

## Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Προηγμένα Συστήματα Πληροφορικής»

### Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	<b>Σχεδιασμός και Ανάπτυξη Εφαρμογής Android για το Νομισματικό Μουσείο Αθηνών</b>  <b>Design and Development of an Android Application for the Numismatic Museum of Athens</b>
Όνοματεπώνυμο Φοιτητή	<b>Χασάπης Παντελής</b>
Πατρώνυμο	<b>Γεώργιος</b>
Αριθμός Μητρώου	<b>ΜΠΣΠ/ 15095</b>
Υπεύθυνος Καθηγητής	<b>Χρήστος Δουληγέρης, Καθηγητής</b>
Επιβλέπων Καθηγητής	<b>Δρ.Σαράντης Μητρόπουλος, Αναπληρωτής Καθηγητής</b>

Ημερομηνία Παράδοσης **Ιούλιος 2018**

**Τριμελής Εξεταστική Επιτροπή**

(υπογραφή)

(υπογραφή)

(υπογραφή)

Χρήστος Δουληγέρης  
Καθηγητής

Δρ.Σαράντης Μητρόπουλος  
Αναπληρωτής Καθηγητής

Παναγιώτης Κοτζανικολάου  
Επίκουρος Καθηγητής

○

Περιεχόμενα	
Ευχαριστίες.....	7
Περίληψη.....	9
Abstract.....	9
1.Εισαγωγή .....	10
1.1 Χώρος του Προβλήματος.....	10
1.2 Στόχοι της εργασίας.....	11
1.3 Δομή της εργασίας και μεθοδολογία που ακολουθήθηκε .....	11
1.4 Παρόμοια Συστήματα .....	11
1.4.1 Louvre Ma Visite.....	12
1.4.2 Dali Museum Virtual Tour .....	12
1.4.3 British Museum Guide.....	13
1.4.4 The Metropolitan Museum of Art .....	14
1.4.5 AMΘ Mobile .....	14
1.5 Σύγκριση Συστημάτων .....	15
2.Φορητές Συσκευές.....	10
2.1 Έξυπνα τηλέφωνα ( Smartphones ).....	16
2.2 Υπολογιστής ταμπλέτα( Tablet ) .....	17
2.3 Λειτουργικά Συστήματα για Κινητά .....	18
2.4 Android και iOS .....	20
2.4.1 Αρνητικά του Android .....	22
2.4.2 Αρνητικά του iOS.....	22
3. Εφαρμογές Έξυπνων τηλεφώνων .....	22
3.1 Ο κατακερματισμός και τα προβλήματα των προγραμματιστών .....	24
3.2 Τύποι εφαρμογών κινητών συσκευών.....	24
3.2.1 Εγγενείς Εφαρμογές( Native App ):.....	24
3.2.2 Εφαρμογές Ιστού ( Web App ):.....	25
3.2.3 Υβριδικές Εφαρμογές ( Native App ):.....	26
3.3 Ανάπτυξη εφαρμογών σε διαφορετικά λειτουργικά συστήματα .....	27
3.4 Επιλογή κατάλληλης μεθόδου .....	28
4. Λειτουργικό Σύστημα Android και Τεχνολογίες που Χρησιμοποιήθηκαν .....	29
4.1 Αρχιτεκτονική Συστήματος Android.....	30
4.1.1 Linux Kernel .....	31
4.1.2 Hardware Abstraction Layer ( HAL ) .....	31
4.1.3 Android Runtime .....	31
4.1.4 Native Libraries.....	31
4.1.5 Java API Framework .....	31
4.1.6 System Apps .....	32
4.2 Δομικά Στοιχεία Μιας Εφαρμογής.....	32
4.3 Κύκλος ζωής των δραστηριοτήτων .....	33
4.4 Fragment.....	34

4.5 Διεπαφή χρήστη-User interface.....	35
4.6 Material Design .....	36
4.7 Δικαιώματα .....	37
4.8 RecyclerView και CardView.....	38
4.9 Αισθητήρες.....	39
4.10 Τεχνολογίες Που Χρησιμοποιήθηκαν .....	40
4.10.1 Βάση Δεδομένων SQLite .....	40
4.10.2 Google Firebase.....	40
4.10.3 Google Maps .....	42
4.10.4 Fabric Crashlytics.....	43
4.10.5 JSON .....	43
4.10.6 Google Geocoder.....	44
4.11 Εργαλεία ανάπτυξης .....	45
4.11.1 Java.....	45
4.11.2 Android SDK .....	45
4.11.3 Android Studio .....	45
4.11.4 Εξομοιωτής Android .....	45
4.11.5 Adobe Photoshop.....	46
4.11.6 Microsoft Visio .....	46
5. Ανάλυση των λειτουργιών της Εφαρμογής .....	47
5.1 Ανάλυση Απαιτήσεων .....	47
5.2 Σχεδιασμός.....	48
5.2.1 Δυνατότητες Λειτουργιών των Χρηστών.....	48
5.2.2 Σχεδιασμός της Διαδικασίας Σύνδεσης και Εγγραφής του Χρήστη .....	51
5.2.3 Σχεδιασμός της Διαδικασίας Αγοράς Ηχητικής Ξενάγησης.....	53
5.2.4 Σχεδιασμός της Διαδικασίας Ενημέρωσης της Τοπικής Βάσης Δεδομένων .....	54
5.2.5 Σχεδιασμός της Τοπικής και της Firebase Βάσης Δεδομένων .....	55
5.2.6 Η MVC Αρχιτεκτονική .....	57
6. Παρουσίαση της Εφαρμογής .....	58
6.1 Η εφαρμογή στις κινητές συσκευές.....	59
6.1.2 Κεντρική Διεπαφή .....	60
6.1.3 Διεπαφή Συλλογών .....	60
6.1.4 Διεπαφή Ηχητικής Ξενάγησης.....	62
6.1.5 Διεπαφή Αγαπημένων.....	63
6.1.6 Διεπαφή Πλάνου Ορόφων .....	64
6.1.7 Διεπαφή Εισιτήρια/Πρόσβαση.....	65
6.1.8 Διεπαφή Επικοινωνίας και Πρόσβασης .....	65
6.1.9 Διεπαφή Εκδηλώσεων.....	65
6.1.10 Διεπαφή Χρονολογίου .....	66
6.1.11 Διεπαφή Portal.....	67
6.1.12 Διεπαφή Χάρτη .....	69

6.2 Η διεπαφή του Google Firebase.....	70
6.2.1 Authentication.....	70
6.2.2 Database.....	71
6.2.3 Storage .....	72
6.2.5 Ειδοποιήσεις και Remote Config.....	73
7. Αξιολόγηση της Εφαρμογής.....	75
7.1 Αποτελέσματα ερωτηματολογίου .....	75
8. Συμπεράσματα - Μελλοντικές Προτάσεις .....	84
8.1 Συμπεράσματα.....	84
8.2 Μελλοντικές Επεκτάσεις.....	84
Βιβλιογραφία.....	86
ΠΑΡΑΡΤΗΜΑ .....	86
Επεξήγηση κώδικα .....	86
1. Δημιουργία Base Activity και Fragment .....	87
2. Έλεγχος Νεότερης Έκδοσης της Βάσης Δεδομένων. ....	88
3. Αγορά της ηχητικής ξενάγησης .....	91
4. Λήψη των αρχείων της ηχητικής ξενάγησης.....	91
5. Αποστολή και λήψη πληροφοριών με Google Map.....	93
6. Εκδηλώσεις .....	96

