεί ότι η πληρότητα είναι κάτι που επιτυγχάνεται σταδιακά.

Προκειμένου να επιτύχουμε το API να έχει τα παραπάνω χαρακτηριστικά ακολουθήθηκε μία σειρά από κανόνες που συνθέτουν τη REST αρχιτεκτονική των APIs. Τα REST APIs χρησιμοποιούνται σχεδόν δύο δεκαετίες και είναι ευρέως αποδεκτά.

6.2 REST API

6.2.1 Ορισμός



Εικόνα 6-1 Επικοινωνία διακομιστή-πελάτη μέσω διαδικτύου με χρήση REST

Ο όρος «REST» προέρχεται από τα αρχικά του όρου «Representational State Transfer». Ο συγκεκριμένος όρος ορίστηκε και χρησιμοποιήθηκε για πρώτη φορά το 2000 από τον Roy T. Fielding στη διδακτορική διατριβή του, με τίτλο «Architectural Styles and the Design of Network-based Software Architectures» [18]. Η διατριβή του Fielding παρουσιάζει και εξηγεί τις αρχές που διέπουν τη REST αρχιτεκτονική, οι οποίες από το 1994 είναι ευρέως γνωστές ως «HTTP object model». Οι συγκεκριμένες αρχές χρησιμοποιήθηκαν για το σχεδιασμό του HTTP 1.1 [19] και των Uniform Resource Identifiers (URI) standards.

Η εν λόγω αρχιτεκτονική σχεδίαση ορίζει ένα σύνολο κανόνων και ιδιοτήτων, που μεταφέρονται μέσω HTTP. Οι διαδικτυακές (web) υπηρεσίες που ακολουθούν REST αρχιτεκτονική — συνήθως αναφερόμαστε σε αυτές με τον όρο RESTful υπηρεσίες - κάνουν εφικτή τη διαλειτουργικότητα και τη συνεργασία μεταξύ υπολογιστικών συστημάτων μέσω του διαδικτύου. Τέτοιες εφαρμογές δίνουν τη δυνατότητα σε συστήματα που το επιδιώκουν, να έχουν πρόσβαση και δυνατότητα επεξεργασίας των textual representations των web resources με ομοιογενή τρόπο και με σαφώς προκαθορισμένες stateless διαδικασίες.

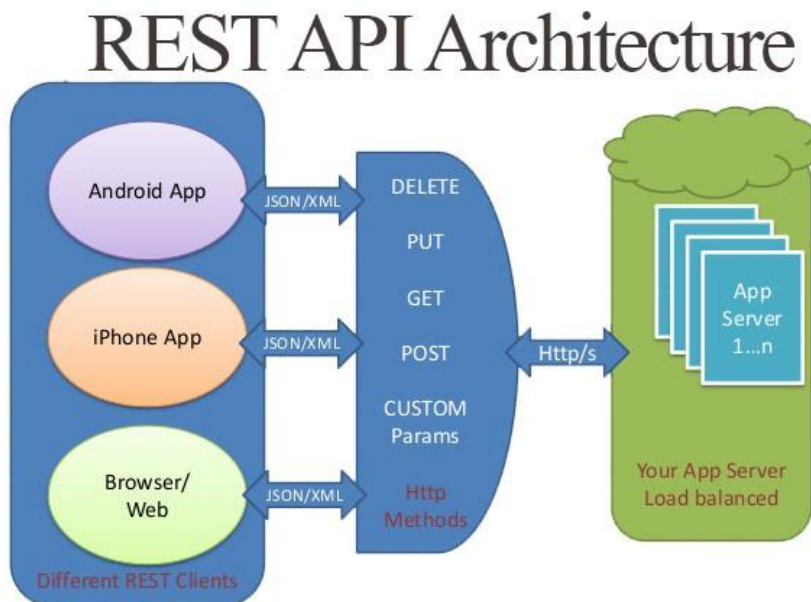
Ο όρος «web resources» αρχικά ορίστηκε για το World Wide Web ως έγγραφα ή αρχεία τα οποία ταυτοποιούνται από το URL τους. Όμως, στις μέρες μας η σημασία του όρου έχει διευρυνθεί. Ο ορισμός περιλαμβάνει κάθε οντότητα που μπορεί να ταυτοποιηθεί, ονομασθεί, τοποθετηθεί ή να γίνει η διαχείρισή του με οποιονδήποτε τρόπο στο διαδίκτυο. Όταν μία RESTful web υπηρεσία, λάβει κλήσεις που κατευθύνονται προς το URI ενός πόρου (resource) θα στείλει μία απάντηση, η οποία μπορεί να έχει μορφή HTML, XML, JSON ή ακόμα κι οποιαδήποτε άλλη. Η απάντηση μπορεί να επιβεβαιώνει ότι κάποια τροποποίηση πραγματοποιήθηκε στο αποθηκευμένο resource. Επιπλέον, η απάντηση μπορεί να διαθέτει και hypertext links για άλλα resources ή για κάποια συλλογή από resources που σχετίζονται με το αρχικό resource (HATEOAS). Όταν χρησιμοποιείται πρωτόκολλο HTTP, που είναι η συνηθισμένη περίπτωση, οι διαθέσιμες ενέργειες που μπορούν να εκτελεστούν είναι GET, POST, PUT, DELETE και άλλες όμοιες που έχουν ορισθεί ως CRUD HTTP μέθοδοι.

Χρησιμοποιώντας ένα stateless πρωτόκολλο [20] που περιλαμβάνει καθορισμένες ενέργειες, τα REST συστήματα στοχεύουν να εξασφαλίσουν βέλτιστη απόδοση, αξιοπιστία και τη δυνατότητα να αυξηθούν σε μέγεθος επαναχρησιμοποιώντας components [21], τα οποία μπορούν να χρησιμοποιηθούν και να αλλαχθούν χωρίς να επηρεάσουν το σύνολο του συστήματος, ακόμα και όταν αυτό είναι σε λειτουργία [22].

6.2.2 Περιγραφή της αρχιτεκτονικής

Σε αυτό το κεφάλαιο περιγράφονται οι αρχές που διέπουν την REST αρχιτεκτονική και οι κανόνες που τη διαφυλάττουν, όπως περιγράφονται στην προαναφερόμενη διδακτορική διατριβή του Fielding [18].

6.2.2.1 Client – Server



Εικόνα 6-2 Αρχιτεκτονική REST API

Ο πρώτος κανόνας που ορίζεται, αφορά το διαχωρισμό μεταξύ των στόχων του διακομιστή (server) και του προγράμματος-πελάτη (client), ο οποίος αναφέρεται ως «separation of concerns». Διαχωρίζοντας τον στόχο της εξυπηρέτησης των αναγκών του χρήστη μέσω της διεπαφής που χρησιμοποιεί, από το στόχο της σωστής αποθήκευσης και διαχείρισης των δεδομένων, επιτυγχάνουμε τη χρήση πολλαπλών διαφορετικών διεπαφών και βελτιώνουμε τη δυνατότητα επέκτασης του συστήματος του server. Η δυνατότητα επέκτασης προκύπτει επειδή τα components που σχεδιάζονται ακολουθώντας αυτήν την αρχή είναι πιο απλά και μπορούν να εξελιχθούν ανεξάρτητα.

6.2.2.2 Stateless

Ο επόμενος κανόνας που τίθεται αφορά την επικοινωνία του server με τον client και συγκεκριμένα την καταγραφή της κατάστασης της μεταξύ τους αλληλεπίδρασης. Με βάση αυτόν τον κανόνα, κάθε request που στέλνει ο client είναι απαραίτητο να περιλαμβάνει όλα τα απαραίτητα στοιχεία που χρειάζεται ο server για να καταλάβει αυτό το request. Όλα τα στοιχεία σχετικά με την κατάσταση της επικοινωνίας (session state) κρατούνται από τον client και όχι από τον server.

Από αυτόν τον κανόνα προκύπτουν και ορισμένες σημαντικές ιδιότητες που ένα REST σύστημα θα πρέπει να επιδιώκει:

- **Ορατότητα (Visibility)**

Η βελτίωση της ορατότητας του συστήματος προκύπτει καθώς ένα εξωτερικό σύστημα δεν χρειάζεται να βρει δεδομένα πέρα από αυτά που περιέχονται στο request για να μπορέσει να κατανοήσει το σκοπό του request.

- **Αξιοπιστία (Reliability)**

Η βελτίωση της αξιοπιστίας προκύπτει επειδή με αυτήν αρχιτεκτονική είναι δυνατή πιο εύκολα η ανάκαμψη και η διαχείριση σφαλμάτων.

- **Επεκτασιμότητα (Scalability)**

Η επεκτασιμότητα διευκολύνεται επειδή το σύστημα του server δεν χρειάζεται να θυμάται επιπλέον στοιχεία για την κατάσταση της επικοινωνίας του με τον client. Εξαιτίας αυτής της τακτικής, ο server μπορεί γρήγορα να απελευθερώσει πόρους. Επιπλέον, επισημαίνεται ότι έτσι απλοποιείται η υλοποίηση του συστήματος, αφού ο server δεν χρειάζεται να διαχειρίζεται τη χρήση πόρων που ζητούνται από πολλαπλά (επί μέρους) requests.

6.2.2.3 Προσωρινή μνήμη (cache)

Με στόχο να επιτευχθεί αποδοτικότερη χρήση του δικτύου προστίθεται κι ένας κανόνας σχετικά με την προσωρινή αποθήκευση δεδομένων μεταξύ του stateless server και του client. Η ύπαρξη αυτής της μνήμης καθιστά απαραίτητο τα δεδομένα στις απαντήσεις του server, να έχουν χαρακτηριστεί έμμεσα ή άμεσα ως στοιχεία που μπορούν ή όχι να αποθηκευτούν στην ενδιάμεση προσωρινή μνήμη.

Προσθέτοντας μία ενδιάμεση προσωρινή μνήμη για τα δεδομένα που ζητούνται επιτυγχάνουμε μείωση ή ακόμα και εκμηδένιση των ενεργειών που φτάνουν στο server, με αποτέλεσμα τη βελτίωση της απόδοσης και της επεκτασιμότητας του συστήματος. Ακόμα κι ο ίδιος ο χρήστης αντιλαμβάνεται ότι το σύστημα είναι πιο γρήγορο αφού μειώνονται οι καθυστερήσεις που προκύπτουν από τη διαχείριση και τη μετακίνηση δεδομένων. Όμως θα πρέπει να δοθεί προσοχή στη χρήση της προσωρινής μνήμης καθώς μπορεί να μειωθεί η αξιοπιστία του συστήματος σε περίπτωση που τα δεδομένα που λαμβάνουν οι clients περιέχουν σημαντικές διαφορές από αυτά εμπεριέχει ο server.

6.2.2.4 Ομοίμορφη διεπαφή (Uniform interface)

Το βασικό σημείο το οποίο διαφοροποιεί την REST αρχιτεκτονική από άλλες είναι η βαρύτητα που δίνεται στην ομοιομορφία των διεπαφών των επιμέρους components. Εφαρμόζοντας ορισμένες γενικές αρχές στη διεπαφή του κάθε component μπορεί να επιτευχθεί απλοποίηση της αρχιτεκτονικής του συνολικού συστήματος και επιπλέον καθορίζεται ένας πιο

ξεκάθαρος τρόπος αλληλεπίδρασης μεταξύ των components. Ο τρόπος υλοποίησης διαχωρίζεται από τον τρόπο που προσφέρεται το service και έτσι είναι πολύ πιο εύκολα επιτεύξιμη η ανεξάρτητη εξέλιξη τους. Επειδή όμως η πληροφορία μεταφέρεται με μία συγκεκριμένη μορφή, που ακολουθεί αυτές τις γενικές αρχές και όχι με μορφή που να διευκολύνει τις ανάγκες της συγκεκριμένης εφαρμογής, η αποδοτικότητα του συστήματος, σε ορισμένες περιπτώσεις, μπορεί να είναι μειωμένη.

Προκειμένου να αναπτυχθεί μία ομοιόμορφη διεπαφή, θα πρέπει να εφαρμοσθούν ορισμένοι κανόνες για την καθοδήγηση της συμπεριφοράς, σύμφωνα με την οποία αλληλοεπιδρούν τα components. Συγκεκριμένα, τίθενται οι εξής κανόνες σχεδίασης της ομοιόμορφης διεπαφής:

- **Ταυτοποίηση των πόρων (Identification of resources)**

Στα πλαίσια της REST αρχιτεκτονικής ο όρος «πόρος» είναι μια αφηρημένη έννοια που περιγράφει οτιδήποτε μπορεί να ορισθεί για το πληροφοριακό σύστημα. Κάθε πόρος πρέπει να έχει ένα μοναδικό αναγνωριστικό. Στα συστήματα που χρησιμοποιούν το διαδίκτυο οι πόροι ταυτοποιούνται (με μοναδικό τρόπο) από το URI τους. Να σημειωθεί ότι η αναπαράσταση του πόρου που περιλαμβάνεται στο response του server προς τον client είναι εξ' ορισμού διαφορετική από τον ίδιο τον πόρο. Για παράδειγμα, ο server μπορεί να στέλνει δεδομένα από τις βάσεις του σε μορφή HTML, XML, JSON χωρίς όμως να είναι αποθηκευμένα σε αυτή τη μορφή στο εσωτερικό σύστημα του server.

- **Διαχείριση των πόρων με χρήση των αναπαραστάσεων τους (Manipulation of resources through representations)**

Όταν ο client λάβει μία αναπαράσταση του πόρου που ζήτησε, συμπεριλαμβανομένων και των μεταδεδομένων του πόρου αυτού, έχει επαρκείς πληροφορίες ώστε να μπορεί να τροποποιήσει ή να διαγράψει τον πόρο αυτό.

- **Αυτό-περιγραφικά μηνύματα (Self-descriptive messages)**

Κάθε μήνυμα θα πρέπει να διαθέτει όλες τις απαραίτητες πληροφορίες για το πως μπορεί αυτό το μήνυμα να το διαχειριστεί οποιοδήποτε άλλο σύστημα. Χαρακτηριστικό παράδειγμα εφαρμογής αυτού του κανόνα είναι η χρήση του πεδίου «media type».

- **HATEOAS: Hypermedia as the engine of application**

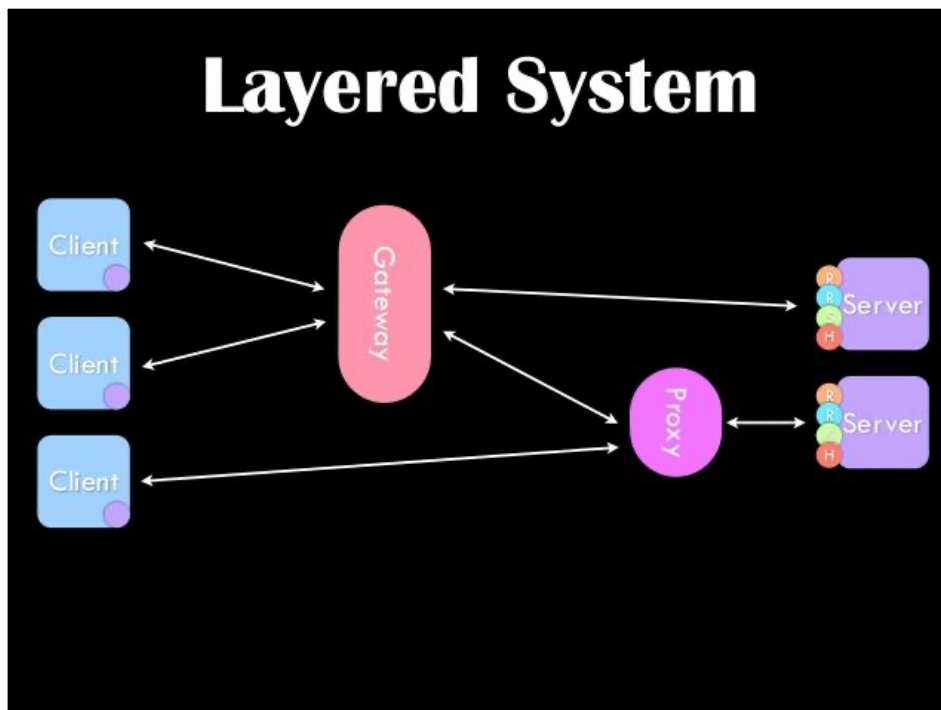
Όταν ένας client αποκτήσει πρόσβαση σε μία REST εφαρμογή, θα πρέπει να έχει τη δυνατότητα να εντοπίσει όλους τους διαθέσιμους πόρους και ενέργειες που επιθυμεί. Αυτό υλοποιείται με την προσθήκη υπερσυνδέσμων (hyperlinks) στις απαντήσεις που του στέλνει ο server. Η λήψη των υπερσυνδέσμων συνεπώς γίνεται σταδιακά. Ακολουθώντας αυτόν τον κανόνα οι clients δεν χρειάζεται να καταγραφούν στον κώδικά τους πληροφορίες σχετικά με τη μορφή της REST διεπαφής που αλληλοεπιδρούν γιατί σε κάθε βήμα που κάνουν βρίσκουν τους απαραίτητους υπερσυνδέσμους για τα επόμενα (μορφή των URLs).

6.2.2.5 Σύστημα με πολλαπλά επίπεδα (Layered system)

Με τον όρο «layered system» (πολύ-επίπεδο σύστημα) περιγράφεται η αρχιτεκτονική δομή που αποτελείται από επίπεδα που έχουν τοποθετηθεί ιεραρχικά. Το κάθε component μπορεί να αντιληφθεί μόνο το πιο κοντινό σε αυτό επίπεδο και να αλληλοεπιδράσει με αυτό. Με αυτήν την διαμοιρασμένη προσέγγιση μειώνεται η πολυπλοκότητα του συστήματος και προάγεται η ανεξαρτησία των επί μέρους στοιχείων του συστήματος.

Τα επίπεδα αυτά μπορούν να αξιοποιηθούν για την απομόνωση παλαιότερων υπηρεσιών του συστήματος ή των clients που είναι σε εξέλιξη ή κατάργηση τους. Επιπλέον, στα πλαίσια αυτής της δομής, διαμεσολαβητές μπορούν να χρησιμοποιηθούν ανάμεσα στους clients και τον server. Αυτοί οι διαμεσολαβητές δεν γίνονται αντιληπτοί από τους clients και μπορούν να χρησιμοποιηθούν για να καλυφθούν ανάγκες που προκύπτουν σε συστήματα μεγάλης κλίμακας, όπως load balancing [23], καθώς και για την εφαρμογή πολιτικών ασφαλείας.

Το κύριο μειονέκτημα αυτής της δομής είναι ότι με τη χρήση του προστίθενται παραπάνω διαδικασίες διαχείρισης των δεδομένων καθώς και μία αντίστοιχη χρονική καθυστέρηση (overhead και latency). Για την αντιστάθμιση αυτών των μειονεκτημάτων υπάρχουν διάφορες τεχνικές όπως π.χ. η χρήση προσωρινής μνήμης σε επιλεγμένα σημεία της ιεραρχικής δομής, όπως αναφέρθηκε και προηγουμένως.



Εικόνα 6-3 Σύστημα με πολλαπλά επίπεδα

6.2.2.6 Αποστολή κώδικα κατόπιν αιτήματος (Code-On-Demand)

Η αποστολή κώδικα κατόπιν αιτήματος είναι ένας προαιρετικός κανόνας της αρχιτεκτονικής. Ακολουθώντας αυτόν τον κανόνα οι clients κατασκευάζονται ώστε να έχουν πρόσβαση σε δεδομένα αλλά και γενικότερα σε πόρους χωρίς να γνωρίζουν πως να τα/τους διαχειριστούν. Αντ' αυτού ο client στέλνει ένα request στον server, ώστε να λάβει κώδικα που υλοποιεί τη διαχείριση και αφού τον λάβει, τον αξιοποιεί εκτελώντας τον τοπικά.

