ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Υλοποίηση εφαρμογής Android με υπηρεσίες εντοπισμού θέσης και μηχανικής μάθησης για την υποβοήθηση επισκεπτών μουσείου

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

ΚΟΥΒΑΤΖΗ ΚΩΝΣΤΑΝΤΙΝΟΥ

Επιβλέπων: Γεώργιος Στάμου

Αναπληρωτής καθηγητής Ε.Μ.Π

Αθήνα, Ιούλιος 2018

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

Υλοποίηση εφαρμογής Android με υπηρεσίες εντοπισμού θέσης και μηχανικής μάθησης για την υποβοήθηση επισκεπτών μουσείου

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

ΚΟΥΒΑΤΖΗ ΚΩΝΣΤΑΝΤΙΝΟΥ

Επιβλέπων: Γεώργιος Στάμου

Αναπληρωτής Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή επιτροπή την 3η Ιουλίου 2018:

(Υπογραφή)	(Υπογραφή)	(Υπογραφή)
.....	.....	.....
Γεώργιος Στάμου	Ανδρέας Σταφυλοπάτης	Παναγιώτης Τσανάκας
Αναπληρωτής Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2018

(Υπογραφή)

ΚΟΥΒΑΤΖΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π

Περίληψη

Η παρούσα διπλωματική στοχεύει στην ανάπτυξη μιας εφαρμογής, η οποία θα λειτουργεί ως μέσο για να βοηθήσει τον χρήστη της να αποκτήσει μια καλύτερη εμπειρία – πέρα από την παραδοσιακή – στον χώρο του Μουσείου Μπενάκη. Η χρήση του κινητού τηλεφώνου ενδείκνυται για αυτήν την προσέγγιση, καθώς αποτελεί ένα μη συμβατικό μέσο. Η εφαρμογή δεν έχει όμως ως μόνο στόχο το να κάνει μια επίσκεψη σε ένα μουσείο πιο ευχάριστη, αλλά στοχεύει και στο να βελτιώσει την κατανόηση και αφομοίωση των πληροφοριών που παρέχονται στο μουσείο. Αυτό επιτυγχάνεται μέσα από τα παιχνίδια της κατηγορίας <<σοβαρού σκοπού>> (serious games).

Η αναλυτική ανασκόπηση της βιβλιογραφίας που προηγείται του κυρίως μέρους της εργασίας καλύπτει τις βασικές πτυχές της διαδικασίας ανάπτυξης λογισμικού, των τεχνολογιών που εξετάστηκαν και τα βασικά σημεία της υλοποίησης της εφαρμογής.

Η εφαρμογή έχει σχεδιαστεί έτσι ώστε να λειτουργεί στις συσκευές κινητών τηλεφώνων που χρησιμοποιούν το λογισμικό Android, ενώ για την ανάπτυξή της χρησιμοποιήθηκε η γλώσσα προγραμματισμού Java.

Στο πλαίσιο της διπλωματικής υλοποιήθηκε αρχικά το γραφικό περιβάλλον του χρήστη της εφαρμογής. Στο ξεκίνημα, οι οθόνες υλοποιήθηκαν στατικά και στη συνέχεια τροποποιήθηκαν έτσι ώστε να υποστηρίζουν μια τοπική βάση δεδομένων τύπου SQLite. Πλέον όλα τα στοιχεία της εφαρμογής περιέχονται στη βάση δεδομένων και φορτώνονται στο γραφικό περιβάλλον δυναμικά.

Στην συνέχεια υλοποιήθηκε ο εντοπισμός της τοποθεσίας του χρήστη στο μουσείο. Για να επιτευχθεί αυτό, εξετάστηκαν διάφορες τεχνολογίες, όπως τα beacons, αλλά και η χρήση του μαγνητικού πεδίου της γης, μέσω της τεχνολογίας που παρέχει η Indoor Atlas. Η χρήση της τεχνολογίας Indoor Atlas οδηγούσε σε λιγότερο κόστος και σε ικανοποιητικά αποτελέσματα, όταν χρησιμοποιούνταν στη συσκευή στην οποία σχεδιάστηκε. Ωστόσο, όταν ελέχθηκε σε διαφορετικές συσκευές, η απόδοση και η ακρίβειά της μειώθηκε αισθητά. Από την άλλη πλευρά, η τεχνολογία των beacons προσέφερε από την αρχή τα ίδια σχεδόν ικανοποιητικά αποτελέσματα σε όποια συσκευή και αν εξετάστηκε. Συνεπώς, για την τελική υλοποίηση επιλέχθηκε η τεχνολογία των beacons για λόγους απόδοσης.

Εξαιτίας όμως του γεγονότος ότι σε μερικές περιπτώσεις η τεχνολογία των beacons δεν απέδιδε ικανοποιητικά, αποφασίστηκε να υλοποιηθεί και ένας εναλλακτικός τρόπος εντοπισμού της τοποθεσίας του χρήστη. Ο τρόπος, που επιλέχθηκε, ήταν η αναγνώριση της εικόνας των εκθεμάτων με τη χρήση της μηχανικής εκμάθησης μέσω του εργαλείου Tensorflow. Για την υλοποίησή της, έγινε

επανεκπαίδευση του τελευταίου layer ενός ήδη υπάρχοντος συνελιξιακού μοντέλου. Η τελική ακρίβεια που επιτεύχθηκε πλησίασε το 100% κυρίως λόγω του ότι οι εικόνες που θέλουμε να αναγνωρίσουμε είναι συγκεκριμένες. Παρόλα αυτά, νέα χαρακτηριστικά και βελτιώσεις μπορούν να προστεθούν στο μέλλον, έτσι ώστε να ενισχυθεί η εμπορικότητά της.

Λέξεις κλειδιά: εφαρμογή μουσείου, Android, εφαρμογή κινητού τηλεφώνου, παιχνιδοποίηση, beacon, τοποθεσία, indoor positioning, βάση δεδομένων, μηχανική μάθηση.

Abstract

This thesis targets the development of an mobile app, that will function as a mean to help the user gain a better experience at the Benaki Museum. The use of mobile phones is indicated for this approach since it is a non conventional mean. This app does not have as a single target the improvement of a museum visit but also targets the improvement of the understanding and the absorption of the information presented at the museum. This is accomplished through the category of games called serious games.

An analytical sum of the references that proceeds the main part of the thesis covers the main targets of the app, the basic technologies used and the main points of the implementation of the app.

The app is designed for mobile devices which run the android software and for the development of the app the programming language Java was used.

In the scope of this thesis the graphical user interface was first created. At first, the screens were created from a static content but afterwards they were modified to support a local SQLite database. Now all the contents of the app are saved in the database and are loaded dynamically when needed.

Afterwards the location of the user's position into the museum was implemented. For this purpose, multiple technologies were examined such as the beacons and the magnetic field of the earth, through the technology offered from Indoor Atlas. The use of the Indoor Atlas technology led to fewer expenses and sufficient results when it was designed and tested on the same device. But when it was designed and tested on different devices the accuracy was not the same. On the other hand, the technology of beacons showed the same results from the beginning no matter the device. So, for the final implementation of the app it was decided that the beacons were the best fit.

Due to the fact that in some cases beacons were not performing as expected, it was desided to implement another alternative way of locating the user's position at the museum. The selected way was the image recognition with the help of machine learning and the software of Tensorflow. For the implelmentation, the last layer of an already trained convolutional neural network was retrained. The final accuracy was very close to 100% because the images that need to be recognised were standard. Finally, new characteristics and improvments can be added in the future to improve the marketability of the app.

Keywords: museum application, beacons, Android, mobile app, indoor positioning, gamefication, database, location,machine learning.

Ευχαριστήριο Σημείωμα

Τελειώνοντας την διπλωματική εργασία μου, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κο Γεώργιο Στάμου για την καθοδήγησή του, αλλά και για την ευκαιρία που μου έδωσε να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα ως τελευταία υποχρέωσή για την ολοκλήρωση του πτυχίου μου.

Στην συνέχεια, θα ήθελα να ευχαριστήσω την κα Μαρία Ράλλη και τα υπόλοιπα μέλη του *Εργαστηρίου Επεξεργασίας Εικόνας Βίντεο και Πολυμέσων* για τις συμβουλές που μου έδωσαν, αλλά και για την απαραίτητη βιβλιογραφία που μου παρείχαν, όποτε χρειάστηκε.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου για όλη την στήριξη που μου παρείχε κατά την διάρκεια της εκπόνησης της διπλωματικής μου εργασίας.

Ευελπιστώ το παρόν κείμενο και η παρούσα εφαρμογή να φανούν χρήσιμα σε μελλοντικούς αναγνώστες και χρήστες.

Πίνακας Περιεχομένων

Πίνακας Εικόνων

1

Εισαγωγή

Οι τεχνολογικές εξελίξεις των τελευταίων δεκαετιών έχουν οδηγήσει στην ταχύτατη ανάπτυξη της βιομηχανίας των κινητών τηλεφώνων. Ταυτόχρονα η αυξημένη ζήτηση για εφαρμογές κινητών τηλεφώνων έχει οδηγήσει στην αντικατάσταση των παλαιών κινητών με τα σύγχρονα έξυπνα κινητά (smartphones). Η ευκολία στη χρησιμοποίηση και στην ανάπτυξη των εφαρμογών λόγω της τεράστιας δημοφιλίας τους έχει καταστήσει τις εφαρμογές κινητών τηλεφώνων σχεδόν απαραίτητες για την καθημερινότητά μας.

Μια τεχνολογία που έχει αναπτυχθεί αρκετά λόγω των έξυπνων κινητών είναι το Παγκόσμιο Σύστημα Εντοπισμού θέσης (GPS). Χάρη σε αυτό, η πλοήγηση στους εξωτερικούς χώρους έχει απλοποιηθεί σημαντικά. Παρά την τεράστια εξέλιξή του, όμως, οι δυνατότητές του περιορίζονται σε εξωτερικούς χώρους. Αντιθέτως, οι αντίστοιχες τεχνολογίες για εσωτερικούς χώρους, μόλις τα τελευταία χρόνια έχουν αρχίσει να αναπτύσσονται.

Μια ακόμα τεχνολογία που έχει κάνει την εμφάνισή της τα τελευταία χρόνια είναι η αναγνώριση εικόνας μέσω της μηχανικής μάθησης. Η ανάπτυξη νευρωνικών δικτύων έχει προσδώσει στους υπολογιστές μερικά ανθρώπινα χαρακτηριστικά. Αυτά τα χαρακτηριστικά τούς επιτρέπουν να αναγνωρίζουν εικόνες και να κατατάσσουν αντικείμενα σε κατηγορίες.

Τις παραπάνω τεχνολογικές εξελίξεις μπορούν να εκμεταλλευτούν μουσεία ή χώροι πολιτισμικής κληρονομιάς και να προσφέρουν νέες ευκαιρίες εκπαίδευσης και νέα σενάρια μάθησης. Η χρησιμοποίηση μιας φορητής συσκευής μπορεί να επηρεάσει θετικά τη συμπεριφορά του επισκέπτη και να βελτιώσει την αλληλεπίδρασή του με τον χώρο του μουσείου. Παρόλα αυτά, η ανάπτυξη ψηφιακών εργαλείων που μπορούν να χρησιμοποιηθούν από μουσεία ή πολιτισμικούς χώρους εξακολουθεί να αποτελεί πρόκληση.

Η πρόκληση αυτή έρχεται να συμπληρώσει και να επεκτείνει, σε άλλες κατευθύνσεις, αλλαγές που έχουν γίνει στον τρόπο αντιμετώπισης της πολιτισμικής κληρονομιάς από τον ψηφιακό κόσμο. ΄Ηδη μέσω της Europeana, της ευρωπαϊκής ψηφιακής βιβλιοθήκης που λειτουργεί από το 2008, μέλος της οποίας είναι και το Μουσείο Μπενάκη, συγκεντρώνονται και αποθηκεύονται σε μια τεράστια βάση δεδομένων διάφορα τεκμήρια πολιτισμού, όπως βιβλία, χάρτες, φωτογραφίες, ηχογραφήσεις, πίνακες κτλ. Το επόμενο βήμα είναι η «δημοσιοποίηση» και η προσφορά αυτών των θησαυρών στο κοινό με έναν άλλο τρόπο. Τα μουσεία λειτουργώντας ως ένας ζωντανός χώρος πολιτισμικής κληρονομιάς μπορούν να κινηθούν σε αυτήν την κατεύθυνση, προσφέροντας σε ένα ολοένα και πιο εξοικειωμένο με τα κινητά κοινό, όχι μόνο ευχάριστες περιδιαβάσεις στον χώρο τους, αλλά και μορφωτικά αγαθά που αποκτώνται και μέσα από παιχνίδια της κατηγορίας <<σοβαρού σκοπού>> (serious games).

Έναν τέτοιο στόχο πιστεύω ότι μπορεί να υλοποιήσει η ανάπτυξη της εφαρμογής για το Μουσείο Μπενάκη Ισλαμικής Τέχνης που παρουσιάζεται. Στα επόμενα κεφάλαια θα παρουσιαστεί η εξέλιξή της με λεπτομέρειες, θα αναλυθούν οι τεχνολογίες που χρησιμοποιήθηκαν, καθώς και οι τεχνικές δυσκολίες που προέκυψαν. Τέλος, θα παρουσιαστούν συνοπτικά και κάποιες προτάσεις για μελλοντικές επεκτάσεις της εφαρμογής.

1.1 Στόχος

Η εφαρμογή θα επιτρέπει στον χρήστη να περιηγηθεί στο Μουσείο μέσω των περιηγήσεων που θα του προτείνει, ενώ ταυτόχρονα θα του παρέχει πληροφορίες για τα εκθέματα που υπάρχουν. Επιπροσθέτως μέσω τεχνολογιών

indoor positioning θα του παρέχει την δυνατότητα ενός παιχνιδιού κρυμμένου θησαυρού. Στόχος της εφαρμογής αυτής είναι να παρέχει στον χρήστη μια πλήρη και ολοκληρωμένη εμπειρία του Μουσείου Μπενάκη χρησιμοποιώντας τις τελευταίες τεχνολογικές εξελίξεις.

Στόχος αυτής της διπλωματικής είναι να παρουσιάσει και να αναλύσει σε ένα ικανοποιητικό επίπεδο τις διάφορες τεχνολογίες που εξετάστηκαν, να τονίσει τα θετικά και τα αρνητικά χαρακτηριστικά τους, να σχολιάσει τις σχεδιαστικές αποφάσεις που ελήφθησαν και τέλος να αναλύσει κάποια βασικά σημεία της υλοποίησης της εφαρμογής μαζί με μερικές μελλοντικές επεκτάσεις.

1.3 Οργάνωση του Τόμου

Εκτός του παρόντος 1ου κεφαλαίου που αποτελεί την εισαγωγή της, η εργασία έχει χωριστεί σε 5 επιπλέον κεφάλαια, το περιεχόμενο των οποίων παρουσιάζεται συνοπτικά παρακάτω.

Στο 2ο κεφάλαιο, ο χρήστης θα συναντήσει και θα εξοικειωθεί με τους βασικούς ορισμούς που χρησιμοποιούνται κατά κύριο λόγο στην εργασία. Επίσης γίνεται μια συνοπτική εισαγωγή στις τεχνολογίες που εξετάστηκαν για την συγκεκριμένη εργασία με σκοπό την καλύτερη κατανόηση των ακολούθων κεφαλαίων από τον αναγνώστη.

Στο 3ο κεφάλαιο, αναλύεται η διαδικασία σχεδίασης της εφαρμογής: πιο συγκεκριμένα παρέχονται πολλαπλά διαγράμματα για την πλήρη και αναλυτική κατανόηση της λειτουργίας της εφαρμογής. Επίσης, αναλύονται οι τεχνολογίες που χρησιμοποιήθηκαν, αλλά και αυτές που εξετάστηκαν, αλλά τελικά δεν συμπεριλήφθηκαν στην τελική υλοποίηση της εφαρμογής για διαφορετικούς λόγους η κάθε μια. Επιπροσθέτως, σε αυτό το κεφάλαιο γίνεται παρουσίαση των οθονών της εφαρμογής μαζί με μία συνοπτική ανάλυση τους.

Στο 4ο κεφάλαιο, ο χρήστης συναντά λεπτομέρειες για την υλοποίηση των πιο σημαντικών σημείων της κάθε οθόνης, αλλά και της εφαρμογής γενικότερα. Το κεφάλαιο αυτό περιέχει επίσης λεπτομέρειες για τον έλεγχο της ομαλής λειτουργίας της εφαρμογής σε όλες τις συσκευές τύπου Android μαζί με μερικά στατιστικά για την λειτουργία της.

Το 5ο κεφάλαιο αποτελεί μια σύνοψη και μια αξιολόγηση της εφαρμογής. Επίσης περιέχει προτάσεις για την βελτίωση και την επέκτασή της με την προσθήκη νέων χαρακτηριστικών και τεχνολογιών.

Τέλος, στο 6ο κεφάλαιο παρουσιάζεται η βασική βιβλιογραφία στην οποία βασίστηκε η παρούσα διπλωματική.

2

Θεωρητικό Υπόβαθρο

Το μεγαλύτερο μέρος των πληροφοριών του παρόντος κεφαλαίου έχει συλλεχθεί από ιστοσελίδες στο internet. Στην σημερινή εποχή το διαδίκτυο αποτελεί την κύρια πηγή αναζήτησης κάθε ερασιτέχνη, αλλά και επαγγελματία προγραμματιστή.

Στόχος αυτού του κεφαλαίου είναι να εξοικειώσει τον αναγνώστη με τις βασικές τεχνολογίες και ορολογίες που θα αναφερθούν στη συνέχεια της διπλωματικής, έτσι ώστε να διευκολυνθεί η κατανόησή τους. Παρακάτω αναφέρονται οι κύριες τεχνολογίες που εξετάστηκαν για την ανάπτυξη της εφαρμογής, ακόμα και αν στο τέλος δεν συμπεριλήφθηκαν στην τελική της υλοποίηση για διάφορους λόγους, όπως θα εξηγηθούν στο οικείο κεφάλαιο.

2.1 Android- Τι είναι το Android;

Το Android είναι ένα λογισμικό για κινητά τηλέφωνα, tablets και για μία μεγάλη γκάμα από wearable συσκευές μέχρι συσκευές ψυχαγωγίας αυτοκινήτου. Ξεκίνησε το 2003 και είναι το πιο διαδεδομένο σύστημα λογισμικού στον κόσμο αυτή την στιγμή.

Το Android είναι ένα open source project, πού εισήχθη από την Google, παρότι δεν της ανήκει. Η Google το χρησιμοποιεί σαν βάση για να δημιουργήσει διαφορετικές εκδόσεις του, οι οποίες στη συνέχεια χρησιμοποιούνται από άλλες εταιρίες. Συνήθως δημιουργεί μία νέα έκδοση κάθε χρόνο. Παρακάτω παρουσιάζονται οι εκδόσεις που έχουν κυκλοφορήσει μέχρι σήμερα με το όνομά τους, τον αριθμό της έκδοσής τους και την ημερομηνία κυκλοφορίας τους.