## Πίνακας Εικόνων

Εικόνα 1. Στιγμιότυπα του Louvre Ma Visite.....	12
Εικόνα 2. Μουσείο Σαλβαδρό Νταλί.....	74
Εικόνα 3. Οθόνες του Dali Museum.....	14
Εικόνα 4. Στιγμιότυπα του British Guide.....	14
Εικόνα 5. Στιγμιότυπα του The Met.....	15
Εικόνα 6. Διεπαφές του AMΘ.....	15
Εικόνα 7. Το iOS και οι συσκευές της Apple.....	19
Εικόνα 8. Windows Phone 7 μέχρι τα Windows Mobile 10.....	19
Εικόνα 9. Λειτουργικά Συστήματα Android και iOS.....	20
Εικόνα 10. Γραφικό Περιβάλλον Android και iOS.....	22
Εικόνα 11. Framework για δημιουργία εφαρμογών ιστού.....	25
Εικόνα 12. React Native στο Instagram.....	28
Εικόνα 13. Open Handset Alliance.....	29
Εικόνα 14. Αρχιτεκτονική του Android.....	30
Εικόνα 15. Κύκλος ζωής δραστηριότητας.....	33
Εικόνα 16. Ευέλικτη σχεδίαση διεπαφών.....	35
Εικόνα 17. Ιεραρχία διεπαφών.....	35
Εικόνα 18. Παράδειγμα XML κώδικα και το οπτικό αποτέλεσμα.....	36
Εικόνα 19. Gmail App, Πριν και Μετά το Material Design.....	36
Εικόνα 20. Το μοντέλο δικαιωμάτων πριν και μετά το Android 6.0.....	37
Εικόνα 21. Μοντέλο ενός RecyclerView.....	38
Εικόνα 22. Παράδειγμα Χρήσης Cardview με RecyclerView.....	38
Εικόνα 23. Παράδειγμα Χρήσης Cardview στο Google App.....	39
Εικόνα 24. Παράδειγμα JSON δέντρου της Firebase.....	42
Εικόνα 25. Παράδειγμα ενός σφάλματος σε Android App.....	44
Εικόνα 26. Αλγόριθμος δημιουργίας αντικειμένου σε αρχείο JSON.....	45
Εικόνα 27. Αλγόριθμος δημιουργίας πίνακα σε αρχείο JSON.....	45
Εικόνα 28. Χρήση Δεικτών στο Google Map.....	45
Εικόνα 29. Η κλήση προς το διακομιστή της Google.....	47
Εικόνα 30. Εξομοιωτής Android και οι ρυθμίσεις του.....	47
Εικόνα 31. Στιγμιότυπο του Microsoft Visio.....	48
Εικόνα 32. Η MVC Αρχιτεκτονική.....	57
Εικόνα 33. Η MVC Αρχιτεκτονική στο Android.....	58
Εικόνα 34. Αρχική Διεπαφή.....	60
Εικόνα 35. Κεντρική Διεπαφή.....	61
Εικόνα 36. Λίστα της Συλλογής και η διεπαφή ανάλογα με τη κατηγορία.....	61
Εικόνα 37. Παράθυρα ενημέρωσης του χρήστη και διεπαφή PayPal.....	62
Εικόνα 38. Διεπαφή φωτογραφίας και οριζόντιο γκάλερι.....	63
Εικόνα 39. Διεπαφή φωτογραφίας και οριζόντιο γκάλερι.....	63
Εικόνα 40. Στιγμιότυπα Διεπαφής Αγαπημένων.....	64
Εικόνα 41. Στιγμιότυπα Διεπαφής Πλάνου Ορόφων.....	65
Εικόνα 42. Στιγμιότυπα Διεπαφών.....	65
Εικόνα 43. Στιγμιότυπα Διεπαφής Εκδήλωσης.....	66
Εικόνα 44. Στιγμιότυπα Διεπαφής Χρονολόγιου.....	67
Εικόνα 45. Στιγμιότυπα Διεπαφής Portal.....	67
Εικόνα 46. Στιγμιότυπα Διεπαφής Portal.....	68
Εικόνα 47. Στιγμιότυπα Διεπαφής Portal Μενού.....	68
Εικόνα 48. Στιγμιότυπα Αλλαγής Κωδικού.....	69
Εικόνα 49. Δημιουργία ανάρτησης και σχολίων.....	69
Εικόνα 50. Αναζήτηση μέσω διεύθυνσης.....	70
Εικόνα 51. Αναζήτηση μέσω GPS.....	71
Εικόνα 52. Επιλογές Δημιουργίας Λογαριασμού.....	71
Εικόνα 53. Διαχείριση Λογαριασμών Χρηστών.....	72
Εικόνα 54. Βάση Δεδομένων Firebase.....	72
Εικόνα 55. Μηνιαίο Όριο Χρήσης της Βάσης Δεδομένων.....	73
Εικόνα 56. Firebase Storage.....	73
Εικόνα 57. Ενεργοί Χρήστες το διάστημα 1/10-31/12.....	73
Εικόνα 58. Πληροφορίες σχετικά με την εφαρμογή.....	74
Εικόνα 59. Δημιουργία και Διαχείριση Notification.....	74
Εικόνα 60. Προσθήκη Παραμέτρων στο Remote Config.....	74

## Λίστα Διαγραμμάτων

Διάγραμμα 1. Συνολικός αριθμών εφαρμογών ανά κατάσταση μέχρι το Μάιο του 2017 .....	17
Διάγραμμα 2. Παγκόσμιο μερίδιο αγοράς Smartphones μέχρι Q2 2017.....	18
Διάγραμμα 3. Εκδόσεις Λογισμικού iOS και Android μέχρι της 20 Ιανουαρίου 2018.....	21
Διάγραμμα 4. Χαρακτηρίστηκα μιας εφαρμογής στο κάθε λειτουργικό σύστημα.....	23
Διάγραμμα 5. Το κόστος παραμετροποίησης ανάμεσα σε εφαρμογές .....	27
Διάγραμμα 6. Μερίδιο Αγοράς Λειτουργικών Συστημάτων από το 2010 έως το 2017. ....	30
Διάγραμμα 7. Διάγραμμα περίπτωσης χρήσης του επισκέπτη.....	43
Διάγραμμα 8. Διάγραμμα περίπτωσης χρήσης του διαχειριστή. ....	44
Διάγραμμα 9. Διάγραμμα Ακολουθίας Εγγραφής Χρήστη.....	44
Διάγραμμα 10. Διάγραμμα Δραστηριοτήτων Εισόδου Χρήστη. ....	46
Διάγραμμα 11. Διάγραμμα Δραστηριοτήτων Αγοράς και Αναπαραγωγής Ηχητικής Ξενάγησης. ....	47
Διάγραμμα 12. Διάγραμμα Δραστηριοτήτων Ενημέρωσης Τοπικής Βάσης Δεδομένων .....	48
Διάγραμμα 13. Μοντέλο της Τοπικής Βάσης Δεδομένων .....	49
Διάγραμμα 14. Cloud Βάση Δεδομένων στη Firebase.....	50
Διάγραμμα 15. Κατανομή δείγματος βάση το φύλο.....	75
Διάγραμμα 16. Κατανομή δείγματος με βάση την ηλικία.....	76
Διάγραμμα 17. Μορφωτικό επίπεδο.....	76
Διάγραμμα 18. Εργασιακή απασχόληση.....	76
Διάγραμμα 19. Ερωτήσεις 5 και 6.....	77
Διάγραμμα 20. Μηνιαίες αγορές των χρηστών.....	77
Διάγραμμα 21. Συνήθειες των χρηστών.....	77
Διάγραμμα 22. Βαθμός εξοικείωσης με το Android.....	78
Διάγραμμα 23. Πλοήγηση και Περιεχόμενο της εφαρμογής.....	78
Διάγραμμα 24. Ερωτήσεις 12 και 13.....	78
Διάγραμμα 25. Ερωτήσεις 14 και 15.....	79
Διάγραμμα 26. Έμφαση στο κείμενο .....	79
Διάγραμμα 27. Ευκολία στην αναζήτηση πληροφορίας.....	79
Διάγραμμα 28. Ανάγκη για διαρκή σύνδεση στο διαδίκτυο.....	80
Διάγραμμα 29. Ανάγκη για περαιτέρω λειτουργίες.....	80
Διάγραμμα 30. Ανάγκη για είσοδο με social λογαριασμό.....	81
Διάγραμμα 31. Χρήση της εφαρμογής στο Νομισματικό Μουσείο.....	81
Διάγραμμα 32. Χρήση της ηχητικής ξενάγησης.....	81
Διάγραμμα 33. Χρήση του Portal.....	82
Διάγραμμα 34. Χρήση των ειδοποιήσεων.....	82
Διάγραμμα 35. Προβλήματα κατά την περιήγηση.....	83
Διάγραμμα 36. Καταγραφές ασταθειών μέσω Fabric.....	83
Διάγραμμα 37. Επίσκεψη στο Νομισματικό Μουσείο.....	83

## Λίστα Πινάκων

Πίνακας 1. Χαρακτηρίστηκα μιας εφαρμογής στο κάθε λειτουργικό σύστημα.....	23
--	----

## Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον κ. Δουλουγέρη για την ανάθεση της μεταπτυχιακής μου διατριβής, καθώς και τον Δρ. Μητρόπουλο για την επίβλεψη και την επιστημονικά άρτια καθοδήγησή του κατά τη διάρκεια του σχεδιασμού και της ανάπτυξης αυτής της εργασίας, αλλά και για τις καίριες συμβουλές που μου έδωσε για τη βελτίωση της εφαρμογής που αναπτύχθηκε.

Τέλος, ευχαριστίες οφείλω στην οικογένειά μου και στους φίλους μου, για την ηθική υποστήριξη και βοήθεια όλα τα χρόνια των σπουδών, όπως επίσης για την κατανόηση και την ενθάρρυνση κατά τη διάρκεια εκπόνησης της μεταπτυχιακής διατριβής.