Στα προτερήματα της αποστολής κώδικα με αυτόν τον τρόπο περιλαμβάνονται η διευκόλυνση της επέκτασης και παραμετροποίησης των clients. Εξετάζοντας για το αν θα χρησιμοποιηθεί αυτός ο κανόνας θα πρέπει να συνυπολογιστούν και οι περιορισμοί μίας τέτοιας τακτικής. Ο σημαντικότερος, σύμφωνα με τον Fielding, είναι η έλλειψη ορατότητας του κώδικα που τρέχει ο client τοπικά, που μπορεί να οδηγήσει σε προβλήματα σχετικά με την ασφάλεια του συστήματος και του χρήστη.

6.3 Περιγραφή του προτεινόμενου API με OpenAPI Specification

6.3.1 OpenAPI Specification (OAS)

Για την περιγραφή του API χρησιμοποιήθηκε το OpenAPI Specification (OAS), παλαιότερα γνωστό ως Swagger Specification. Το OAS προδιαγράφει έναν τρόπο αποτύπωσης των RESTful APIs, ανεξάρτητο από τις γλώσσες και τις τεχνολογίες που χρησιμοποιούνται για την υλοποίηση ή την κατανάλωσή τους. Αυτή η αποτύπωση είναι κατανοητή και αξιοποιήσιμη τόσο από υπολογιστικά συστήματα, όσο και από ανθρώπους. Η περιγραφή των διαθέσιμων πόρων και των δυνατοτήτων προς τους καταναλωτές (clients) γίνεται δυνατή χωρίς να χρειάζεται να εξετασθεί ο κώδικας της υπηρεσίας ή άλλα έγγραφα που την περιγράφουν. Επομένως, με το OAS μπορεί να περιγράψει το API με τέτοιο τρόπο ώστε ο καταναλωτής αυτού να το καταλάβει και να καταστεί δυνατή η αλληλεπίδραση του με την απομακρυσμένη υπηρεσία, με την ελάχιστη δυνατή προσπάθεια.

Μία καταγραφή, όταν έχει γίνει σύμφωνα με το OAS, μπορεί να χρησιμοποιηθεί από εργαλεία που παράγουν αυτοματοποιημένα διάφορους χρήσιμους πόρους [24], όπως:

- Την απεικόνιση του API
- Τον κώδικα που υλοποιεί ή καταναλώνει αυτό το API σε διάφορες γλώσσες
- API tests

6.3.2 Προτεινόμενο API

Στο API προτείνεται ο καταναλωτής να αλληλεπιδρά με το διακομιστή με τη χρήση HTTP λειτουργιών. Επιλέχθηκε οι λειτουργίες αυτές να είναι προκαθορισμένες και ευρέως χρησιμοποιούμενες στο χώρο της πληροφορικής. Συνεπώς, αξιοποιήθηκαν οι μέθοδοι που περιγράφονται στα RFC 2616 [19] και RFC 5789 [25]. Συγκεκριμένα, οι HTTP μέθοδοι που κρίθηκαν χρήσιμες για την εξυπηρέτηση των αναγκών των χρηστών-αιμοδοτών είναι: GET, POST, PATCH.

Το προτεινόμενο API θα πρέπει να περιέχει endpoints σύμφωνα με τους πόρους και τις λειτουργίες που διατίθενται στους καταναλωτές. Η μορφή τους χρειάζεται να είναι απλή και οι απαντήσεις τους προς τους καταναλωτές κατανοητές. Επομένως, για κάθε συλλογή από πόρους του ίδιου είδους δημιουργήθηκε ένα endpoint, που η ονομασία του περιγράφει το είδος του πόρου σε πληθυντικό αριθμό. Π.χ. «/donations», «/regions», «/users». Όταν ένα endpoint σχετίζεται με ένα συγκεκριμένο πόρο, που αποτελεί μέλος της συλλογής που προσφέρουν τα παραπάνω endpoints, η μορφή του αποτελείται από 2 μέρη: το όνομα της συλλογής και το μοναδικό του αναγνωριστικό. Π.χ. «/donations/1287», «/regions/12», «/users/123».

Για τις συλλογές που περιέχουν μεγάλο αριθμό μελών ή προβλέπεται να αποκτήσουν πολλά μέλη προτάθηκε να είναι δυνατή η ανάκτηση πόρων ενός είδους που τηρούν συγκεκριμένα κριτήρια καθώς και η ανάκτηση ορισμένου αριθμού πόρων. Ενδεικτικά προτείνεται να υποστηρίζεται από το διακομιστή, ως προαιρετική παράμετρος (optional query parameter), ο όρος «fields». Ο καταναλωτής μπορεί να χρησιμοποιεί τον όρο αυτό για να απαριθμήσει ποια από τα διαθέσιμα στοιχεία των πόρων χρειάζεται. Συμπληρωματικά, μπορεί να χρησιμοποιηθεί ο header «If-Modified-Since», όπως ορίζεται στο RFC 7232 [26]. Με αυτόν τον τρόπο ο καταναλωτής μπορεί να πληροφορηθεί αν τα στοιχεία/πόροι που έχει αποκτήσει στο παρελθόν από ένα endpoint έχουν μείνει αμετάβλητα και μπορεί να τα παρουσιάσει στον χρήστη ή χρειάζεται να τα ζητήσει εκ νέου.

Στα πλαίσια της προσπάθειας μείωσης του μεγέθους των πληροφοριών που αποστέλλονται από το διακομιστή, περιορίστηκαν τα στοιχεία σε αυτά που είναι απαραίτητα για τους καταναλωτές, αποφεύγοντας επαναλήψεις ή συγχωνεύσεις πληροφοριών. Αυτή η εργασία είχε σημασία καθώς η προς υλοποίηση εφαρμογή πολλές φορές θα λειτουργεί συνδεδεμένη σε δίκτυο χαμηλής ταχύτητας.

Προκειμένου οι πόροι που παρέχονται από το διακομιστή να είναι εύκολα κατανοητοί μειώθηκε ο βαθμός εμφώλευσης και μετονομάστηκαν κλειδιά και endpoints. Για να είναι προβλέψιμος ο τρόπος πρόσβασης των καταναλωτών σε πόρους και πληροφορίες προτείνεται όλα τα διαθέσιμα endpoints να έχουν κοινό το αρχικό κομμάτι του URL τους (base URL).

Επιπλέον, παρατηρήθηκε ότι οι πόροι είναι διαθέσιμοι σε δύο γλώσσες. Για να ενισχύσουμε την προβλεψιμότητα και την καλή επικοινωνία μεταξύ διακομιστή και καταναλωτή όσον αφορά τη γλώσσα, προτείνεται η υποστήριξη του header «Accept-Language» [27] και την αποστολή του header «Content-Language» [28] μαζί με κάθε απάντηση.

Με τα παραπάνω επιτεύχθηκε η σχεδίαση ενός εύκολα κατανοητού API, το οποίο ακολουθεί τους κανόνες που ορίζονται στη REST αρχιτεκτονική και μπορεί να υποστηρίξει εφαρμογές-καταναλωτές που έχουν πρόσβαση σε δίκτυο χαμηλής ταχύτητας. Στη συνέχεια περιλαμβάνεται η πλήρης αποτύπωση του API, δομημένη σύμφωνα με το OpenAPI Specification.

```

openapi: 3.0.1
info:
  title: Blood Donor Registry: Proposed API for donors'
application
  version: 1.0.0
servers:
- url: https://service.bloodonorregistry.gr/v2
paths:
  /v2/donations:
    get:
      description: The donations that the user has done in
the past.
      operationId: getDonations
      parameters:
        - name: Accept-Language
          in: header
          description: |
            The languages that the client is able to understand,
and which locale variant is preferred.
          schema:
            type: string
        - name: If-Modified-Since
          in: header
          description: |
            Makes the request conditional. The server will send
back the data, with a 200 status, only if the list or its items
have been modified after the given date, otherwise will respond
with the 304 "Not Modified" status.
          schema:
            type: string
      responses:
        200:
          description: OK
          headers:
            Last-Modified:
              description: Date and time at which the origin
server believes
                the resource was last modified.

```



```

    schema:
      type: string
      format: date-time
  Content-Language:
    description: The language of the content.
    schema:
      type: string
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/Donations'
      example:
        - id: donation_11
          donatedAt: 2010-07-17
          occasion: RELATIVE
          hospitalName: Η ΣΩΤΗΡΙΑ
          collectedComponent: Platelets
        - id: donation_111
          donatedAt: 2009-05-31
          occasion: BLOODDONORASSOCIATION
          hospitalName: ΕΥΑΓΓΕΛΙΣΜΟΣ
          collectedComponent: Blood
  304:
    description: Not Modified
    content: {}
  401:
    description: Unauthorized
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/ErrorMsg'
        example:
          error: Unauthorized
  security:
    - token: []

```

```

/v2/municipalities:
  get:
    description: Get the list of municipalities. No token
is required.
    operationId: getMunicipalities
    parameters:
      - name: Accept-Language
        in: header
        description: |
          The languages that the client is able to understand,
and which locale variant is preferred.
        schema:
          type: string
      - name: If-Modified-Since
        in: header
        description: |
          Makes the request conditional. The server will send
back the data, with a 200 status, only if the list or its items
have been modified after the given date, otherwise will respond
with the 304 "Not Modified" status.
        schema:
          type: string
      - name: region_id
        in: query
        description: Get the list of municipalities that are
located in the region
          with the given ID.
        schema:
          type: integer
    responses:
      200:
        description: OK
        headers:
          Last-Modified:
            description: Date and time at which the origin
server believes the resource
              was last modified.

```

```

        schema:
          type: string
      Content-Language:
        description: The language of the content.
        schema:
          type: string
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Municipalities'
        example:
          - id: 123
            name: ΖΩΓΡΑΦΟΥ
          - id: 345
            name: ΔΟΞΑΤΟΥ
    304:
      description: Not Modified
      content: {}
/v2/regions:
  get:
    description: Get the list of regions. Only Greek regions
are included. No token
    is required.
    operationId: getRegions
    parameters:
      - name: Accept-Language
        in: header
        description: |
          The languages that the client is able to understand,
and which locale variant is preferred.
        schema:
          type: string
      - name: If-Modified-Since
        in: header
        description: |

```

Makes the request conditional. The server will send back the data, with a 200 status, only if the list or its items have been modified after the given date, otherwise will respond with the 304 "Not Modified" status.

```
    schema:
      type: string
  responses:
    200:
      description: OK
      headers:
        Last-Modified:
          description: Date and time at which the origin
server believes the resource
          was last modified.
          schema:
            type: string
        Content-Language:
          description: The language of the content.
          schema:
            type: string
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Regions'
          example:
            - id: 12
              name: AN. MAKEΔONΙΑΣ & ΘΡΑΚΗΣ
            - id: 34
              name: ΑΤΤΙΚΗΣ
    304:
      description: Not Modified
      content: {}
/v2/coverages:
  get:
    description: Get the list of coverages for the given
user.
    operationId: getCoverages
```

```

parameters:
  - name: Accept-Language
    in: header
    description: |
      The languages that the client is able to understand,
      and which locale variant is preferred.
    schema:
      type: string
  - name: If-Modified-Since
    in: header
    description: |
      Makes the request conditional. The server will send
      back the data, with a 200 status, only if the list or its items
      have been modified after the given date, otherwise will respond
      with the 304 "Not Modified" status.
    schema:
      type: string
responses:
  200:
    description: OK
    headers:
      Last-Modified:
        description: Date and time at which the origin
server believes the resource
        was last modified.
        schema:
          type: string
      Content-Language:
        description: The language of the content.
        schema:
          type: string
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Coverages'
        example:
          - id: kalipsi_11

```

```

        reservedAt: 2010-07-17
        numOfBags: 3
        hospitalName: Η ΣΩΤΗΡΙΑ
    - id: kalipsi_111
      reservedAt: 2009-05-31
      numOfBags: 2
      hospitalName: ΕΥΑΓΓΕΛΙΣΜΟΣ
    - id: kalipsi_211
      reservedAt: 2014-12-11
      numOfBags: 3
      hospitalName: ΑΓΙΟΣ ΑΝΤΡΕΑΣ
304:
  description: Not Modified
  content: {}
401:
  description: Unauthorized
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/ErrorMsg'
      example:
        error: Unauthorized
security:
  - token: []
/v2/users/{id}:
  get:
    description: |
      Get user related info. Includes personal data, public
      records that the user has presented, their address in Greece
      etc.
    operationId: getUser
    parameters:
      - name: id
        in: path
        required: true
        schema:

```

```

    type: string
  - name: Accept-Language
    in: header
    description: |
      The languages that the client is able to understand,
      and which locale variant is preferred.
    schema:
      type: string
  - name: If-Modified-Since
    in: header
    description: |
      Makes the request conditional. The server will send
      back the data, with a 200 status, only if the list or its items
      have been modified after the given date, otherwise will respond
      with the 304 "Not Modified" status.
    schema:
      type: string
responses:
  200:
    description: OK
    headers:
      Last-Modified:
        description: Date and time at which the origin
server believes the resource
          was last modified.
        schema:
          type: string
      Content-Language:
        description: The language of the content.
        schema:
          type: string
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/User'
        example:
          donorCardID: 311700793038

```

```
donorCardStatus: APPROVED
registrationNumber: "1700793035"
socialSecurityNumber: "22028602054"
birthCountry: ΕΛΛΑΔΑ
birthDate: 1986-02-22
birthRegion: ΑΝ. ΜΑΚΕΔΟΝΙΑΣ & ΘΡΑΚΗΣ
majorBloodGroup: A
cellNumber: "6971231234"
education: LEVEL7
municipality:
  id: "9186"
  name: ΑΘΗΝΑΙΩΝ
emailValid: true
email: john@example.org
fathersName: ΓΕΩΡΓΙΟΣ
firstName: ΙΩΑΝΝΗΣ
gender: MALE
id: 12345
identityDocuments:
  type: ast
  number: AX 2134
  issueCountry: ΕΛΛΑΔΑ
lastName: ΝΤΟΠΟΥΛΟΣ
mothersName: ANNA
phoneNumber: "2103217123"
region:
  id: 9
  name: ΑΤΤΙΚΗΣ
rhesus: POSITIVE
smsValid: true
status: ACTIVE
street: ΜΠΟΥΜΠΟΥΛΙΝΑΣ
streetNumber: "10"
type: BloodDonorUser
username: johnny
```



```

        zipcode: "14440"
    304:
        description: Not Modified
        content: {}
    401:
        description: Unauthorized
        content:
            application/json:
                schema:
                    $ref: '#/components/schemas/ErrorMsg'
                example:
                    error: Unauthorized
    security:
        - token: []
    patch:
        description: |
            Update any of the contact info, including home
            address, phone and/or e-mail address.
        operationId: updateContactInfo
        parameters:
            - name: id
              in: path
              required: true
              schema:
                  type: string
            - name: Accept-Language
              in: header
              description: |
                  The languages that the client is able to understand,
                  and which locale variant is preferred.
              schema:
                  type: string
        requestBody:
            description: The data that is requested to be modified
            for the user.
            content:

```

```

    application/json:
      schema:
        $ref: '#/components/schemas/UserContactInfo'
    text/plain:
      schema:
        $ref: '#/components/schemas/UserContactInfo'
  required: false
responses:
  200:
    description: OK
    headers:
      Content-Language:
        description: The language of the content.
        schema:
          type: string
    content: {}
  401:
    description: Unauthorized
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/ErrorMsg'
        example:
          error: Unauthorized
  security:
    - token: []
/v2/nextDonation:
  get:
    description: Get the date that the user is able to make
their next donation
again.
    operationId: getNextDonation
  responses:
    200:
      description: OK
      headers:

```

```

        Content-Language:
          description: The language of the content.
          schema:
            type: string
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/NextDonation'
        example:
          donateAt: 2018-01-04T22:00:00.000+0000
/v2/alert:
  get:
    description: |
      Get the user's notification preferences. Informs
      whether they are notified when they are next able to donate, and
      the method that will be used to notify them (e-mail or sms).
    operationId: getAlert
    responses:
      200:
        description: OK
        headers:
          Content-Language:
            description: The language of the content.
            schema:
              type: string
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Alert'
            example:
              status: ACTIVE
              email: john@example.org
/v2/previousRecords:
  get:
    description: Hospital records that were kept before the
    national donation system

```

```

    for each blood donor.
  operationId: getPreviousRecords
  parameters:
  - name: Accept-Language
    in: header
    description: |
      The languages that the client is able to understand,
and which locale variant is preferred.
    schema:
      type: string
  - name: If-Modified-Since
    in: header
    description: |
      Makes the request conditional. The server will send
back the data, with a 200 status, only if the list or its items
have been modified after the given date, otherwise will respond
with the 304 "Not Modified" status.
    schema:
      type: string
  responses:
    200:
      description: OK
      headers:
        Last-Modified:
          description: Date and time at which the origin
server believes the resource
was last modified.
          schema:
            type: string
        Content-Language:
          description: The language of the content.
          schema:
            type: string
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/PreviousRecords'

```

```

example:
- id: 253
  hospitalName: ΙΠΠΟΚΡΑΤΕΙΟ
  volunteerDonations: 1
  patientDonations: 0
  coverages: 3
  associationDonations: 0
  registrationNumber: 1550793035
- id: 2553
  hospitalName: ΑΧΕΠΑ
  volunteerDonations: 0
  patientDonations: 0
  coverages: 0
  associationDonations: 8
  associationName: Σύλλογος Εργαζομένων
  registrationNumber: 1550793035
- id: 4853
  hospitalName: ΑΓΙΑ ΣΟΦΙΑ
  volunteerDonations: 0
  patientDonations: 5
  coverages: 6
  associationDonations: 0

304:
  description: Not Modified
  content: {}

401:
  description: Unauthorized
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/ErrorMsg'
      example:
        error: Unauthorized

security:
- token: []

```

```

/v2/cardApplications:
  get:
    description: Each donor can apply for a donor card. May
have more than one application.
    operationId: getCardApplications
    parameters:
      - name: Accept-Language
        in: header
        description: |
          The languages that the client is able to understand,
and which locale variant is preferred.
        schema:
          type: string
      - name: If-Modified-Since
        in: header
        description: |
          Makes the request conditional. The server will send
back the data, with a 200 status, only if the list or its items
have been modified after the given date, otherwise will respond
with the 304 "Not Modified" status.
        schema:
          type: string
      - name: fields
        in: query
        description: |
          The fields that would be included in every
application object in the response array. If this parameter is
not used the response includes every available attribute.
        schema:
          type: string
    responses:
      200:
        description: OK
        headers:
          Last-Modified:
            description: Date and time at which the origin
server believes the resource
              was last modified.