Code name	Version number	Initial release date
(No codename)[2]	1.0	September 23, 2008
(Internally known as "Petit Four")[2]	1.1	February 9, 2009
Cupcake	1.5	April 27, 2009
Donut[3]	1.6	September 15, 2009
Eclair[4]	2.0 – 2.1	October 26, 2009
Froyo[5]	2.2 – 2.2.3	May 20, 2010
Gingerbread[6]	2.3 – 2.3.7	December 6, 2010
Honeycomb[7]	3.0 – 3.2.6	February 22, 2011
Ice Cream Sandwich[8]	4.0 – 4.0.4	October 18, 2011
Jelly Bean[9]	4.1 – 4.3.1	July 9, 2012
KitKat[10]	4.4 – 4.4.4	October 31, 2013
Lollipop[12]	5.0 – 5.1.1	November 12, 2014
Marshmallow[14]	6.0 – 6.0.1	October 5, 2015
Nougat[15]	7.0 – 7.1.2	August 22, 2016
Oreo[16]	**8.0 – 8.1**	August 21, 2017
Android P[17]	9	May 8, 2018 (beta)

Εικόνα 1: Android versions

Ως λειτουργικό σύστημα, η δουλειά του Android είναι να λειτουργεί σαν ένας μεταφραστής μεταξύ του χρήστη και της συσκευής. Όταν τραβηχτεί μια φωτογραφία, το Android παρέχει το κουμπί για να τραβηχτεί η φωτογραφία και λέει στο κινητό τι να κάνει, όταν ο χρήστης το πατήσει. Όταν πραγματοποιείται ή λαμβάνεται μια κλήση, το Android λέει στο κινητό πώς να το κάνει. Στην ουσία λειτουργεί, όπως και τα Windows για έναν υπολογιστή.

2.2 Android *SDK*

Το Android Software Development Kit (SDK) είναι ένα σύνολο εργαλείων που επιτρέπει στους προγραμματιστές να δημιουργήσουν εφαρμογές για το λειτουργικό σύστημα Android. Περιλαμβάνει τα εξής παρακάτω στοιχεία:

1. Απαιτούμενες βιβλιοθήκες
2. Αποσφαλματοποιητή - Debugger
3. Προσομοιωτή - Emulator
4. Παραδείγματα πηγαίου κώδικα- Sample Source Code
5. Βιβλιογραφία-Documentation

Κάθε φορά που μια νέα έκδοση Android κυκλοφορεί, συνεπάγεται μια αντίστοιχη νέα έκδοση του Android SDK. Συνεπώς, αν κάποιος προγραμματιστής επιθυμεί να φτιάξει μια εφαρμογή με τις τελευταίες προσθήκες του Android, θα πρέπει να έχει την τελευταία έκδοση του Android SDK στο συγκεκριμένο κινητό. Όμως παρόλο που το SDK μπορεί να χρησιμοποιηθεί και από το command prompt, η πιο συνηθισμένη μέθοδος είναι μέσω ενός Integrated development environment(IDE). Παρακάτω ακολουθεί συνοπτική παρουσίαση του IDE που χρησιμοποιήθηκε(Android Studio).

2.3 Android *studio*

Το Android studio αποτελεί το επίσημο IDE για το λογισμικό Android. Είναι βασισμένο στο IntelliJ IDEA και χρησιμοποιείται για να κατασκευαστεί η πλειοψηφία των εφαρμογών που χρησιμοποιεί ο μέσος άνθρωπος σε μια μέρα. Ως γλώσσα προγραμματισμού χρησιμοποιεί την Java, η οποία όμως εγκαθίσταται ξεχωριστά στο κάθε μηχάνημα.

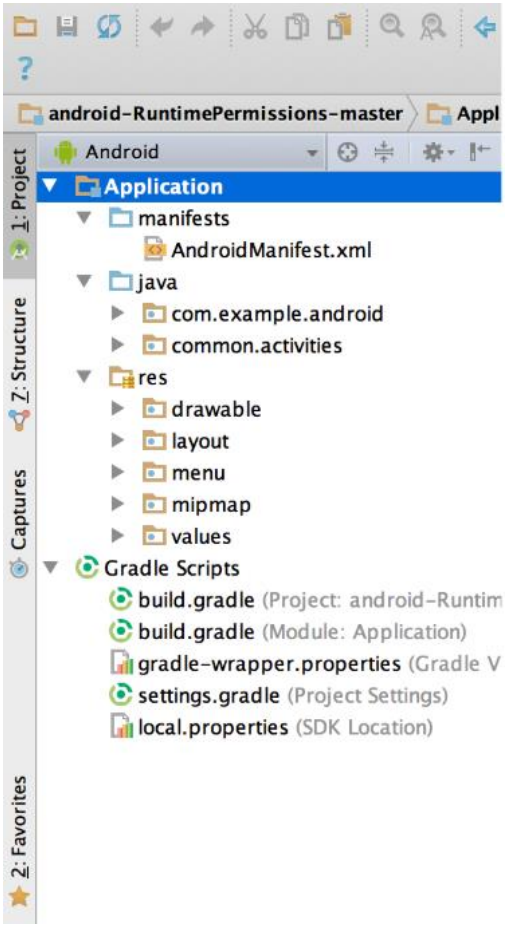
Εικόνα 2: Android Studio

Προσφέρει όλες τις δυνατότητες του IntelliJ IDEA, καθώς και ένα ευέλικτο σύστημα βασισμένο σε gradle, ταχύτατους προσομοιωτές, ένα ενοποιημένο

περιβάλλον στο οποίο μπορεί να αναπτυχθούν εφαρμογές για όλες τις διαφορετικές εκδόσεις android, δείγματα κώδικα και σύνδεση με το Github (έτσι ώστε να είναι πιο εύκολη η κατασκευή συνηθισμένων εφαρμογών ή η εισαγωγή κώδικα), εκτενή εργαλεία και frameworks, εργαλεία για τον έλεγχο της απόδοσης, της χρησιμοποίησης, της συμβατότητας εκδόσεων και άλλων προβλημάτων και τέλος υποστήριξη για C++ και NDK.

2.3.1 Δομή του project

Κάθε project στο Android Studio περιέχει ένα ή περισσότερα modules με πηγαίο κώδικα και resource files. Οι υποστηριζόμενοι τύποι modules είναι Android app module, Library module, Google App engine module. Εξ ορισμού, το Android studio τοποθετεί τα αρχεία της εφαρμογής στο Android studio project view, όπως φαίνεται παρακάτω.



Εικόνα 3: Android Project Structure

Αυτή η δομή χρησιμοποιείται έτσι ώστε να παρέχεται γρήγορη πρόσβαση στα κύρια αρχεία του προγράμματος. Όλα τα built files μπορούν να είναι

προσβάσιμα στο φάκελο Gradle Scripts και το module κάθε εφαρμογής περιέχει τους ακόλουθους φάκελους:

Manifests: Περιέχει το AndroidManifest.xml.

Java: Περιέχει τους πηγαίους κώδικες σε Java, συμπεριλαμβανομένων και των JUnit test codes.

Res: Περιέχει τα υπόλοιπα στοιχεία του προγράμματος εκτός από κώδικες όπως XML layouts, UI strings και Bitmap images.