## Περίληψη

Η παρούσα μεταπτυχιακή διατριβή έχει ως κύριο στόχο την ανάπτυξη μιας ψηφιακής εφαρμογής σε περιβάλλον Android. Η εφαρμογή αφορά το Νομισματικό μουσείο. Μέσω της εφαρμογής ο επισκέπτης θα έχει τη δυνατότητα να πλοηγηθεί και να ξεναγηθεί στους χώρους του μουσείου μέσω της έξυπνης συσκευής του. Επιπλέον, στην εφαρμογή έχουν προστεθεί λειτουργίες που κάνουν ευκολότερη την εμπειρία επίσκεψης του χρήστη στο μουσείο.

Αν θελήσουμε να συνοψίσουμε τα βασικά στοιχεία αυτής της διπλωματικής εργασίας, αυτά επικεντρώνονται σε δύο άξονες. Ο πρώτος περιστρέφεται γύρω από τις λειτουργικές παραμέτρους και την αισθητική παρουσίαση της εφαρμογής, που οριοθετεί την ευχρηστία και την ευκολία πρόσβασης μέσω των διεπαφών ενός έξυπνου κινητού τηλεφώνου. Ο δεύτερος αφορά την παρουσία των λειτουργιών σύννεφου (cloud) που αξιοποιήθηκαν στην εφαρμογή μέσω του Google Firebase.

Τέλος περιγράφονται λεπτομερώς και αναλύονται όλα τα στάδια της δημιουργίας της εφαρμογής, από τη μελέτη των λειτουργικών απαιτήσεων και την αρχιτεκτονική του συστήματος και την τεχνική υλοποίηση, μέχρι την παρουσίαση των συμπερασμάτων και των μελλοντικών προοπτικών της.

## Abstract

The purpose of this master thesis is to develop a digital application for the Android platform. The application concerns the Numismatic Museum. Through the application the visitor will be able to navigate and tour virtually in the museum through his smartphone. In addition, features that make it easier for a user to visit the museum have been added to the application.

If we want to summarize the core elements of this diploma thesis, we focus on two axes. The first revolves around the operating parameters and the aesthetic presentation of the application, which may limit the usability and the ease of access through the interfaces of a smartphone. The second concerns the presence of cloud features that were exploited in the application through the use of Google Firebase.

Finally, all the stages are described in detail and the creation of the application is analyzed, from the study of the functional requirements and the architecture of the system and the technical implementation until the presentation of its conclusions and future perspectives.

# 1.Εισαγωγή

Στη σημερινή εποχή η ραγδαία αύξηση των κινητών συσκευών και η ανάγκη για υπερπληροφόρηση, έχει οδηγήσει στην αυξανόμενη εμφάνιση σύγχρονων ηλεκτρονικών εφαρμογών και υπηρεσιών, που έχουν ως στόχο να διευκολύνουν την απομακρυσμένη επικοινωνία και να παρέχουν στους χρήστες τη δυνατότητα να πλοηγηθούν και να λάβουν την απαραίτητη πληροφορία που επιθυμούν ταχύτατα. Σε αυτό έχει συμβάλει ιδιαίτερα το διαδίκτυο και η άνθιση της υπολογιστικής νέφους (cloud computing) τα τελευταία χρόνια, τεχνολογίες οι οποίες έχουν δώσει τη δυνατότητα στο τελικό χρήστη σε οποιοδήποτε σημείο του κόσμου βρίσκεται να έχει άμεσα αποτελέσματα και επαρκή πληροφόρηση.

Όσο αναφορά τα μουσεία για τα οποία θα αναφερθούμε, τα τελευταία χρόνια έχουν δημιουργήσει τους κατάλληλους ιστότοπους όπου οι επισκέπτες μπορούν να βρουν τις κατάλληλες πληροφορίες που τους ενδιαφέρουν αναφορικά με το περιεχόμενο του μουσείου, τη πρόσβαση προς το μουσείο και τα στοιχεία επικοινωνίας. Όμως είτε κάποια μουσεία δεν έχουν δημιουργήσει τους πλέον «φιλικούς» ιστότοπους είτε το περιεχόμενο που περιέχουν δεν ανταποκρίνεται στα σημερινά πρότυπα, όπου χαρακτηρίζεται από την αμεσότητα και την επαρκή πληροφορία.

Στην παρούσα μεταπτυχιακή διατριβή θα παρουσιαστεί η ανάπτυξη της εφαρμογής σε περιβάλλον Android για το Νομισματικό Μουσείο Αθηνών και πώς οι νέες τεχνολογίες θα βοηθήσουν το τελικό χρήστη στην καλύτερη ενημέρωση του και περιήγηση στους χώρους του μουσείου.

Στο παρόν κεφάλαιο θα αναφερθούμε στο σκοπό εκπόνησης της διπλωματικής εργασίας, στο χώρο προβλήματος και σε παρόμοιες εφαρμογές Ελληνικών και ξένων μουσείων.

## 1.1 Χώρος του Προβλήματος

Τις προηγούμενες δεκαετίες όταν ακόμα η τεχνολογία δεν είχε παρεισφρήσει στη καθημερινότητά μας, τα μουσεία δεν κατείχαν τα κατάλληλα μέσα για την απαραίτητη ενημέρωση του επισκέπτη, ο οποίος μπορούσε να ενημερωθεί μόνο τηλεφωνικά ή μέσω κάποιων ενημερωτικών φυλλαδίων σχετικά με το μουσείο. Οι ξεναγήσεις γινόντουσαν μέσω φυλλαδίων του μουσείου είτε μέσω ξεναγού. Τα επόμενα χρόνια πριν αρχίσει να διαδίδεται το διαδίκτυο άρχισαν σιγά-σιγά να προστίθενται ψηφιακά μέσα στα μουσεία, για παράδειγμα υπολογιστές, προβολικά και ξεναγήσεις μέσω ηχητικών συσκευών τύπου Walkman.

Στη συνέχεια το διαδίκτυο ήρθε για να εκσυγχρονίσει τα προαναφερθέντα, σταδιακά άρχισαν οι δημιουργίες ιστοσελίδων και με την εξέλιξη των φορητών συσκευών δημιουργήθηκαν οι κατάλληλες εφαρμογές σε περιβάλλον Android/iOS/Windows Phone όπως τις γνωρίζουμε σήμερα. Μέσω των εφαρμογών αυτών πλέον μπορεί ο επισκέπτης να ξεναγηθεί στο χώρο του μουσείου μέσω τρισδιάστατης απεικόνισης, να εστιάσει περισσότερο σε κάποιο έκθεμα που εντυπωσιάστηκε, να ενημερωθεί για κάποια εκδήλωση που θα πραγματοποιηθεί, να λάβει πληροφορίες αναφορικά με το μουσείο και να επιτευχθεί ηχητική ξεναγήση ανάλογα με τη γλώσσα που έχουν επιλέξει.

Ο επισκέπτης μέσω των κινητών εφαρμογών μπορεί να οργανώσει καλύτερα το χρόνο του στο μουσείο, να εστιάσει στα εκθέματα που ενδιαφέρεται, να περιηγηθεί στις υπηρεσίες του μουσείου απομακρυσμένα, ακόμα και από το σπίτι του. Δεν χρειάζεται

πλέον υπολογιστής απλά μια φορητή συσκευή που έχει πρόσβαση στο διαδίκτυο, ταχύτερη εξυπηρέτηση και μείωση του χρόνου αναμονής στα εκδοτήρια. Προσφέρονται επίσης, διαδικτυακές αποκλειστικές προσφορές στους επισκέπτες ώστε να προτείνουν το μουσείο και σε κάποιο φιλικό πρόσωπο ή να τους δοθεί το κίνητρο να επισκεφτούν ξανά το μουσείο.

## 1.2 Στόχοι της εργασίας

Αντικείμενο της παρούσας μεταπτυχιακής διατριβής αποτελεί η μελέτη και ανάπτυξη της εφαρμογής του Νομισματικού Μουσείου Αθηνών με λειτουργικό σύστημα Android με τους πλέον ενδεδειγμένους τρόπους σχεδίασης και συγχρόνων λειτουργιών. Και στόχος να βελτιώσουμε την εμπειρία ξενάγησης του χρήστη. Το βασικό χαρακτηριστικό της εφαρμογής είναι να είναι εύχρηστη και φιλική στο τελικό χρήστη, να κάνει χρήση των κατάλληλων πόρων ώστε να μην επιβαρύνει τις επιδόσεις της συσκευής, να προσαρμόζεται σε κάθε τύπου συσκευή (tablet/smartphone) ανάλογα με τις διαστάσεις της οθόνης, να παρέχει τη κατάλληλη πληροφόρηση και να αξιοποιούνται νέες τεχνολογίες της υπολογιστικής νέφους(Cloud).