```

```

    schema:
      type: string
  Content-Language:
    description: The language of the content.
    schema:
      type: string
  content:
    application/json:
      schema:
        $ref:
'#/components/schemas/CardApplications'
      example:
        - id: 253
          status: PENDING
          donorCardID: 643803103867
          photo: data:image/png;base64,skipped the rest
string
          email: john@example.org
          cellNumber: "6901231231"
          mothersName: ANNA
          mothersNameLatin: ANNA
          fathersName: ΓΕΩΡΓΙΟΣ
          fathersNameLatin: GEORGE
          firstName: ΙΩΑΝΝΗΣ
          firstNameLatin: JOHN
          lastName: ΝΤΟΠΟΥΛΟΣ
          lastNameLatin: DOPOULOS
          rhesus: POSITIVE
          majorBloodGroup: A
        - id: 253
          status: REJECTED
          rejectionReason: Not good photo
          submittedAt: 12-12-12
          donorCardID: 643803103867
          photo: data:image/png;base64,skipped the rest
string

```

```
email: john@example.org
cellNumber: "6901231231"
mothersName: ANNA
mothersNameLatin: ANNA
fathersName: ΓΕΩΡΓΙΟΣ
fathersNameLatin: GEORGE
firstName: ΙΩΑΝΝΗΣ
firstNameLatin: JOHN
lastName: ΝΤΟΠΟΥΛΟΣ
lastNameLatin: DOPOULOS
rhesus: POSITIVE
majorBloodGroup: A
distributionCenter:
  id: 9012
  email: info@distribution.gr
  street: ΑΓ. ΓΕΩΡΓΙΟΥ
  streetNumber: "1"
  zipcode: "10440"
  phoneNumber: "2101234567"
  visitingHours: 'Δευτέρα, Τετάρτη: 09:00-
16:00, Τρίτη, Πέμπτη, Παρασκευή:
  09:00-21:00, Σάββατο: 09:00-15:00'
  municipality:
    id: 111
    name: ΚΑΒΑΛΑΣ
    regionID: 1
  region:
    id: 1
    name: ΑΝ. ΜΑΚΕΔΟΝΙΑΣ & ΘΡΑΚΗΣ

304:
  description: Not Modified
  content: {}

401:
  description: Unauthorized
  content:
    application/json:
```



```

        schema:
          $ref: '#/components/schemas/ErrorMsg'
        example:
          error: Unauthorized
    security:
      - token: []
/v2/cardApplications/{id}:
  get:
    description: Each donor can apply for a donor card. May
have more than one application.
    operationId: getCardApplication
    parameters:
      - name: Accept-Language
        in: header
        description: |
          The languages that the client is able to understand,
and which locale variant is preferred.
        schema:
          type: string
      - name: If-Modified-Since
        in: header
        description: |
          Makes the request conditional. The server will send
back the data, with a 200 status, only if the list or its items
have been modified after the given date, otherwise will respond
with the 304 "Not Modified" status.
        schema:
          type: string
      - name: id
        in: path
        description: The detailed list of attributes of an
application.
        required: true
        schema:
          type: string
    responses:

```

```

200:
  description: OK
  headers:
    Last-Modified:
      description: Date and time at which the origin
server believes the resource
      was last modified.
      schema:
        type: string
    Content-Language:
      description: The language of the content.
      schema:
        type: string
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/CardApplication'
      example:
        id: 253
        status: SENT
        rejectionReason: Not good photo
        submittedAt: 12-12-12
        donorCardID: 643803103867
        photo: data:image/png;base64,skipped the rest
string

        email: john@example.org
        cellNumber: "6901231231"
        mothersName: ANNA
        mothersNameLatin: ANNA
        fathersName: ΓΕΩΡΓΙΟΣ
        fathersNameLatin: GEORGE
        firstName: ΙΩΑΝΝΗΣ
        firstNameLatin: JOHN
        lastName: ΝΤΟΠΟΥΛΟΣ
        lastNameLatin: DOPOULOS
        rhesus: POSITIVE

```

```

majorBloodGroup: A
distributionCenter:
  id: 9012
  email: info@distribution.gr
  street: ΑΓ. ΓΕΩΡΓΙΟΥ
  streetNumber: "1"
  zipcode: "10440"
  phoneNumber: "2101234567"
  visitingHours: 'Δευτέρα, Τετάρτη: 09:00-
16:00, Τρίτη, Πέμπτη, Παρασκευή:
  09:00-21:00, Σάββατο: 09:00-15:00'
municipality:
  id: 111
  name: ΚΑΒΑΛΑΣ
  regionID: 1
region:
  id: 1
  name: ΑΝ. ΜΑΚΕΔΟΝΙΑΣ & ΘΡΑΚΗΣ

304:
  description: Not Modified
  content: {}

401:
  description: Unauthorized
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/ErrorMsg'
      example:
        error: Unauthorized

security:
  - token: []

/captcha:
  get:
    summary: Get CAPTCHA data.
    description: |

```

Get CAPTCHA image encoded in base64 and its ID. Use the ID and the CAPTCHA answer that a user can read in the image, for the requests that is necessary.

No token is required.

operationId: getCaptcha

responses:

200:

description: OK

content:

application/json:

schema:

\$ref: '#/components/schemas/Captcha'

text/html:

schema:

\$ref: '#/components/schemas/Captcha'

id:

example: "91606034"

img:

example: PNG encoded in Base64. String is

skipped.

/contact:

post:

summary: Send a message from the user to the helpdesk.

description: |

A user can send a message.

E-mail or phoneNumber is required.

No token is required.

operationId: sendIssue

requestBody:

description: The content of the communication.

content:

application/json:

schema:

\$ref: '#/components/schemas/Message'

text/plain:

schema:

```

        $ref: '#/components/schemas/Message'
    required: false
responses:
  200:
    description: OK
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/MessageKey'
      text/html:
        schema:
          $ref: '#/components/schemas/MessageKey'
    key:
      example: BDR-19954
  400:
    description: Bad request
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/ErrorMsg'
      text/html:
        schema:
          $ref: '#/components/schemas/ErrorMsg'
    error:
      example: invalid captcha
components:
  schemas:
    ErrorMsg:
      type: object
      properties:
        error:
          type: string
    Captcha:
      type: object
      properties:

```

```
    id:
      type: string
    img:
      type: string
  Message:
    required:
      - captchaAnswer
      - captchaId
      - subject
      - type
    type: object
    properties:
      captchaId:
        type: integer
      captchaAnswer:
        type: string
      description:
        type: string
      email:
        type: string
      firstName:
        type: string
      lastName:
        type: string
      phoneNumber:
        type: string
      subject:
        type: string
      type:
        type: string
  MessageKey:
    type: object
    properties:
      key:
        type: string
```

```

Donation:
  type: object
  properties:
    id:
      type: number
    collectedComponent:
      type: string
      enum:
        - Blood
        - Platelets
    donatedAt:
      type: string
      format: date
    hospitalName:
      type: string
    occasion:
      type: string
      enum:
        - BLOODDONORASSOCIATION
        - MILITARY
        - RELATIVE
        - VOLUNTEER

Donations:
  type: array
  items:
    $ref: '#/components/schemas/Donation'

Coverage:
  type: object
  properties:
    id:
      type: integer
    hospitalName:
      type: string
    numOfBags:
      type: number

```

```

    reservedAt:
      type: string
      format: date
  Coverages:
    type: array
    items:
      $ref: '#/components/schemas/Coverage'
  User:
    type: object
    properties:
      birthCountry:
        type: string
      birthDate:
        type: string
      birthRegion:
        type: string
        description: 'Note: If the birthCountry is not
Greece the birthRegion is
        not available'
        nullable: true
      cellNumber:
        type: string
      donorCardID:
        type: integer
      donorCardStatus:
        type: string
        enum:
          - APPROVED
          - CANCELLED
          - DELIVERED
          - LOST
          - PENDING
          - READY
          - REJECTED
          - SENT
          - SUBMITTED

```



```
education:
  type: string
  enum:
    - LEVEL1
    - LEVEL2
    - LEVEL3
    - LEVEL4
    - LEVEL5
    - LEVEL6
    - LEVEL7
email:
  type: string
emailValid:
  type: boolean
fathersName:
  type: string
firstName:
  type: string
gender:
  type: string
id:
  type: integer
identityDocuments:
  type: array
  items:
    $ref: '#/components/schemas/IdentityDocument'
lastName:
  type: string
majorBloodGroup:
  type: string
mothersName:
  type: string
municipality:
  $ref: '#/components/schemas/Municipality'
phoneNumber:
```

```
    type: string
  region:
    $ref: '#/components/schemas/Region'
  registrationNumber:
    type: string
  rhesus:
    type: string
    enum:
      - POSITIVE
      - NEGATIVE
  smsValid:
    type: boolean
  socialSecurityNumber:
    type: string
  status:
    type: string
    enum:
      - ACTIVE
      - LOCKED
      - DISABLED
  street:
    type: string
  streetNumber:
    type: string
  type:
    type: string
    enum:
      - BloodDonorUser
  username:
    type: string
  zipcode:
    type: string
UserContactInfo:
  type: object
  properties:
```

```
municipality:
  $ref: '#/components/schemas/Municipality'
street:
  type: string
streetNumber:
  type: string
zipcode:
  type: string
cellNumber:
  type: string
phoneNumber:
  type: string
email:
  type: string
IdentityDocument:
  type: object
  properties:
    id:
      type: string
    issueCountry:
      type: string
    number:
      type: string
    type:
      type: string
      enum:
        - aad
        - aap
        - aat
        - adb
        - adt
        - afm
        - aka
        - ama
        - amka
```

```
- ast
Municipality:
  type: object
  properties:
    id:
      type: integer
    name:
      type: string
Municipalities:
  type: array
  items:
    $ref: '#/components/schemas/Municipality'
Region:
  type: object
  properties:
    id:
      type: integer
    name:
      type: string
Regions:
  type: array
  items:
    $ref: '#/components/schemas/Region'
NextDonation:
  type: object
  properties:
    donateAt:
      type: string
      format: date
Alert:
  type: object
  properties:
    status:
      type: string
      enum:
```

```

- DISABLED
- ACTIVE
email:
  type: string
  description: If "email" is returned, the user will
receive the alert in
  their e-mail.
mobile:
  type: string
  description: If "mobile" is returned, the user will
receive the alert in
  their mobile.
HospitalRecord:
  type: object
  properties:
    id:
      type: integer
    hospitalName:
      type: string
    volunteerDonations:
      type: integer
    patientDonations:
      type: integer
    coverages:
      type: integer
    associationDonations:
      type: integer
    associationName:
      type: string
    registrationNumber:
      type: string
      description: Blood donor ID number at the current
hospital
CardApplication:
  type: object
  properties:

```

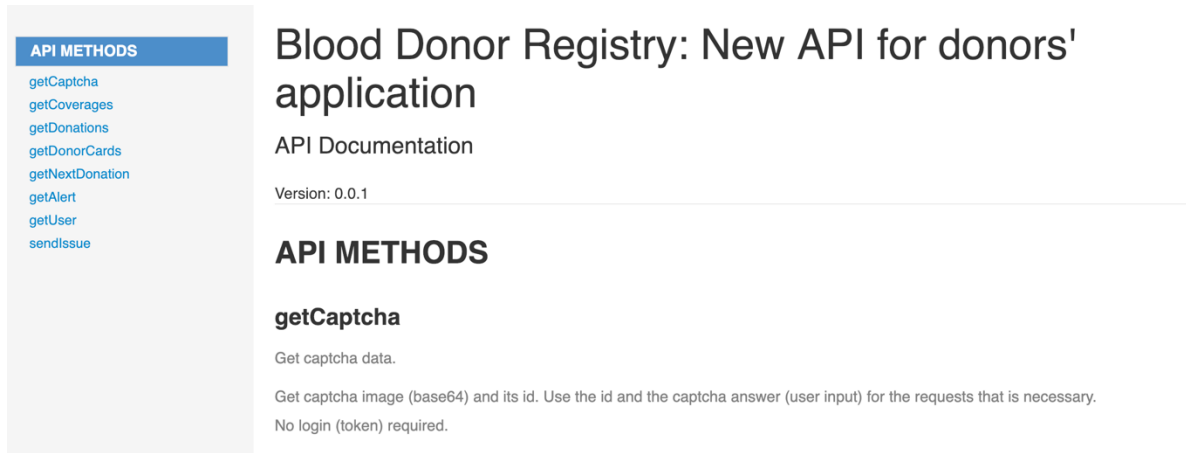
```
id:
  type: integer
status:
  type: string
  enum:
    - PENDING
    - SUBMITTED
    - APPROVED
    - SENT
    - READY
    - DELIVERED
    - LOST
    - REJECTED
    - CANCELLED
rejectionReason:
  type: string
  description: Not null only when status is REJECTED.
submittedAt:
  type: string
  format: date
donorCardID:
  type: integer
photo:
  type: string
  description: PNG image, encoded in Base64.
email:
  type: string
cellNumber:
  type: string
mothersName:
  type: string
fathersName:
  type: string
mothersNameLatin:
  type: string
```

```
fathersNameLatin:
  type: string
firstName:
  type: string
firstNameLatin:
  type: string
lastName:
  type: string
lastNameLatin:
  type: string
rhesus:
  type: string
majorBloodGroup:
  type: string
distributionPoint:
  $ref: '#/components/schemas/DistributionPoint'
CardApplications:
  type: array
  items:
    $ref: '#/components/schemas/CardApplication'
DistributionPoint:
  type: object
  properties:
    id:
      type: integer
    email:
      type: string
    street:
      type: string
    streetNumber:
      type: string
    zipcode:
      type: string
    visitingHours:
      type: string
```

```
    municipality:
      $ref: '#/components/schemas/Municipality'
    region:
      $ref: '#/components/schemas/Region'
  PreviousRecords:
    type: array
    items:
      $ref: '#/components/schemas/HospitalRecord'
  securitySchemes:
    token:
      type: apiKey
      name: X-Auth-Token
      in: header
```


Για να είναι ακόμα πιο εύκολη η κατανόηση του API αξιοποιήθηκε η δυνατότητα του OAS για την παραγωγή στατικής σελίδας, η οποία με εύληπτο και όμορφο τρόπο κάνει διαθέσιμες όλες τις σχεδιαστικές αποφάσεις. Η σελίδα αυτή είναι διαθέσιμη ηλεκτρονικά στη διεύθυνση <https://athinab.github.io/bdr-api-docs/>.

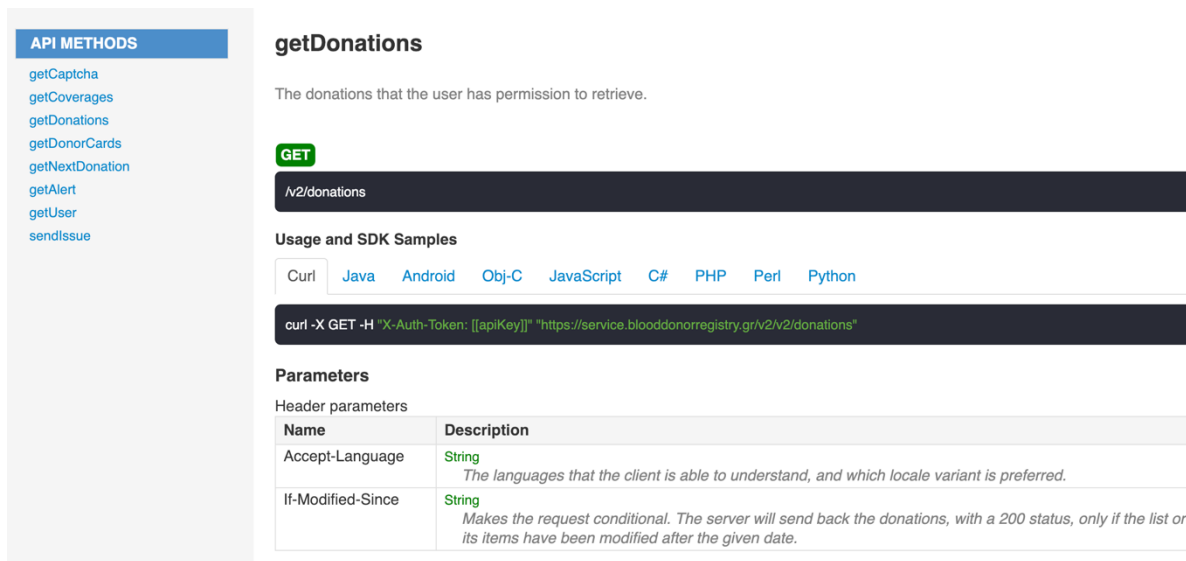
Παρατίθενται ορισμένα στιγμιότυπα της ιστοσελίδας:



The screenshot shows the 'API METHODS' section of the documentation. A sidebar on the left lists various methods: `getCaptcha`, `getCoverages`, `getDonations`, `getDonorCards`, `getNextDonation`, `getAlert`, `getUser`, and `sendIssue`. The main content area is titled 'Blood Donor Registry: New API for donors' application' and 'API Documentation'. Below this, it shows 'Version: 0.0.1' and another 'API METHODS' heading. The selected method is `getCaptcha`, with a description: 'Get captcha data.' and a note: 'Get captcha image (base64) and its id. Use the id and the captcha answer (user input) for the requests that is necessary. No login (token) required.'

Εικόνα 6-4 Επισκόπηση της ιστοσελίδας

Η ιστοσελίδα έχει την ακόλουθη δομή: αριστερά είναι διαθέσιμες οι δυνατές μέθοδοι που μπορεί να εκτελέσει ο client και δεξιά βρίσκεται το πλήρες περιεχόμενο κάθε μεθόδου.



The screenshot shows the 'getDonations' method page. The sidebar on the left is the same as in the previous image. The main content area is titled 'getDonations' and has a description: 'The donations that the user has permission to retrieve.' Below this, there is a 'GET' status indicator and the endpoint `/v2/donations`. Under 'Usage and SDK Samples', there are tabs for different languages: Curl, Java, Android, Obj-C, JavaScript, C#, PHP, Perl, Python. A code block shows a curl command: `curl -X GET -H 'X-Auth-Token: [[apiKey]]' "https://service.bloodonorregistry.gr/v2/v2/donations"`. The 'Parameters' section is titled 'Header parameters' and contains a table with two rows:

Name	Description
Accept-Language	String <i>The languages that the client is able to understand, and which locale variant is preferred.</i>
If-Modified-Since	String <i>Makes the request conditional. The server will send back the donations, with a 200 status, only if the list or its items have been modified after the given date.</i>

Εικόνα 6-5 Περιγραφή της δομής των requests και παραδείγματα εκτέλεσης τους με διάφορες γλώσσες και SDKs

API METHODS

- getCaptcha
- getCoverages
- getDonations
- getDonorCards
- getNextDonation
- getAlert
- getUser
- sendIssue

Responses

Status: 200 - OK

Schema [Response Example](#) [Headers](#)

```

▼ [
  ▼ {
    id: number
    collectedComponent: string
                        Enum: Blood, Platelets
    donatedAt: string (date)
    hospitalName: string
    occasion: string
              Enum: BLOODDONORASSOCIATION, MILITARY, RELATIVE, VOLUNTEER
  }
]

```

Status: 304 - Not Modified

Status: 401 - Unauthorized

Schema [Response Example](#)

```

▼ {
  error: string
}

```

Εικόνα 6-6 Περιγραφή και παραδείγματα των απαντήσεων των requests

7 Υλοποίηση

7.1 Χαρακτηριστικά εφαρμογής

Προτού γίνει η επιλογή των τεχνολογιών για την υλοποίηση της εφαρμογής, συγκεντρώθηκαν στοιχεία από την αντίστοιχη διαδικτυακή εφαρμογή και προέκυψαν ορισμένες σχεδιαστικές αποφάσεις για την οργάνωση της πληροφορίας και των λειτουργιών. Η διαδικασία αυτή περιλάμβανε τον εντοπισμό των αναγκών του χρήστη και μία ποιοτική ανάλυση χαρακτηριστικών που καλείται να έχει η διεπαφή της εφαρμογής για κινητά. Ο λόγος που έγινε αυτή η ανάλυση πριν την επιλογή των τεχνολογιών είναι για να είναι δυνατός ο εντοπισμός των τεχνικών αναγκών της εφαρμογής και η διαμόρφωση των κατάλληλων κριτηρίων επιλογής.