2.4 *XML*- Extensible Markup Language

Η XML επιτρέπει μια πιο ακριβή, ευέλικτη και εξελίξιμη αναγνώριση πληροφοριών στο διαδίκτυο βελτιώνοντας έτσι την λειτουργικότητα του. Είναι επεκτάσιμη (Extensible) γιατί δεν έχει ένα στάνταρ format σαν την HTML (η οποία είναι μια predefined markup language). Η XML είναι μια μεταγλώσσα- μια γλώσσα που περιγράφει άλλες γλώσσες- η οποία σου επιτρέπει να σχεδιάσεις τις δικές σου markup γλώσσες για διαφορετικού τύπου αρχεία.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.kouva.test1.beacon"
    android:orientation="vertical"
    android:background="#FF69B4"
    android:id="@+id/lay">
    <com.wang.avi.AVLoadingIndicatorView
        android:id="@+id/avi"
        android:layout_width="200dp"
        android:layout_height="200dp"
        style="@style/AVLoadingIndicatorView"
        android:visibility="visible"
        app:indicatorName="BallClipRotateIndicator"
        app:indicatorColor="#000"
        android:layout_gravity="center"
        android:layout_marginTop="100dp"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Searching for question..."
        android:textColor="#000"
        android:layout_marginTop="100dp"
        android:textSize="20dp"
        android:textStyle="bold"
        android:layout_gravity="center"/>
</LinearLayout>
```

Εικόνα 4: XML example

Στο Android χρησιμοποιείται για τη σχεδίαση του γραφικού περιβάλλοντος. Όλη η διεπαφή του χρήστη και η διάταξη της εφαρμογής είναι σχεδιασμένα με την χρήση της XML. Σε αντίθεση με την Java, η οποία είναι η ραχοκοκαλιά της

εφαρμογής, η XML βοηθάει στον σχεδιασμό της εφαρμογής, στο πώς θα φαίνονται και το πού θα τοποθετηθούν οι εικόνες, τα κουμπιά κτλ. Επιπλέον η XML μπορεί να χρησιμοποιηθεί για την ανάλυση δεδομένων είτε από την βάση δεδομένων είτε από κάποιον server.

2.5 SQLite

Η SQLite είναι μία βιβλιοθήκη που υλοποιεί μια αυτόνομη, χωρίς server μηχανή βάσης δεδομένων τύπου SQL. Ο κώδικας της SQLite είναι αναρτημένος σε ένα δημόσιο domain και συνεπώς η χρήση της είναι δωρεάν για όλους τους σκοπούς, ιδιωτικούς ή εμπορικούς. Η SQL είναι η πιο ευρέως διαδεδομένη βάση δεδομένων στον κόσμο με αμέτρητα project, συμπεριλαμβανομένων και μερικών εξαιρετικά υψηλού προφίλ. Η SQLite είναι μια ενσωματωμένη μηχανή βάσης δεδομένων SQL. Σε αντίθεση όμως με τις περισσότερες SQL βάσεις δεδομένων, δεν υποστηρίζει σύνδεση με κάποιο server. Η SQLite πραγματοποιεί τις εγγραφές και τις αναγνώσεις κατευθείαν από τα τοπικά αρχεία δίσκου και ολόκληρη η βάση δεδομένων με τα tables,τα indices, τα views και τα triggers βρίσκονται σε ένα τοπικό αρχείο στον δίσκο. Το file format της, τής επιτρέπει να χρησιμοποιηθεί σε διαφορετικές αρχιτεκτονικές, καθώς και να μεταφερθεί ανάμεσα σε τέτοιες(πχ. από 32-bit σε 64-bit και το αντίστροφο ή από big-endian σε little-endian και το αντίστροφο). Αυτά τα χαρακτηριστικά καθιστούν δημοφιλή την SQLite στις εφαρμογές κινητών, αλλά σε καμία περίπτωση δεν λειτουργεί ως αντικαταστάτης της Oracle.

Εικόνα 5: SQL Database

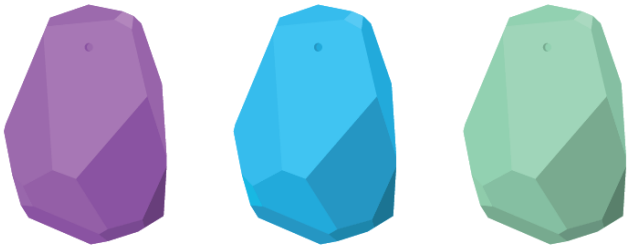
Η SQLite είναι μια βιβλιοθήκη, που με όλα τα χαρακτηριστικά της μπορεί να καταλαμβάνει λιγότερα από 500KiB ανάλογα με την πλατφόρμα και τη βελτιστοποίηση της μεταγλώττισης. Σε γενικές γραμμές, η SQLite τρέχει πιο

γρήγορα με όσο περισσότερη μνήμη έχει στην διάθεσή της. Παρόλα αυτά οι επιδόσεις της είναι αρκετά καλές, ακόμα και σε περιπτώσεις στις οποίες υπάρχουν χαμηλά επίπεδα μνήμης. Επιπλέον, ανάλογα με την χρήση της μπορεί να είναι πιο αποδοτική από το direct σύστημα αρχείων.

Πριν την έκδοσή της, η SQLite τεσταρίστηκε εκτενώς και γι' αυτό τα επίπεδα αξιοπιστίας της είναι τόσο υψηλά. Εξάλλου το μεγαλύτερο κομμάτι του κώδικά της στοχεύει πλήρως στον έλεγχο και στην επαλήθευση της. Επιπλέον, είναι ACID (Atomicity, Consistency, Isolation, Durability), ακόμα και αν κάποιο transaction διακοπεί από έλλειψη μνήμης ή μπαταρίας.

2.6 Beacons

Ένα beacon είναι ένας μικρός ραδιοπομπός σήματος Bluetooth. Λειτουργεί σαν φάρος, και εκπέμπει το σήμα του έτσι ώστε να μπορεί να εντοπιστεί από τις άλλες συσκευές που είναι κοντά του. Το σήμα του εκπέμπεται κάθε μερικά κλάσματα του δευτερολέπτου και αποτελείται από μερικά γράμματα και μερικούς αριθμούς. Όταν μια συσκευή είναι στην εμβέλειά του μπορεί να λάβει αυτό το σήμα και ανάλογα με την ισχύ που το λαμβάνει μπορεί να έχει και μια εκτίμηση για την ακτίνα όπου βρίσκεται.



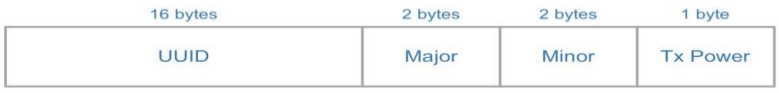
Εικόνα 6: Beacons

Μια σημαντική διαφορά στη χρήση του Bluetooth από τα beacons όμως έγκειται στο γεγονός ότι δεν το χρησιμοποιούν με τον κλασσικό τρόπο της σύνδεσης μεταξύ δύο ή περισσότερων συσκευών. Αντιθέτως, χρησιμοποιούν το Bluetooth σαν κάτι που ονομάζεται "undirected advertising", δηλαδή "διαφημίζουν" τα δεδομένα τους στον αέρα, χωρίς συγκεκριμένο προορισμό και σύνδεση. Αν κάποια συσκευή είναι σε εμβέλεια, μπορεί να τα λάβει χωρίς να δημιουργηθεί κάποια σύνδεση.

Καθώς τα beacons γίνονται όλο και πιο δημοφιλή στον τομέα του indoor positioning, αρκετές εταιρίες έχουν εισέλθει στην αγορά της παροχής τους. Όπως γίνεται εύκολα κατανοητό, οι βασικές αρχές των beacons είναι ίδιες για όλες τις εταιρίες αλλά στον τομέα του λογισμικού, της κατανάλωσης ενέργειας και της τεχνολογίας εντοπισμού παρατηρούνται αρκετές αποκλίσεις. Στην παρούσα διπλωματική θα εστιάσουμε στα beacons που είναι κατασκευασμένα από την Estimote, καθώς αυτά χρησιμοποιήθηκαν για την ανάπτυξη και τον έλεγχο της εφαρμογής.

Κάθε beacon έχει μια συγκεκριμένη περιοχή στην οποία εκπέμπει το σήμα του. Αυτή η περιοχή μπορεί να οριστεί με αρκετούς τρόπους. Στην Estimote μπορείς να προσθέσεις διάφορα tags στα beacons (πχ Καλώς Ήρθατε, Ευχαριστούμε για την προτίμηση σας κτλ) και ανάλογα με το αν μπαίνεις στην εμβέλειά τους ή βγαίνεις, να υπάρχει μια αντίστοιχη ενέργεια. Ένας άλλος τρόπος είναι μέσω κάποιων ατομικών χαρακτηριστικών του κάθε beacon. Τα αναγνωριστικά κάθε beacon αποτελούνται από 3 αριθμούς:

- UUID: 16 bytes, συνήθως αναπαρίσταται σαν String πχ “B9407F30-F5F8-466E-AFF9-25556B57FE6D”.
- Major number: 2 bytes ή ένας μη προσιμασμένος πχ ένας αριθμός από το 1 έως το 65535.
- Minor number: 2 bytes το ίδιο σαν το Major.



Εικόνα 7: Beacons identifiers

Το UUID είναι ορισμένο εξ αρχής, ενώ τα major και minor numbers είναι τυχαίοι αριθμοί. Οι τρόποι ορισμού μιας περιοχής με βάση τα ατομικά χαρακτηριστικά μιας ομάδας beacon είναι οι εξής:

1. Μέσω του UUID: Κάθε beacon έχει ένα προεπιλεγμένο UUID το οποίο όμως μπορεί να αλλάξει ο ιδιοκτήτης του. Μία περιοχή θα μπορούσε να είναι όλα τα beacons στα οποία δεν έχει γίνει κάποια αλλαγή στο UUID τους.

2. Μέσω του UUID και του Major: Αποτελείται από όλα τα beacons των οποίων ο UUID δεν έχει αλλάξει και έχουν έναν συγκεκριμένο Major.

3. Μέσω του UUID, του Major και του Minor: Αποτελείται από ένα συγκεκριμένο beacon με αυτούς τους τρείς αριθμούς.

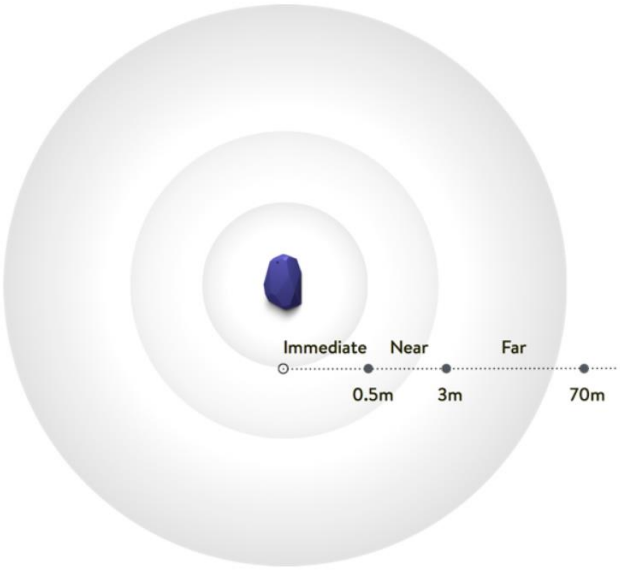
Μπορούμε με δύο τρόπους να χρησιμοποιήσουμε τις περιοχές των beacons στο indoor positioning. Ο πρώτος είναι το beacon region monitoring.

Το monitoring επιτρέπει στην εφαρμογή να γνωρίζει το πότε εισέρχεται και πότε εξέρχεται από μία περιοχή beacon. Το σημαντικό με αυτή την τεχνική είναι ότι δεν απαιτεί από την εφαρμογή να τρέχει εκείνη την στιγμή. Θα μπορούσε κάλλιστα να είναι στο παρασκήνιο ή να μην τρέχει καθόλου. Για παράδειγμα ένας επισκέπτης εισέρχεται στο μουσείο και μπαίνει στο range των beacons της εισόδου του μουσείου. Τότε η εφαρμογή το καταλαβαίνει και τρέχει από μόνη της για να τον ενημερώσει για την παροχή audio guide ή για κάποια ειδική έκθεση που λαμβάνει χώρο εκείνη την στιγμή.

Εικόνα 8: Beacon monitoring

Τα μειονεκτήματα του monitoring εντοπίζονται στο ότι δεν καταλαβαίνει αμέσως τις αλλαγές στις περιοχές, λόγω του ότι χρησιμοποιεί low-power scanning. Επίσης δεν προσφέρει καμία πληροφορία για την απόσταση του χρήστη από συγκεκριμένα beacons, παρά μόνο αναγνωρίζει γεγονότα εισόδου και εξόδου.

Ο δεύτερος τρόπος είναι το beacon region ranging. Σε αντίθεση με το monitoring το ranging επιστρέφει μία λίστα με όλα τα beacon στην περιοχή και με μία προσέγγιση για την απόστασή τους.



Εικόνα 9: Beacon ranging

Για παράδειγμα, με το beacon ranging ο επισκέπτης θα μπορούσε να πλησιάσει το beacon ενός εκθέματος και η εφαρμογή να του παρέχει αμέσως πληροφορίες γι' αυτό το έκθεμα, αφού θα λαμβάνει πιο ισχυρά το σήμα του συγκεκριμένου beacon. Βέβαια τα αποτελέσματα αυτής της τεχνικής μπορεί να επηρεάζονται από τη θέση του τηλεφώνου, αλλά και από τη θέση του beacon, μιας και για τα βέλτιστα αποτελέσματα θα πρέπει να υπάρχει μια γραμμή επικοινωνίας χωρίς παρεμβολές ανάμεσα στο τηλέφωνο και το beacon. Επίσης το ranging καταναλώνει σημαντικά μεγαλύτερο κομμάτι της μπαταρίας των beacon από το monitoring.

2.6 Indoor Atlas

Η τεχνολογία indoor atlas αναπτύχθηκε από το πανεπιστήμιο του Oulu της Φιλανδίας και παρέχει μια διαφορετική λύση για το indoor positioning. Έχει εμπνευστεί από το ζωϊκό βασίλειο και από το πώς τα ζώα βρίσκουν τον προσανατολισμό τους μέσω του βαρυντικού πεδίου της γης και στη συνέχεια εξελίχθηκε με την προσθήκη διαφόρων τεχνολογιών της σημερινής εποχής.

Εικόνα 10: Indoor Atlas

Οι τεχνολογίες που συνδυάζει είναι οι εξής: Bluetooth beacons, Wi-Fi signal και μαγνητικά σήματα. Οι πληροφορίες που λαμβάνει από τις παραπάνω

τεχνολογίες συνδυάζονται με πληροφορίες που είναι αποθηκευμένες στο cloud του indoor atlas και με τη χρήση machine learning παρέχεται μία εκτίμηση για τη θέση του χρήστη στον χώρο.

Εικόνα 11: Mapping quality

Πιο συγκεκριμένα, κάθε σύγχρονο κτήριο στη Γη έχει ένα μοναδικό μαγνητικό αποτύπωμα, χάρη στα υλικά από τα οποία είναι κατασκευασμένο, τα οποία είναι κυρίως μεταλλικά. Έτσι με τη χρήση διαφόρων μαγνητικών τεχνολογιών που υπάρχουν στα σημερινά smartphones (πχ. πυξίδα, γυροσκόπιο,), η εφαρμογή μπορεί να δημιουργήσει ένα map με βάση το μαγνητικό πεδίο μέσα στο κάθε κτήριο και να προσφέρει μια εκτίμηση για τη θέση του χρήστη. Η χρήση των beacons, όπως έχει αναλυθεί παραπάνω, μπορεί να βελτιώσει την ακρίβεια που παρέχει η εφαρμογή μαζί με τον τριγωνισμό του σήματος Wi-Fi, μια από τις πρώτες τεχνολογίες που χρησιμοποιήθηκαν στο indoor positioning.

Η τεχνολογία αυτή έχει ιδιαίτερη χρήση σε μεγάλους χώρους, όπως νοσοκομεία, αεροδρόμια ή shopping malls, που μπορούν να παρέχουν τις παραπάνω πληροφορίες σε πλεονάζοντα βαθμό.

2.7 Machine learning

Στις μέρες μας ο τομέας του machine learning είναι ίσως ο πιο αναπτυσόμενος τομέας σε ότι αφορά την πληροφορική. Αποτελεί την προσπάθεια της εκμάθησης ενός υπολογιστή έτσι ώστε να μάθει και να συμπεριφέρεται όπως κάνουν οι άνθρωποι. Επίσης, περιλαμβάνει την βελτίωση της συμπεριφοράς των υπολογιστών με την παροχή νέων δεδομένων και πληροφοριών.

2.7.1 Tensorflow

Είναι μια τεχνολογία ανοιχτού λογισμικού που αναπτύχθηκε από την Google, για αριθμητικούς υπολογισμούς μέσω διαγραμμάτων ροής δεδομένων. Αρχικά σχεδιάστηκε για ερευνητικούς σκοπούς στους τομείς του machine learning και των deep neural networks, αλλά μπορεί να χρησιμοποιηθεί σε ένα μεγάλο εύρος τομέων. Υποστηρίζεται από πολλές πλατφόρμες και τρέχει σχεδόν παντού (GPUs και CPUs), συμπεριλαμβανομένων και των κινητών τηλεφώνων. Για την ανάπτυξη του έχει χρησιμοποιηθεί η γλώσσα προγραμματισμού C++, αλλά στο front-end υποστηρίζεται και η Python. Οι κύριες εφαρμογές του είναι:

- Αναγνώριση Ήχου/Φωνής.
- Αναγνώριση Εικόνας.
- Video Detection.
- Time series.
- Αναγνώριση Κειμένου.

3

Σχεδιασμός

Οι εφαρμογές κινητών τηλεφώνων εξαρτώνται πλέον σε τεράστιο βαθμό από την ύπαρξη ενός φιλικού περιβάλλοντος χρήστη. Μια εφαρμογή είναι χρήσιμη, μόνο εάν ο χρήστης μπορεί να τη χρησιμοποιήσει με ευκολία.

Σε αυτό το κεφάλαιο παρουσιάζονται τα βασικά συστατικά που έπαιξαν ρόλο στη σχεδίαση και ανάπτυξη της εφαρμογής. Επίσης παρέχονται διαγράμματα με σκοπό την καλύτερη κατανόηση της λειτουργίας και της δομής της εφαρμογής, καθώς τις περισσότερες φορές μια γραφική απεικόνιση είναι πολύ πιο αποδοτική από μια παράγραφο.

3.1 Ιστορίες