## 1.3 Δομή της εργασίας και μεθοδολογία που ακολουθήθηκε

Στην παρούσα μεταπτυχιακή διατριβή, αρχικά θα αναφερθούμε σε παρόμοια συστήματα, αναφέροντας πληροφορίες και θα κάνουμε μια σύγκριση μεταξύ τους. Στη συνέχεια θα γίνει αναφορά σχετικά με τις φορητές συσκευές, του τύπους των εφαρμογών που μπορούν να αναπτυχθούν και ο τρόπος επιλογής ανάλογα με τις ανάγκες που απαιτούνται για να υλοποιηθεί μια εφαρμογή, έπειτα θα γίνει μια μικρή παρουσίαση του λειτουργικού συστήματος Android, τα δομικά στοιχεία που χρησιμοποιήθηκαν, το Material Design της Google και οι τεχνολογίες που αξιοποιήθηκαν ώστε να ολοκληρωθεί η εφαρμογή, όπως η SQL-Lite βάση δεδομένων, η Firebase κλπ.

Επιπλέον, παρουσιάζεται η ανάλυση απαιτήσεων, η περιγραφή της σχεδίασης και ο τρόπος υλοποίησης μέσω κατάλληλων διαγραμμάτων UML. Έπειτα, γίνεται η παρουσίαση της λειτουργικότητας της εφαρμογής μέσω των κατάλληλων στιγμιότυπων και περιγραφή αυτών. Το κεφάλαιο μπορεί να θεωρηθεί ως εγχειρίδιο χρήσης.

Στο επόμενο κεφάλαιο, ακολουθεί η αξιολόγηση της εφαρμογής. Για την αξιολόγηση δημιουργήθηκε ένα ερωτηματολόγιο με σκοπό τη βελτίωση της εφαρμογής, μέσω του Google Docs διανεμήθηκε σε εικοσιπέντε άτομα, ώστε να λάβουμε τις απαντήσεις. Από τους ερωτώμενους ζητήθηκε να απαντήσουν το κατά πόσο ορισμένες λειτουργίες τις θεωρούν χρήσιμες και τυχόν βελτιώσεις που θα χρειαστούν.

Στο τελευταίο κεφάλαιο, παρουσιάζονται τα συμπεράσματα που προέκυψαν από την υλοποίηση της διπλωματικής και την αξιολόγηση της. Επίσης γίνεται αναφορά σε μελλοντικές βελτιώσεις βάσει νεότερων τεχνολογιών και των αποτελεσμάτων της αξιολόγησης.

Τέλος, υπάρχει στο παράρτημα παρουσιάζονται τα σημαντικότερα κομμάτια του κώδικα της εφαρμογής. Γίνεται ανάλυση αυτών και υπάρχουν στιγμιότυπα από το Android Studio.

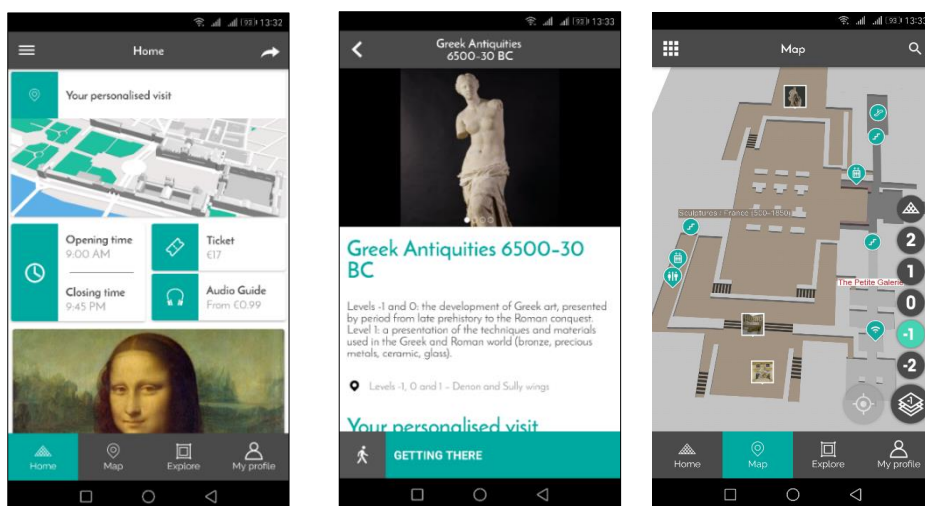
## 1.4 Παρόμοια Συστήματα

Σε αυτό το κεφάλαιο θα παρουσιαστούν εφαρμογές διάσημων μουσείων του εξωτερικού και η μοναδική στην Ελλάδα εφαρμογή για το Αρχαιολογικό Μουσείο της Θεσσαλονίκης, θα

αναφερθούν οι καινοτομίες και οι παροχές τους. Όλες οι εφαρμογές αφορούν το λειτουργικό σύστημα Android.

#### 1.4.1 Louvre Ma Visite

Το μουσείο του Λούβρου σε συνεργία με την SmartApps έχει αναπτύξει τη δικιά της εφαρμογή/ξεναγό. Μέσω αυτής της υλοποίησης ο επισκέπτης έχει τη δυνατότητα να πλοηγηθεί μέσω τρισδιάστατης απεικόνισης του πλάνου ορόφων στο μουσείο για καλύτερη κατανόηση του χώρου, να εντοπίσει τα εκθέματα που τον ενδιαφέρουν, ταυτόχρονα μπορεί να προβεί σε αγορά του ηχητικού ξεναγού που διαθέτει για ηχητική ξενάγηση για κάθε έκθεμα. Συνολικά διαθέτει εξακόσια αρχεία ήχου.



Εικόνα 1. Στιγμιότυπα του Louvre Ma Visite (<https://play.google.com/>)

Ένα από τα βασικά χαρακτηριστικά της εφαρμογής είναι η μη διαρκής σύνδεση στο διαδίκτυο. Επίσης, είναι εύχρηστη και το περιβάλλον χρήσης της είναι ιδιαίτερα φιλικό. Άλλο ένα χαρακτηριστικό της γνώρισμα είναι η παροχή ηχητικής ξενάγησης εξακοσίων εκθεμάτων καθώς και η τρισδιάστατη και διαδραστική απεικόνιση 60.000 m<sup>2</sup> τ.μ. του μουσείου. Παρέχονται, ακόμη, οδηγίες κατεύθυνσης στον επισκέπτη για το χώρο μέσω GPS. Τέλος, η εφαρμογή είναι συμβατή σε συσκευές με λογισμικό μεγαλύτερο του Android 4.3 και iOS 8.0 και διατίθεται σε 7 διαφορετικές γλώσσες.

#### 1.4.2 Dali Museum Virtual Tour

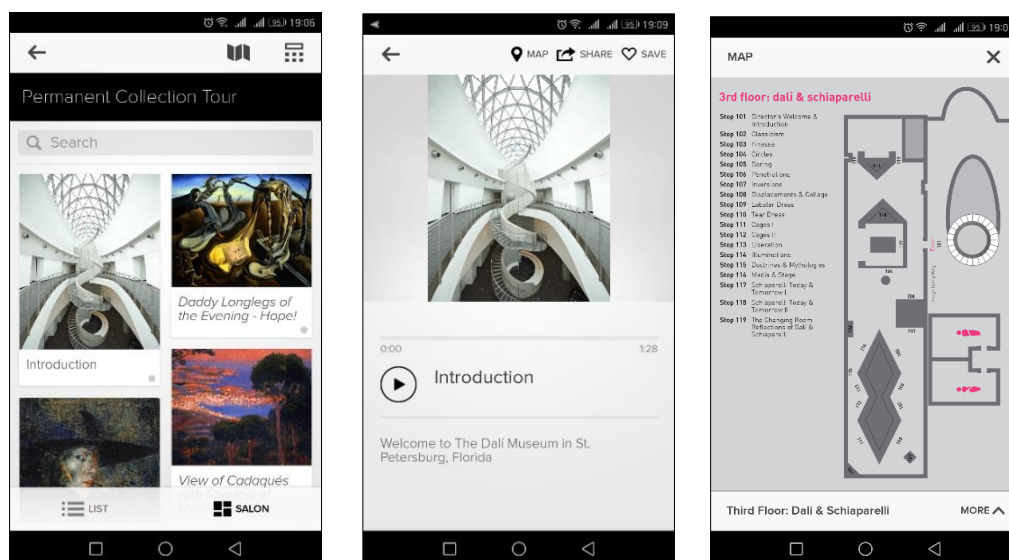


Εικόνα 2. Μουσείο Σαλβαδόρ Νταλί (<https://thedali.org/virtual-tour/>)

Η παρούσα εφαρμογή αφορά το The Dali Museum που βρίσκεται στην Αγία Πετρούπολη της Φλόριντα των ΗΠΑ, και παρουσιάζει τη καριέρα του γνωστού ζωγράφου Σαλβαδόρ Νταλί. Πρόκειται για μια ψηφιακή ξενάγηση που δίνει τη δυνατότητα στους επισκέπτες να βιώσουν τις εμμονές, το χιούμορ και το ταλέντο του Νταλί μέσω αυτής της εφαρμογής.