Η πρώτη παρατήρηση που προέκυψε ήταν ότι η εφαρμογή περιλαμβάνει λίστες από ομάδες στοιχείων όπως είναι η λίστα αιμοδοσιών, καλύψεων, αιτήσεων για έκδοση ταυτότητας αιμοδότη κ.ο.κ. Οι λίστες περιλαμβάνουν περιορισμένο αριθμό μελών, επομένως δεν απαιτείται καταμερισμός τους και σταδιακή εμφάνιση τους (pagination).

Ορισμένες ομάδες αποτελούνται από πολυάριθμα στοιχεία. Σε αυτές τις περιπτώσεις επιλέχθηκε να δοθεί στο χρήστη άμεση πρόσβαση στα αντιπροσωπευτικά στοιχεία της ομάδας και τα υπόλοιπα να είναι διαθέσιμα σε επόμενη οθόνη. Για παράδειγμα, δημιουργήθηκε μία λίστα με εμφανή κάποια από τα στοιχεία της κάθε αίτησης για ταυτότητα και μετά από επιλογή του χρήστη παρουσιάζονται τα πλήρη της στοιχεία.

Προκειμένου ο χρήστης να ξεχωρίσει εύκολα κάθε ομάδα-μέλος από τη λίστα που βλέπει, επιλέχθηκαν αντιπροσωπευτικά εικονίδια και χρωματικός κώδικας. Παραδείγματα χρήσης αυτών των δύο, είναι τα διαφορετικά εικονίδια που τοποθετήθηκαν στο μενού πλοήγησης και ο χρωματισμός της αίτησης για ταυτότητα, με βάση το στάδιο στο οποίο βρίσκεται.

Πέρα από τις παραπάνω οθόνες, η εφαρμογή έχει φόρμες που είτε ανανεώνουν ήδη υπάρχοντα στοιχεία είτε προσθέτουν νέα στοιχεία στο διακομιστή.

Επίσης, παρατηρήθηκε ότι κατά τη λειτουργία της εφαρμογής δεν απαιτούνται πολύπλοκοι υπολογισμοί που θα χρειάζονταν πολύτιμους υπολογιστικούς πόρους από τη συσκευή του χρήστη. Επιπλέον, οι οθόνες της εφαρμογής δεν έχουν ανάγκη από τη διάταξη πολλών στοιχείων ή μεγάλων και απαιτητικών πολυμέσων, που θα μπορούσαν να βαρύνουν τη σχεδίαση μίας οθόνης.

Η εφαρμογή χρειάζεται και τη λειτουργικότητα της λήψης φωτογραφίας και την αποστολή της στο διακομιστή για την υποστήριξη της διαδικασίας υποβολής αίτησης για ταυτότητα εθελοντή αιμοδότη.

Η προς υλοποίηση εφαρμογή θα λειτουργεί σε Android λειτουργικό. Αυτό επηρεάζει τον τρόπο διάταξης ορισμένων στοιχείων των οθονών όπως παραδείγματος χάριν το μενού πλοήγησης, το οποίο τοποθετήθηκε στα αριστερά σε μία περιοχή που ανοιγοκλείνει με συγκεκριμένες χειρονομίες (navigation drawer).

Η εφαρμογή θα είναι καλό να είναι διαθέσιμη τουλάχιστον στα ελληνικά και στα αγγλικά ώστε να είναι δυνατή η χρήση της ανεξάρτητα από την γλώσσα του αιμοδότη.

Ένα άλλο πολύ σημαντικό σημείο για μία εφαρμογή τέτοιου είδους είναι η διασφάλιση της ασφάλειας των στοιχείων του χρήστη.

Τέλος, κατά την αναζήτηση των κατάλληλων τεχνολογιών για την υλοποίηση της εφαρμογής θα ήταν καλό να ληφθεί υπόψιν ότι στο μέλλον θα χρειαστεί να εξυπηρετηθούν και οι ανάγκες χρηστών που χρησιμοποιούν iOS συστήματα.

7.2 Διαθέσιμες τεχνολογίες

Για την ανάπτυξη της εφαρμογής εξετάστηκαν τρεις εναλλακτικές γλώσσες και τα αντίστοιχα οικοσυστήματα, που θα μπορούσαν να χρησιμοποιηθούν:

- Java
- Υβριδικές τεχνολογίες όπως Apache Cordova, PhoneGap
- React Native

Οι παραπάνω επιλογές εξετάστηκαν με βάση την ευκολία υλοποίησης, τη δυνατότητα υλοποίησης των ζητούμενων λειτουργιών αποδοτικά, την καθαρότητα του κώδικα, το χρόνο εκμάθησης και την επαναχρησιμοποίηση του κώδικα για άλλο λειτουργικό.

7.2.1 Java για την υλοποίηση Android εφαρμογών

Το Android είναι λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας το οποίο τρέχει τον πυρήνα του λειτουργικού Linux. Επιτρέπει τη δημιουργία εφαρμογών με τη χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών λογισμικού που διανέμονται κυρίως μέσω της Google.

Οι εφαρμογές που χρησιμοποιούνται σε μία συσκευή που εκτελεί το λειτουργικό Android γράφονται χρησιμοποιώντας το κιτ ανάπτυξης λογισμικού Android (SDK) και, συχνά, τη γλώσσα προγραμματισμού Java. Η Java μπορεί να συνδυαστεί με C/C++. Επίσης, τα τελευταία χρόνια το Android υποστηρίζει την υλοποίηση εφαρμογών με χρήση της γλώσσας Go και της γλώσσας Kotlin.

Το Android SDK περιλαμβάνει ένα πλήρες σύνολο εργαλείων ανάπτυξης, συμπεριλαμβανομένου ενός προγράμματος εντοπισμού σφαλμάτων, βιβλιοθηκών λογισμικού, ενός εξομοιωτή βασισμένου σε QEMU, τεκμηρίωσης, δείγματα κώδικα και μαθήματα. Αρχικά, το υποστηριζόμενο ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) της Google ήταν το Eclipse χρησιμοποιώντας το πρόσθετο εργαλείων ανάπτυξης Android (ADT). Τον Δεκέμβριο του 2014, η Google κυκλοφόρησε το Android Studio, βασισμένο στην IntelliJ IDEA, ως πρωταρχικό IDE για ανάπτυξη εφαρμογών Android. Άλλα εργαλεία ανάπτυξης είναι διαθέσιμα, συμπεριλαμβανομένου ενός κιτ φυσικής ανάπτυξης (NDK) για εφαρμογές ή επεκτάσεις σε C ή C++, Google App Inventor, ένα οπτικό περιβάλλον για αρχάριους προγραμματιστές και διάφορες πλατφόρμες εφαρμογών διαδικτυακών εφαρμογών [29].

7.2.2 Υβριδικές τεχνολογίες για εφαρμογές Android και iOS

Μία επιλογή για την υλοποίηση εφαρμογών για κινητές συσκευές είναι η χρήση υβριδικών τεχνολογιών, όπως το PhoneGap και το Cordova. Με αυτές τις τεχνολογίες μπορεί να γραφτεί ο κώδικας της εφαρμογής με HTML, JavaScript και CSS. Το PhoneGap ή το Cordova αναλαμβάνει και εκτελεί αυτόν τον κώδικα μέσα σε WebViews. Τα WebViews είναι components που προσφέρονται από τα λειτουργικά Android και iOS και επιτρέπουν την λειτουργία ιστοσελίδων και web εφαρμογών μέσα από την εφαρμογή και όχι μέσα από τον browser [30], [31].

Το καλό με αυτές τις τεχνολογίες είναι η επαναχρησιμοποίηση του κώδικα. Δεν χρειάζεται να αναπτυχθούν οι ίδιες λειτουργικότητες σε διαφορετικές γλώσσες ώστε να εξυπηρετήσουν τα διαφορετικά λειτουργικά συστήματα που χρησιμοποιούν οι χρήστες.

Το κύριο μειονέκτημα είναι το κόστος λειτουργίας τέτοιων εφαρμογών. Η εκτέλεση της JavaScript και όλη η διαχείριση της διεπαφής γίνεται πάνω σε ένα thread και έτσι προκύπτει ο κίνδυνος η απόκριση της εφαρμογής να είναι χαμηλή, ιδιαίτερα σε περιπτώσεις πολύπλοκων διεπαφών ή πιο δαπανηρών υπολογιστικών διαδικασιών. Ένα άλλο σημείο που χρειάζεται προσοχή είναι η εμφάνιση τέτοιων εφαρμογών ώστε να είναι εναρμονισμένη με το λειτουργικό στο οποίο λειτουργεί και επομένως να είναι γνώριμη για το χρήστη. Τέλος, ένα άλλο μειονέκτημα είναι ότι σε σχέση με τις native εφαρμογές, οι υβριδικές εφαρμογές είναι μεγαλύτερες σε μέγεθος .

7.2.3 React Native για εφαρμογές Android και iOS

Το React Native ακολουθεί μία διαφορετική προσέγγιση για υβριδικές εφαρμογές. Συνδυάζει το development native εφαρμογών με το development διεπαφών με JavaScript. Με αυτόν τον τρόπο, λειτουργίες που έχουν μεγάλες υπολογιστικές απαιτήσεις ή που είναι στενά συνδεδεμένες με το λειτουργικό σύστημα μπορούν να υλοποιηθούν ως native modules για iOS ή Android, ενώ οι υπόλοιπες λειτουργίες μπορούν να υλοποιηθούν σε JavaScript και να τρέχουν και στα δύο εν λόγω λειτουργικά συστήματα.

Σε αντίθεση με το PhoneGap και το Cordova, τα οποία χρησιμοποιούνται για την υλοποίηση cross-platform εφαρμογών, το React Native δεν χρησιμοποιεί Web Views. Αντί το React Native να χρησιμοποιεί Web Views, προσφέρει πρόσβαση στους native mobile controllers, καταφέροντας οι εφαρμογές που υλοποιούνται με αυτό να έχουν αποδοτικότερη λειτουργία και επιπλέον δίνουν την αίσθηση native εφαρμογής στους χρήστες.

7.3 Επιλεγμένες τεχνολογίες και εργαλεία

Λαμβάνοντας όλα τα παραπάνω και διερευνώντας τις τεχνολογίες που χρησιμοποιούνται αυτήν την περίοδο στο χώρο της πληροφορικής κάναμε τις παρακάτω επιλογές:

Η εφαρμογή αυτή υλοποιήθηκε με χρήση του συστήματος Git. Το Git είναι ένα σύστημα ελέγχου εκδόσεων (λέγεται και σύστημα ελέγχου αναθεωρήσεων ή σύστημα ελέγχου πηγαίου κώδικα) με έμφαση στην ταχύτητα και στην ακεραιότητα των δεδομένων καθώς και στην υποστήριξη για καταναμημένες μη γραμμικές ροές εργασίας. Το Git άρχισε να αναπτύσσεται το 2005 για να καλύψει τις ανάγκες της ανάπτυξης του πυρήνα Linux και έχει γίνει από τότε το πιο διαδεδομένο σύστημα ελέγχου εκδόσεων για ανάπτυξη λογισμικού. [32]

Η κύρια γλώσσα που χρησιμοποιήθηκε για τη υλοποίηση είναι η ECMAScript (JavaScript). Προκειμένου να λειτουργεί η JavaScript σε κινητές συσκευές χρησιμοποιήθηκε η βιβλιοθήκη React Native, η οποία «αναλαμβάνει» την επικοινωνία με το API του λειτουργικού συστήματος. Επιπλέον, η εφαρμογή αναπτύχθηκε με χρήση Node.js και npm. Το πρώτο δίνει τη δυνατότητα να λειτουργεί η JavaScript έξω από τον περιηγητή του διαδικτύου (browser) [33], [34] και το δεύτερο χρησιμεύει στη διαχείριση και οργάνωση των βιβλιοθηκών που χρησιμοποιούνται κατά την ανάπτυξη μίας JavaScript εφαρμογής [35]. Ο κώδικας που γράφεται βασίζεται πάνω στη βιβλιοθήκη React [36]. Η βιβλιοθήκη React καθορίζει τη μορφή του κώδικα και προσφέρει δυνατότητες που σχετίζονται με την διαμόρφωση των οθονών και τις λειτουργίες που διατίθενται στο χρήστη μέσα από αυτές. Για τη διαχείριση των πληροφοριών που αποθηκεύει τοπικά η εφαρμογή χρησιμοποιήθηκε η βιβλιοθήκη Redux [37], η οποία προσφέρει εργαλεία για καλύτερη διαχείριση της ροής της πληροφορίας μεταξύ

ανεξάρτητων components του React. Η υλοποίηση στηρίχθηκε και στη βιβλιοθήκη Flow [38], η οποία εκτελεί στατικό έλεγχο του κώδικα και στη βιβλιοθήκη ESLint [39], που στοχεύει στην εξασφάλιση ενός συνεπούς στυλ γραφής κώδικα και στην αποφυγή χρήσης της γλώσσας με επισφαλή τρόπο. Τέλος, προκειμένου να γραφτούν tests, που να διασφαλίζουν την ορθή λειτουργία της εφαρμογής κατά τη διάρκεια της ενίσχυσης των λειτουργιών της, χρησιμοποιήθηκε η βιβλιοθήκη Jest [40].

7.4 React και React Native

Το React είναι μία βιβλιοθήκη γραμμένη σε JavaScript. Υλοποιήθηκε από τους μηχανικούς του Facebook, οι οποίοι στόχευαν στη δημιουργία δυναμικών διεπαφών, υψηλής απόδοσης και ταχύτατης απόκρισης. Προκειμένου να επιτύχουν αυτό το στόχο και ταυτόχρονα να βελτιστοποιήσουν την ίδια τη διαδικασία υλοποίησης αποφάσισαν να εισάγουν την XHP μέσα στο σύστημα διαχείρισης της JavaScript. Η XHP είναι η γλώσσα σήμανσης που έχει σχεδιασθεί και αυτή από το Facebook.

Το αποτέλεσμα αυτής της προσπάθειας έγινε γνωστό ευρέως το 2011, με τη δημοσίευση του React. Το React χρησιμοποιεί ένα νέο μηχανισμό για το rendering των ιστοσελίδων, που επιτυγχάνει ανανέωση του περιεχομένου που παρουσιάζεται στο χρήστη και απόκριση στις ενέργειες του γρήγορα και αποδοτικά. Η βιβλιοθήκη έγινε λογισμικό ανοιχτού κώδικα το 2013 και εκ τότε η δημοτικότητα και η χρήση της αυξάνεται με μεγάλους ρυθμούς.

Δύο χρόνια αργότερα η ίδια ομάδα μηχανικών έδωσε στη δημοσιότητα το React Native, το οποίο παρουσίασε μία νέα προσέγγιση στις υβριδικές εφαρμογές, οι οποίες λειτουργούν σε περιβάλλον iOS και Android. Το React Native χρησιμοποιεί παρόμοιες αρχές με το React και έγινε αποδεκτό αντίστοιχα σύντομα από την κοινότητα των μηχανικών όπως και από ένα ευρύ κύκλο εταιριών. Μερικά χαρακτηριστικά παραδείγματα εφαρμογών που αξιοποίησαν το React Native και έχουν μεγάλο αριθμό χρηστών είναι [41]:

- Facebook
- Facebook Analytics
- Microsoft Office
- Microsoft Outlook
- Instagram
- Shopify
- Discord
- Bloomberg
- Skype
- Tesla
- Pinterest

7.4.1 Αρχιτεκτονική

Όπως αναφέρθηκε και νωρίτερα, το React Native δεν λειτουργεί με Web Views όπως το PhoneGap ή το Cordova. Αντίθετα, οι React Native εφαρμογές έχουν άμεση πρόσβαση στα APIs και views που διαθέτει το λειτουργικό σύστημα, πάνω στο οποίο λειτουργούν, με στόχο την αποδοτική τους λειτουργία και την αίσθηση ότι η εφαρμογή είναι native.

Αυτός ο τρόπος λειτουργίας δεν επιτυγχάνεται με μεταγλώττιση της JavaScript σε Java και Objective C. Αυτό που υλοποιεί το React Native είναι η αναπαράσταση των React components σε native views και components. Π.χ. το native component TextInput έχει ένα αντίστοιχο React Native component και έτσι μπορεί να χρησιμοποιηθεί από JavaScript κώδικα όπως κάθε άλλο React component που χρησιμοποιείται σε μία διαδικτυακή εφαρμογή.

Πιο αναλυτικά, η πλατφόρμα React Native χωρίζεται στα εξής 3 τμήματα:

1. Native κώδικας και modules

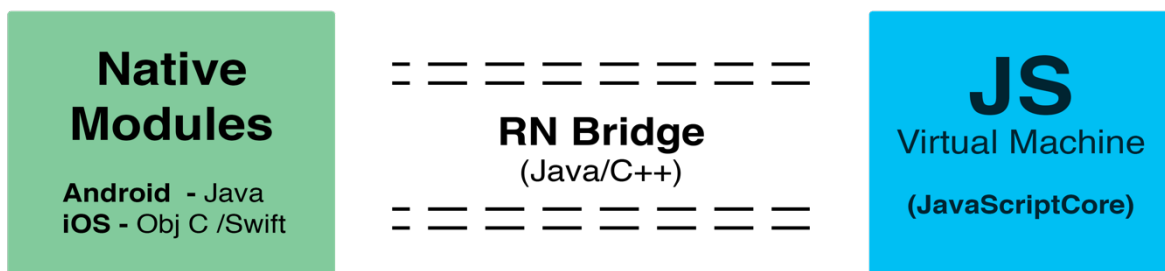
Ο native κώδικας είναι γραμμένος σε Objective C ή Swift αν πρόκειται για εφαρμογή που λειτουργεί σε λειτουργικό iOS ή σε Java αν η εφαρμογή λειτουργεί σε Android.

2. Εικονική μηχανή JavaScript

Στο εικονικό μηχανήμα λειτουργεί όλος ο JavaScript κώδικας. Στις συσκευές που τρέχουν React Native εφαρμογές χρησιμοποιείται η JavaScriptCore, που είναι το engine που χρησιμοποιείται από τον φυλλομετρητή Safari. Η JavaScriptCore υπάρχει στο iOS (μετά την έκδοση 7), οπότε οι React Native εφαρμογές μπορούν να την αξιοποιήσουν, ενώ διαφορετικά εμπεριέχεται στο πακέτο της εφαρμογής. Το αρνητικό σε αυτήν την περίπτωση είναι ότι το μέγεθος της εφαρμογής αυξάνεται ακόμα και κατά 4 MB.