Κατά μέσο όρο, για τους περισσότερους επισκέπτες ενός μουσείου μια ξενάγηση στον χώρο τους περιλαμβάνει μια απλή και γρήγορη παρατήρηση των εκθεμάτων τους. Συνεπώς, μετά το τέλος της περιήγησης, ο επισκέπτης δυσκολεύεται να συγκρατήσει τις πληροφορίες που τον ενδιαφέρουν. Επίσης είναι δύσκολο σε τόσο λίγο χρόνο, ένας μη εξοικειωμένος επισκέπτης να κατανοήσει από μόνος του τις λεπτομέρειες και το χρονικό πλαίσιο των εκθεμάτων. Γι' αυτό τον λόγο, η ύπαρξη μιας ιστορίας και ενός παιχνιδιού μπορεί να οδηγήσει στην

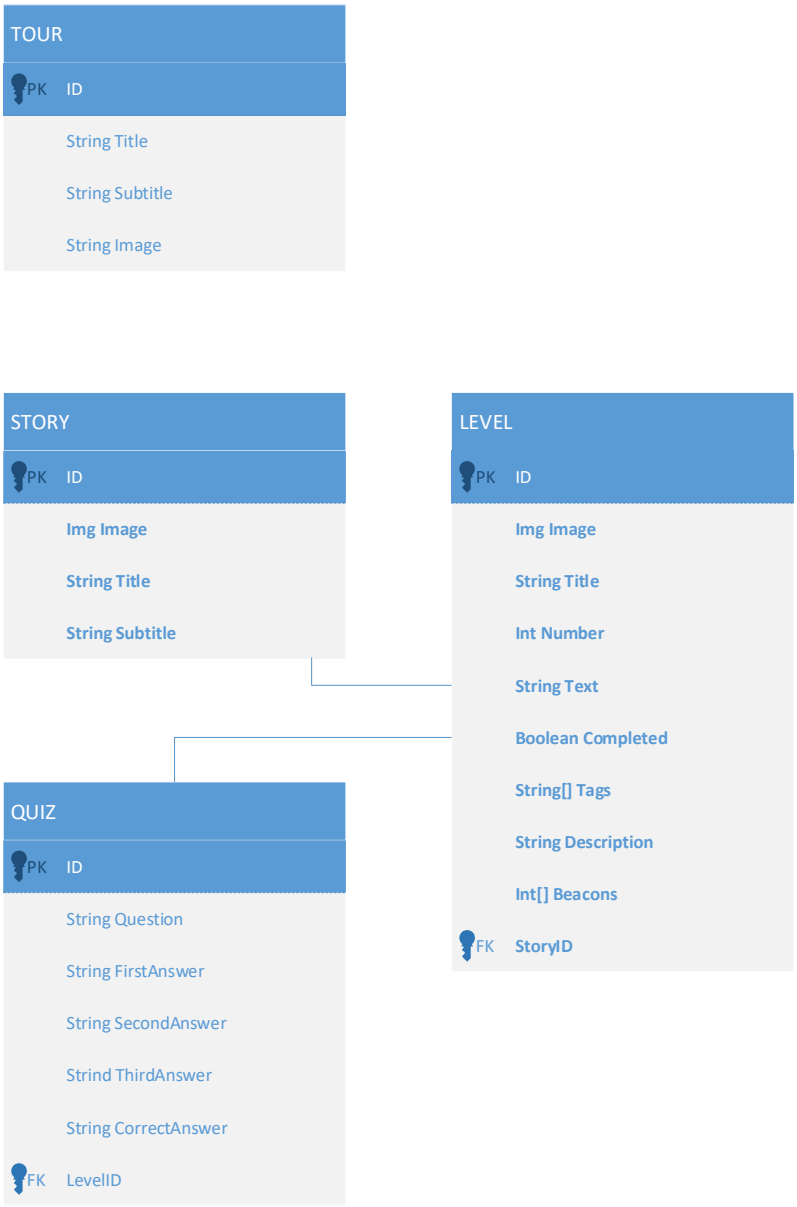
κατακόρυφη αύξηση του ενδιαφέροντος και της εμπειρίας του επισκέπτη, ενώ ταυτόχρονα θα του προσδώσει και περισσότερη διασκέδαση.

Για να πραγματοποιηθεί ο παραπάνω σκοπός, αποφασίστηκε να υλοποιηθεί, ένα παιχνίδι σαν κυνήγι θησαυρού, και να συμπεριληφθεί στην εφαρμογή. Σε αυτό το παιχνίδι, ο χρήστης θα ακολουθεί τον Αχμέτ, ο οποίος ξεκινάει με το άλογό του, για να βρει την ιδανική κοπέλα και να την παντρευτεί. Το παιχνίδι αυτό στο αρχικό του στάδιο έχει σχεδιαστεί έτσι ώστε να παίζεται από παιδιά νηπιαγωγείου με την επίβλεψη του προσωπικού του Μουσείου, με βάση τους πίνακες του τρίτου και του τέταρτου ορόφου. Πιο αναλυτικά το παιχνίδι περιλαμβάνει:

- Διάδραση των παιδιών με τα εκθέματα του μουσείου για την εξέλιξη της ιστορίας.
- Εύρεση εκθεμάτων μέσα στο μουσείο με βάση τη φωτογραφία τους.
- Εξαγωγή περισσότερων πληροφοριών για τα συγκεκριμένα εκθέματα.
- Εύρεση στοιχείων κοντά ή μέσα σε ένα συγκεκριμένο έκθεμα.
- Απάντηση σε γρίφο για τη συνέχεια της ιστορίας.

3.2 Βάση δεδομένων – SQLite

Στο παρακάτω σχήμα φαίνεται το διάγραμμα σχέσεων-οντοτήτων (entity-relation diagram) στο οποίο βασίστηκε η τύπου SQL βάση δεδομένων της εφαρμογής. Εδώ χρειάζεται να υπενθυμίσουμε ότι η βάση δεδομένων είναι τοπική στην μνήμη της εφαρμογής και δεν συνδέεται με κάποιον server προς το παρόν. Κάθε table έχει ως primary key ένα ακέραιο ID, το οποίο δημιουργείται αυτόματα, μόλις δημιουργήσουμε το table. Στο table TOURS αποθηκεύονται πληροφορίες για τις ξεναγήσεις που προσφέρει το μουσείο. Προς το παρόν περιέχει μόνο τον τίτλο, μια μικρή περιγραφή (υπότιτλο) και μια ενδεικτική εικόνα. Στο table STORY αποθηκεύονται πληροφορίες για κάθε ιστορία που προσφέρεται. Πιο συγκεκριμένα, κάθε ιστορία θα έχει έναν τίτλο, μια σύντομη περιγραφή (υπότιτλο) και μια εικόνα. Στην συνέχεια υπάρχει το table LEVEL, όπου περιλαμβάνει τα επίπεδα της κάθε ιστορίας του table STORY. Ειδικότερα, κάθε διαφορετικό επίπεδο θα έχει την εικόνα του πίνακα ή του εκθέματος που θα αντιπροσωπεύει, το όνομα αυτού, έναν ακέραιο που θα δείχνει τη θέση του στην ιστορία, το κείμενο που θα χρησιμοποιηθεί για την αναζήτηση του εκθέματος, μια μεταβλητή λογικού τύπου για το αν έχει ολοκληρωθεί από τον χρήστη ή όχι, έναν πίνακα από λέξεις κλειδιά-tags που θα χαρακτηρίζουν το έκθεμα, έναν πίνακα με τα κοντινά σε αυτό beacons και τέλος έναν ακέραιο σαν foreign key που θα το συνδέει με την ιστορία του στο table STORY. Τέλος το τελευταίο table θα είναι το QUIZ, το οποίο θα αποθηκεύει την ερώτηση του κάθε LEVEL, δηλαδή θα έχει ένα πεδίο με την ερώτηση, στην συνέχεια 3 πεδία με κάθε πιθανή απάντηση, και ακόμα ένα με το ποια είναι η σωστή. Επίσης, θα έχει και αυτό έναν ακέραιο σαν foreign key, έτσι ώστε να συνδέεται με το LEVEL που αντιπροσωπεύει.



Εικόνα 12: Database schema

3.3 UML

Ένα UML διάγραμμα είναι ένα διάγραμμα βασισμένο στην γλώσσα UML (Unified Modeling Language) με σκοπό να αναπαραστήσει οπτικά ένα σύστημα με τους κύριους του ρόλους, δράσεις, κλάσεις, έτσι ώστε να είναι πιο εύκολη η κατανόηση, η συντήρηση και η επέκτασή του. Τα UML διαγράμματα είναι μια σχετικά μοντέρνα προσέγγιση, βασισμένη σε μια διαγραμματική απεικόνιση των στοιχείων του λογισμικού. Η κύρια χρήση τους είναι αδιαμφισβήτητα στον τομέα του λογισμικού, παρόλα αυτά σήμερα αρχίζουν και βρίσκουν εφαρμογή και σε διάφορες επιχειρησιακές δραστηριότητες και workflows. Για παράδειγμα, τα διαγράμματα δραστηριότητας, ένας τύπος διαγράμματος UML, μπορούν να

χρησιμοποιηθούν για την αντικατάσταση των διαγραμμάτων ροής, καθώς προσφέρουν έναν πιο σταθερό τρόπο αναπαράστασης και μια ευρεία γκάμα χαρακτηριστικών, τα οποία βελτιώνουν την αναγνωσιμότητα και την αποτελεσματικότητα.

3.3.1 Διάγραμμα κλάσεων-Class diagram

Τα διαγράμματα κλάσεων είναι τα πιο δημοφιλή διαγράμματα σε ότι αφορά το λογισμικό. Αυτό οφείλεται στο γεγονός ότι στις μέρες μας οι οδηγούμενες από το γεγονός γλώσσες (Object-Oriented languages) είναι πολύ διαδεδομένες και χρησιμοποιούνται κατά κόρον.

Το ακόλουθο διάγραμμα δεν είναι το πλήρες διάγραμμα κλάσεων για λόγους απλότητας. Στο παρακάτω διάγραμμα παρουσιάζονται συνοπτικά οι κυριότερες κλάσεις της εφαρμογής και οι σχέσεις μεταξύ τους. Το μπλε χρώμα συμβολίζει ότι η κλάση πρόκειται για δραστηριότητα, το κίτρινο ότι αποτελεί adapter και το πορτοκαλί ότι αποτελεί κλάση τύπου δεδομένων. Παρατηρείται ότι η κύρια κλάση είναι η Database Adapter, μιας και σε αυτή περιέχονται όλα τα έτοιμα query που χρησιμοποιούνται από όλες τις άλλες κλάσεις με σκοπό να πάρουν πληροφορίες από τη βάση δεδομένων. Η αρχική κλάση που δημιουργείται με την έναρξη της εφαρμογής είναι η Main Activity.

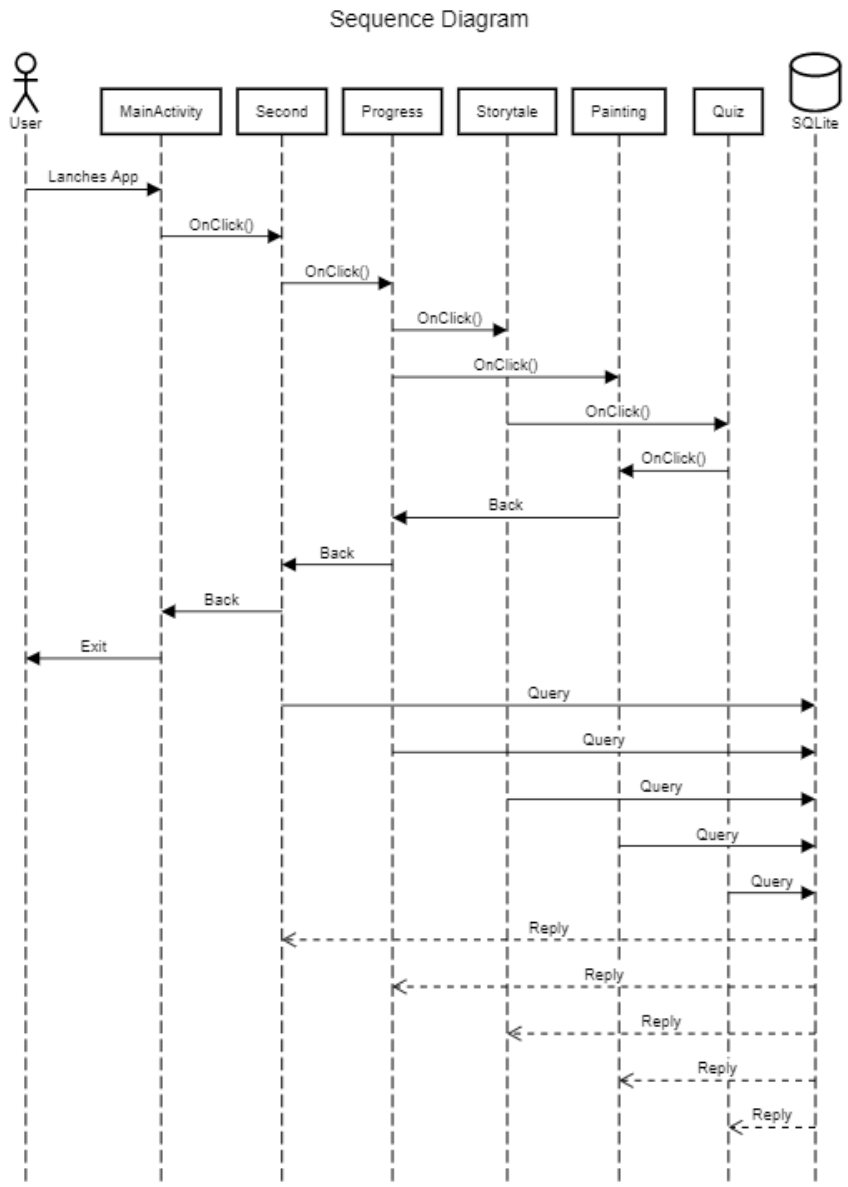
Για τη δημιουργία του διαγράμματος κλάσεων χρησιμοποιήθηκε το Code Iris το οποίο είναι ένα plugin εργαλείο του Android studio και στη συνέχεια επιλέχθηκε το κομμάτι με τις παρακάτω κλάσεις, από τις 47 που υπήρχαν συνολικά στο project.

Εικόνα 13: Class diagram

3.3.2 Ακολουθιακό διάγραμμα-Sequence diagram

Τα ακολουθιακά διαγράμματα (sequence diagrams) δείχνουν το πώς αλληλεπιδρούν τα στοιχεία με βάση μια χρονική ακολουθία. Αναπαριστούν τα αντικείμενα και τις κλάσεις που παίρνουν μέρος σε μία εφαρμογή και τους τύπους μηνυμάτων που ανταλλάσσονται μεταξύ τους, έτσι ώστε η εφαρμογή να λειτουργήσει σύμφωνα με τις προδιαγραφές της. Επίσης είναι συχνά συνδεδεμένα με τα διαγράμματα χρήσης (use case), ενώ μερικές φορές μπορεί να χρησιμοποιηθούν ως event diagrams ή event scenarios.

Το παρακάτω διάγραμμα αποτελεί το ακολουθιακό διάγραμμα της εφαρμογής: Τα ορθογώνια παραλληλόγραμμα απεικονίζουν τις κλάσεις/αντικείμενα που χρησιμοποιούνται, ο άνθρωπος απεικονίζει τον χρήστη που εκκινεί την εφαρμογή, τα βέλη χωρίς διακεκομμένες γραμμές και με μαυρισμένο το βέλος σημαίνουν σύγχρονο μήνυμα, ενώ τα βέλη με διακεκομμένες γραμμές σημαίνουν απάντηση. Το παρακάτω διάγραμμα επιτρέπει στον χρήστη την κατανόηση ενός απλού runtime σεναρίου υπό τη μορφή διαγράμματος.

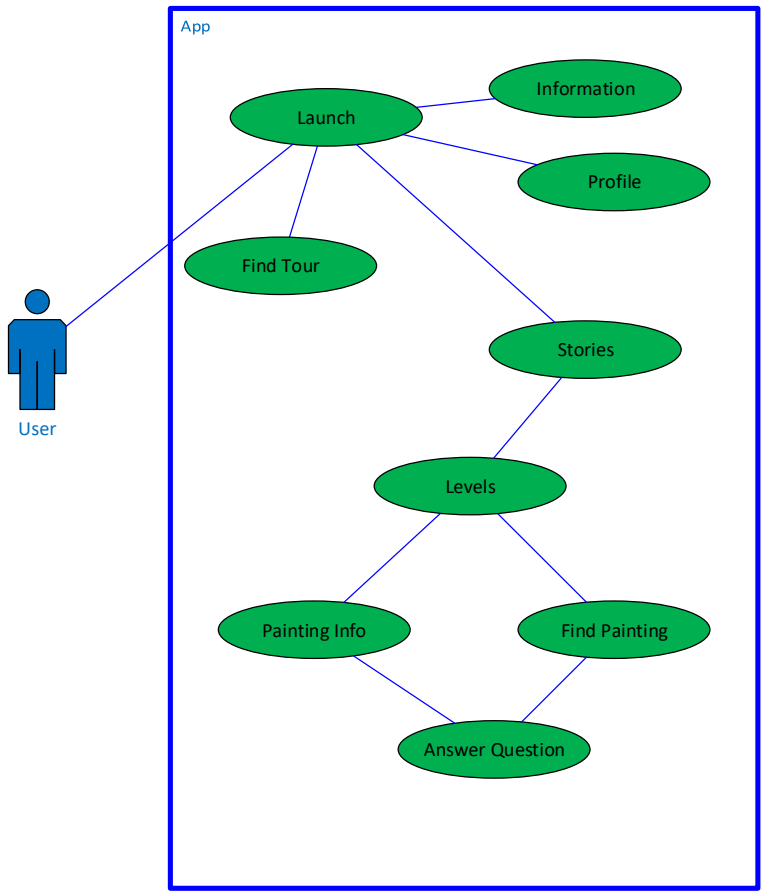


Εικόνα 14: Sequence diagram

3.3.3 Διάγραμμα Περιπτώσεων χρήσης - Use-case diagram

Τα διαγράμματα περιπτώσεων χρήσης στην πιο απλή τους μορφή αποτελούν μια αναπαράσταση της αλληλεπίδρασης ενός χρήστη με ένα σύστημα η οποία δείχνει τις σχέσεις μεταξύ του χρήστη και των διαφόρων περιπτώσεων χρήσης στις οποίες συμμετέχει. Τα πλεονεκτήματα τέτοιου τύπου διαγραμμάτων εντοπίζονται κυρίως στην απλότητά τους και στο γεγονός ότι μπορούν να προσφέρουν μια οπτική του τι μπορεί να κάνει το σύστημα σε μια πολύ απλή και κατανοητή μορφή. Γιαυτό και έχουν χαρακτηριστεί και ως η ιδανική μορφή επικοινωνίας ανάμεσα σε προγραμματιστές και μετόχους.

Παρακάτω παρατίθεται το διάγραμμα περιπτώσεων χρήσης της εφαρμογής.



Εικόνα 15: Use-case diagram

3.4 Περιβάλλον χρήστη – User Interface

Στην σημερινή εποχή η οποία χαρακτηρίζεται από πληθώρα εφαρμογών κινητού τηλεφώνου, το User Interface έχει χαρακτηριστεί από πολλούς προγραμματιστές ως ένα από τα πιο σημαντικά κομμάτια μιας εφαρμογής. Είναι άλλωστε η πρώτη επαφή του χρήστη με την εφαρμογή και θα πρέπει να εξασφαλιστεί η ικανοποίηση, αλλά και η αφοσίωση του χρήστη μέσω της εύκολης χρήσης της. Ένα καλά σχεδιασμένο περιβάλλον χρήστη θα οδηγήσει σίγουρα στην αύξηση του αριθμού χρηστών, αλλά θα οδηγήσει και στην αύξηση του μέσου χρόνου χρησιμοποίησης της εφαρμογής από αυτούς.

3.4.1 Περιβάλλον χρήστη εφαρμογής

Για να διευκολυνθεί κατά το μέγιστο ο χρήστης, έχει επιλεχθεί να χρησιμοποιηθεί σε αρκετά σημεία της εφαρμογής το TabHost. Το TabHost επιτρέπει την παρουσίαση διαφορετικών layouts στην ίδια ουσιαστικά δραστηριότητα-οθόνη. Για συμβατότητα με την τεχνολογία Android, το TabHost τοποθετήθηκε στην

κορυφή της οθόνης, καθώς στην πλειοψηφία των εφαρμογών υπάρχει εκεί. Παρέχει μια σύνδεση με 4 λειτουργίες της εφαρμογής.

- Περιηγήσεις
- Πληροφορίες
- Ιστορίες
- Προφίλ

Εικόνα 16: TabHost

Επίσης για μεγαλύτερη διευκόλυνση του χρήστη έχουν τοποθετηθεί εικόνες στο TabHost που να αντιπροσωπεύουν την κάθε λειτουργία.

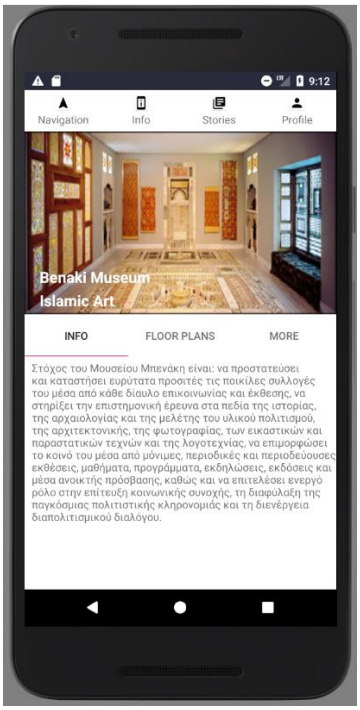
3.4.2 Περιηγήσεις

Η οθόνη των περιηγήσεων έχει σχεδιαστεί, έτσι ώστε να παίρνει όλες τις προσφερόμενες περιηγήσεις δυναμικά από την SQLite βάση δεδομένων και να τις εμφανίζει στον χρήστη σαν μία Scrollable Recycle View μέσω ενός Adapter. Ο χρήστης θα μπορεί να κάνει Scroll μέχρι να βρει την επιθυμητή περιήγηση και να κάνει κλικ πάνω της για ξεκινήσει. Επίσης ορίστηκε ως αρχική οθόνη της εφαρμογής μετά την εκκίνησή της.

Εικόνα 17: Οθόνη περιηγήσεων

3.4.3 Πληροφορίες

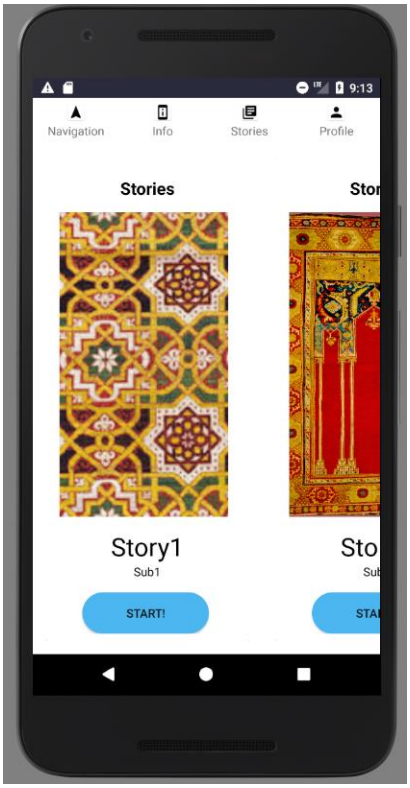
Στην οθόνη των πληροφοριών, η οποία είναι η επόμενη στο TabHost μετά την περιήγηση, θα παρέχονται πληροφορίες για το Μουσείο σε ένα ξεχωριστό/εσωτερικό TabHost, όπως ο σκοπός του, το ωράριο του, τα τηλέφωνα επικοινωνίας κτλ. Επίσης θα υπάρχουν ενδεικτικά πλάνα ορόφων και ένα κουμπί για την τοποθεσία του που θα οδηγεί στους χάρτες της Google για ευκολότερη πρόσβαση στο Μουσείο. Το εσωτερικό TabHost δημιουργήθηκε με σκοπό την διευκόλυνση του χρήστη, καθώς οι απαραίτητες πληροφορίες για το Μουσείο δεν χωρούσαν σε μία οθόνη, συνεπώς με τον χωρισμό σε 3 ουσιαστικά οθόνες μειώθηκε αισθητά το Scroll που θα κάνει ο χρήστης για να βρει τη ζητούμενη πληροφορία.



Εικόνα 18: Οθόνη Πληροφοριών

3.4.4 Ιστορίες

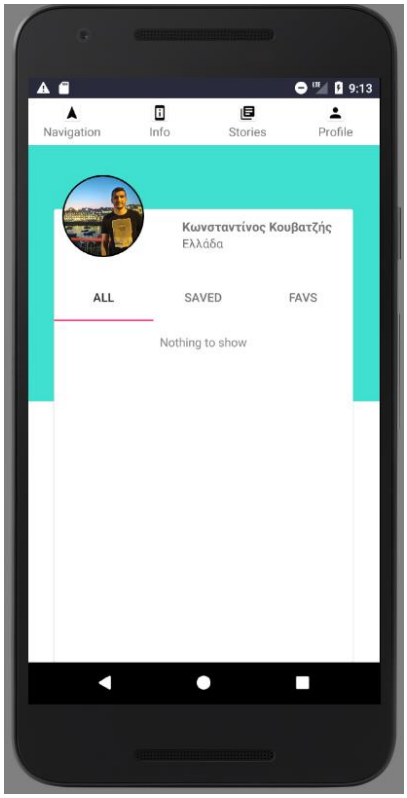
Στην οθόνη αυτή παρουσιάζονται οι *Ιστορίες* που λειτουργούν σαν παιχνίδι χαμένου θησαυρού. Όπως και στην οθόνη των περιηγήσεων, έτσι και εδώ οι ιστορίες φορτώνονται δυναμικά από την βάση δεδομένων και παρουσιάζονται στον χρήστη μέσω ενός Adapter. Εδώ δεν υλοποιήθηκε το κάθετο Recycle View, καθώς ο προβλεπόμενος αριθμός ιστοριών δεν είναι μεγάλος και συνεπώς προτιμήθηκε το οριζόντιο για να προσφέρει μια μικρή διαφοροποίηση στην εφαρμογή. Επίσης έχει υλοποιηθεί εστίαση σε κάθε εικόνα ιστορίας, υπό τα πρότυπα του play store, έτσι ώστε αν το Scroll του χρήστη δεν καταλήξει ακριβώς σε κάποια εικόνα να γίνει άμεσο focus σε αυτή την εικόνα. Κάνοντας κλικ στο κουμπί Εκκίνηση! ο χρήστης θα οδηγείται στη οθόνη της προόδου του στη συγκεκριμένη ιστορία.



Εικόνα 19: Οθόνη Ιστοριών

3.4.5 Προφίλ

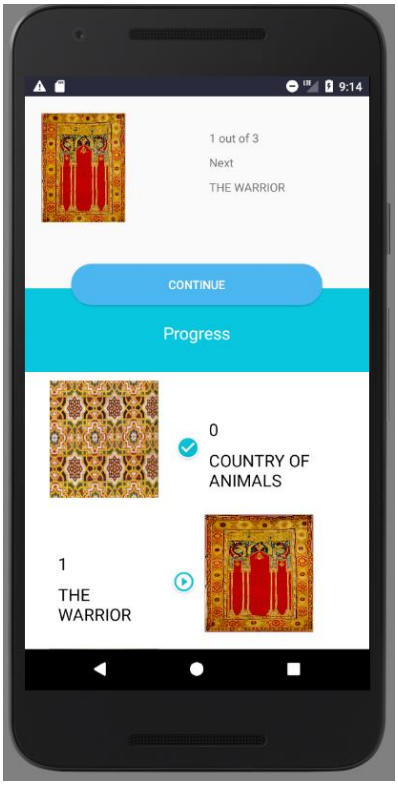
Στην οθόνη του προφίλ, ο χρήστης θα μπορεί να προσθέτει την εικόνα του, η οποία θα εμφανίζεται στρογγυλεμένη μαζί με το όνομα του και τη χώρα του. Επίσης και σε αυτή την οθόνη έχει υλοποιηθεί ένα TabHost για διευκόλυνση του χρήστη. Στο συγκεκριμένο θα εμφανίζονται τα εκθέματα που αρέσουν στον χρήστη, εκείνα που έχει αποφασίσει να αποθηκεύσει στην συσκευή του, αλλά και ο συνδυασμός αυτών των δύο.



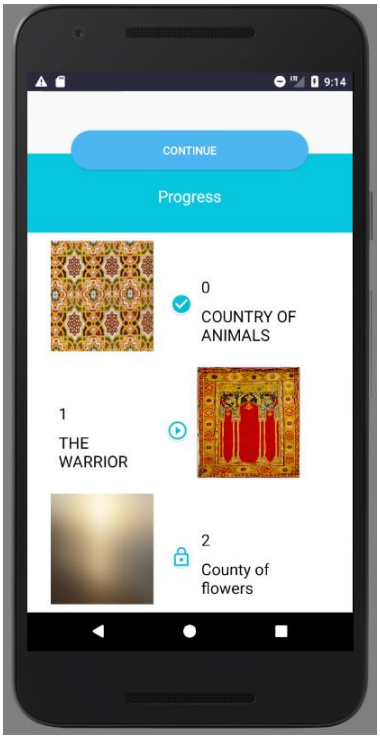
Εικόνα 20: Οθόνη Προφίλ

3.4.6 Πρόοδος

Ο χρήστης θα μπορεί να προηγηθεί στην οθόνη της προόδου του, αφού κάνει κλικ σε κάποια από τις ιστορίες του Μουσείου. Τα επίπεδα της κάθε ιστορίας μαζί με την εικόνα της, την περιγραφή της, το αν είναι ολοκληρωμένη ή όχι κτλ. φορτώνονται πάλι δυναμικά από τη βάση δεδομένων. Επίσης, όπως φαίνεται στην παρακάτω εικόνα, ανάλογα με το αν η θέση του επιπέδου είναι άρτια ή περιττή χρησιμοποιείται διαφορετικό layout. Στα μη ολοκληρωμένα επίπεδα, η εικόνα εμφανίζεται θολή και το κουμπί έναρξης είναι κλειδωμένο, στα ολοκληρωμένα επίπεδα η εικόνα εμφανίζεται κανονικά μαζί με το χαρακτηριστικό τικ της ολοκλήρωσης. Σε περίπτωση που πατηθεί το τικ, ο χρήστης θα οδηγείται στην οθόνη του εκθέματος η οποία αναλύεται παρακάτω. Με την επιλογή των κουμπιών συνέχεια ή παίξε (το οποίο προφανώς είναι μοναδικό σε κάθε ιστορία) ο χρήστης θα οδηγείται στην οθόνη με το κείμενο του επιπέδου. Τέλος στο πάνω μέρος της οθόνης παρέχονται μερικά στοιχεία για την πρόοδο της Ιστορίας, όπως ο επόμενος πίνακας, ο αριθμός των ολοκληρωμένων επιπέδων και το όνομα του επόμενου επιπέδου.