Διαθέτει βασικά έργα κλειδιά από κάθε στιγμή της καλλιτεχνικής του δραστηριότητας. Η εφαρμογή βρίσκεται στο Google Play Store όπου οι επισκέπτες μπορούν να την κατεβάσουν και να περιηγηθούν στους χώρους του μουσείου. Στα θετικά της εφαρμογής συμπεριλαμβάνεται η λειτουργία της χωρίς σύνδεση στο διαδίκτυο. Στα αρνητικά βρίσκουμε ότι το πλάνο ορόφων του μουσείου είναι φωτογραφία, δεν είναι διαδραστικό και περιέχει λιγιστές λειτουργίες.

Η συγκεκριμένη εφαρμογή παρουσιάζει ποικίλα χαρακτηριστικά. Αρχικά, το περιβάλλον χρήσης είναι εύχρηστο και μοντέρνο. Δεν απαιτεί διαρκή σύνδεση στο διαδίκτυο και παρέχει δωρεάν ηχητική περιγραφή των εκθεμάτων. Επιπλέον, δίνει τη δυνατότητα στους επισκέπτες να επιλέξουν μεταξύ τεσσάρων διαφορετικών γλωσσών και να κοινοποιήσουν τα εκθέματά τους στα κοινωνικά δίκτυα. Άλλη μια παροχή της εφαρμογής αυτής, είναι ένα παιχνίδι ερωτήσεων καθώς και η συμβατότητα με μεγάλο εύρος συσκευών.

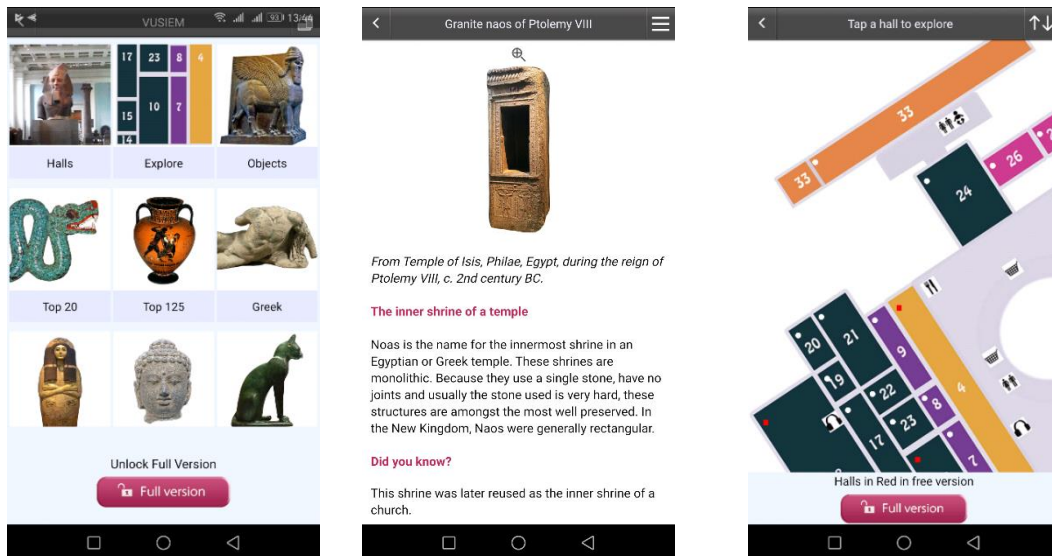


Εικόνα 3. Οθόνες του Dali Museum. (<https://play.google.com/>)

### 1.4.3 British Museum Guide

Η αντίστοιχη υλοποίηση για το Βρετανικό Μουσείο καλύπτει όλα τα εκθέματα που βρίσκονται στο μουσείο, όπως τα σχετικά με την Αρχαία Ελλάδα και Αίγυπτο και πολλά άλλα. Διαθέτει και αυτό τρισδιάστατη απεικόνιση του πλάνου ορόφων και ηχητική ξενάγηση. Όμως η σχεδίαση της εφαρμογής δεν έχει ακολουθήσει τις σχεδιαστικές τάσεις της Google το Material Design, για το οποίο θα αναφερθούμε στα επόμενα κεφάλαια.

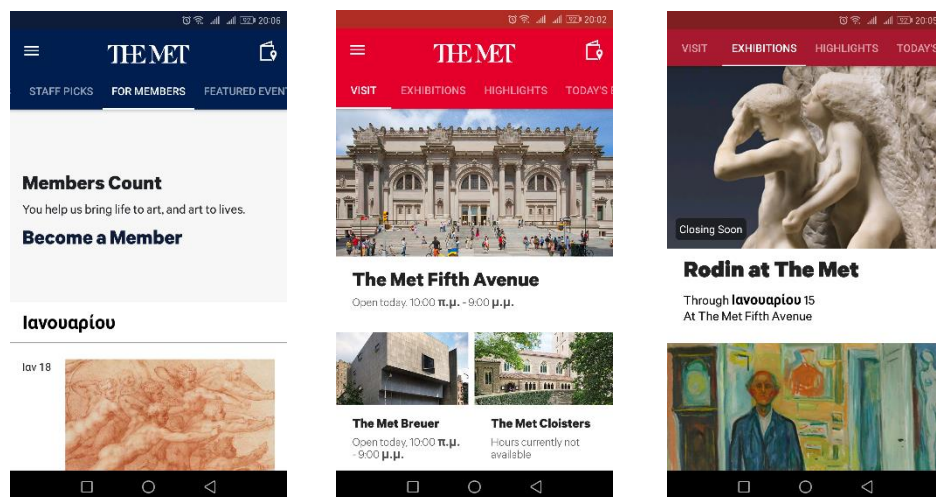
Οι διεπαφές και η συνολική εικόνα της εφαρμογής παραπέμπουν σε απαρχαιωμένη πλέον σχεδίαση, σε σύγκριση με τις δύο προηγούμενες εφαρμογές που είναι φιλικές προς το χρήστη. Επιπλέον, δεν έχει γίνει η απαραίτητη βελτιστοποίηση ώστε να μην διαπιστώνονται καθυστερήσεις στο μενού της εφαρμογής.



Εικόνα 4. Στιγμιότυπα του British Guide. (<https://play.google.com/>)

#### 1.4.4 The Metropolitan Museum of Art

Το Μητροπολιτικό Μουσείο Τέχνης της Νέας Υόρκης είναι ένα από τα μεγαλύτερα και σημαντικότερα μουσεία τέχνης. Ιδρύθηκε το 1872 και περιέχει συλλογές με περισσότερα από δύο εκατομμύρια έργα τέχνης. Η υψηλή επισκεψιμότητα και ακολουθώντας τις τάσεις της εποχής, το μουσείο το 2015 διέθεσε στο κοινό τις εφαρμογές για κινητές συσκευές με λειτουργικό σύστημα Android και iOS. Κύρια χαρακτηριστικά των εφαρμογών είναι η φιλικότητα προς το χρήστη, το γεγονός ότι διαθέτει τρισδιάστατη απεικόνιση και δυνατότητα για προσθήκη στίγματος στο χάρτη για ευκολία εύρεσης εκθεμάτων και σύστημα προτάσεων βάσει των συνηθειών του χρήστη. Αρνητικό της εφαρμογής είναι η απαίτηση για συνεχή σύνδεση στο διαδίκτυο για τη λειτουργία της.