3. Γέφυρα (React Native bridge)

Η γέφυρα είναι γραμμένη σε C++/Java και μεταφέρει μηνύματα μεταξύ του native και του JavaScript thread. Όταν ο προγραμματιστής εκτελέσει τις εντολές που «χτίζουν» και «πακετάρουν» τον κώδικα που έγραψε, θα συγκεντρωθεί όλος ο JavaScript κώδικας σε ένα αρχείο. Όταν η εφαρμογή εκκινήσει το πρώτο πράγμα που θα φορτωθεί θα είναι το native σημείο έναρξης. Τότε το native thread θα δημιουργήσει (spawn) το JavaScript νήμα (thread), που θα εκτελέσει τον κώδικα JavaScript. Μόλις ξεκινήσει να εκτελείται ο JavaScript κώδικας αρχίζει η αποστολή οδηγιών στο native thread μέσω της γέφυρας. Οι οδηγίες καθορίζουν ποια views χρειάζεται να εμφανιστούν στην οθόνη της συσκευής, ποιες πληροφορίες χρειάζεται να αντληθούν από το hardware κ.α. Παραδείγματος χάριν, το JavaScript thread θέλει να εμφανιστεί ένα view και ένα κείμενο, οπότε στέλνει ένα μήνυμα που περιγράφει όλα αυτά που χρειάζεται. Το native thread θα εκτελέσει αυτά που ζητούνται και θα ενημερώσει το JavaScript thread για όποιες πράξεις έχει ολοκληρώσει.



Εικόνα 7-1 Τα τρία βασικά συστατικά του React Native

Πιο αναλυτικά, για την εφαρμογή το React Native θα δημιουργήσει 3 threads:

1. Κύριο thread (Native)

Αυτό είναι το thread που δημιουργείται κατά την έναρξη της εφαρμογής. Φορτώνει τον κώδικα της εφαρμογής και εκκινεί το thread που εκτελεί τον JavaScript κώδικα.

Επιπλέον, αυτό το thread περιμένει, ή αν θέλετε «ακροάζεται», τη διεπαφή ώστε να αντιληφθεί και να καταγράψει τις κινήσεις του χρήστη σε αυτήν. Αφού εντοπίσει κάποια κίνηση του χρήστη πάνω στη διεπαφή ενημερώνει το JavaScript thread μέσω της γέφυρας. Όταν η JavaScript λάβει την ενημέρωση στέλνει τι χρειάζεται να εμφανιστεί στην οθόνη. Αυτήν την πληροφορία τη χρησιμοποιεί ένα «shadow node thread», το οποίο υπολογίζει τη διάταξη των στοιχείων που θα προβληθούν στην οθόνη. Το «shadow node thread» λειτουργεί σαν μία μαθηματική μηχανή που υπολογίζει τις θέσεις των views. Αυτές οι οδηγίες αποστέλλονται στο native thread που θα εμφανίσουν τα ζητούμενα views με τον ζητούμενο τρόπο.

2. JavaScript thread

Το JavaScript thread είναι αυτό που περιέχει όλη τη νοοτροπία, τη λογική και τους κανόνες της εκάστοτε εφαρμογής (business logic).

3. Custom native modules

Πέρα από τα threads που εκκινεί το React Native είναι δυνατό να υλοποιήσουν οι προγραμματιστές custom native modules, τα οποία λειτουργούν παράλληλα με άλλα threads, επιτυγχάνοντας την πιο αποδοτική και γρήγορη λειτουργία της εφαρμογής. Χαρακτηριστικό παράδειγμα τέτοιας λειτουργίας είναι τα κινούμενα γραφικά τα οποία λειτουργούν σε ξεχωριστό thread ώστε να μην επιβαρύνεται αυτό στο οποίο εκτελείται ο JavaScript κώδικας [42] [43].

Το React Native είναι σε μία διαδικασία συνεχούς ανάπτυξης και βελτίωσης. Μία από τις νεότερες και μεγάλες αλλαγές σχετίζεται με την αρχιτεκτονική της διαχείρισης της διεπαφής. Η νέα προσθήκη, το Fabric, επιτρέπει την εκτέλεση ενημερώσεων που έχουν υψηλή προτεραιότητα, στη διεπαφή με έναν σύγχρονο μηχανισμό.

7.4.2 Πλεονεκτήματα και μειονεκτήματα React Native

Στη συνέχεια ακολουθούν ορισμένα από τα πλεονεκτήματα του React Native που εντοπίστηκαν κατά τη διάρκεια έρευνας αλλά και κατά την υλοποίηση της εφαρμογής [44] [45].

1. Επαναχρησιμοποιούμενα αυτόνομα components

Κύριο χαρακτηριστικό της αρχιτεκτονικής του React, που υιοθέτησε και το React Native, είναι η δυνατότητα να επαναχρησιμοποιηθεί κώδικας σε πολλά σημεία εύκολα. Αυτό γίνεται με τη δημιουργία components. Κάθε component είναι ένα ανεξάρτητο σύνολο κώδικα. Με χρήση αυτού του μηχανισμού, όταν τροποποιηθεί ο κώδικας ενός component δεν επηρεάζει τον κώδικα σε άλλα σημεία, εφ' όσον δεν έχει τροποποιηθεί η διεπαφή που παρουσιάζει προς τα έξω.

2. Ροή δεδομένων προς μία κατεύθυνση (one-direction data flow)

Η ροή των δεδομένων στο React έχει κατεύθυνση από τα parent components στα παιδιά τους και όχι το αντίθετο. Αυτό σημαίνει ότι οποιαδήποτε αλλαγή προκύψει στην κατάσταση του παιδιού component δεν θα επηρεάσει το parent component.

3. Προγραμματισμός με JavaScript

Η JavaScript είναι μία γλώσσα προγραμματισμού που χρησιμοποιείται από πολύ μεγάλο ποσοστό προγραμματιστών. Αυτή τη στιγμή η JavaScript χρησιμοποιείται όχι μόνο για την υλοποίηση web clients αλλά και στην υλοποίηση του backend πολλών υπηρεσιών. Επομένως, με το React Native δίνεται η δυνατότητα σε όλο αυτό τον

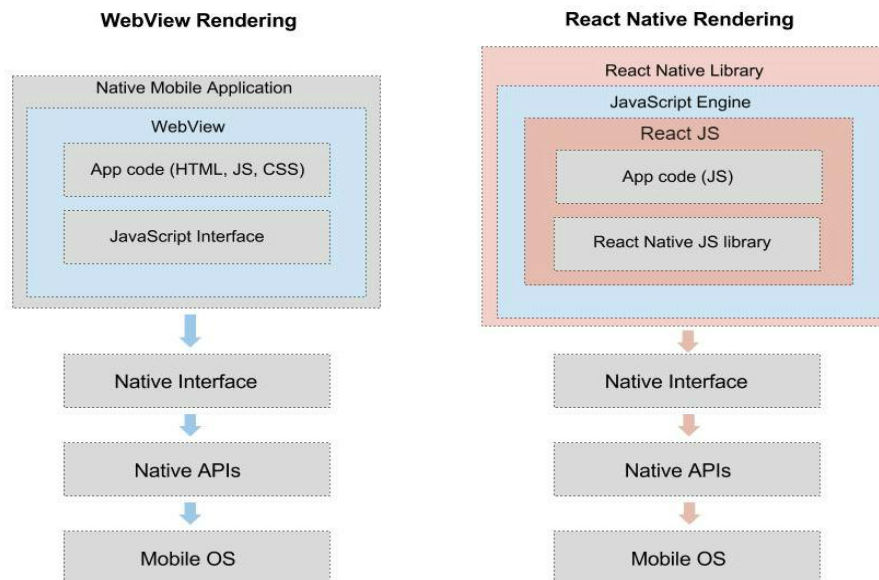
κόσμο να αξιοποιήσει τις γνώσεις του και να υλοποιήσει mobile clients σε λιγότερο χρόνο.

4. Γρήγορη εποπτεία των αλλαγών κατά την ανάπτυξη

Ο προγραμματιστής δεν χρειάζεται να ακολουθεί χρονοβόρες διαδικασίες με επανεκκινήσεις και επαναφορτώσεις για να δει κάθε αλλαγή που πραγματοποιεί. Μία απλή ανανέωση, όπως αυτή που πραγματοποιεί για την ανανέωση μίας ιστοσελίδας στο φυλλομετρητή είναι αρκετή και σε ορισμένες περιπτώσεις δεν είναι καν απαραίτητη καθώς πολλές φορές οι αλλαγές εμφανίζονται αυτόματα.

5. Δυνατότητα χρήσης native modules και native controls

Όπως προαναφέρθηκε, το React Native δεν κάνει render τον κώδικα με χρήση WebView αλλά αξιοποιεί τα native APIs του εκάστοτε λειτουργικού. Ο λόγος που επιλέχθηκε μία τέτοια προσέγγιση από τους μηχανικούς του React Native σχετίζεται με την απόδοση, καθώς το WebView είναι πολύ κοστοβόρος μηχανισμός. Η λύση που δόθηκε για αυτό είναι το React Native να επικοινωνεί με συγκεκριμένα native components του iOS και Android.



Εικόνα 7-2 Μηχανισμός εμφάνισης στοιχείων με χρήση WebViews σε αντιπαραβολή με τη χρήση React Native

Το React Native προσφέρει τη δυνατότητα χρήσης των native modules, χωρίς ο ίδιος ο προγραμματιστής να γράψει σε native γλώσσα, αλλά αρκεί να χρησιμοποιήσει τη JavaScript και το API που προσφέρει το React Native. Στις περιπτώσεις όμως που απαιτείται η υλοποίηση μίας λειτουργικότητας που έχει μεγαλύτερες υπολογιστικές απαιτήσεις κατά την εκτέλεση της, μπορεί να υλοποιήσει αυτή τη λειτουργικότητα σε native κώδικα ή εάν υπάρχει ήδη υλοποιημένη αυτή η λειτουργικότητα να την ενσωματώσει και αυτή να συλλειτουργήσει με τον JavaScript κώδικα.

6. Επαναχρησιμοποίηση κώδικα σε διαφορετικά λειτουργικά και περιβάλλοντα

Το React Native προσφέρει κώδικα που λειτουργεί και σε Android και iOS χωρίς κάποια τροποποίηση κατά την ανάπτυξη. Επειδή αυτά τα λειτουργικά έχουν διαφορετικά APIs καθώς και διαφορετικό τρόπο προτεινόμενης παρουσίασης των

εφαρμογών δεν είναι δυνατή η επαναχρησιμοποίηση του συνόλου του κώδικα. Όμως υπολογίζεται ότι μπορεί να ανέλθει στο 80%. Για παράδειγμα η εφαρμογή του Facebook «Ads Manager» χρησιμοποιεί το 87% του κώδικα και για το Android και για το iOS. Τέλος, υπάρχει η δυνατότητα τροποποίησης του κώδικα ώστε να μπορεί να χρησιμοποιηθεί για λειτουργία του και μέσω του περιηγητή ιστού.

7. Το React Native περιλαμβάνει στοιχεία και λειτουργικότητες του React

Με αυτόν τρόπο οι προγραμματιστές μπορούν να αξιοποιήσουν τη JavaScript προγραμματίζοντας με δηλωτικό τρόπο. Το React δίνει τη δυνατότητα χρήσης components, τα οποία έχουν και συγκεκριμένα στάδια κατά τη διάρκεια της ζωής τους και ο προγραμματιστής με ευκολία μπορεί να συνδέσει τις λειτουργικότητες που επιθυμεί με αυτά τα στάδια.

8. Οι εφαρμογές που αναπτύσσονται με το React Native δίνουν την αίσθηση στο χρήστη ότι έχουν υλοποιηθεί με native τεχνολογίες.

Αυτό είναι ένα σημαντικό προτέρημα των εφαρμογών που έχουν αναπτυχθεί με χρήση του React Native, καθώς οι χρήστες τους νιώθουν ότι η εφαρμογή έχει χαρακτηριστικά και λειτουργίες που έχουν δει και σε άλλες εφαρμογές.

9. Αποδοτική και γρήγορη λειτουργία των υλοποιημένων εφαρμογών

Επειδή η λογική της εφαρμογής (business logic) είναι συγκεντρωμένη στο JavaScript thread αλλά το κύριο thread είναι υπεύθυνο για την παρουσίαση των οθονών της εφαρμογής έχουμε αποδοτικότερη χρήση των υπολογιστικών πόρων.

10. Αξιοποίηση εργαλείων JavaScript για στατική ανάλυση του κώδικα

Με την στατική ανάλυση του κώδικα μπορούν να αποφευχθούν συνήθη προγραμματιστικά λάθη και να επιτευχθεί ομοιόμορφο στυλ κώδικα.

11. Ανοιχτό λογισμικό

Ο κώδικας του React Native είναι πλήρως ανοιχτός και διαθέσιμος σε κάθε ενδιαφερόμενο, όπως και πολλές βιβλιοθήκες και εργαλεία που αναπτύσσονται με τη χρήση του.

Τα μειονεκτήματα που εντοπίστηκαν είναι τα ακόλουθα:

1. Υψηλοί ρυθμοί εξέλιξης

Τα τελευταία χρόνια η ανάπτυξη του κώδικα του React Native αλλά και του React είναι ταχύτατη. Μέρα με τη μέρα προστίθενται νέες δυνατότητες και βελτιώνονται οι υπάρχουσες. Ως αποτέλεσμα, οι προγραμματιστές που εργάζονται με τις εν λόγω βιβλιοθήκες είναι απαραίτητο να είναι σε μία συνεχή διαδικασία μάθησης και προσαρμογής στις νέες δυνατότητές τους.

2. Αυξημένος βαθμός δυσκολίας για την ανάπτυξη απαιτητικών ή ιδιαίτερων εφαρμογών

Όταν μία εφαρμογή έχει μεγαλύτερες απαιτήσεις υπολογιστικών πόρων λόγω π.χ. πολύπλοκων υπολογισμών ή λόγω αυξημένων απαιτήσεων στη διαχείριση των ενεργειών του χρήστη με τη διεπαφή, ο προγραμματιστής καλείται να αναπτύξει τις πιο απαιτητικές λειτουργίες σε native κώδικα. Αυτό, όπως προαναφέρθηκε, έχει το θετικό ότι δεν περιορίζονται οι δυνατότητες υλοποίησης αλλά μπορεί ταυτόχρονα να

αυξήσει το βαθμό δυσκολίας της υλοποίησης. Ιδιαίτερα εάν ο προγραμματιστής έχει εμπειρία μόνο σε JavaScript οικοσυστήματα.

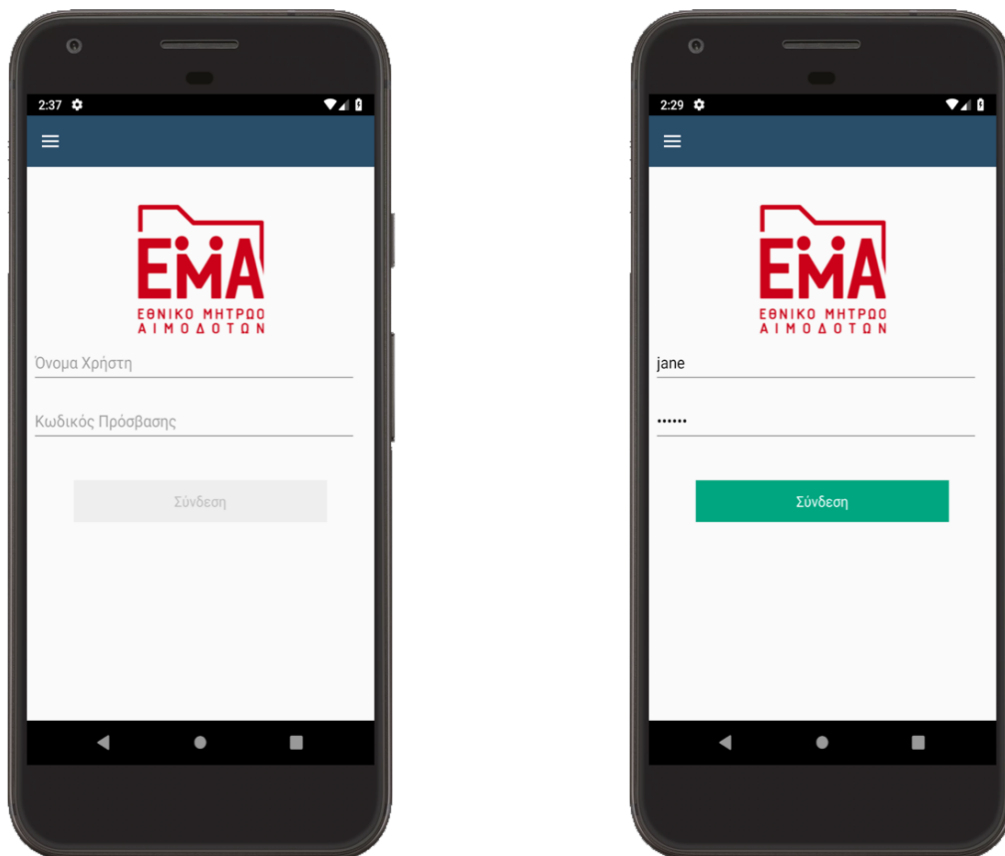
3. Δύσκολη ανάπτυξη βιβλιοθηκών

Προκειμένου να αναπτυχθούν βιβλιοθήκες που χρησιμοποιούνται με το React Native είναι χρήσιμο οι προγραμματιστές που τις υλοποιούν να έχουν καλή γνώση ανάπτυξης εφαρμογών για Android και για iOS και με React Native. Επειδή πολλές φορές οι γνώσεις σε κάποιο από τα τρία υστερούν, μπορεί η υλοποιημένη βιβλιοθήκη να έχει περισσότερα προβλήματα κατά την εκτέλεση της εφαρμογής σε κάποιο από τα λειτουργικά.

8 Παρουσίαση υλοποιημένης εφαρμογής

Η εφαρμογή κατά την εκκίνησή της παρουσιάζει στο χρήστη τη δυνατότητα σύνδεσης με το όνομα χρήστη και τον κωδικό πρόσβασης που έχει ορίσει κατά την εγγραφή του. Στις επόμενες εικόνες παρουσιάζεται η οθόνη σύνδεσης στην αρχική της κατάσταση αλλά και αφού συμπληρώσει ο χρήστης τα στοιχεία σύνδεσης του. Στις εν λόγω εικόνες, τα εξής στοιχεία σχεδιάστηκαν και υλοποιήθηκαν με στόχο την καλή εμπειρία χρήστη:

- Το κουμπί «Σύνδεση» είναι απενεργοποιημένο μέχρι ο χρήστης να συμπληρώσει όνομα χρήστη και κωδικό πρόσβασης. Έτσι, η εφαρμογή προσπαθεί να μην αφήνει περιθώρια λάθους στους χρήστες και για αυτό δεν επιτρέπει την προσπάθεια σύνδεσης εάν δεν συμπληρωθούν τα απαραίτητα στοιχεία.
- Υπάρχουν labels (επεξηγηματικά στοιχεία κειμένου), τα οποία καθοδηγούν το χρήστη στο να συμπληρώσει τα σωστά στοιχεία προκειμένου να συνδεθεί στο σύστημα του Εθνικού Μητρώου Αιμοδοτών.

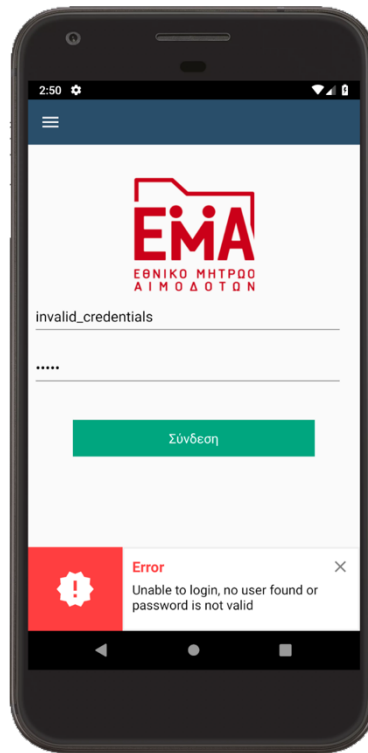


Εικόνα 8-1 Οθόνη σύνδεσης χρήστη σε αρχική κατάσταση και αφού εισάγει ο χρήστης τα στοιχεία του

Επιπλέον, εάν συμπληρώσει τα στοιχεία αλλά τα στοιχεία είναι λανθασμένα, ο χρήστης ενημερώνεται με την εμφάνιση ενός παραθύρου που περιλαμβάνει περιγραφή του λάθους που προέκυψε.