Εικόνα 21: Οθόνη Προόδου 1

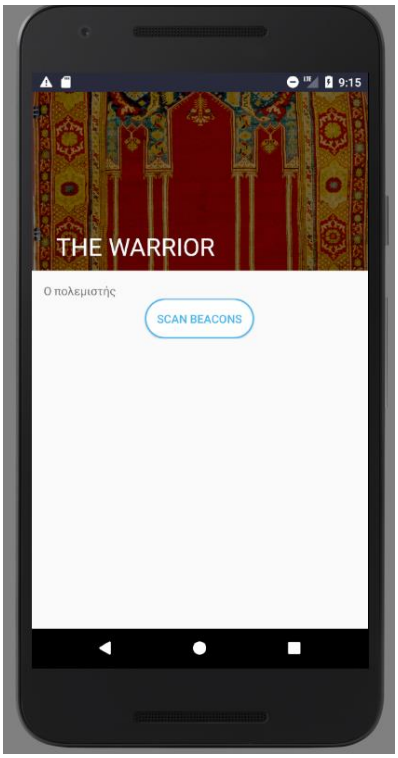


Εικόνα 22: Οθόνη Προόδου 2

3.4.7 Κείμενο Ιστορίας

Εδώ ο χρήστης θα μπορεί να δει το κείμενο του επιπέδου που έχει επιλέξει, καθώς και της εικόνας του εκθέματος, έτσι ώστε να το αναζητήσει στο Μουσείο. Πάλι όλα τα στοιχεία για αυτήν την οθόνη έρχονται δυναμικά από τη βάση

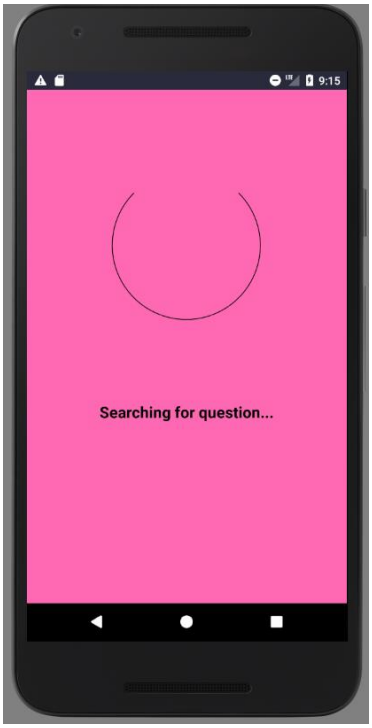
δεδομένων μας. Όταν ο χρήστης πιστεύει ότι έχει εντοπίσει το σωστό έκθεμα, θα πρέπει να πατήσει το κουμπί Scan Beacons, έτσι ώστε να προχωρήσει στην οθόνη αναζήτησης του σωστού beacon.



Εικόνα 23: Οθόνη Κειμένου Ιστορίας

3.4.8 Αναζήτηση

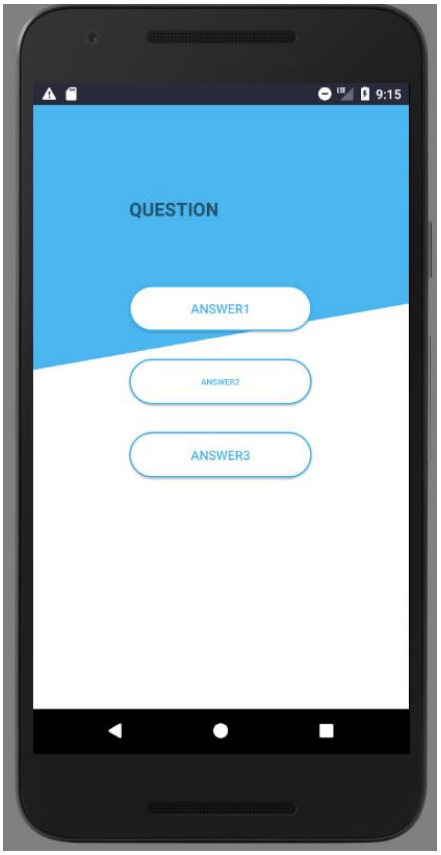
Στην παρούσα οθόνη, η οποία οπτικά είναι η πιο απλή της εφαρμογής, καθώς η κύρια λειτουργικότητά της είναι στο background, η εφαρμογή λαμβάνει το σήμα από τα beacons και αξιολογεί αν ο χρήστης είναι στο σωστό σημείο για το επίπεδο που παίζει αυτή την στιγμή. Ο έλεγχος αυτός θα κρατάει μερικά δευτερόλεπτα, καθώς θέλουμε να αποφύγουμε λανθασμένα συμπεράσματα από το ranging των beacons. Σε περίπτωση που δεν ληφθούν τα σωστά Bluetooth σήματα, τότε ο χρήστης θα οδηγείται πάλι στην οθόνη του Κειμένου της Ιστορίας, έτσι ώστε να αναζητήσει ξανά το σωστό έκθεμα. Σε αντίθετη περίπτωση, θα προχωρά στην οθόνη της ερώτησης για το έκθεμα.



Εικόνα 24: Οθόνη Αναζήτησης

3.4.9 Ερώτηση

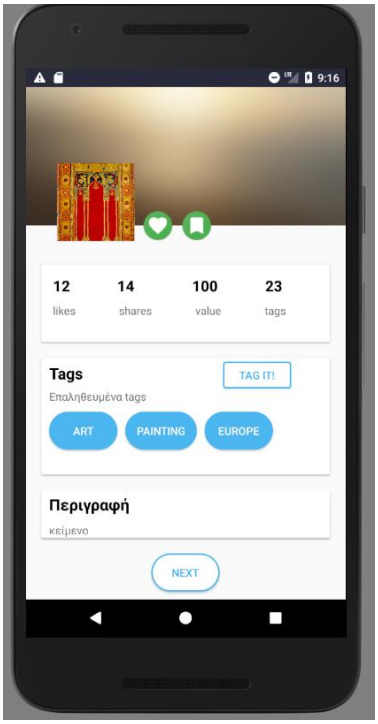
Η συγκεκριμένη οθόνη αποτελεί την τελευταία οθόνη για την ολοκλήρωση ενός επιπέδου. Και αυτή η οθόνη είναι σχετικά απλή, όπως και εκείνη της αναζήτησης. Εδώ ο χρήστης θα βλέπει μια ερώτηση για το έκθεμα, που εντόπισε, μαζί με τρεις απαντήσεις, από τις οποίες μόνο η μια θα είναι σωστή. Αν πατήσει λανθασμένη απάντηση, τότε το μπλε περίγραμμά της θα γίνεται κόκκινο και θα πρέπει να επιλέξει μια άλλη απάντηση. Σε περίπτωση που επιλέξει τη σωστή απάντηση, το περίγραμμα θα γίνεται πράσινο για μερικά δευτερόλεπτα και στη συνέχεια ο χρήστης θα μεταφέρεται στην οθόνη του εκθέματος για να δει περισσότερες πληροφορίες για το έκθεμα.



Εικόνα 25: Οθόνη Ερώτησης

3.4.10 Οθόνη Εκθέματος

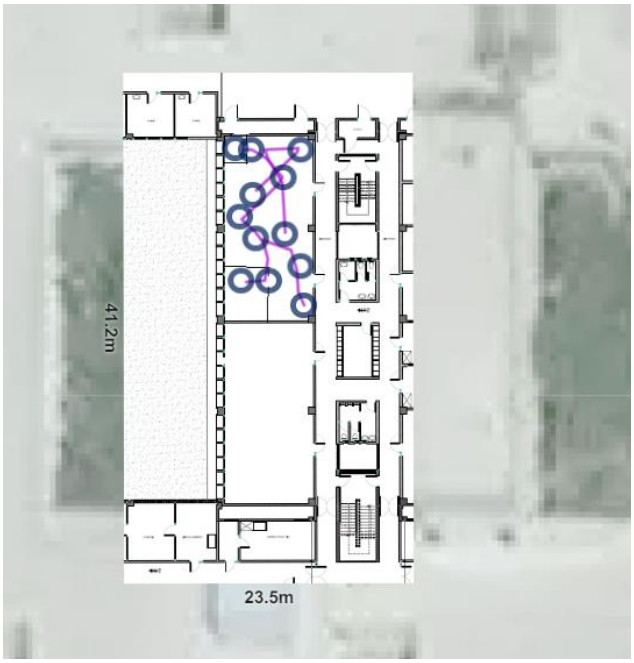
Σε αυτή την οθόνη ο χρήστης μπορεί να βρεθεί με διαφορετικούς τρόπους. Ο πρώτος είναι μετά από την επιλογή μιας εικόνας ενός ολοκληρωμένου επιπέδου, ενώ ο δεύτερος είναι μέσω της σωστής απάντησης του στην ερώτηση κάποιου επιπέδου. Σε κάθε περίπτωση, αυτή η οθόνη παρέχει πληροφορίες για το επιλεχθέν έκθεμα. Ο τίτλος του, ο δημιουργός του, κάποια tags που αποτελούνται από λέξεις κλειδιά για το έκθεμα και μια σύντομη περιγραφή του είναι μερικά από αυτά. Επιπλέον, υπάρχει κατάλληλο κουμπί, έτσι ώστε ο χρήστης να μπορεί να αποθηκεύσει την εικόνα στη συσκευή του σε περίπτωση που το επιθυμεί. Θα μπορεί ταυτόχρονα να πατήσει το κουμπί “Μου αρέσει” και να αποθηκεύσει την εικόνα αυτή στο προφίλ του, έτσι ώστε να έχει πιο εύκολη πρόσβαση σε αυτήν αργότερα. Τέλος, πατώντας το κουμπί “Επόμενο” θα μεταφερθεί στην οθόνη με τα επίπεδα της ιστορίας, έτσι ώστε να συνεχίσει την αναζήτηση των επόμενων εκθεμάτων στο Μουσείο.



Εικόνα 26: Οθόνη Εκθέματος

3.5 Indoor Atlas

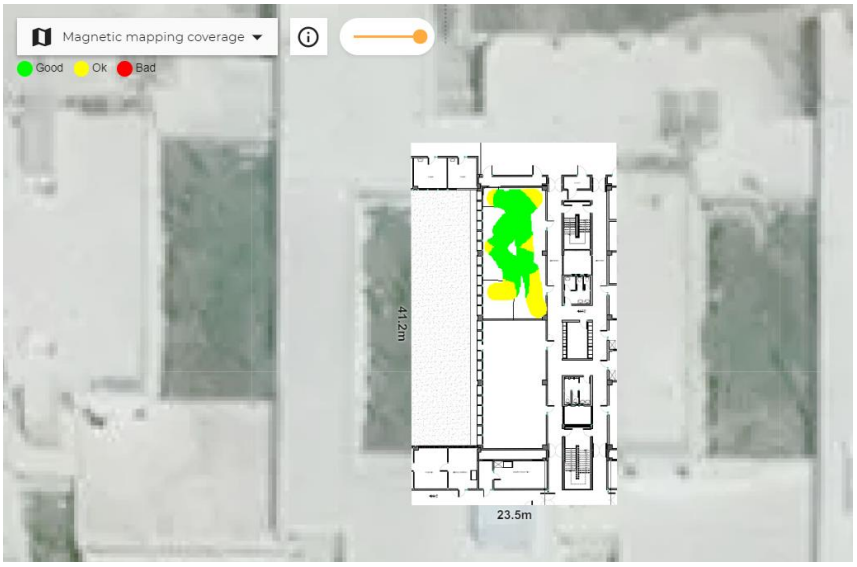
Για την μέθοδο του indoor positioning της εφαρμογής αρχικά εξετάστηκε η τεχνολογία του indoor atlas. Αφού δημιουργήθηκε ένας λογαριασμός στην ιστοσελίδα, ακολουθήθηκαν τα βήματα για τη διαδικασία του mapping. Μια διαδικασία που δημιουργεί ένα μοναδικό αποτύπωμα για κάθε σημείο του χώρου, στον οποίο αργότερα θα χρειαστεί να πλοηγηθούμε και το οποίο αποτελείται από δεδομένα, όπως η ισχύς του σήματος Wi-Fi, η ισχύς του σήματος των κοντινών beacons και το μαγνητικό αποτύπωμα του σημείου. Η διαδικασία του mapping υλοποιήθηκε στο εργαστήριο IVML στα Παλαιά Κτήρια των Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου. Για τη διαδικασία του mapping απαιτείται ένα σχεδιάγραμμα του χώρου, ένα κινητό τηλέφωνο με φυσικό αισθητήρα γυροσκοπίου και μαγνητόμετρου και τέλος το κατέβασμα της εφαρμογής map creator2 στο κινητό του χρήστη. Ο χρήστης αρχικά τοποθετεί το σχεδιάγραμμα στο κατάλληλο σημείο στον χάρτη, μέσω του Google maps, και στη συνέχεια ορίζει waypoints μεταξύ των οποίων περπατάει, ενώ η εφαρμογή συλλέγει δεδομένα για τα αποτυπώματα των σημείων που περνάει.



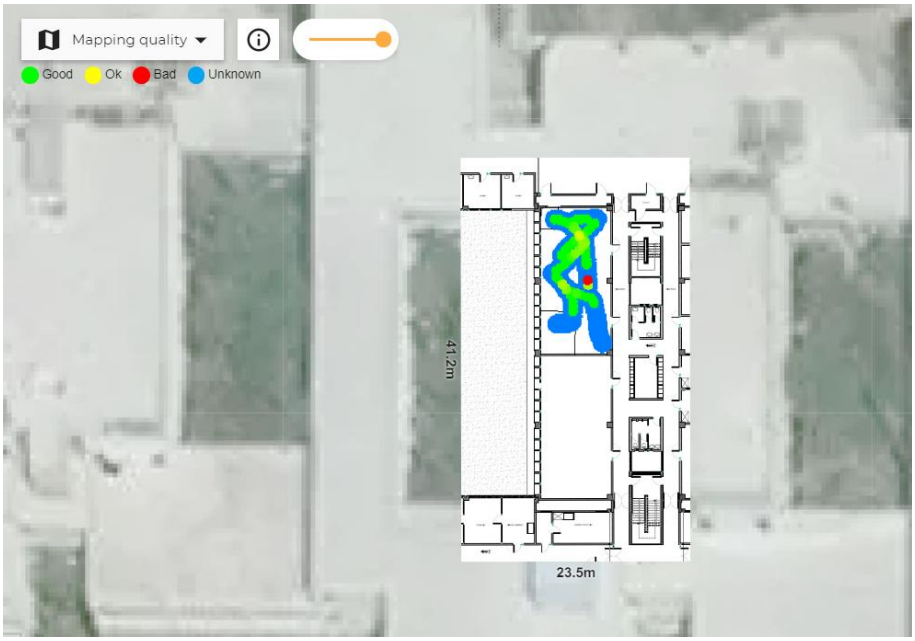
Εικόνα 27: Paths and Checkpoints

Μόλις τελειώσει η διαδικασία του mapping, τα δεδομένα που συλλέχθηκαν ανεβαίνουν στον cloud server της indoor atlas και στη συνέχει ο χρήστης μπορεί να πραγματοποιήσει τον έλεγχο της ποιότητας του mapping που δημιούργησε. Σε αυτό το σημείο, πρέπει να τονιστεί ότι, παρότι τα αποτελέσματα του testing στη συσκευή όπου πραγματοποιήθηκε το mapping ήταν ικανοποιητικά, όταν αυτή η τεχνολογία ελέγχθηκε σε άλλες συσκευές, οι αποκλίσεις στην ακρίβεια ήταν αρκετά μεγάλες και ουσιαστικά απαγορευτικές για τη χρησιμοποίηση της. Πιο συγκεκριμένα η έλλειψη του γυροσκοπίου από αρκετά κινητά τηλέφωνα οδηγούσε σε κατακόρυφη πτώση της ακρίβειας. Παρακάτω παρατίθενται μερικές εικόνες από το αποτελέσματα της διαδικασίας του mapping και της αξιολόγησης της ποιότητας των δεδομένων που συλλέχθηκαν με βάση τις διάφορες τεχνολογίες.

Εικόνα 28: Map coverage



Εικόνα 29: Magnetic map coverage



Εικόνα 30: Mapping quality

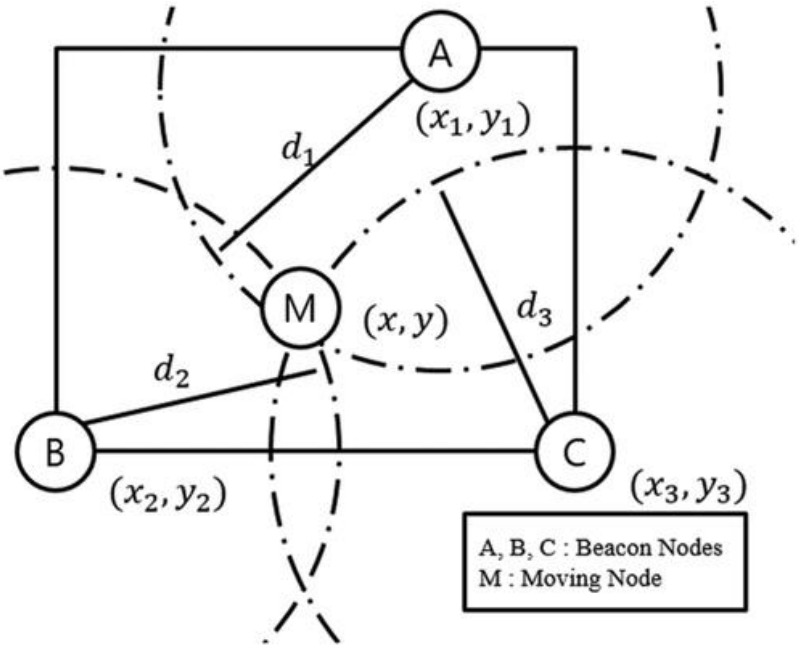
3.6 Beacons

Η χρήση των beacons για τη λειτουργία του Indoor positioning μπορεί να υλοποιηθεί με πολλές μεθόδους, η ακρίβεια και η αποτελεσματικότητα των οποίων μπορεί να διαφέρουν από εφαρμογή σε εφαρμογή. Για την επιλογή της κατάλληλης για την συγκεκριμένη εφαρμογή εξετάστηκαν οι εξής μέθοδοι:

3.6.1 Triangulation

Ο τριγωνισμός σε δύο διαστάσεις για να λειτουργήσει απαιτεί την παρουσία 3 beacons και ενός κινούμενου χρήστη για να εφαρμοστεί και αποτελεί ίσως τη

σημαντικότερη και πιο παλιά μέθοδο εντοπισμού θέσης. Η μέθοδος βρίσκει ένα σημείο στο οποίο συναντώνται όλοι οι κύκλοι, οι οποίοι έχουν κέντρο ένα από τα beacons και ακτίνα την απόσταση αυτού του beacon από τον χρήστη. Βέβαια σε περίπτωση που δεν υπάρχει σημείο όπου τέμνονται και οι τρείς κύκλοι, η μέθοδος επιστρέφει μια περιοχή στην οποία ο χρήστης βρίσκεται.



Εικόνα 31: Triangulation

Στην περίπτωση των beacons, τα αποτελέσματα της μεθόδου του τριγωνισμού δεν είναι καθόλου ικανοποιητικά. Αυτό οφείλεται κυρίως στο γεγονός ότι τα beacons δεν είναι φτιαγμένα για να παρέχουν μια ακριβή τοποθεσία, αλλά περισσότερο μια ένδειξη του πόσο κοντά είσαι σε αυτά.

3.6.2 Ranging

Με αυτήν τη μέθοδο χρησιμοποιήθηκε το ranging που παρέχουν τα beacons. Κάθε έκθεμα έχει μια λίστα από τα κοντινά σε αυτό beacons και την απόσταση κάθε beacon από αυτό (Immediate, Near, Far). Όταν ο χρήστης φτάσει στην οθόνη αναζήτησης beacon, τότε, για ένα χρονικό διάστημα της τάξης των 5 δευτερολέπτων, η συσκευή στην οποία τρέχει η εφαρμογή, παίρνει μετρήσεις και στο αξιολογεί αν ο χρήστης είναι στο σωστό σημείο. Πιο συγκεκριμένα ελέγχει αν λαμβάνει τουλάχιστον το απαιτούμενο σήμα από τα beacons που έχουν οριστεί ως κοντινά σε αυτό το έκθεμα. Το διάστημα των 5 δευτερολέπτων έχει οριστεί, γιατί, αφού μιλάμε για σήματα Bluetooth, σημαίνει ότι είναι εύκολο να αλλοιωθούν. Για παράδειγμα, αν ανάμεσα στο beacon και στον χρήστη παρεμβάλλεται κάποιο άλλο έκθεμα, τότε το σήμα θα φτάνει εξασθενημένο και θα οδηγήσει σε λανθασμένη εκτίμηση. Επίσης, σε περίπτωση που το μουσείο έχει αρκετό κόσμο, πάλι ένα μικρότερο χρονικό διάστημα ίσως δημιουργούσε προβλήματα. Ένα μεγαλύτερο

χρονικό διάστημα θα οδηγούσε πιθανώς σε μεγαλύτερη ακρίβεια, αλλά θεωρήθηκε ότι θα μείωνε αρκετά την εμπειρία αλληλεπίδρασης του χρήστη με την εφαρμογή, καθώς θα παρέμενε ανενεργός για μεγάλο χρονικό διάστημα.

3.7 Αναγνώριση Εικόνας

Λόγω της μη επαρκείς ακρίβειας των beacon στην προκυμένη εφαρμογή αποφασίστηκε να υλοποιηθεί μια εναλλακτική μέθοδος απόφασης του αν ο χρήστης είναι στο σωστό έκθεμα. Η μέθοδος που επιλέχθηκε είναι η αναγνώριση εικόνας με την κάμερα του κινητού τηλεφώνου του χρήστη μέσω του machine learning με την χρήση της τεχνολογίας Tensorflow. Η συγκεκριμένη μέθοδος είναι ιδανική για να συμπληρώσει την μέθοδο των beacon καθώς κάθε σύγχρονο κινητό τηλέφωνο διαθέτει κάμερα, συνεπώς δεν απαιτεί κάποιον επιπλέον αισθητήρα hardware. Για την υλοποίηση της, επεξεργάστηκαν οι εικόνες των εκθεμάτων του μουσείου μέσω του 3D Painting των Windows 10 έτσι ώστε να έχουμε εικόνες από κάθε οπτική γωνία και να οδηγήσουν σε καλύτερα ποσοστά αναγνώρισης. Επίσης, στις εικόνες τοποθετήθηκε και διαφορετικού χρώματος φόντο έτσι ώστε να καλυφθεί και η περίπτωση του χρώματος των τοίχων του μουσείου.

3.8 Από το Design στην Υλοποίηση

Σε αυτό το σημείο έχουμε αρκετό υλικό από την Σχεδίαση του project, έτσι ώστε να προχωρήσουμε στην Υλοποίηση και στον Έλεγχό του. Στο επόμενο κεφάλαιο παρατίθενται πληροφορίες για τα κυριότερα σημεία της υλοποίησης, μαζί με τις τεχνολογίες και τα εργαλεία που τελικά χρησιμοποιήθηκαν από το θεωρητικό υπόβαθρο.

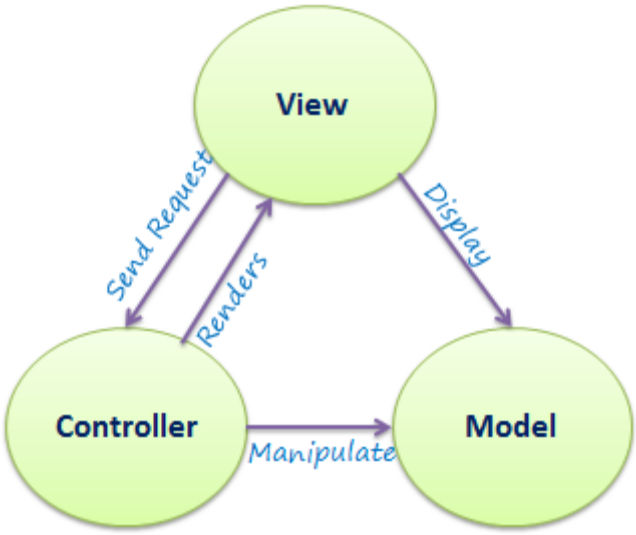
4

Υλοποίηση

Σε αυτό το κεφάλαιο γίνεται αναφορά στα τεχνολογικά εργαλεία του τελικά συμπεριελήφθησαν στην υλοποίηση της εφαρμογής, στις τεχνολογίες που χρησιμοποιήθηκαν, καθώς και σε μερικές λεπτομέρειες για τα κύρια σημεία της εφαρμογής.

4.1 Αρχιτεκτονική

Το μοντέλο αρχιτεκτονικής Model-View-Controller (MVC) είναι ένα από τα πιο γνωστά και πλέον διαδεδομένα μοντέλα αρχιτεκτονικής για την ανάπτυξη εφαρμογών. Χρησιμοποιείται για την ανάπτυξη του User Interface. Σύμφωνα με αυτό, τα αντικείμενα της εφαρμογής κατατάσσονται σε 3 κατηγορίες, προωθώντας έτσι την ευελιξία, την εύκολη συνεργασία ανάμεσα τους και την επαναχρησιμοποίησή τους. Οι ρόλοι της κάθε κατηγορίας είναι οι εξής:



Εικόνα 32: Model-View-Controller

Model: Τα models ορίζουν ποια δεδομένα θα πρέπει να περιέχει η εφαρμογή. Σε περίπτωση που τα δεδομένα αλλάξουν, τότε το Model θα ενημερώσει το View, ώστε να απεικονίσει αυτή την αλλαγή, και μερικές φορές τον Controller. Για παράδειγμα, στην εφαρμογή μας μπορεί κάποιο έκθεμα να αντικατασταθεί ή να προστεθεί στη βάση δεδομένων μας. Αυτή είναι μια αλλαγή στο Model της εφαρμογής.

View: Το View αποφασίζει πώς θα απεικονίζονται τα δεδομένα της εφαρμογής στον χρήστη. Στην εφαρμογή μας, το View αποφασίζει για το πώς θα εμφανιστεί η κάθε πληροφορία της βάσης δεδομένων μας στον χρήστη και λαμβάνει αυτές τις πληροφορίες από το Model.

Controller: Οι controllers περιέχουν τη λογική που ενημερώνει το Model ή/και το View, με βάση κάποια δράση του χρήστη. Για παράδειγμα, στην εφαρμογή μας, όταν ο χρήστης ολοκληρώσει κάποιο επίπεδο βρίσκοντας τον σωστό πίνακα, τότε πρέπει να ενημερωθεί και το Model, έτσι ώστε να αποθηκεύσει ότι ο χρήστης ολοκλήρωσε αυτό το επίπεδο, αλλά και το View, έτσι ώστε το συγκεκριμένο επίπεδο να έχει το χαρακτηριστικό ''τικ'' και να ξεκλειδωθεί το επόμενο.

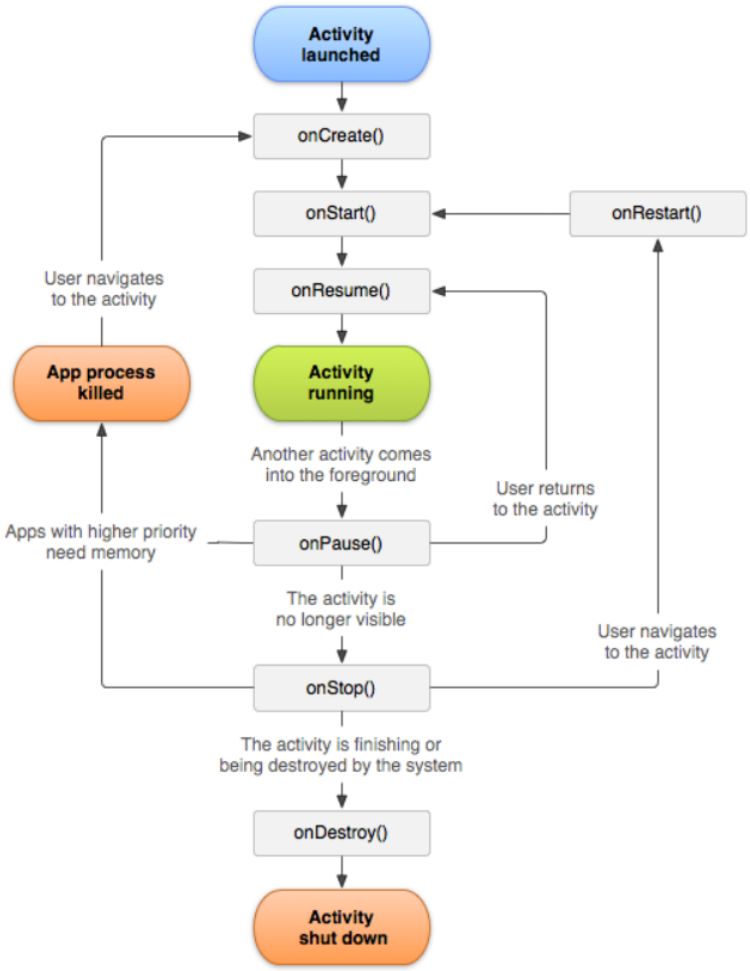
4.2 Κύρια σημεία κώδικα της εφαρμογής

Παρακάτω αναλύονται συνοπτικά και με παρουσία μερικών γραμμών κώδικα, τα κυριότερα σημεία του κώδικα της εφαρμογής.

4.2.1 Activity

Κάθε οθόνη της εφαρμογής αποτελεί μια δραστηριότητα η οποία ξεκινά να τρέχει κάθε φορά που μεταβαίνουμε στην αντίστοιχη οθόνη. Μια δραστηριότητα μπορεί να δημιουργηθεί την στιγμή της μετάβασης ή να <<ξυπνήσει>> τότε, αν έχει ξαναχρησιμοποιηθεί στο παρελθόν. Όπως φαίνεται από το παρακάτω σχήμα, η

μέθοδος onCreate() θα τρέξει, μόνο την πρώτη φορά που θα δημιουργηθεί η δραστηριότητα, και είναι εκεί όπου ορίζεται το layout της.



Εικόνα 33: Activity lifecycle

Έτσι κάθε onCreate() μέθοδος κάθε δραστηριότητάς μας περιλαμβάνει τις παρακάτω γραμμές, οι οποίες αντιστοιχούν τον κώδικα σε Java. Ο κώδικας σε Java ορίζει τη λειτουργικότητα της δραστηριότητας και ο κώδικας σε XML περιγράφει το γραφικό περιβάλλον της συγκεκριμένης δραστηριότητας.

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

}
```