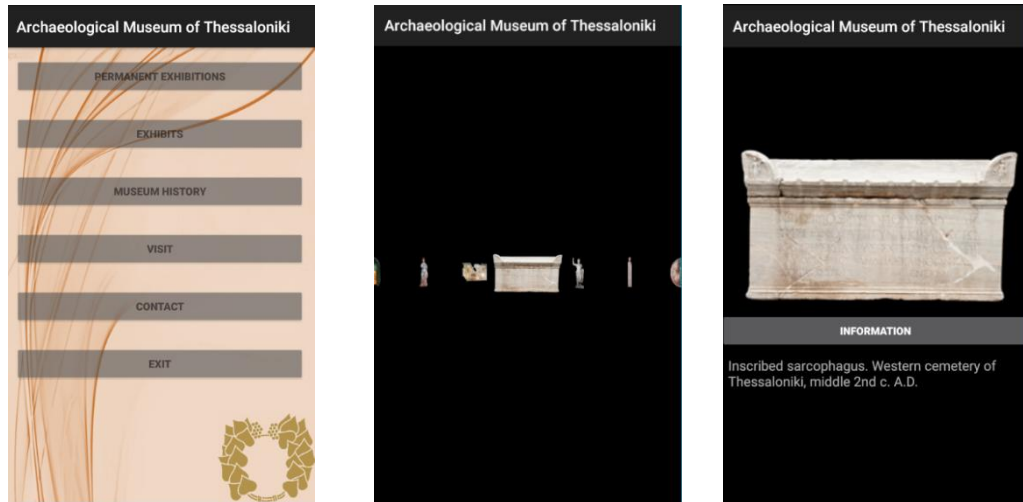


Εικόνα 5. Στιγμιότυπα του The Met (<https://play.google.com/>)

#### 1.4.5 AMΘ Mobile

Το Αρχαιολογικό Μουσείο της Θεσσαλονίκης, το οποίο είναι το πρώτο μουσείο στην Ελλάδα που αποκτά τη δικιά της κινητή εφαρμογή και έχει πλέον παρουσία και στο Play Store της Google. Η εφαρμογή δημιουργήθηκε από φοιτητές στο πλαίσιο της πτυχιακής τους εργασίας. Οι επισκέπτες έχουν τη δυνατότητα να τη κατεβάσουν στο κινητό τους και

να πλοηγηθούν στις λειτουργίες της εφαρμογής. Μπορούν να ενημερωθούν αναφορικά με το ωράριο λειτουργίας, το κόστος των εισιτηρίων, και να λάβουν ενημερώσεις και ανακοινώσεις σχετικά με τις μόνιμες εκθέσεις του μουσείου. Επιπλέον ενημερώνει τους επισκέπτες με τους τρόπους πρόσβασης στο μουσείο.



Εικόνα 6. Διεπαφές του ΑΜΘ (<https://play.google.com/>)

### 1.5 Σύγκριση Συστημάτων

Θα προσπαθήσουμε να συγκρίνουμε τις εφαρμογές που παρουσιάστηκαν. Αρχικά οι εφαρμογές που ξεχωρίζουν από τις προαναφερθείσες είναι του μουσείου του Λούβρου και του μουσείου Σαλβαδόρ Νταλί. Οι δύο αυτές εφαρμογές ξεχωρίζουν λόγω των περισσότερων λειτουργιών, της μοντέρνας σχεδίασης και του εύχρηστου περιβάλλοντος χρήσης. Η εφαρμογή του Μητροπολιτικού Μουσείου Τέχνης της νέας Υόρκης δεν απαιτεί διαρκή σύνδεση στο διαδίκτυο, το οποίο δίνει τη δυνατότητα στον επισκέπτη να διαθέτει συσκευή που δεν περιέχει 3G/4G συνδεσιμότητα. Διαθέτουν πληθώρα εκθεμάτων με ηχητική ξενάγηση και υποστηρίζουν διαφορετικές γλώσσες. Στα αρνητικά τους είναι, η απουσία 3-D απεικόνισης του Dali Museum και το πλάνο ορόφων είναι απλά μια φωτογραφία, ο χρήστης δεν μπορεί να επιλέξει και να πλοηγηθεί απευθείας σε κάποιο έκθεμα ή δωμάτιο που επιθυμεί και θα πρέπει να βρει το κατάλληλο έκθεμα/δωμάτιο εξερευνώντας την εφαρμογή.

Το μουσείο του Λούβρου διαθέτει σχετικά επιβαρυσμένο UI, το οποίο σε συσκευές με χαμηλή επεξεργαστική ισχύ δημιουργούνται καθυστερήσεις. Από την άλλη η εφαρμογή του Βρετανικού Μουσείου, διαθέτει απαραιτημένη σχεδίαση και διαπιστώνονται καθυστερήσεις κατά την περιήγηση στις διεπαφές. Παρότι την έλλειψη ανανεωμένης σχεδίασης, διαθέτει μεγάλο αριθμό εκθεμάτων. Επίσης, διαθέτει ηχητική ξενάγηση και ηχητική ανάγνωση των κειμένων για άτομα με προβλήματα υγείας για τη καλύτερη διευκόλυνσή τους. Επιπλέον, η εφαρμογή του Μητροπολιτικού Μουσείου Τέχνης της Νέας Υόρκης ακολουθεί τη σχεδιαστικούς κανόνες του Material Design της Google, διαθέτει αρκετές λειτουργίες όπως και τα προαναφερθέντα μουσεία. Σε σχέση με τις άλλες εφαρμογές περιέχει τη μεγαλύτερη ενσωμάτωση των κοινωνικών μέσων δικτύωσης στην εφαρμογή και τη δυνατότητα δημιουργίας λογαριασμού ώστε να προτείνει στους επισκέπτες βάσει παραμέτρων ποια δωμάτια εκθέματα να επισκεφτούν βάσει των αγαπημένων τους κατηγοριών και ανάλογα με το χρόνο που διαθέτουν.

## 2. Φορητές συσκευές

Οι φορητές συσκευές αποτελούν ένα ευρύ σύνολο. Στην ενότητα αυτή παρουσιάζονται οι δύο επικρατέστερες φορητές συσκευές (Smartphones, tablet) με τα χαρακτηριστικά τους καθώς και οι ανάγκες που επιχειρούν να ικανοποιήσουν, οι οποίες ταυτίζονται αλλά και ποικίλουν σε μεγάλο βαθμό.

### 2.1 Έξυπνα τηλέφωνα(Smartphones)

Τα έξυπνα τηλέφωνα έκαναν την εμφάνισή τους στα τέλη της δεκαετίας του 2000.

Πρόκειται για την εξέλιξη των κλασικών κινητών τηλεφώνων και παρέχουν μεγαλύτερη υπολογιστική ισχύ και συνδεσιμότητα. Στη σημερινή εποχή τα έξυπνα τηλέφωνα προσφέρουν δυνατότητες αναπαραγωγής πολυμέσων, λήψης φωτογραφιών και βίντεο σε υψηλή ανάλυση, πλοήγησης μέσω GPS και γρήγορης πρόσβασης στο διαδίκτυο μέσω τεχνολογιών, Wi-Fi ή 4G.

Τα συμβατικά κινητά έχουν τη δυνατότητα να τρέχουν εφαρμογές βασισμένες σε πλατφόρμες Java Micro Edition και BREW, ενώ τα έξυπνα τηλέφωνα επιτρέπουν στους χρήστες να εγκαταστήσουν εφαρμογές ανάλογα με το λειτουργικό σύστημα της συσκευής. Στη διάρκεια των τελευταίων χρόνων, αναπτύχθηκαν αρκετά λειτουργικά συστήματα. Τα δημοφιλέστερα είναι το BlackBerry OS, το iOS, το Symbian OS, το Windows Phone και το Android. Από τα προαναφερθέντα λειτουργικά συστήματα, δύο από αυτά κατάφεραν να επικρατήσουν, δηλαδή το Android και το iOS, για τα οποία θα γίνει περαιτέρω ανάλυση στο επόμενο κεφάλαιο.

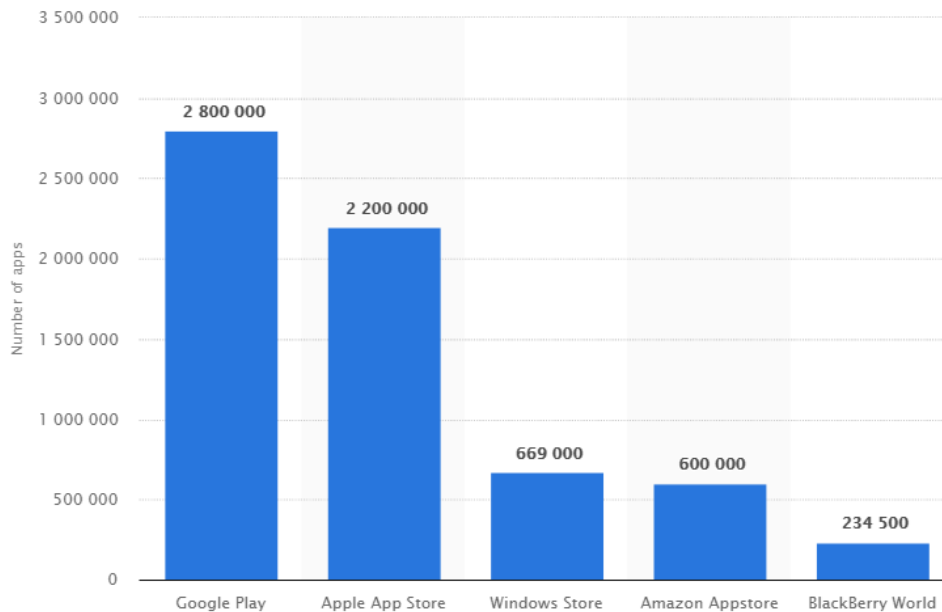
Κύριο χαρακτηριστικό των έξυπνων τηλεφώνων είναι ότι δίνουν τη δυνατότητα στο χρήστη να έχει πρόσβαση σε λειτουργίες και προγράμματα που μέχρι να εμφανιστούν οι έξυπνες συσκευές είχαν μόνο οι υπολογιστές. Πλέον διαθέτουν οθόνες αφής για ευκολότερη χρήση, υποστήριξη 3G/4G τεχνολογίας σύνδεσης, ενσωματωμένο Wi-Fi για ασύρματη συνδεσιμότητα με το διαδίκτυο μέσω των κατάλληλων σημείων πρόσβασης, αισθητήρες θέσης, κίνησης και περιβάλλοντος. Διαθέτουν μεγάλο αποθηκευτικό χώρο ώστε ο χρήστης να έχει τη δυνατότητα να κάνει λήψη των εφαρμογών που του χρειάζονται, αλλά και για να διατηρεί τα απαραίτητα έγγραφα, φωτογραφίες και μουσική στη συσκευή του. Ο χρήστης μπορεί να επιλέξει και να εγκαταστήσει ανάμεσα σε ένα μεγάλο εύρος εφαρμογών, το οποίο έχει ως αποτέλεσμα οι χρήστες να μπορούν να εξατομικεύσουν το εύρος των εφαρμογών στις συσκευές τους αυτές, με τέτοιο τρόπο ώστε να ταιριάζουν στον τρόπο ζωής τους και στη δουλειά τους.