Το παράθυρο εμφάνισης σημαντικών πληροφοριών είναι ένα βοήθημα που χρησιμοποιείται σε όλες τις οθόνες της εφαρμογής ώστε να ενημερώνει το χρήστη είτε για σφάλματα που

προέκυψαν είτε για την επιτυχή ολοκλήρωση ενεργειών, όπως είναι για παράδειγμα η ενημέρωση για επιτυχή αποστολή μηνύματος στο γραφείο αρωγής χρηστών.



Εικόνα 8-2 Σφάλμα κατά την προσπάθεια σύνδεσης χρήστη

Επιπλέον, στις περισσότερες οθόνες, πάνω αριστερά, τοποθετήθηκε ένα εικονίδιο με τρεις οριζόντιες γραμμές. Αυτό είναι ένα χαρακτηριστικό εικονίδιο (hamburger icon), το οποίο λειτουργεί ως κουμπί που κατά την πίεση του εμφανίζει ένα πλαϊνό μενού πλοήγησης. Η θέση αυτού του κουμπιού, το εικονίδιο και η λειτουργικότητα που εκτελεί συνθέτουν μία πρακτική που καλούνται οι προγραμματιστές να την ακολουθήσουν όταν αναπτύσσουν εφαρμογές Android ώστε ο χρήστης να μπορεί εύκολα να προβλέψει πως θα πλοηγηθεί σε αυτές. Στην προκειμένη περίπτωση, ακόμα και χωρίς να συνδεθεί ο χρήστης μπορεί να πλοηγηθεί σε τρεις οθόνες οι οποίες του δίνουν τις εξής δυνατότητες:

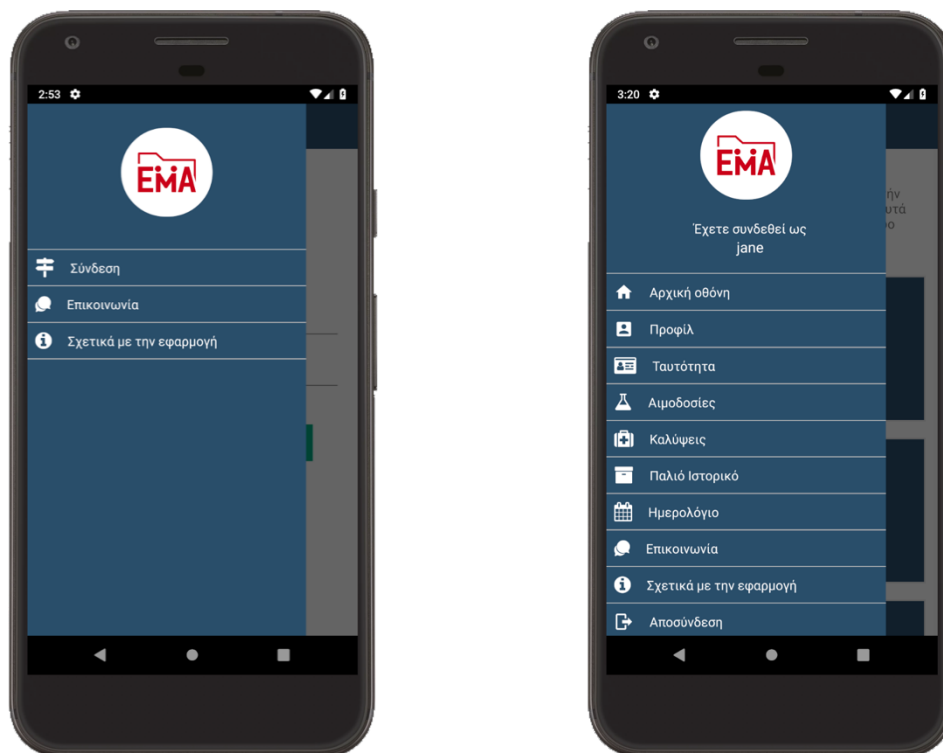
- Να επικοινωνήσει με το γραφείο αρωγής χρηστών
- Να δει περισσότερα στοιχεία σχετικά με την εφαρμογή, το σύνολο της υπηρεσίας καθώς και σχετικές νομικές πληροφορίες.
- Να συνδεθεί στην εφαρμογή

Σύμφωνα με όσα αναφέρονται και παραπάνω, το μενού πλοήγησης και το εικονίδιο είναι σταθερά στοιχεία της εφαρμογής, στα οποία έχει πρόσβαση ο χρήστης είτε είναι συνδεδεμένος, είτε όχι. Όταν όμως ο χρήστης είναι συνδεδεμένος, είναι απαραίτητο να έχει πρόσβαση σε περισσότερα στοιχεία. Τα επιπλέον στοιχεία που έχει πρόσβαση όταν συνδεθεί είναι:

- Το προφίλ του
- Οι αιτήσεις που έχει κάνει για έκδοση ταυτότητας αιμοδότη και τα πλήρη στοιχεία της ενεργής ταυτότητας του

- Το ιστορικό των αιμοδοσιών του
- Το ιστορικό των καλύψεων που έχει πραγματοποιήσει
- Την ημερομηνία της επόμενης δυνατής αιμοδοσίας και στοιχεία που σχετίζονται με τις ειδοποιήσεις που λαμβάνει ο χρήστης

Επίσης, από το πλευρικό μενού δίνεται και η δυνατότητα αποσύνδεσης.

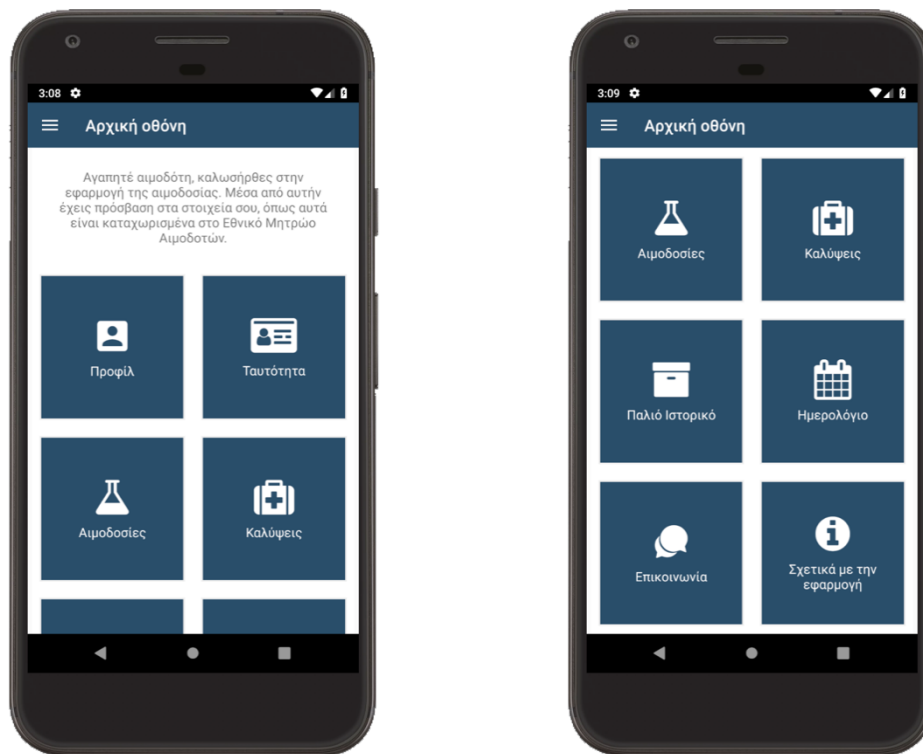


Εικόνα 8-3 Διαφορετικές επιλογές πλοήγησης ανάμεσα σε συνδεδεμένους και μη χρήστες

Μετά την είσοδο του ο χρήστης, οδηγείται στην αρχική οθόνη, όπου περιλαμβάνει ένα σύντομο μήνυμα καλωσορίσματος και όλες τις διαθέσιμες κατηγορίες πληροφοριών. Κάθε κατηγορία έχει αποτυπωθεί σε μία οθόνη που περιέχει τις αντίστοιχες πληροφορίες. Οι κατηγορίες εμφανίζονται στην αρχική οθόνη με τη μορφή κουμπιού, το οποίο αν το επιλέξει ο χρήστης, θα μεταφερθεί στην αντίστοιχη οθόνη.

Ο λόγος που δημιουργήθηκε η αρχική οθόνη με το παραπάνω περιεχόμενο είναι επειδή υπήρξε ως προτεραιότητα κατά το σχεδιασμό της εφαρμογής η εύκολη πρόσβαση στις διαθέσιμες πληροφορίες, ακόμα και αν ο χρήστης της εφαρμογής δεν είναι εξοικειωμένος με τη χρήση εφαρμογών κινητών τερματικών. Για το λόγο αυτό, παρατίθενται μπροστά του όλες οι πιθανές επιλογές με μεγάλα κουμπιά (το μέγεθος επιλέχθηκε ώστε να διευκολυνθεί το πάτημα τους), τα οποία περιέχουν το αντιπροσωπευτικό εικονίδιο και μία σύντομη περιγραφή για να είναι άμεσα κατανοητά. Όπως φαίνεται στην εικόνα που ακολουθεί, τα κουμπιά που παρατίθενται δίνουν τις ίδιες επιλογές στο χρήστη, όπως και το πλαϊνό μενού. Τα ίδια εικονίδια, οι ίδιες περιγραφές και τα ίδια χρώματα έχουν επιλεγεί ώστε το ανθρώπινο μυαλό να καταλάβει γρήγορα ότι μπορεί να εκτελέσει την ίδια λειτουργικότητα και από την οθόνη με τα μεγάλα κουμπιά αλλά και από τις επιλογές του πλαϊνού μενού.

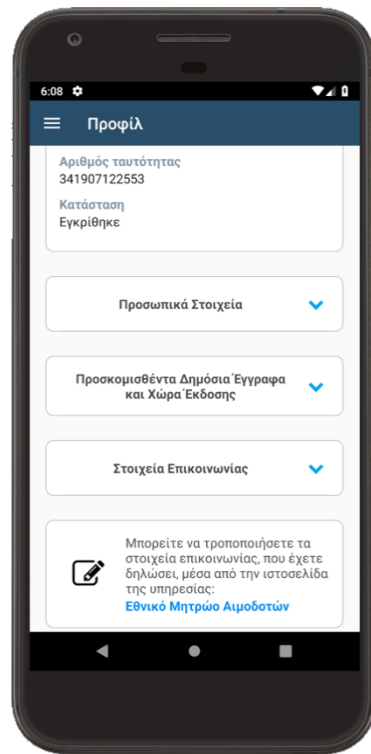
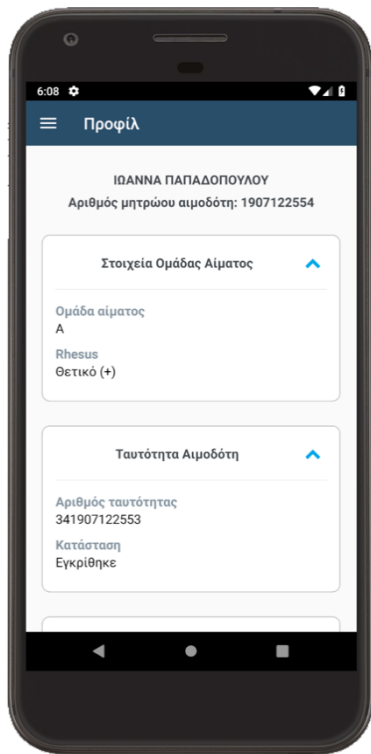
Σημειώνεται ότι στο πλαϊνό μενού υπάρχει ένα επιπλέον κουμπί στο τέλος τις λίστας. Πρόκειται για το κουμπί «Αποσύνδεση», το οποίο έχει τοποθετηθεί χαμηλά καθώς είναι μία λειτουργία που δεν χρησιμοποιείται συχνά από τους χρήστες. Επίσης, μετά από λίγες ώρες η εφαρμογή αποσυνδέει το χρήστη για λόγους ασφαλείας και έχει επιλεγθεί όταν ο χρήστης επιλέξει να τερματίσει τη λειτουργία της εφαρμογής επίσης να τον αποσυνδέει.



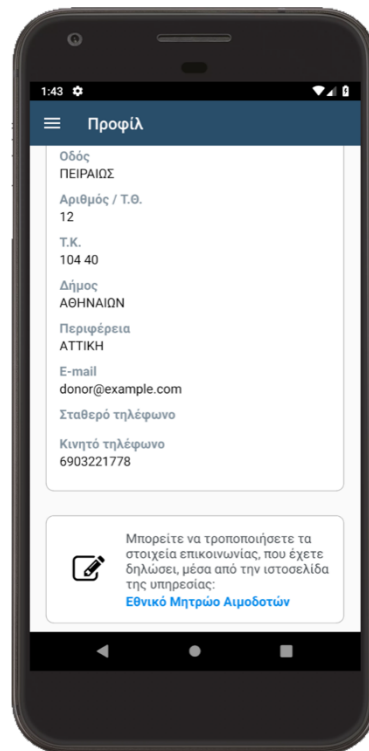
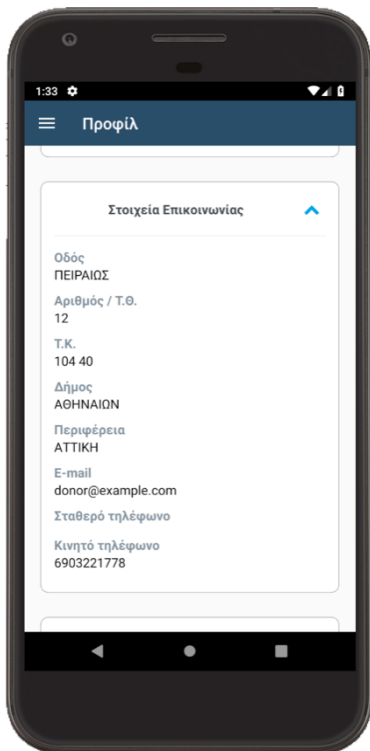
Εικόνα 8-4 Αρχική οθόνη

Στην οθόνη που παρουσιάζονται τα στοιχεία του προφίλ, στην επάνω πλευρά της οθόνης, έχει τοποθετηθεί με τρόπο που ξεχωρίζει, το ονοματεπώνυμο του χρήστη και ο αριθμός μητρώου του στο Εθνικό Μητρώο Αιμοδοτών, τα οποία είναι καίρια στοιχεία ταυτοποίησης.

Επειδή το προφίλ περιλαμβάνει μεγάλο πλήθος πληροφοριών, οι πληροφορίες έχουν ομαδοποιηθεί και διαχωριστεί σε κάρτες. Μετά τα στοιχεία ταυτοποίησης έχουν τοποθετηθεί τα στοιχεία αίματος του αιμοδότη και ορισμένα σημαντικά στοιχεία της ταυτότητας αιμοδότη. Κατά την κύλιση της οθόνης προς τα κάτω, παρουσιάζονται άλλες τέσσερις κάρτες. Στις τρεις που ακολουθούν είναι εμφανής μόνο ο τίτλος. Εάν ο χρήστης θέλει να δει τις αναλυτικές πληροφορίες που περιλαμβάνουν, μπορεί να επιλέξει το αντίστοιχο κουμπί που αντιπροσωπεύεται από ένα βέλος και να πραγματοποιήσει «ξετύλιγμα» της εκάστοτε κάρτας. Οι κάρτες περιέχουν τα προσωπικά στοιχεία του χρήστη, ποια έγγραφα έχει προσκομίσει για να ταυτοποιηθεί καθώς και τα στοιχεία επικοινωνίας που έχει δηλώσει στο Εθνικό Μητρώο Αιμοδοτών. Ένα από τα σενάρια χρήσης που μελετήθηκαν, αφορά την επιθυμία του χρήστη να τροποποιήσει τα στοιχεία επικοινωνίας του. Αυτό μπορεί να το πραγματοποιήσει με χρήση του φυλλομετρητή του. Για αυτό η τελευταία κάρτα, που βρίσκεται μετά τα στοιχεία επικοινωνίας, ενημερώνει το χρήστη για τον τρόπο που μπορεί να πραγματοποιήσει την επιθυμητή τροποποίηση και του δίνει και τον αντίστοιχο υπερσύνδεσμο ώστε να επιταχύνει τη διαδικασία.



Εικόνα 8-5 Προφίλ χρήστη



Εικόνα 8-6 Προφίλ χρήστη

Η επόμενη οθόνη παρουσιάζει τη λίστα με τις αιτήσεις για έκδοση ταυτότητας αιμοδότη που έχει πραγματοποιήσει ο χρήστης. Ο χρήστης μπορεί να μην έχει ακόμα πραγματοποιήσει αίτηση για την έκδοση ταυτότητας αιμοδότη ή να έχει κάνει μία ή ακόμα και περισσότερες αιτήσεις. Η τελευταία περίπτωση προκύπτει εάν κάποια αίτηση του χρήστη έχει απορριφθεί, ακυρωθεί ή χαθεί.

Στην οθόνη εμφανίζονται τα πιο σημαντικά στοιχεία της κάθε αίτησης/ταυτότητας. Συγκεκριμένα εμφανίζονται:

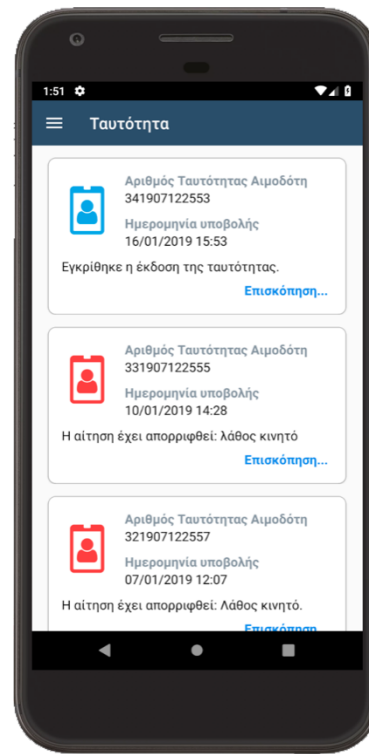
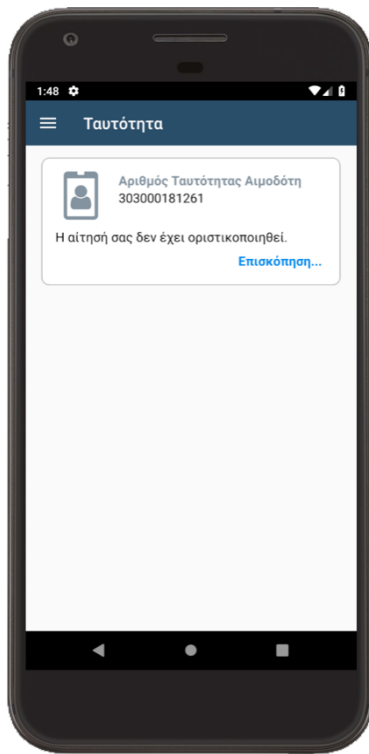
- Ο αριθμός ταυτότητας αιμοδότη
- Η ημερομηνία υποβολής της αίτησης, εκτός εάν η αίτηση έχει αποθηκευτεί προσωρινά και δεν έχει υποβληθεί
- Η κατάσταση στην οποία βρίσκεται η ταυτότητα.