4.2.2 ViewGroups and Layouts

Τα αντικείμενα View χρησιμοποιούνται από το Android με σκοπό την αναπαράσταση αντικειμένων στην οθόνη μιας συσκευής Android. Αν και μπορούν να οριστούν και μέσα από τον κώδικα Java της εφαρμογής, συνήθως ορίζονται από

αρχεία XML. Παραδείγματα μερικών View είναι οι εικόνες, τα κείμενα και τα κουμπιά μιας εφαρμογής.

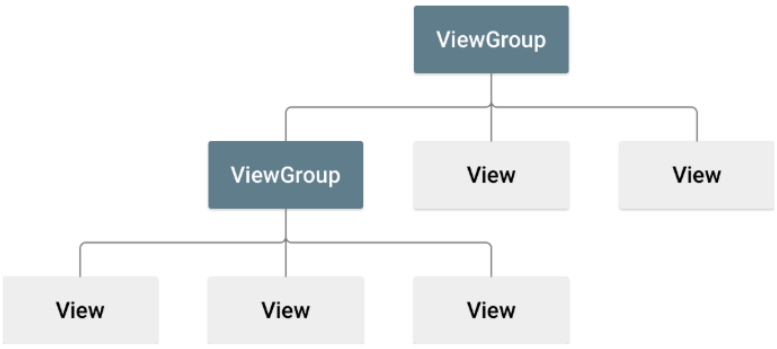
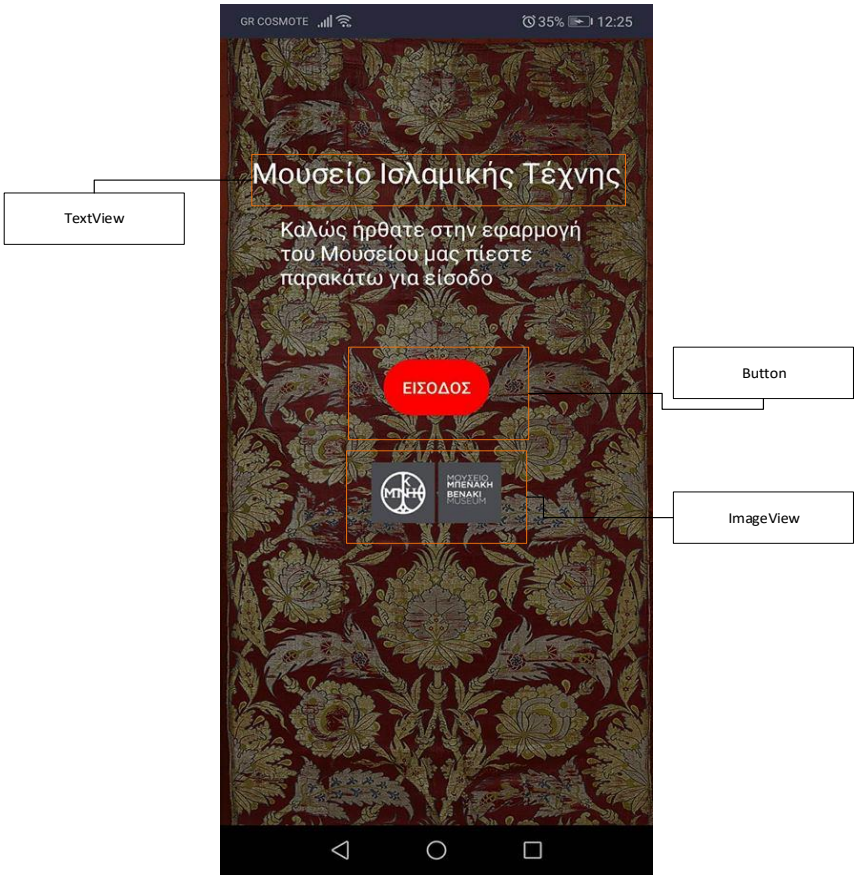


Figure 1. Illustration of a view hierarchy, which defines a UI layout

Το Layout από την άλλη είναι αόρατο στην οθόνη της εφαρμογής, αλλά η χρήση του είναι απολύτως απαραίτητη. Πιο συγκεκριμένα, η κύρια δουλειά ενός Layout είναι να ομαδοποιεί τα Views που είναι εμφωλευμένα σε αυτό και να τούς υποδεικνύει με ποιο τρόπο και με ποια σχέση μεταξύ τους θα εμφανιστούν στην οθόνη. Παραδείγματα Layouts αποτελούν τα Linear Layout, Relative Layout και FrameLayout.

Παρακάτω, παρατίθεται η οθόνη εισόδου της εφαρμογής μαζί με το αρχείο xml που περιέχει ένα Linear Layout. Στο συγκεκριμένο layout υπάρχει ένα μοναδικό id, έτσι ώστε να καθίσταται δυνατή η επεξεργασία του και από τον κώδικα Java. Επίσης, οι παράμετροι layout_height και layout_width είναι υποχρεωτικοί για το κάθε layout, καθώς ορίζουν το μέγεθός του. Σε αυτήν την περίπτωση έχει οριστεί ως match_parent που ουσιαστικά σημαίνει ότι πρέπει να καλύπτει όλη την οθόνη, αφού είναι το πρώτο στοιχείο του αρχείου xml. Άξιο τονισμού είναι και το πεδίο orientation που στην συγκεκριμένη περίπτωση έχει την τιμή vertical. Αυτό ορίζει το πώς θα εμφανίζονται γραμμικά τα <<παιδιά>> αυτού του layout. Οι δύο επιλογές είναι vertical και horizontal. Επίσης, μπορούν να παρατηρηθούν και τα πεδία Imageview, Textview και Button τα οποία αναλύονται παρακάτω.



Εικόνα 34: Views στην κύρια οθόνη

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/im"
    android:orientation="vertical"
    tools:context="com.example.kouva.test1.MainActivity">
    <TextView
        android:layout_width="230dp"
        android:layout_height="85dp"
        android:text="Μουσείο Ισλαμικής Τέχνης"
        android:textSize="25sp"
        android:textColor="#EEFDD7"
        android:id="@+id/textView2" />
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Είσοδος"
        android:textColor="#EEFDD7"
        android:background="@drawable/buttonred" />
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="145dp"
        android:layout_height="50dp"
        app:srcCompat="@mipmap/benakinewlogo"/>
</LinearLayout>
```

Εικόνα 35: XML κώδικας για την εικόνα 37

4.2.3 ImageView

Όπως αφήνει και το όνομα να υπονοηθεί, το ImageView έχει σχεδιαστεί, ώστε να αναπαριστά εικόνες στην οθόνη. Οι εικόνες αυτές μπορεί να είναι αποθηκευμένες στην μνήμη του τηλεφώνου ή να κατεβάζονται από τον ιστό. Προσφέρονται αρκετές επιλογές για τη διαφορετική εμφάνιση της κάθε εικόνας, όπως το μέγεθός της ή το πώς θα κάνει scale για να εφαρμοστεί στο συγκεκριμένο μέγεθος.

```
<ImageView
    android:id="@+id/imageView"
    android:layout_width="145dp"
    android:layout_height="50dp"
    app:srcCompat="@mipmap/benakinewlogo"/>
```

Εικόνα 36: ImageView

4.2.4 TextView

Το TextView χρησιμοποιείται για να εμφανίζει κάποιο κείμενο στην οθόνη της εφαρμογής. Παρόλο που ακούγεται σαν ένα απλό feature, περιέχει μια σύνθετη λογική, καθώς είναι ικανό για την εμφάνιση κειμένου, υπερσυνδέσμων, τηλεφώνων, email και άλλων τύπων κειμένου, που όμως απαιτούν συγκεκριμένη εμφάνιση.

```
<TextView
    android:layout_width="230dp"
    android:layout_height="85dp"
    android:text="Μουσείο Ισλαμικής Τέχνης"
    android:textSize="25sp"
    android:textColor="#EEFDD7"
    android:id="@+id/textView2" />
```

Εικόνα 37: TextView

4.2.5 Button

Η κλάση του Button είναι από τις πιο σημαντικές κλάσεις, καθώς παρέχει έλεγχο της εφαρμογής στον χρήστη. Περιμένει το κλικ του χρήστη και στη συνέχεια πραγματοποιεί μια δράση η οποία έχει οριστεί στον κώδικα της εφαρμογής. Αυτή η δράση μπορεί να είναι η εμφάνιση ενός απλού μηνύματος ή ακόμα και η έναρξη μιας καινούργιας δραστηριότητας.

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Είσοδος"
    android:textColor="#EEFDD7"
    android:background="@drawable/buttonred" />
```

Εικόνα 38: Button

4.2.6 TabHost

Στο Android, κάθε TabHost χρησιμοποιείται για να απεικονιστούν διαφορετικά παράθυρα στην ίδια δραστηριότητα. Κάθε TabHost έχει δύο <<παιδιά>>, από τα οποία το ένα θέτει τα labels των tabs και το άλλο το περιεχόμενο του κάθε παραθύρου. Αυτός είναι ένας ιδιαίτερα αποτελεσματικός τρόπος για να απεικονιστούν πολλές πληροφορίες σε μία μόνο δραστηριότητα.

```
<?xml version="1.0" encoding="UTF-8"?>
<TabHost xmlns:android="http://schemas.android.com/apk/res/android"
android:id="@android:id/tabhost"
android:layout_width="fill_parent"
android:layout_height="fill_parent">

<LinearLayout
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:orientation="vertical">

<FrameLayout
android:id="@android:id/tabcontent"
android:layout_width="fill_parent"
android:layout_height="0dip"
android:layout_weight="1" />

<TabWidget
android:id="@android:id/tabs"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:layout_marginBottom="-4dp"
android:layout_weight="0" />

</LinearLayout>

</TabHost>
```

Εικόνα 39: TabHost XML

```
TabHost tabHost = (TabHost)findViewById(android.R.id.tabhost); // initiate TabHost
TabHost.TabSpec tabSpec = tabHost.newTabSpec("tab1"); // Create a new TabSpec using tab host
tabSpec.setIndicator("Tab 1"); // set the “Tab 1” as an indicator for a tab
```

Εικόνα 40: TabHost Java

Βέβαια, πάλι το documentation του Android μάς παρέχει πολλές εναλλακτικές για την υλοποίησή του. Στο παράδειγμά μας, δεν θέλαμε να χρησιμοποιήσουμε το standard TabHost, γιατί στα labels μπορούσαμε να τοποθετήσουμε μόνο μια εικόνα ή μόνο κείμενο, ενώ για ένα βελτιωμένο UI

χρειαζόμασταν και τα δύο. Έτσι δημιουργήσαμε ένα Custom TabHost, το οποίο μπορεί να τα έχει και τα δύο.