Οι εφαρμογές διατίθενται από τα ηλεκτρονικά καταστήματα των λειτουργικών συστημάτων αλλά και από τρίτους. Το μεγαλύτερο ποσοστό των εφαρμογών που διατίθενται είναι δωρεάν για λήψη. Στο Google Play Store ένα ποσοστό 94% των συνολικών εφαρμογών είναι προς δωρεάν λήψη σύμφωνα με έρευνα<sup>1</sup> στο 4ο τρίμηνο του 2017 και υπάρχουν εφαρμογές όπου ο χρήστης θα πρέπει να καταβάλει το αντίστοιχο αντίτιπο σε ποσοστό 6%. Θα πρέπει να ληφθεί υπόψιν ότι για ολοένα και περισσότερες εφαρμογές πλέον, οι προγραμματιστές διαθέτουν δωρεάν την εφαρμογή αλλά για να μπορέσει ο χρήστης να έχει πρόσβαση σε όλες τις λειτουργίες της εφαρμογής θα πρέπει να προβεί σε αγορά αυτών.

<sup>1</sup> <https://www.statista.com/statistics/266211/distribution-of-free-and-paid-android-apps/>



Σύμφωνα με τις επίσημες ανακοινώσεις των εταιριών, η Apple μέχρι και το Μάιο του 2017 είχε συνολικά περισσότερες από 2.2 εκατομμύρια εφαρμογές στο κατάστημά της, ενώ η Google το Ιανουάριο του 2018 ανακοίνωσε ότι το Δεκέμβρη του 2017 οι εφαρμογές στο Google Play Store υπερβαίνουν το 3.5 εκατομμύρια. Ο παρακάτω πίνακας είναι μέρος έρευνας μέχρι το Μάιο του 2017 όπου παρατηρούμε των αριθμών των εφαρμογών στα εκάστοτε ηλεκτρονικά καταστήματα.



*Διάγραμμα 1. Συνολικός αριθμών εφαρμογών ανά κατάστημα μέχρι το Μάιο του 2017(<https://www.statista.com> )*

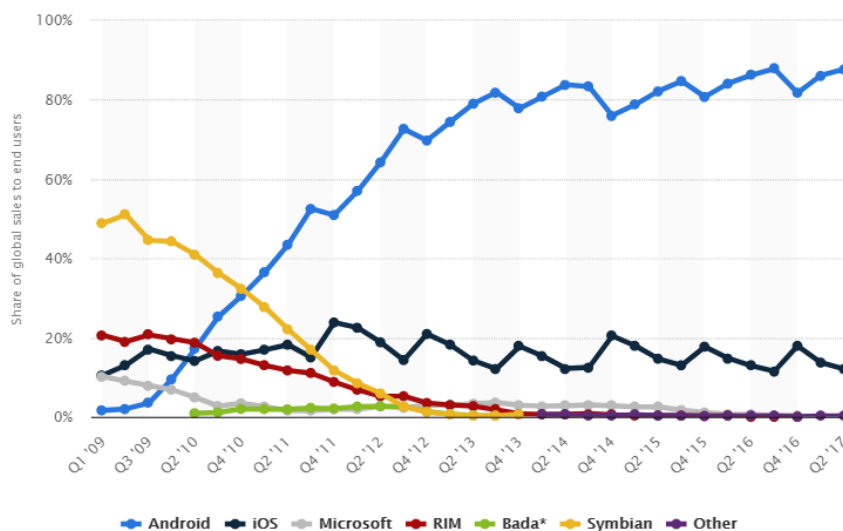
Οι χρήστες έξυπνων τηλεφώνων αυξήθηκαν σημαντικά διότι η χρήση εκείνων είναι γρήγορη και εύκολη. Παράλληλα, μέσω αυτών έχουν την δυνατότητα να περιηγηθούν εύκολα στο διαδίκτυο, όταν είναι μακριά από ηλεκτρονικό υπολογιστή. Τα έξυπνα κινητά, ακόμη, διαθέτουν άμεση πρόσβαση στο ηλεκτρονικό ταχυδρομείο και στα κοινωνικά δίκτυα καθώς και μεγάλη ποικιλία στις εφαρμογές. Μπορεί, τέλος να προσαρμοστεί ανάλογα με τις ανάγκες των χρηστών.

## 2.2 Υπολογιστής ταμπλέτα(Tablet)

Ο υπολογιστής ταμπλέτα είναι ένα έξυπνο τηλέφωνο με μεγαλύτερη οθόνη αλλά μικρότερη από τους υπολογιστή. Οι διαστάσεις τους συνήθως κυμαίνονται μεταξύ των 7 και 12 ιντσών. Οι δυνατότητες είναι ίδιες με αυτές των έξυπνων κινητών. Κύρια διαφορά είναι ότι υπάρχουν ταμπλέτες που δεν διαθέτουν τεχνολογία 3G/4G. Στα πλεονεκτήματά τους είναι ότι λόγω των μεγαλύτερων διαστάσεων διαθέτουν μεγαλύτερη μπαταρία και έχουν καλύτερη αυτονομία, διαθέτουν περισσότερες θύρες επέκτασης και είναι πιο βολικά σε παράλληλες εργασίες λόγω της μεγαλύτερης οθόνης. Τα λειτουργικά συστήματα στις ταμπλέτες επικεντρώνονται κυρίως σε Android και iOS, αλλά τα τελευταία χρόνια η Microsoft έχει λανσάρει τη σειρά Surface κυρίως για επαγγελματίες. Τα Surface είναι οι πρώτες συσκευές ταμπλέτες που χρησιμοποιούν πλήρες λειτουργικό σύστημα Windows 10 χωρίς ελλείψεις. Επίσης, με τη προσθήκη πληκτρολογίου γίνεται ένας «φορητός υπολογιστής» μικρότερων διαστάσεων.

### 2.3 Λειτουργικά Συστήματα για Κινητά

Κατά τη διαδικασία αγοράς ενός έξυπνου τηλεφώνου, σημαντικό ρόλο για την επιλογή του είναι το λειτουργικό σύστημα. Την παρούσα χρονική περίοδο οι χρήστες έχουν περιορισμένες επιλογές λειτουργικών συστημάτων. Τα προηγούμενα χρόνια, κατά την άνθιση των έξυπνων τηλεφώνων ολοένα και περισσότεροι κατασκευαστές διέθεταν το δικό τους λειτουργικό σύστημα, όμως δεν κατάφεραν να επικρατήσουν λόγω των λειτουργικών συστημάτων iOS, Android και Windows Phone. Παρακάτω θα γίνει μια μικρή αναφορά στα λειτουργικά συστήματα που υπάρχουν στην αγορά αυτή την περίοδο.