Επιπλέον, έχει προστεθεί ένα εικονίδιο το οποίο χρωματίζεται ανάλογα με την κατάσταση στην οποία βρίσκεται η αίτηση. Οι επιλεγμένοι χρωματισμοί είναι:

- Γκρι όταν βρίσκεται σε εκκρεμότητα η υποβολή της
- Γαλάζιο όταν έχει υποβληθεί ή όταν έχει εγκριθεί ή έχει αποσταλεί προς εκτύπωση
- Πράσινο όταν είναι έτοιμη για παραλαβή ή έχει παραληφθεί
- Πορτοκαλί όταν έχει χαθεί ή ακυρωθεί
- Κόκκινο εάν έχει απορριφθεί

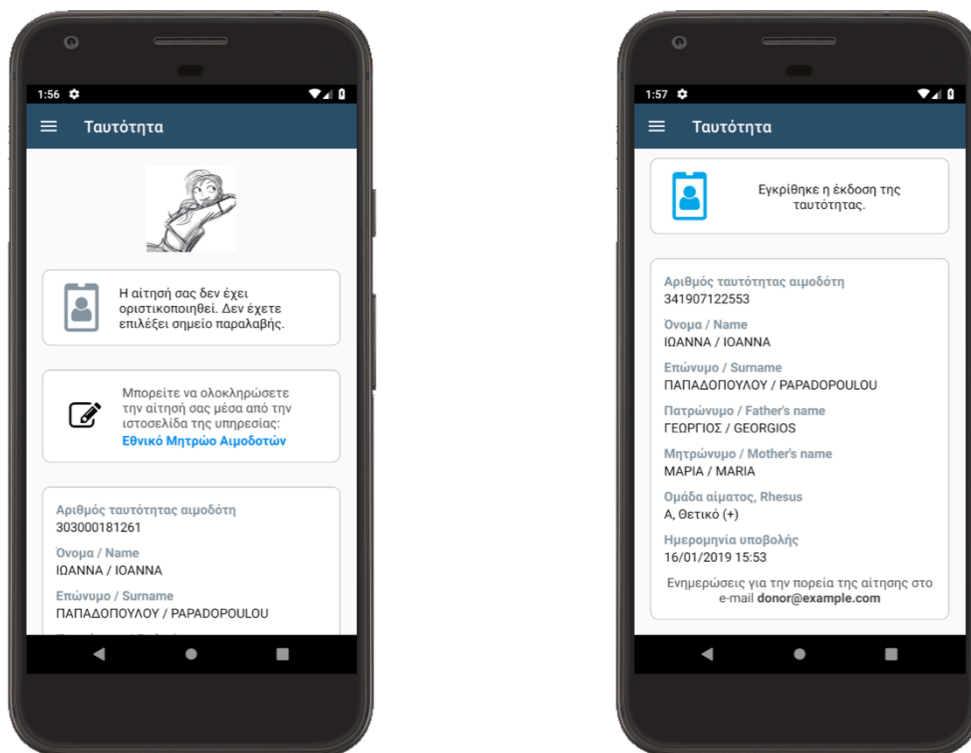
Η λογική, σύμφωνα με την οποία επιλέχθηκαν αυτά τα χρώματα στοχεύει στην γρήγορη κατανόηση από το χρήστη του τι συμβαίνει με την ταυτότητα του. Το γκρι υπογραμμίζει ότι η αίτηση δεν είναι ενεργή και απαιτούνται επιπλέον ενέργειες από το χρήστη. Το γαλάζιο δείχνει ότι η αίτηση βρίσκεται σε μία ουδέτερη κατάσταση, όπου δεν απαιτείται κάποια ενέργεια από το χρήστη. Το πράσινο δείχνει ότι πλέον η αίτηση είναι σε μία θετική κατάσταση, διαφορετική από την κατάσταση που έχει χρώμα γαλάζιο γιατί είτε απαιτείται ενέργεια από το χρήστη (παραλαβή της ταυτότητας), είτε είναι στο τελικό στάδιο όπου έχει παραληφθεί. Το πορτοκαλί χρησιμοποιήθηκε για τις καταστάσεις που απαιτείται η προσοχή του χρήστη αλλά όχι επειδή ο ίδιος έχει κάνει κάποια εσφαλμένη καταχώριση. Τέλος, το κόκκινο χρησιμοποιήθηκε για να τονίσει ότι προέκυψε ένα σφάλμα, για το οποίο καλείται ο χρήστης να δράσει.

Κάθε αίτηση για έκδοση ταυτότητας έχει ένα κουμπί «Επισκόπηση», που οδηγεί στην οθόνη που παρουσιάζει τα λεπτομερή στοιχεία της.

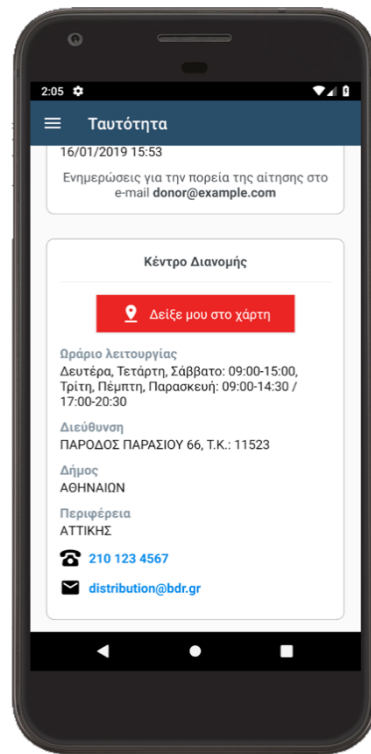
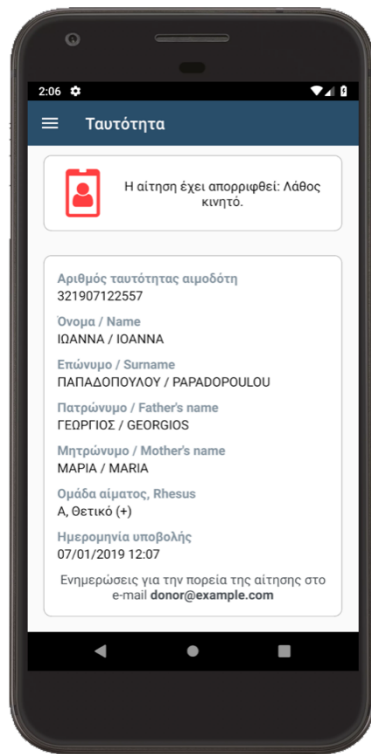


Εικόνα 8-7 Λίστα αιτήσεων έκδοσης ταυτότητας αιμοδότη

Στην οθόνη που παρουσιάζονται οι λεπτομέρειες κάθε αίτησης για ταυτότητα χρησιμοποιούνται οι κάρτες που χρησιμοποιούνται και στο προφίλ χρήστη για την ομαδοποίηση των στοιχείων. Η δομή της οθόνης της αίτησης περιλαμβάνει στην πάνω πλευρά τη φωτογραφία της ταυτότητας. Έπειτα, έχει τοποθετηθεί μία κάρτα που δίνει πληροφορίες για την κατάσταση της αίτησης. Εάν η αίτηση δεν έχει υποβληθεί οριστικά, έχει τοποθετηθεί στη συνέχεια μία κάρτα με οδηγίες για να μπορέσει ο χρήστης να προχωρήσει στην ολοκλήρωση της υποβολής. Έπειτα παρατίθενται στοιχεία όπως το ονοματεπώνυμο που θα τυπωθεί στην ταυτότητα, η ημερομηνία υποβολής της αίτησης (αν έχει υποβληθεί). Στο τέλος της ίδιας κάρτας έχουν τοποθετηθεί πληροφορίες για τον τρόπο που ο χρήστης θα ενημερώνεται για την εξέλιξη της διαδικασίας έκδοσης της ταυτότητάς του. Η επόμενη καρτέλα περιέχει όλες τις απαραίτητες πληροφορίες για το σημείο διανομής της ταυτότητας, όπως το ωράριο λειτουργίας και στοιχεία επικοινωνίας. Επιπλέον, περιέχει ένα κουμπί, το οποίο, όταν το πατήσει ο χρήστης, ανοίγει την εφαρμογή χαρτών του κινητού για να παρουσιάσει την ακριβή διεύθυνση του σημείου από όπου ο χρήστης μπορεί να παραλάβει την ταυτότητα του.



Εικόνα 8-8 Λεπτομέρειες αίτησης έκδοσης ταυτότητας αιμοδότη



Εικόνα 8-9 Λεπτομέρειες αίτησης έκδοσης ταυτότητας αιμοδότη

Οι οθόνες που ακολουθούν περιέχουν τα στοιχεία που σχετίζονται με τις αιμοδοσίες που έχει πραγματοποιήσει ο χρήστης από όταν τέθηκε σε λειτουργία το Εθνικό Μητρώο Αιμοδοτών. Τα στοιχεία μοιράζονται σε τρεις καρτέλες.

Η πρώτη καρτέλα περιέχει δύο κάρτες. Η μία αναφέρει τον συνολικό αριθμό αιμοδοσιών του χρήστη από το 2015, έτος που ξεκίνησε η παραγωγική λειτουργία του Εθνικού Μητρώου Αιμοδοτών. Στη δεύτερη έχει τοποθετηθεί ένα διάγραμμα πίτα με τις αιμοδοσίες του χρήστη, κατηγοριοποιημένες με βάση την αιτία που πραγματοποιήθηκαν.

Η επόμενη καρτέλα περιέχει ένα ραβδόγραμμα με τον αριθμό των αιμοδοσιών του χρήστη ανά έτος.

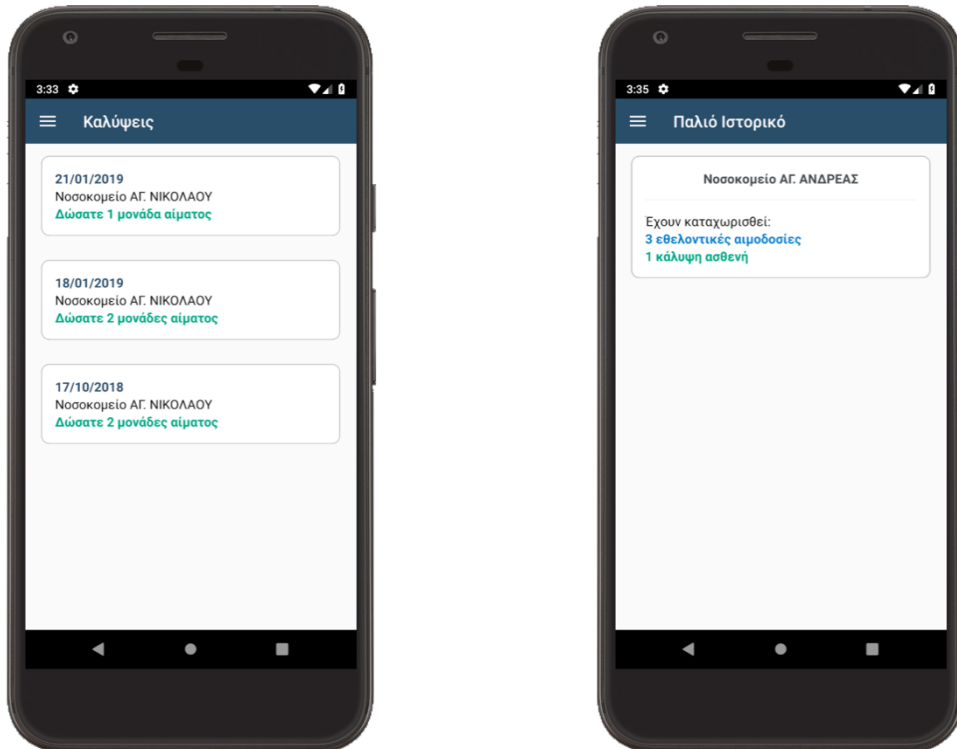
Στην τρίτη καρτέλα περιέχονται τα αναλυτικά στοιχεία των αιμοδοσιών, οι οποίες παρατίθενται ταξινομημένες, με την πιο πρόσφατη στην κορυφή της λίστας. Συγκεκριμένα, για κάθε αιμοδοσία ο χρήστης μπορεί να δει την ημερομηνία που πραγματοποιήθηκε, το κέντρο αιμοδοσίας όπου έγινε και τον λόγο που πραγματοποιήθηκε. Κάθε λόγος έχει αντιστοιχισθεί σε ένα χρώμα. Το χρώμα χρησιμοποιείται για το χρωματισμό του εικονιδίου που βρίσκεται στα αριστερά κάθε αιμοδοσίας και στο γράφημα πίτα της πρώτης καρτέλας. Το χρώμα παραμένει σταθερό έτσι ώστε να δημιουργείται ένα μοτίβο, εύκολα και γρήγορα αναγνωρίσιμο από τον χρήστη της εφαρμογής.



Εικόνα 8-10 Καρτέλες με τα στοιχεία των αιμοδοσιών χρήστη

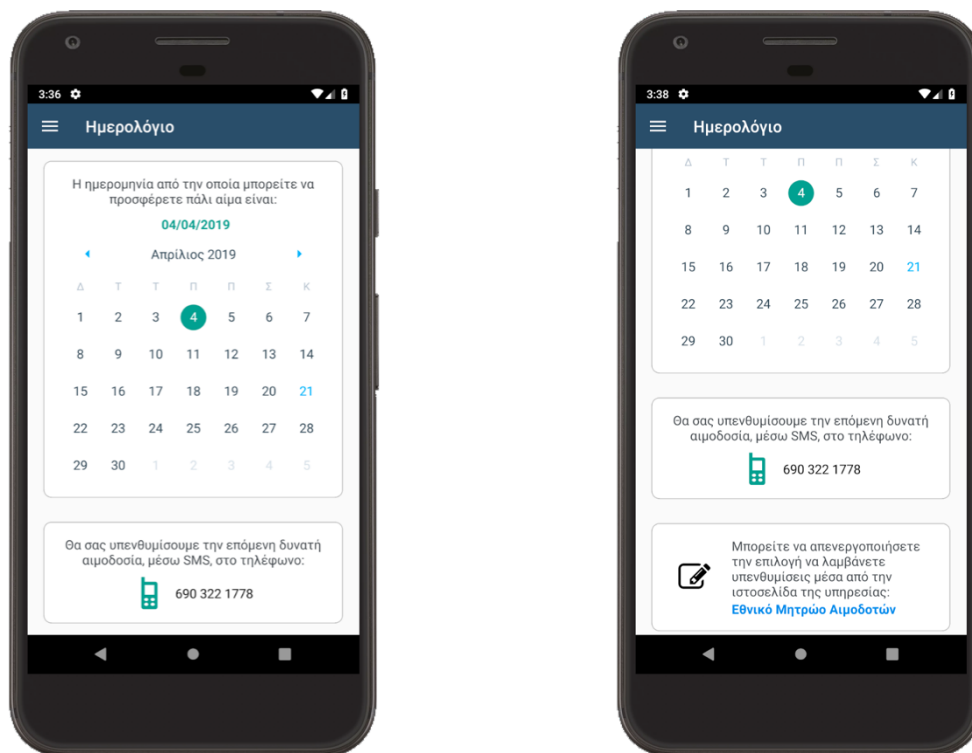
Η οθόνη που παρουσιάζει τις καλύψεις είναι μία απλή λίστα. Για κάθε κάλυψη εμφανίζεται η ημερομηνία πραγματοποίησής της, η μονάδα αιμοδοσίας όπου έγινε και πόσες μονάδες αίματος δόθηκαν.

Αντίστοιχη δομή έχει και η οθόνη που περιλαμβάνει τα στοιχεία των αιμοδοσιών που έκανε ο αιμοδότης πριν το 2015. Τα διαθέσιμα στοιχεία για εκείνη την περίοδο δεν περιλαμβάνουν τις ημερομηνίες των αιμοδοσιών αλλά πόσες αιμοδοσίες έχουν γίνει ανά λόγο και σε ποιο νοσοκομείο. Ουσιαστικά, αποτελεί μία αποτύπωση των σχετικών ιστορικών δεδομένων που κρατούσε κάθε νοσοκομείο για τον αιμοδότη. Να σημειωθεί ότι ο χρωματικός κώδικας που αναφέρθηκε για τις καρτέλες των αιμοδοσιών χρησιμοποιείται και στην οθόνη «Παλιό ιστορικό».



Εικόνα 8-11 Αριστερά οι καλύψεις του χρήστη και δεξιά οι αιμοδοσίες του πριν τη λειτουργία του Εθνικού Μητρώου Αιμοδοτών

Η οθόνη «Ημερολόγιο», που ακολουθεί, παρέχει πληροφορίες στον αιμοδότη σχετικά με την επόμενη δυνατή αιμοδοσία. Συγκεκριμένα, περιέχει την ημερομηνία, ύστερα από την οποία ο αιμοδότης μπορεί να δώσει εκ νέου αίμα (έχει παρέλθει το απαιτούμενο χρονικό διάστημα από την προηγούμενη αιμοδοσία του που έχει οριστεί από το Ε.ΚΕ.Α.). Επιπλέον, επειδή ο χρήστης έχει τη δυνατότητα να ενημερωθεί, μέσω e-mail ή μηνύματος sms για αυτήν την ημερομηνία, υπάρχουν και πληροφορίες για τις επιλογές που έχει κάνει σχετικά με τις ειδοποιήσεις και πώς είναι δυνατό να τις τροποποιήσει.



Εικόνα 8-12 Ημερολόγιο

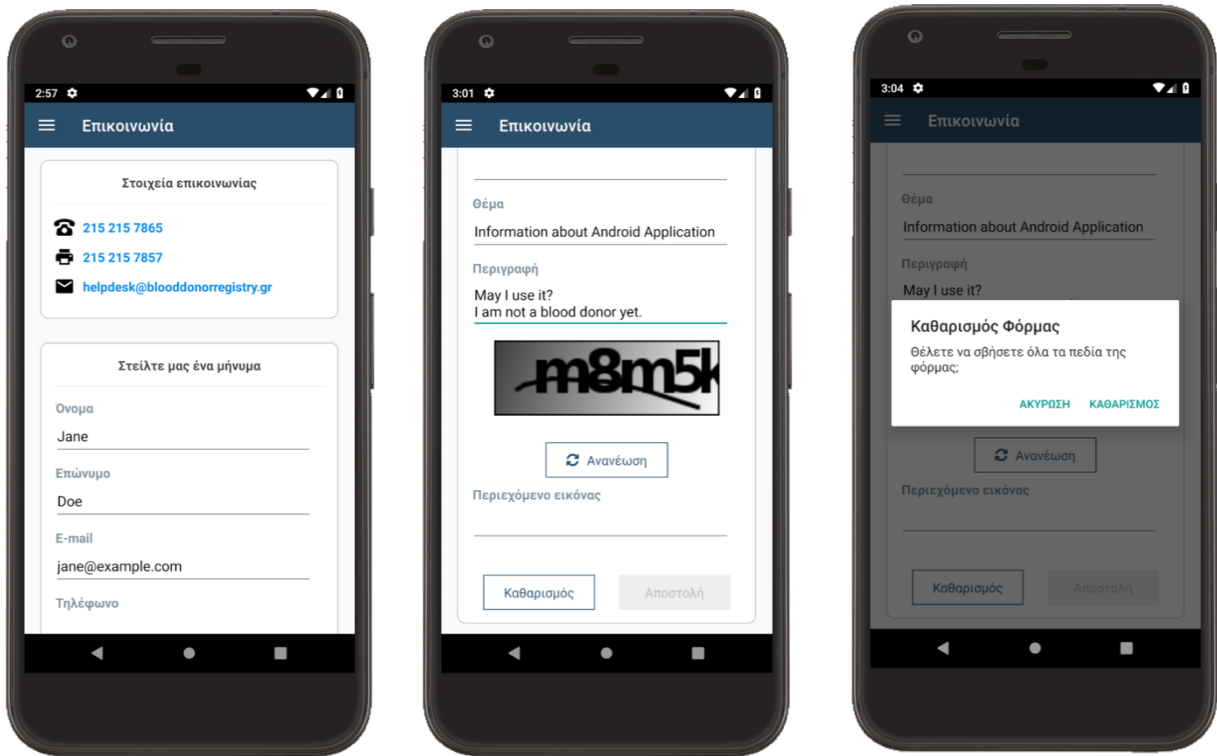
Η οθόνη «Επικοινωνία» είναι μία από τις οθόνες που είναι προσιτές σε όλους τους χρήστες, συνδεδεμένους και μη. Τα στοιχεία της είναι πάλι ομαδοποιημένα με κάρτες. Στην επάνω πλευρά της, περιλαμβάνονται τα στοιχεία επικοινωνίας του γραφείου αρωγής της υπηρεσίας. Ο χρήστης μπορεί να πατήσει πάνω στον τηλεφωνικό αριθμό, στον αριθμό του φαξ ή στη διεύθυνση e-mail που είναι χρωματισμένα με γαλάζιο χρώμα και να θέσει σε λειτουργία την αντίστοιχη εφαρμογή του Android συστήματος. Αυτή η λειτουργία υλοποιήθηκε για να βελτιώσει την εμπειρία του χρήστη.