4.2.7 Recycle View

Η κλάση Recycle View αποτελεί μια επέκταση της κλάσης List View, η οποία αναπαριστά κάποια δεδομένα με τη μορφή λίστας. Για τη δημιουργία μιας recycle view, πρέπει να ανατεθεί ένας layout manager, στην περίπτωση μας ένας LinearLayoutManager. Στην συνέχεια, κάθε στοιχείο στη λίστα μας αναπαρίσταται ως ένα συγκεκριμένο view. Το μεγάλο πλεονέκτημα της recycle view έναντι της list view βρίσκεται σε αυτό το σημείο, καθώς η πρώτη υπολογίζει μόνο τα views που θα είναι ορατά από τον χρήστη και όχι όλα. Έτσι, όταν ο χρήστης κάνει scroll και πρέπει να εμφανιστούν νέα δεδομένα, τότε μόνο θα υπολογιστούν. Αυτό κάνει ουσιαστικά την εφαρμογή πιο ελαφριά και πιο γρήγορη. Τα δεδομένα που πρέπει να απεικονιστούν συνδέονται με τα κατάλληλα views μέσω ενός Adapter.

```
<android.support.v7.widget.RecyclerView
    android:layout_width="match_parent"
    android:id="@+id/tour_list"
    android:layout_height="match_parent"
    android:smoothScrollbar="true"
    android:scrollingCache="true"
    android:animationCache="true">
```

Εικόνα 41: RecycleView XML

```
RecyclerView tour = (RecyclerView) findViewById(R.id.tour_list);
tour.setHasFixedSize(true);
LinearLayoutManager manager = new LinearLayoutManager( context: this,LinearLayoutManager.VERTICAL,  reverseLayout: false);
tour.setLayoutManager(manager);
TourListAdapter listAdapter = new
        TourListAdapter( context: second.this, Title, Subtitle, Image);
//list=(RecyclerView)findViewById(R.id.tour_list);
tour.setAdapter(listAdapter);
```

Εικόνα 42: RecycleView Java

4.2.8 Database

Ως database της εφαρμογής επιλέχθηκε η SQLite, όπως έχει αναφερθεί και αναλυθεί παραπάνω. Για την χρήση της δημιουργήθηκαν δύο κλάσεις. Η πρώτη κλάση είναι η DatabaseHelper.java η οποία δημιουργεί και ανανεώνει, αν χρειάζεται, τα tables. Η μέθοδος onCreate() που υλοποιεί τρέχει, μόνο όταν τρέχει για πρώτη φορά η εφαρμογή στο κινητό τηλέφωνο. Με αυτό τον τρόπο μάς επιτρέπει να αρχικοποιήσουμε τη βάση δεδομένων μας στις default τιμές.

```
@Override
public void onCreate(SQLiteDatabase _db) {
    try {
        _db.execSQL(DatabaseAdapter.DATABASE_CREATE);
        _db.execSQL("INSERT INTO TOUR(Title, Subtitle, Image ) VALUES('Tour1', 'This is a tour', 'ifasmagarifalla' )");
        _db.execSQL("INSERT INTO TOUR(Title, Subtitle, Image ) VALUES('Tour2', 'This is a tour', 'ifasmagarifalla' )");
        _db.execSQL("INSERT INTO TOUR(Title, Subtitle, Image ) VALUES('Tour3', 'This is a tour', 'story2' )");
        _db.execSQL("INSERT INTO TOUR(Title, Subtitle, Image ) VALUES('Tour4', 'This is a tour', 'ifasmagarifalla' )");
        _db.execSQL("INSERT INTO TOUR(Title, Subtitle, Image ) VALUES('Tour5', 'This is a tour', 'ifasmagarifalla' )");
        _db.execSQL("create table STORY ( ID integer primary key autoincrement, Title text, Subtitle text, Image text); ");
```

Εικόνα 43: Μέθοδος onCreate()

Η μέθοδος onUpdate() που επίσης υλοποιεί χρησιμοποιείται, μόνο αν υπάρξει κάποια αλλαγή στο version της database και έτσι κάνει την κατάλληλη ενημέρωση. Αυτή η μέθοδος είναι χρήσιμη σε περίπτωση που θέλουμε να προσθέσουμε κάποιο table ή να επεξεργαστούμε κάποιο ήδη υπάρχον.

Από την άλλη μεριά, η κλάση DatabaseAdapter.java ορίζει τη βάση δεδομένων μας, την έκδοσή της, καθώς και μερικές βασικές μεθόδους, όπως την open() και την close(). Η κύρια χρησιμότητά της στον κώδικα μας είναι ότι σε αυτή την κλάση ορίζονται και εκτελούνται όλα τα έτοιμα queries, που χρειάζονται για να πάρουμε δεδομένα από τη βάση δεδομένων κατά το runtime. Για παράδειγμα, αν θέλουμε να βρούμε τον τίτλο του επιπέδου μιας ιστορίας από το id του θα χρησιμοποιήσουμε τον παρακάτω κώδικα που ανοίγει τη βάση δεδομένων και εκτελεί το κατάλληλο query.

```
databaseAdapter =new DatabaseAdapter(getApplicationContext());

databaseAdapter=databaseAdapter.open();
title.setText(databaseAdapter.findLevelTitle(value));
```

Εικόνα 44: Εκτέλεση Query

```
public String findLevelTitle(int id){
    String title;
    String c = String.valueOf(id);
    db=dbHelper.getReadableDatabase();
    Cursor cursor=db.query( table: "LEVEL" , columns: null , selection: "ID=?" ,new String[] {c}, groupBy: null , having: null , orderBy: null );
    if(cursor.getCount()<1) // UserName Not Exist
        return "not found";
    cursor.moveToFirst();
    title=cursor.getString(cursor.getColumnIndex( columnName: "Title"));
    return title;
}
```

Εικόνα 45: Ορισμός query

4.2.9 Beacons

Η τεχνολογία των beacons μπορεί να χρησιμοποιείται μόνο από την δραστηριότητα beacon μέσα στον κώδικα της εφαρμογής, αλλά είναι μια από τις πιο σημαντικές και άξιες προσοχής τεχνολογίες της εφαρμογής. Για τη χρήση των beacons απαιτείται από τον χρήστη να ενεργοποιήσει το Bluetooth της συσκευής του, καθώς και την τοποθεσία του. Από το Android 6.0 και μετά, αυτές οι άδειες πρέπει να παρέχονται κατά το runtime της εφαρμογής και όχι όταν ο χρήστης κατεβάσει την εφαρμογή από το play store. Έτσι πρέπει να δηλώσουμε την αίτηση για το άνοιγμα του Bluetooth και της τοποθεσίας στο Android manifest και, όταν τρέξει για πρώτη φορά αυτή η δραστηριότητα, να την ζητήσει από τον χρήστη.

```
    SystemRequirementsChecker.checkWithDefaultDialogs( activity: this);
<uses-permission android:name="android.permission.BLUETOOTH"/>
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
```

Εικόνα 46: Permissions

Στη συνέχεια, πρέπει να ορίσουμε την περιοχή των beacons που μας ενδιαφέρουν και να πάρουμε από την βάση δεδομένων τους major αριθμούς και την απόσταση του κάθε beacon του πίνακα που πρέπει ο χρήστης να βρει σε αυτό το επίπεδο.

```
major=databaseAdapter.findBeacon(value);
System.out.println("Major is: "+ major);

beaconManager = new BeaconManager(getApplicationContext());
region = new BeaconRegion( identifier: "ranged region",
        UUID.fromString("B9407F30-F5F8-466E-AFF9-25556B57FE6D"), major: null, minor: null);
```

Εικόνα 47: Beacon region

Το επόμενο βήμα είναι να αρχίσουμε να ακούμε τα σήματα που λαμβάνουν από την περιοχή των beacons που ορίσαμε. Όπως έχει αναφερθεί, ακούμε για 5 δευτερόλεπτα και αν ικανοποιηθεί η συνθήκη εντοπισμού, τότε ένα flag τίθεται στο 1 έτσι, ώστε όταν λήξει ο χρόνος να προχωρήσουμε στην ερώτηση, ενώ αλλιώς να επιστρέψουμε στην ιστορία του πίνακα και ο χρήστης να ψάξει ξανά τον σωστό πίνακα.

```
beaconManager.setRangingListener(new BeaconManager.BeaconRangingListener() {
    @Override
    public void onBeaconsDiscovered(BeaconRegion region, List<Beacon> list) {
        if (!list.isEmpty()) {
            System.out.println("Getting from beacons:");
            for (int i=0; i<list.size(); i++){
                if(list.get(i).getMajor()==major && com.estimote.coresdk.observation.region.RegionUtils.computeProximity(list.get(i)).toString().equals("IMMEDIATE") ){
                    f=1;
                }
            }

        }
    }
});
```

Εικόνα 48: Beacon scanning

Ταυτόχρονα με την προηγούμενη διαδικασία, πρέπει να αρχίσουμε να μετράμε αντίστροφα για τα 5 δευτερόλεπτα και, μόλις τελειώσουν, να δούμε αν ο χρήστης τελικά βρήκε τον πίνακα ή όχι. Αυτή η διαδικασία γίνεται με το παρακάτω κομμάτι κώδικα.

```
    new Handler().postDelayed(() → {
            if(f==1){
                Intent i = new Intent(getApplicationContext(), quiz1.class);
                Bundle b = new Bundle();
                b.putInt("key", value); //Your id
                i.putExtras(b);
                startActivity(i);
                finish();
            }else{
                Toast.makeText(getApplicationContext(), text: "Wrong Painting.Please try again",Toast.LENGTH_LONG).show();
                finish();
            }
    }, TIME_OUT);
}
```

Εικόνα 49: Timer

Ένα ακόμα σημαντικό σημείο των beacons είναι ότι το ranging είναι αρκετά energy consuming. Συνεπώς, όταν η δραστηριότητα μας σταματήσει να είναι η ενεργή δραστηριότητα, θα πρέπει να σταματήσουμε το ranging για να είναι αποδοτικό. Αυτό επιτυγχάνεται με την προσθήκη μερικών γραμμών κώδικα στην onPause() μέθοδο της δραστηριότητας.

```
@Override
protected void onPause() {
    beaconManager.stopRanging(region);

    super.onPause();
}
```

Εικόνα 50: Stop ranging

4.2.10 Αναγνώριση εικόνας

Για την υλοποίηση του Image Recognition χρειάστηκε να γίνει επανεκπαίδευση το μοντέλο που επιθυμούμε να κατασκευάσουμε με 4000 βήματα. Το αρχικό μοντέλο τύπου MobileNet αποτελεί ένα συνελιξιακό νευρωνικό δίκτυο που περιελάμβανει 28 επίπεδα και πάνω από 1000 κλάσεις. Μέσω της επανεκπαίδευσης επιρρεάστηκε μόνο το τελικό επίπεδο καθώς οι απαιτήσεις σε χρόνο για την επεξεργασία των πιο κάτω επιπέδων ήταν μεγάλες. Επιπλέον λόγω του ότι οι εικόνες που επιθυμούμε να αναγνωρίζει το μοντέλο είναι σταθερές και όχι αόριστες, η ακρίβεια που επιτεύχθηκε πλησίασε το 100%. Επίσης δημιουργήθηκαν πάνω από 50 διαφορετικές εικόνες του κάθε εκθέματος υπο διαφορετικές γωνίες και οπτικές έτσι ώστε να είναι αποτελεσματική η εκπαίδευση.

```
python -m scripts.retrain \
  --bottleneck_dir=tf_files/bottlenecks \
  --model_dir=tf_files/models/"${ARCHITECTURE}" \
  --summaries_dir=tf_files/training_summaries/"${ARCHITECTURE}" \
  --output_graph=tf_files/retrained_graph.pb \
  --output_labels=tf_files/retrained_labels.txt \
  --architecture="${ARCHITECTURE}" \
  --image_dir=tf_files/flower_photos
```

Η παραπάνω εντολή python τρέχει το script retrain και δημιουργεί το αρχείο retrained_graph.pb, το οποίο είναι ο γράφος για την αναγνώριση του εκθέματος, και το αρχείο retrained_labels.txt το οποίο περιέχει το όνομα του κάθε εκθέματος, πιο συγκεκριμένα το όνομα του φακέλου που περιείχε τις εικόνες του κάθε εκθέματος.

Το μοντέλο που δημιουργήθηκε παραπάνω, είναι έτοιμο για να τρέξει σε κάποιον υπολογιστή που χρησιμοποιεί Tensorflow, αλλά επειδή σε αυτή την περίπτωση η επιθυμητή χρήση του είναι σε κινητό τηλέφωνο πρέπει να το μετατρέψουμε για να είναι συμβατό. Η ανάγκη γι'αυτή την μετατροπή οφείλετε στο ότι οι εφαρμογές κινητών τηλεφώνων έχουν αρκετά πιο περιορισμένη μνήμη από τις εφαρμογές υπολογιστών, έτσι η βιβλιοθήκη Tensorflow για κινητά υποστηρίζει λιγότερες βιβλιοθήκες από τις κανονικές με σκοπό να μειωθεί το συνολικό μέγεθος της.

Για μπορέσει να χρησιμοποιηθεί από συσκευές Android έπρεπε να μετατραπεί το αρχείο .pb σε ένα αρχείο τύπου .lite. Για την μετατροπή αυτή χρησιμοποιήθηκε το TOCO (TensorFlow Lite Optimizing Converter) και η παρακάτω εντολή python.

```
IMAGE_SIZE=224
toco \
  --input_file=tf_files/retrained_graph.pb \
  --output_file=tf_files/optimized_graph.lite \
  --input_format=TENSORFLOW_GRAPHDEF \
  --output_format=TFLITE \
  --input_shape=1,${IMAGE_SIZE},${IMAGE_SIZE},3 \
  --input_array=input \
  --output_array=final_result \
  --inference_type=FLOAT \
  --input_data_type=FLOAT
```

4.3 Performance

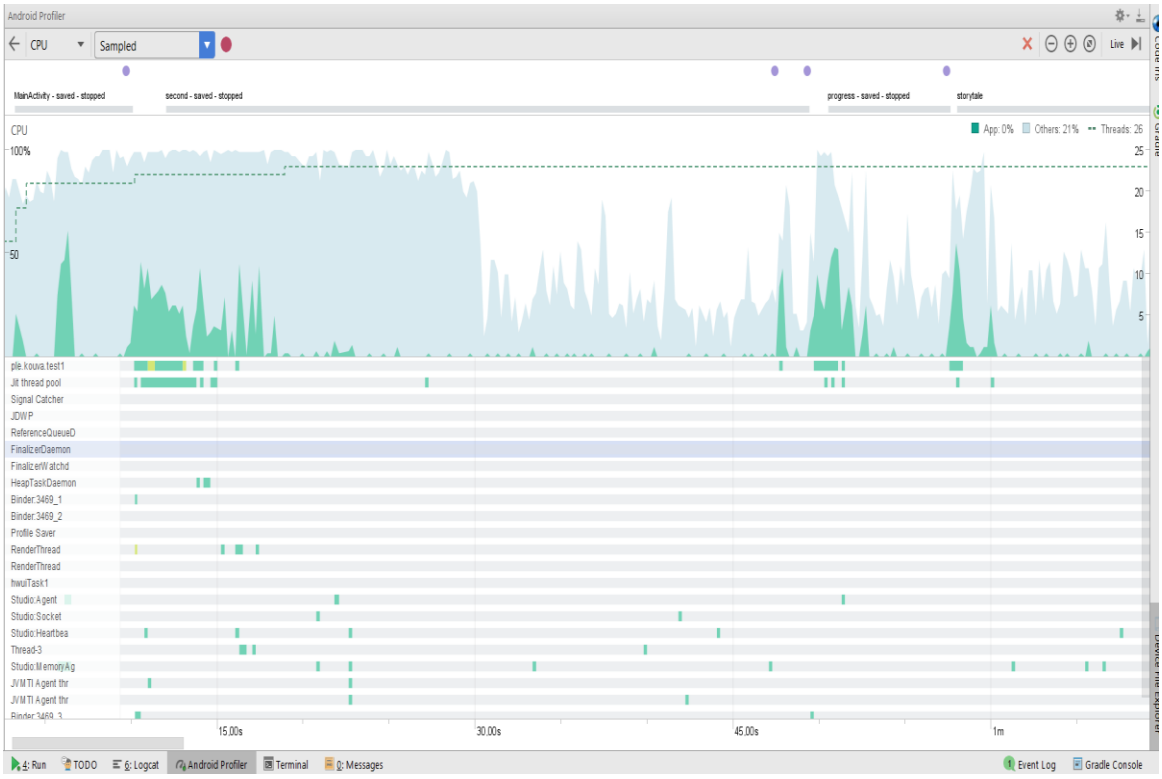
Για να ελέγξουμε αν η εφαρμογή αποδίδει, όπως πρέπει, και αν ικανοποιεί όλες τις απαιτήσεις που έχουν τεθεί, ήταν αναγκαία η διεξαγωγή μερικών διαγνωστικών test. Σε αυτό το κεφάλαιο παρατίθενται οι μέθοδοι με τις οποίες ελέγχθηκε η εφαρμογή, αλλά και τα αποτελέσματα αυτών των ελέγχων. Ο έλεγχος πραγματοποιήθηκε με το built in built in εργαλείο του Android studio, Android Profiler.

4.3.1 CPU Test

Ο έλεγχος της απόδοσης για μια εφαρμογή Android είναι αναγκαίος, καθώς το περιβάλλον και τα resources που θα έχει διαθέσιμα η εφαρμογή σε κάθε συσκευή μπορεί είναι περιορισμένα. Αυτό δεν θα πρέπει να μειώνει σε καμία

περίπτωση αισθητά την απόδοσή της, καθώς κατά την σχεδίαση ο στόχος ήταν να τρέχει αποδοτικά σε όσο το δυνατόν περισσότερες συσκευές.