Μετά τα στοιχεία επικοινωνίας, εμφανίζεται μία φόρμα, την οποία μπορεί να συμπληρώσει ο χρήστης και να στείλει μήνυμα στο γραφείο αρωγής. Προτού σταλεί το μήνυμα, κάθε πεδίο της φόρμας ελέγχεται. Ορισμένα πεδία είναι απαραίτητα να έχουν τιμή και να συμπεριληφθούν στο μήνυμα, όπως το ονοματεπώνυμο του χρήστη. Άλλα στοιχεία πρέπει να έχουν συγκεκριμένη μορφή, όπως η διεύθυνση ηλεκτρονικού ταχυδρομείου.

Μετά τη φόρμα ο χρήστης βλέπει μία εικόνα CAPTCHA και το αντίστοιχο πεδίο προς συμπλήρωση.

Στο τέλος υπάρχουν δύο κουμπιά διαθέσιμα. Το κουμπί στα αριστερά γίνεται ενεργό μόνο όταν η φόρμα είναι έγκυρη και χρωματίζεται με πράσινο χρώμα προκειμένου να αναδείξει ότι υλοποιεί μια σημαντική ενέργεια, την αποστολή του μηνύματος. Στα αριστερά έχει

τοποθετηθεί ένα κουμπί το οποίο ο χρήστης μπορεί να το χρησιμοποιήσει αν θέλει να καθαρίσει όλα τα πεδία της φόρμας, αφού επιβεβαιώσει ότι θέλει να πράξει κάτι τέτοιο.



Εικόνα 8-13 Επικοινωνία χρηστών με το γραφείο αρωγής της υπηρεσίας

Μετά την αποστολή του μηνύματος ο χρήστης θα δει ένα αναδυόμενο παράθυρο, που τον ενημερώνει για την επιτυχία ή την αποτυχία της αποστολής του μηνύματος. Το παράθυρο είναι το ίδιο με αυτό που περιγράφηκε για την εμφάνιση σφαλμάτων κατά την προσπάθεια σύνδεσης του χρήστη. Στην περίπτωση που παρουσιάζει την επιτυχία της αποστολής, το παράθυρο είναι χρωματισμένο πράσινο (θετικό χρώμα) και περιέχει ένα αντίστοιχο εικονίδιο και μήνυμα.

Η τελευταία οθόνη που υλοποιήθηκε, περιέχει χρήσιμες πληροφορίες για την εφαρμογή και την υπηρεσία. Στην εικόνα που ακολουθεί βλέπουμε στην πάνω πλευρά το λογότυπο της υπηρεσίας και στη συνέχεια κουμπιά που οδηγούν το χρήστη σε πληροφορίες που έχουν αναρτηθεί στα κοινωνικά δίκτυα:

- Facebook
- YouTube
- Twitter

Στη συνέχεια έχει τοποθετηθεί η έκδοση της εφαρμογής. Έπειτα, εμφανίζονται τρεις σύνδεσμοι. Ο πρώτος, αν επιλεγθεί, κατεβάζει το αρχείο PDF που περιέχει τους όρους και τις προϋποθέσεις χρήσης της υπηρεσίας. Ο δεύτερος σύνδεσμος όταν επιλεγθεί εμφανίζει στην οθόνη το αναλυτικό κείμενο της άδειας λογισμικού της εφαρμογής. Ο τρίτος μετά την επιλογή του εμφανίζει την πλήρη λίστα με τις βιβλιοθήκες ανοιχτού λογισμικού που έχουν αξιοποιηθεί για την υλοποίηση της εφαρμογής και την άδεια λογισμικού της κάθε μίας. Η οθόνη αυτή είναι διαθέσιμη σε όλους τους χρήστες της εφαρμογής, συνδεδεμένους και μη.



Εικόνα 8-14 Πληροφορίες σχετικά με την εφαρμογή

9 Δυσκολίες που αντιμετωπίστηκαν και μελλοντικές επεκτάσεις

Κατά τη διάρκεια της παρούσας διπλωματικής αντιμετωπίστηκαν ορισμένες δυσκολίες, οι κυριότερες εκ των οποίων αφορούσαν τη διαδικασία προγραμματισμού της εφαρμογής.

Η διαδικασία εντοπισμού σφαλμάτων και η διόρθωση τους ήταν αυξημένης δυσκολίας, ιδιαίτερα όταν δεν ήταν σαφές εάν το σφάλμα προερχόταν από κώδικα Java ή JavaScript. Επίσης, στα πρώτα στάδια υλοποίησης της εφαρμογής, οι βιβλιοθήκες που χρησιμοποιούνταν αλλά και το ίδιο το React Native δημοσίευαν νέες εκδόσεις με μεγάλη συχνότητα χωρίς όμως σαφή βήματα για την υιοθέτηση τους.

Ένα άλλο σημείο, που σε ορισμένες περιπτώσεις, ήταν αυξημένης δυσκολίας αφορούσε την εφαρμογή style στα στοιχεία της διεπαφής. Ο μηχανισμός μορφοποίησης της διεπαφής στο React Native ακολουθεί το API του CSS, που χρησιμοποιείται από του περιηγητές ιστού, όμως δεν είναι υλοποιημένο το σύνολο των δυνατοτήτων του CSS, με αποτέλεσμα ορισμένες οθόνες να μη μπορούν να «στηθούν» με τον επιθυμητό τρόπο.

Όσο αφορά την επέκταση της εφαρμογής, προτού σχεδιαστούν επιπλέον λειτουργικότητες, προτείνεται να πραγματοποιηθεί διερεύνηση και δοκιμές με πραγματικούς χρήστες ώστε να εξασφαλιστεί η ικανοποίηση τους από τις παρεχόμενες λειτουργικότητες.

Από την έως τώρα έρευνα και μελέτη γύρω από τις δυνατότητες της εφαρμογής έχουν εντοπισθεί ορισμένες προσθήκες που θα μπορούσαν να αυξήσουν τη χρηστική της αξία.

Για αρχή, προτείνεται να προστεθούν λειτουργικότητες για την επικαιροποίηση των στοιχείων των χρηστών και η δυνατότητα αίτησης για Ταυτότητα Εθελοντή Αιμοδότη. Οι λειτουργικότητες αυτές είναι διαθέσιμες για τους χρήστες μέσω της διαδικτυακής διεπαφής. Συνεπώς, ένας χρήστης, που έχει εμπειρία με τη διαδικτυακή εφαρμογή, που προϋπάρχει, αναμένει να περιλαμβάνονται και στην εφαρμογή για κινητά τερματικά. Επιπλέον, η αίτηση για Ταυτότητα Αιμοδότη μπορεί να γίνει ακόμα πιο σύντομη για το χρήστη, αν η εφαρμογή προτείνει κοντινά σημεία παραλαβής, με βάση την τοποθεσία του χρήστη.

Μία άλλη χρήσιμη επέκταση της εφαρμογής θα ήταν η ένταξη διαφόρων ενημερώσεων. Για παράδειγμα, ένας μηχανισμός που να ενημερώνει το χρήστη άμεσα ότι η Ταυτότητα του είναι έτοιμη για παραλαβή. Μία άλλη πιθανώς χρήσιμη ενημέρωση σχετίζεται με τη δυνατότητα του χρήστη για νέα δωρεά αίματος. Όταν έχει περάσει το χρονικό διάστημα, που απαιτείται έως την επόμενη αιμοδοσία, θα ήταν χρήσιμο να λαμβάνει μία ενημέρωση και μαζί όλα τα διαθέσιμα σημεία αιμοδοσίας με βάση την τοποθεσία του ή και άλλα κριτήρια που ο ίδιος έχει θέσει. Μία άλλη πιθανή και σημαντική ειδοποίηση αφορά τη στοχευμένη ενημέρωση για ανάγκες αίματος. Τα κριτήρια των αιμοδοτών που θα λάβουν μία τέτοια ενημέρωση θα μπορούσαν να σχετίζονται με τη διεύθυνση κατοικίας τους, το διάστημα που έχει περάσει από την τελευταία τους αιμοδοσία και την ομάδα αίματός τους.

Επίσης, μία άλλη πιθανή επέκταση θα ήταν η προσαρμογή του κώδικα ώστε η εφαρμογή να μπορεί να λειτουργήσει και σε iOS λειτουργικό, ώστε να εξυπηρετεί τους αντίστοιχους αιμοδότες.

Σημαντικό μέλημα όλου του συστήματος τους Εθνικού Μητρώου Αιμοδοτών είναι οι επιμέρους εφαρμογές να διασφαλίζουν την ασφάλεια των στοιχείων των αιμοδοτών. Προκειμένου να εξασφαλίσει και η εφαρμογή για Android συσκευές την ασφάλεια των δεδομένων των χρηστών χωρίς όμως να αυξάνεται η περιπλοκότητα χρήσης της, μπορούν να χρησιμοποιηθούν βιομετρικά στοιχεία ή σύντομος αριθμητικός κωδικός (PIN), τα οποία αποθηκεύονται ασφαλώς στην εφαρμογή και χρησιμοποιούνται στη συνέχεια ώστε να εξασφαλίζουν και την ασφαλή σύνδεση με το Εθνικό Μητρώο Αιμοδοτών [46].

Όσο αφορά τη μελέτη του API του Ενιαίου Μητρώου Αιμοδοτών, ως επόμενο βήμα προτείνεται η υλοποίηση των προτάσεων και στη συνέχεια η μελέτη και ο επανασχεδιασμός του υπόλοιπου API ώστε να αξιοποιηθούν στο έπακρο τα προτερήματα της REST αρχιτεκτονικής.

Βιβλιογραφία

- [1] E.KE.A., [Ηλεκτρονικό]. Available: <http://ekea.gr/our-history/>.
- [2] E.KE.A., «Ιστορία της Αιμοδοσίας» [Ηλεκτρονικό]. Available: <http://ekea.gr/history-blood/>.
- [3] Μηχανή του Χρόνου, «Ηλίας Πολίτης» [Ηλεκτρονικό]. Available: <https://www.mixanitouxronou.gr/ilias-politis-o-anidiotelis-giatros-pou-aftoktonise-gialogous-efthixias/>.
- [4] Ε.Δ.Υ.Τ.Ε., «Εθνικό Μητρώο Αιμοδοτών» [Ηλεκτρονικό]. Available: <https://blooddonorregistry.gr/>.
- [5] Internet World Stats, [Ηλεκτρονικό]. Available: <https://www.internetworldstats.com/stats.htm>.
- [6] «Internet Growth Statistics» [Ηλεκτρονικό]. Available: <https://www.internetworldstats.com/emarketing.htm>.
- [7] Internet World Stats, «Internet in Europe Stats» [Ηλεκτρονικό]. Available: <https://www.internetworldstats.com/stats4.htm>.
- [8] J. Clement, «Global digital population as of October 2019» [Ηλεκτρονικό]. Available: <https://www.statista.com/statistics/617136/digital-population-worldwide/>. [Πρόσβαση 2019].
- [9] «ISO 9241-11» [Ηλεκτρονικό]. Available: <https://www.iso.org/standard/63500.html>.
- [10] Android Open Source Project, «Android Design Principles» [Ηλεκτρονικό]. Available: <http://www.androiddocs.com/design/get-started/principles.html>. [Πρόσβαση 2018].
- [11] Facebook, «Improving User Experience» [Ηλεκτρονικό]. Available: <https://facebook.github.io/react-native/docs/improvingux.html>.
- [12] W3C, «Mobile Web Best Practices 1.0» 2004. [Ηλεκτρονικό]. Available: <https://www.w3.org/TR/mobile-bp/>.
- [13] W3C, «Internationalization» [Ηλεκτρονικό]. Available: <https://www.w3.org/standards/webdesign/i18n>. [Πρόσβαση 2020].
- [14] W3C, [Ηλεκτρονικό]. Available: <https://www.w3.org/International/questions/qa-i18n>. [Πρόσβαση 2020].
- [15] Wikipedia, «Internationalization and localization» [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Internationalization_and_localization. [Πρόσβαση 2020].
- [16] K. Vasudevan, «What is API Design» 2016. [Ηλεκτρονικό]. Available: <https://swagger.io/blog/api-design/what-is-api-design/>.
- [17] K. Vasudevan, «Best Practices in API Design» [Ηλεκτρονικό]. Available: <https://swagger.io/resources/articles/best-practices-in-api-design/>.

- [18] R. T. Fielding, «Doctoral dissertation: Architectural Styles and the Design of Network-based Software Architectures (Section 3.5.3)» University of California, Irvine, 2000.
- [19] Network Working Group, «RFC 2616: Hypertext Transfer Protocol -- HTTP/1.1, Section 9 Method Definitions».
- [20] Wikipedia, «Stateless protocol» [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Stateless_protocol.
- [21] Wikipedia, «Component-based software engineering» [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Component-based_software_engineering#Software_component.
- [22] Wikipedia, «Representational state transfer» [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Representational_state_transfer.
- [23] Wikipedia, «Load balancing (computing)» [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Load_balancing_\(computing\)](https://en.wikipedia.org/wiki/Load_balancing_(computing)).
- [24] OpenAPI Initiative, «OpenAPI Specification» [Ηλεκτρονικό]. Available: <https://swagger.io/specification/>.
- [25] «RFC 5789: PATCH Method for HTTP» Internet Engineering Task Force (IETF), 2010. [Ηλεκτρονικό]. Available: <https://www.rfc-editor.org/rfc/rfc5789>.
- [26] «RFC 7232» Internet Engineering Task Force (IETF), 2014. [Ηλεκτρονικό]. Available: <https://httpwg.org/specs/rfc7232.html#header.if-modified-since>.
- [27] «RFC 7231» Internet Engineering Task Force (IETF), 2014. [Ηλεκτρονικό]. Available: <https://httpwg.org/specs/rfc7231.html#header.accept-language>.
- [28] «RFC 7231» Internet Engineering Task Force (IETF), 2014. [Ηλεκτρονικό]. Available: <https://httpwg.org/specs/rfc7231.html#header.content-language>.
- [29] Wikipedia, «Android» [Ηλεκτρονικό]. Available: <https://el.wikipedia.org/wiki/Android>.
- [30] Android Developers, «WebView» [Ηλεκτρονικό]. Available: <https://developer.android.com/reference/android/webkit/WebView>.
- [31] J. Looper. [Ηλεκτρονικό]. Available: <https://developer.telerik.com/featured/what-is-a-webview/> . [Πρόσβαση 2018].
- [32] Wikipedia, «Git (λογισμικό)» [Ηλεκτρονικό]. Available: [https://el.wikipedia.org/wiki/Git_\(λογισμικό\)](https://el.wikipedia.org/wiki/Git_(λογισμικό)).
- [33] «node.js» [Ηλεκτρονικό]. Available: <https://nodejs.org/en/about/>.
- [34] Wikipedia, «Node.js» wikipedia, [Ηλεκτρονικό]. Available: <https://en.wikipedia.org/wiki/Node.js>.
- [35] «npm» wikipedia, [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Npm_\(software\)](https://en.wikipedia.org/wiki/Npm_(software)).
- [36] «React» [Ηλεκτρονικό]. Available: <https://reactjs.org/>.
- [37] «Redux» [Ηλεκτρονικό]. Available: <https://redux.js.org/>.

- [38] «Flow» [Ηλεκτρονικό]. Available: <https://flow.org/>.
- [39] «ESLint» [Ηλεκτρονικό]. Available: <https://eslint.org/>.
- [40] «Jest» [Ηλεκτρονικό]. Available: <https://jestjs.io/>.
- [41] «Who's using React Native» [Ηλεκτρονικό]. Available: <https://reactnative.dev/showcase>.
- [42] R. Gaba και R. Atul, «React Native Internals» [Ηλεκτρονικό]. Available: <https://www.reactnative.guide/3-react-native-internals/3.1-react-native-internals.html>.
- [43] N. Parashuram, «React Native's New Architecture - React Conf 2018» [Ηλεκτρονικό]. Available: <https://www.youtube.com/watch?v=UcqRXTriUVI>.
- [44] AltexSoft, «The Good and the Bad of ReactJS and React Native» [Ηλεκτρονικό]. Available: <https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-reactjs-and-react-native/>.
- [45] K. Shah, «Advantages and Disadvantages of React Native Development in 2022» [Ηλεκτρονικό]. Available: <https://www.thirdrocktechkno.com/blog/pros-and-cons-of-react-native-development-in-2021/>.
- [46] Diego Poza, «What Is Biometric Authentication? 3 Trends for 2021» auth0, 22 10 2021. [Ηλεκτρονικό]. Available: <https://auth0.com/blog/3-critical-trends-in-biometric-authentication-in-2019/>.
- [47] Ecma International, «ECMAScript Internationalization API Specification» 2012. [Ηλεκτρονικό]. Available: <http://www.ecma-international.org/ecma-402/1.0/index.html#sec-4.1>.
- [48] Ipsos OTX MediaCT, «Our Mobile Planet: Global Smartphone Users» February 2012. [Ηλεκτρονικό]. Available: http://services.google.com/fh/files/blogs/final_global_smartphone_user_study_2012.pdf.
- [49] A. Ντίνης, «Η ιστορία της κινητής τηλεφωνίας: Από το χθες στο σήμερα και επιστροφή στο χθες» 2016. [Ηλεκτρονικό]. Available: <https://www.eleftherostypos.gr/istories/62000-i-istoria-tis-kinitis-tilefonias-apo-to-xthes-sto-simera-kai-epistrofi-sto-xthes/>.
- [50] Wikipedia, «Παγκόσμιος Ιστός» [Ηλεκτρονικό]. Available: https://el.wikipedia.org/wiki/Παγκόσμιος_Ιστός.
- [51] Wikipedia, «Ευχρηστία» [Ηλεκτρονικό]. Available: <https://el.wikipedia.org/wiki/Ευχρηστία>.