Ο έλεγχος έγινε σε μια emulated συσκευή Nexus 5X (API 26). Ο έλεγχος της κατανάλωσης CPU είναι σημαντικός, γιατί έτσι καταλαβαίνουμε τον συνολικό φόρτο εργασίας της εφαρμογής σε ένα κανονικό Nexus 5X. Η παρακάτω γραφική παράσταση δείχνει τα logs του monitor του Android Profiler για λίγο παραπάνω από ένα λεπτό χρήσης της εφαρμογής. Να σημειωθεί ότι οι πράσινες γραμμές δείχνουν την CPU που χρειάζεται η εφαρμογή, ενώ οι μπλε δείχνουν την συνολική CPU που χρησιμοποιείται από το τηλέφωνο. Πάνω από το διάγραμμα της CPU μπορούμε να δούμε το πότε ξεκίνησε να τρέχει η κάθε δραστηριότητα της εφαρμογής.



Εικόνα 51: Χρησιμοποίηση CPU

Παρατηρούμε ότι, όταν μια δραστηριότητα εκκινεί, η χρησιμοποιούμενη CPU από την εφαρμογή αυξάνεται αρκετά. Αυτό είναι λογικό όμως, γιατί πρέπει να εκκινηθεί η καινούργια δραστηριότητα και να σχεδιαστεί το UI. Μετά από λίγα δευτερόλεπτα πάλι, παρατηρείται το ποσοστό της CPU να επιστρέφει στα παλιά χαμηλά του επίπεδα.

4.3.2 Support Multiple screen sizes

Το λογισμικό Android τρέχει σε διάφορες συσκευές που έχουν διαφορετικό μέγεθος οθόνη, αλλά και διαφορετικό density. Παρότι το λογισμικό από μόνο του προσπαθεί να προσαρμοστεί στις οθόνες, σε καμία περίπτωση δεν μπορεί να εγγυηθεί την ομαλή εφαρμογή του UI σε κάθε οθόνη. Η υποστήριξη όλων των

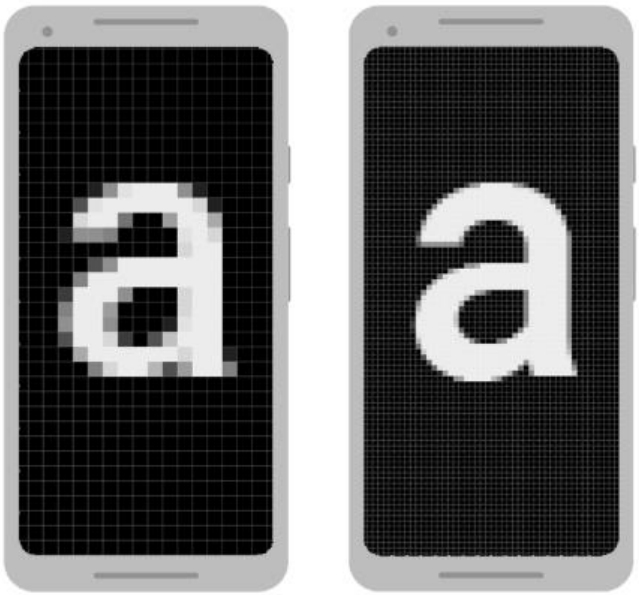
μεγεθών των οθονών αποτελεί τεράστιο πλεονέκτημα για κάθε εφαρμογή που το λαμβάνει υπόψη της, καθώς με αυτόν τον τρόπο αυξάνει τον αριθμό των υποψήφιων χρηστών.

Για το πρόβλημα του διαφορετικού μεγέθους κάθε οθόνης θα πρέπει να λάβουμε υπόψιν μας, όταν φτιάχνουμε το layout, να μην χρησιμοποιούνται συγκεκριμένα νούμερα για τα μεγέθη. Το layout θα πρέπει να είναι ευέλικτο για να μπορεί να προσαρμοστεί σε διαφορετικές οθόνες. Επίσης ο προγραμματιστής της εφαρμογής θα πρέπει να σχεδιάσει διαφορετικά layouts για τα βασικά μεγέθη οθόνης. Αυτά τοποθετούνται στους κατάλληλους φακέλους και, όταν η εφαρμογή τρέχει σε ένα κινητό τηλέφωνο ή tablet, τότε ψάχνει να βρει το κατάλληλο layout γι' αυτή σε αυτούς τους φακέλους. Παρακάτω φαίνονται ενδεικτικά τα βασικά μεγέθη των οθονών με το διάστημα σε ίντσες που τους αναλογεί.

Εικόνα 52: Μεγέθη οθόνης

Μία ακόμα λύση για το πρόβλημα του διαφορετικού μεγέθους των οθονών είναι η παροχή bitmap για τις εικόνες, έτσι ώστε να μπορούν να εφαρμόζουν καλύτερα με τα views των layouts.

Το δεύτερο πρόβλημα είναι ότι εκτός από τις διαφορετικές οθόνες, οι συσκευές Android έχουν και διαφορετική πυκνότητα pixel. Αυτό σημαίνει ότι μια συσκευή μπορεί να έχει 160 pixel σε μια περιοχή και μια άλλη συσκευή να έχει 480 στην ίδια περιοχή. Αυτή η διαφορά μπορεί να οδηγήσει σε μερικές οθόνες οι εικόνες να φαίνονται θολές, καθώς το σύστημα θα τις μεγεθύνει για να καλύψουν τα κατάλληλα pixels.



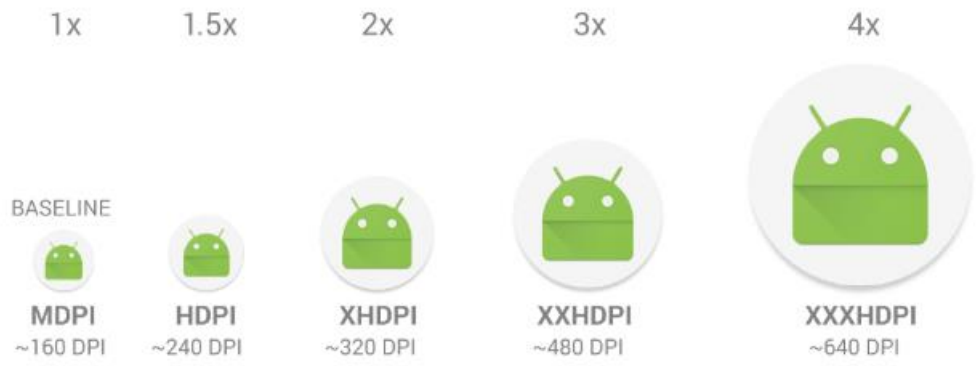
Εικόνα 53: Διαφορετικές πυκνότητες οθόνης

Για τον περιορισμό αυτού του προβλήματος, στα αρχεία xml, όταν δηλώνουμε διαστάσεις, θα πρέπει να δηλώνουμε density-independent pixels (dp) ως μονάδα μέτρησης. Ένα dp είναι μια ψηφιακή μονάδα μέτρησης pixel, περίπου ίση με ένα pixel σε μία οθόνη μέτριας πυκνότητας. Για την μετατροπή των dp σε pixels ακολουθείται ο παρακάτω τύπος.

```
px = dp * (dpi / 160)
```

Εικόνα 54: Τύπος μετατροπής dp σε px

Η βέλτιστη λύση σε αυτό το πρόβλημα όμως είναι η αποθήκευση εικόνων διαφορετικής πυκνότητας pixel στη μνήμη. Το Android χειρίζεται αυτήν την περίπτωση παρόμοια με την περίπτωση των layouts. Βλέπει τι πυκνότητα pixel υπάρχει στην οθόνη της συσκευής και ψάχνει στον κατάλληλο φάκελο να βρει μια εικόνα που να αντιστοιχεί στην αντίστοιχη πυκνότητα.



Εικόνα 55: Παραδείγματα πυκνοτήτων οθόνης

Στην υλοποίηση της παρούσας εφαρμογής ακολουθήθηκαν οι περισσότερες από αυτές τις λύσεις για την αύξηση της συμβατότητας και του αριθμού των υποψήφιων χρηστών. Η μόνη λύση που δεν ακολουθήθηκε ήταν ο σχεδιασμός διαφορετικών layouts για τη λειτουργία landscape του κινητού, καθώς προτιμήθηκε να κλειδωθεί η συγκεκριμένη λειτουργία και να υπάρχει μόνο η portrait. Τα διαφορετικά layouts δοκιμάστηκαν σε διαφορετικά μεγέθη οθονών και σε διαφορετικές πυκνότητες pixel. Μερικές από τις συσκευές που δοκιμάστηκαν μέσα από τους emulators του Android Studio ή μέσα από διαθέσιμες φυσικές συσκευές είναι:

Όνομα	**Μέγεθος οθόνης**	**Ανάλυση**	**Πυκνότητα**
QVGA slider	2,7''	240x320	Ldpi
Nexus 5X	5,2''	1080x1920	420dpi
Nexus 6	5,96''	1440x2560	560dpi
Huawei P20 lite	5,84''	1080x2280	432dpi
Sony Xperia M4 aqua	5,0''	720x1280	293dpi
WVGA (Nexus S)	4,0''	480x800	Hdpi
Nexus 10	10,05''	2560x1600	Xhdpi

5

Επίλογος

Στο κεφάλαιο του επίλογου θα γίνει μια συνοπτική σύνοψη της εμπειρίας της εφαρμογής και θα παρουσιαστούν μερικές ιδέες για μελλοντικές επεκτάσεις της.

5.1 Σύνοψη

Ήξερα εξ αρχής ότι η δημιουργία μιας εφαρμογής κινητού τηλεφώνου δεν θα ήταν μια εύκολη διπλωματική. Όμως, παρόλες τις δυσκολίες που παρουσιάστηκαν κατά τη διάρκεια της ανάπτυξης και του σχεδιασμού, μετά το πέρας της νιώθω ότι έχω αποκομίσει μόνο θετικά στοιχεία. Μέσω αυτής της ευκαιρίας που μου έδωσε το εργαστήριο IVML μπόρεσα και εξέλιξα τις προγραμματιστικές μου ικανότητες, έμαθα να χρησιμοποιώ πολλαπλές νέες τεχνολογίες και οικειοποιήθηκα με τις απίστευτες απαιτήσεις που έχει η ανάπτυξη των εφαρμογών κινητών τηλεφώνων.

Ο στόχος της παρούσας διπλωματικής ήταν να δημιουργήσει μια εφαρμογή που να προσφέρει στον χρήστη μια διαφορετική εμπειρία στον χώρο ενός μουσείου και να αυξήσει την ποσότητα των πληροφοριών που αφομοιώνει ο χρήστης κατά

την παραμονή του σε αυτό. Θεωρώ ότι αυτοί οι στόχοι έχουν επιτευχθεί κατά ένα μεγάλο ποσοστό αν όχι εξ ’ολοκλήρου.

5.2 Μελλοντικές επεκτάσεις

Κάποιες μελλοντικές επεκτάσεις για την βελτίωση ή την ανάπτυξη της εφαρμογής είναι:

5.2.1 Tours

Η προσθήκη ξεναγήσεων οι οποίες θα προτείνουν στον χρήστη διαφορετικές πορείες που μπορεί να ακολουθήσει μέσα στο μουσείο ανάλογα με τα ενδιαφέροντα του. Παραδείγματος χάριν μια πορεία βασισμένη σε χρονολογική σειρά δημιουργίας των εκθεμάτων για έναν χρήστη που θέλει να εντοπίσει την εξέλιξη στην τέχνη ή μια πορεία ανάμεσα στα έργα ενός συγκεκριμένου καλλιτέχνη για έναν άλλο χρήστη που θέλει να μια πιο επικεντρωμένη εμπειρία.

5.2.2 Server Database

Η αντικατάσταση της παρούσας βάσης δεδομένων με μια διαφορετική που θα μπορεί να συνδεθεί με κάποιο server. Αυτή η επέκταση θα προσδώσει στην εφαρμογή πολλές δυνατότητες, μεταξύ των οποίων είναι η δημιουργία προτάσεων εκθεμάτων στον χρήστη με βάση τους πίνακες που του άρεσαν ή με βάση αυτούς που πέρασε την περισσότερη ώρα. Μία άλλη δυνατότητα είναι η δημιουργία λογαριασμού ο οποίος θα προσφέρει μια πιο προσωποποιημένη εμπειρία στον χρήστη, καθώς θα του δίνει και τη δυνατότητα να μεταφέρει τον λογαριασμό του σε άλλες συσκευές.

5.2.3 Ενημερώσεις

Η εφαρμογή θα πρέπει να ενημερώνεται συχνά, έτσι ώστε να μπορεί να παρέχει νέα στοιχεία στον χρήστη, όπως προσθήκη καινούργιων ιστοριών ή ξεναγήσεων, έτσι ώστε να διατηρήσει το ενδιαφέρον του στην εφαρμογή. Επίσης θα πρέπει να υπάρχει συνεχής ενημέρωση, έτσι ώστε να παραμείνει βελτιστοποιημένη και σύγχρονη με τις τελευταίες τεχνολογίες και ενημερώσεις του Android.

5.2.4 Άλλες επεκτάσεις

-Προσθήκη ηχητικής ξενάγησης στις περιηγήσεις που παρέχονται ήδη.

-Αλληλεπίδραση με κοινωνικά δίκτυα μέσω της κοινοποίησης των εκθεμάτων ή της ολοκλήρωσης κάποιας ιστορίας.

-Επέκταση του gamification με την προσθήκη προσκλήσεων, έτσι ώστε να αναπτυχθεί το στοιχείο του υγιούς ανταγωνισμού μεταξύ φίλων.

-Επέκταση του gamification με την προσθήκη βαθμολογίας. Παραδείγματος χάριν ο χρήστης να κερδίζει συγκεκριμένους πόντους, αν βρει το έκθεμα κάτω από ένα ορισμένο χρονικό όριο. Επίσης, να κερδίζει extra πόντους, αν απαντήσει σωστά με

την πρώτη σε μια ερώτηση, λιγότερους extra πόντους, αν απαντήσει με τη δεύτερη και καθόλου αν το βρει με την τρίτη.

-Προσθήκη leaderboards για τους χρήστες με τα υψηλότερα σκορ ανά ιστορία, έτσι ώστε ο κάθε χρήστης να έχει έναν στόχο να ξεπεράσει, αλλά και να μπορεί στο τέλος να αξιολογήσει την προσπάθειά του σε σύγκριση με τους άλλους χρήστες.

-Προσθήκη βαθμολόγησης ιστοριών και ξεναγήσεων με σκοπό τη βελτίωση του περιεχομένου της εφαρμογής μέσω του άμεσου feedback των χρηστών.

6

Βιβλιογραφία

Κεφάλαιο 1 – Εισαγωγή

Omar El Gabry (18/3/2017) Software Engineering — Software Process and Software Process Models (Part 2). 18/6/2018 https://medium.com/omarelgabrys-blog/software-engineering-software-process-and-software-process-models-part-2-4a9d06213fdc

Κεφάλαιο 2 – Θεωρητικό Υπόβαθρο

Cory Schmidt (24/6/2016) What is Android? Here is a complete guide for beginners. 18/6/2018 https://www.androidpit.com/what-is-android

AndroidxCommunity (29/3/2017) What is Android Studio and Android SDK tools? 18/6/2018 http://androiddeveloper.galileo.edu/2017/03/29/android-studio-and-android-sdk-tools/

Techopedia Android SDK. 18/6/2018
https://www.techopedia.com/definition/4220/android-sdk

ALEX MULLIS (11/11/2017) Android Studio tutorial for beginners. 18/6/2018
https://www.androidauthority.com/android-studio-tutorial-beginners-637572/

Android Developers (5/6/2018) Meet Android Studio. 18/6/2018
https://developer.android.com/studio/intro/

Silmaril Consultants What is XML? 18/6/2018 http://xml.silmaril.ie/whatisxml.html

Nitish Chauhan (25/6/2015) What is the use of XML in Android? 18/6/2018
https://www.quora.com/What-is-the-use-of-XML-in-Android

SQLite About SQLite 18/6/2018 https://www.sqlite.org/about.html

kontakt.io What is a beacon? 18/6/2018 https://kontakt.io/beacon-basics/what-is-a-beacon/

Developer Docs Beacons, how do they work? 18/6/2018
https://developer.estimote.com/how-beacons-work/

Estimote What is a beacon region? 18/6/2018
https://community.estimote.com/hc/en-us/articles/203776266-What-is-a-beacon-region-

Estimote What are region Monitoring and Ranging? 18/6/2018
https://community.estimote.com/hc/en-us/articles/203356607-What-are-Region-monitoring-and-Ranging-

Indooratlas How it works. 18/6/2018 http://www.indooratlas.com/positioning-technology/

Daniel Faggella (2/9/2017) What is Machine Learning? 18/6/2018
https://www.techemergence.com/what-is-machine-learning/

Amy Unruh (9/11/2017) What is the TensorFlow machine intelligence platform? 18/6/2018 https://opensource.com/article/17/11/intro-tensorflow

Exastax (3/2/2017) Main Use Cases of TensorFlow 18/6/2018
https://www.exastax.com/deep-learning/top-five-use-cases-of-tensorflow/

Κεφάλαιο 3 – Σχεδιασμός

Noel Ceta (22/2/2018) All You Need to Know About UML Diagrams: Types and 5+ Examples. 18/6/2018 https://tallyfy.com/uml-diagram/

Wikipedia (1/6/2018) Sequence diagram. 18/6/2018
https://en.wikipedia.org/wiki/Sequence_diagram

Wikipedia (20/5/2018) Use case diagram 18/6/2018
https://en.wikipedia.org/wiki/Use_case_diagram

Hyunwook Park, Jaewon Noh, Sunghyun Cho (4/10/2016) Three-dimensional positioning system using Bluetooth low-energy beacons 18/6/2018
http://journals.sagepub.com/doi/full/10.1177/1550147716671720

Developer Docs What is iBeacon? 18/6/2018
https://developer.estimote.com/ibeacon/

Kollol Nath (14/12/2016) The importance of UI and UX in mobile apps. 18/6/2018
http://www.mobiloitte.com/blog/importance-ui-ux-mobile-apps

Κεφάλαιο 4 – Υλοποίηση-Τεστάρισμα

leon-earl (8/6/2018) The theory behind Model View Controller 18/6/2018
https://developer.mozilla.org/en-US/Apps/Fundamentals/Modern_web_app_architecture/MVC_architecture

Paul Trebilcox-Ruiz (4/3/2016) Android From Scratch: Understanding Views And View Groups 18/6/2018 https://code.tutsplus.com/tutorials/android-from-scratch-understanding-views-and-view-groups--cms-26043

Android Developers (27/4/2018) Layouts 18/6/2018
https://developer.android.com/guide/topics/ui/declaring-layout

Android Developers (8/5/2018) Create a List with RecyclerView 18/6/2018
https://developer.android.com/guide/topics/ui/layout/recyclerview

Android Developers (23/4/2018) Screen compatibility overview 18/6/2018
https://developer.android.com/guide/practices/screens_support

Android Developers (23/4/2018) Support different screen sizes 18/6/2018
https://developer.android.com/training/multiscreen/screensizes

Abhiandroid TabHost Tutorial With Example In Android Studio 18/6/2018
http://abhiandroid.com/ui/tabhost

Codelabs developers TensorFlow for Poets 2: TFMobile 18/6/2018
https://codelabs.developers.google.com/codelabs/tensorflow-for-poets-2/#0

Codelabs developers TensorFlow for Poets 2: TFLite Android 18/6/2018
https://codelabs.developers.google.com/codelabs/tensorflow-for-poets-2-tflite/#0