Διάγραμμα 2 .Παγκόσμιο μερίδιο αγοράς Smartphones μέχρι Q2 2017(<https://www.statista.com>)

- Android:**  
 Android είναι το λειτουργικό σύστημα της Google και το οποίο θα μας απασχολήσει στη παρούσα μεταπτυχιακή εργασία. Όπως παρατηρούμε στο παραπάνω διάγραμμα, είναι η πιο γρήγορη αναπτυσσόμενη πλατφόρμα και χρησιμοποιείται από πολλούς κατασκευαστές. ,την αγορά διατίθενται πολλές διαφορετικές συσκευές σε όλες τις κατηγορίες τιμών, ανάλογα με τις απαιτήσεις του χρήστη. Εκτενέστερη αναφορά θα γίνει σε επόμενο κεφάλαιο.
- iOS:**  
 Το 2007 η Apple με την παρουσίαση του πρώτου iPhone σηματοδότησε τη νέα γενιά έξυπνων τηλεφώνων. Η μεγαλύτερη διαφορά που έδωσε και την ώθηση στην εταιρεία ήταν η κατάργηση του φυσικού πληκτρολογίου, η ενσωμάτωση οθόνης αφής και πολλών λειτουργιών που έκανε ευχάριστη και φιλική την ενασχόληση με τα κινητά τηλέφωνα στο τελικό χρήστη.  
 Οι συσκευές της Apple τρέχουν στο δικό της λειτουργικό σύστημα, το iOS. Σήμερα, το iOS βρίσκεται στην ενδέκατη έκδοσή του, και έχει τροποποιηθεί κατάλληλα για να τρέχει και στην ταμπλέτα της Apple το iPad. Το λειτουργικό βασίζεται στο macOS, ο

προγραμματισμός γίνεται σε Objective-C και Swift. Το λειτουργικό αυτό σύστημα είναι κλειστού κώδικα και οι συσκευές που διατίθενται στην αγορά, κατασκευάζονται αποκλειστικά από την ίδια την Apple.



Εικόνα 7. Το iOS και οι συσκευές της Apple. (<https://www.apple.com>)

- **Windows Phone/Mobile:**

Στη προσπάθεια της Microsoft να ανταγωνιστεί την Google και την Apple, το 2010 παρουσίασε το λειτουργικό σύστημα Windows Phone. Πρόκειται για την εξέλιξη των Windows Mobile που μέχρι τότε χρησιμοποιούταν από τους υπολογιστές παλάμης. Η γλώσσα προγραμματισμού που υποστηρίζουν είναι οι C#, Visual Basic, και C++. Η Microsoft μέσω των Windows Phone, προσπάθησε να φέρει όσο πιο κοντά γίνεται το γραφικό περιβάλλον και τις λειτουργίες των σταθερών υπολογιστών στα έξυπνα κινητά της, ώστε οι χρήστες της να έχουν μεγαλύτερη οικειότητα με το λειτουργικό της σύστημα. Το 2015 η Microsoft ανακοίνωσε τη τελευταία έκδοση του λειτουργικού και τα μετονόμασε στα σημερινά Windows Mobile 10. Στη διάρκεια αυτών των χρόνων δεν είχε μεγάλη ανταπόκριση από τους χρήστες, λόγω του μικρού αριθμού εφαρμογών και των μεγάλων ελλείψεων



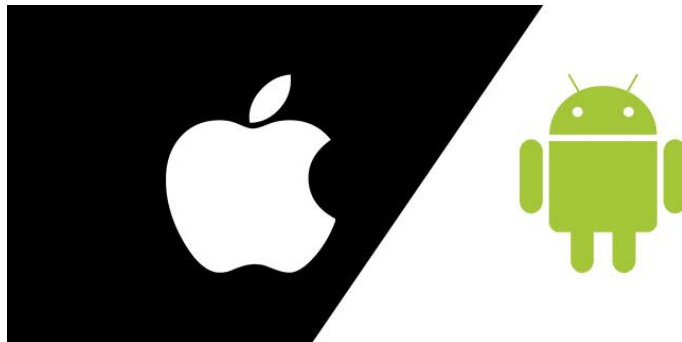
Εικόνα 8. Windows Phone 7 μέχρι τα Windows Mobile 10 (<https://www.microsoft.com>)

σε σχέση με τον ανταγωνισμό, αυτό δημιουργήθηκε κυρίως λόγω της έλλειψης κινήτρου και των κατάλληλων συνθηκών στους προγραμματιστές ώστε να επενδύσουν στο λειτουργικό της Microsoft. Τέλος, τον Οκτώβριο του 2017 ανακοίνωσε επίσημα ότι θα

σταματήσει την περαιτέρω εξέλιξη των Windows Mobile 10 και θα συνεχίζουν να παρέχουν απλά ενημερώσεις ασφαλείας στους χρήστες.

## 2.4 Android και iOS

Η Google με τη συνεχή εξέλιξη του Android τα τελευταία χρόνια κατάφερε να φτάσει αλλά και να ξεπεράσει σε πωλήσεις την Apple. Η ραγδαία εξάπλωση του Android προέκυψε επειδή η Google παρέχει στους κατασκευαστές το λειτουργικό της σύστημα δωρεάν. Αυτό είχε ως αποτέλεσμα στην αγορά να βγαίνουν ολοένα και περισσότερες συσκευές, με τις τιμές να κυμαίνονται από τα 50€ έως τα 1000€, με τον τελικό καταναλωτή να έχει πληθώρα επιλογών για την αγορά νέου κινητού. Επίσης χάρις την τιμολογιακή πολιτική της Apple αυτό έδωσε μεγαλύτερη ώθηση στην αγορά των κινητών Android, διότι με λιγότερα χρήματα είχαν εφάμιλλες επιδόσεις και χαρακτηριστικά. Στο παρόν υποκεφάλαιο θα αναφερθούμε στα θετικά και αρνητικά της κάθε πλατφόρμας.



Εικόνα 9. Λειτουργικά Συστήματα Android και iOS

Από την πλευρά των προγραμματιστών, η ανάπτυξη εφαρμογών σε περιβάλλον Android είναι ευκολότερη, διότι ο υποψήφιος προγραμματιστής Android θα πρέπει να διαθέτει έναν προσωπικό υπολογιστή με οποιοδήποτε λειτουργικό σύστημα, συσκευές Android για δοκιμές που είναι αισθητά οικονομικότερες από τις iOS συσκευές και να διαθέτει ένα λογαριασμό Google στο Play Store, για τον οποίο απαιτείται η εφάπαξ καταβολή ενός τιμήματος της τάξης των 25\$ για να μπορέσει να προσθέσει μια εφαρμογή στο ηλεκτρονικό κατάστημα της Google. Τέλος, η ανάπτυξη γίνεται σε Java, που παραμένει η δημοφιλέστερη γλώσσα προγραμματισμού και διαθέτει μεγάλη κοινότητα υποστήριξης.

Από τη πλευρά της Apple, η ανάπτυξη γίνεται σε Objective C και Swift, το πρόγραμμα ανάπτυξης της Apple το XCode τρέχει αποκλειστικά σε λειτουργικό σύστημα MacOS, για το οποίο απαιτείται αγορά MAC υπολογιστή που κοστίζει τουλάχιστον 1000€, συσκευές iOS για τις δοκιμές και ετήσια συνδρομή 100\$ για να μπορεί να ανεβάζει ο προγραμματιστής τις εφαρμογές του στο App Store.

Αισθητή διαφορά μεταξύ των δύο λειτουργικών συστημάτων αφορά το αγοραστικό κοινό. Οι χρήστες iOS είναι περισσότερο διατεθειμένοι να πληρώσουν για να αγοράσουν μια εφαρμογή από ότι οι χρήστες Android. Αυτό οφείλεται στο γεγονός πως το iPhone καλώς ή κακώς θεωρείται στις μέρες μας ως status symbol, καθώς πολλοί από τους κατόχους του είναι σταθεροί οπαδοί του. Θεωρείται στις μέρες μας ότι κάποιος γίνεται περισσότερο αποδεκτός έχοντας στη κατοχή του κάποιος συσκευή iPhone σε σχέση με Android συσκευή. Τέτοιοι χρήστες δεν είναι επιφυλακτικοί στο να πληρώσουν για να κατεβάσουν μια εφαρμογή και αυτή η συμπεριφορά λειτουργεί προς όφελος των προγραμματιστών.

Αντίθετα, το Android βασίζεται όπως θα δούμε και στη συνέχεια στο Linux και έχει ένα καθαρά open source χαρακτήρα. Πολλοί από τους χρήστες του Android ασπάζονται αυτή τη

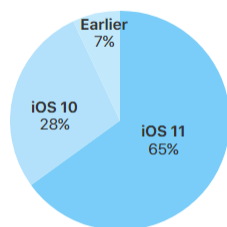
φιλοσοφία και δεν είναι διατεθειμένοι να πληρώσουν για μία εφαρμογή. Θα αγοράζαν μια εφαρμογή μόνο αν είναι εγγυημένα χρήσιμη και προσεγμένη.

### 2.4.1 Αρνητικά του Android

Στο Android υπάρχει πληθώρα συσκευών με διαφορετικά χαρακτηριστικά όπως οθόνες διαφορετικών διαστάσεων και αναλύσεων, και διαφορετικά σετ αισθητήρων, τα οποία οι προγραμματιστές καλούνται να λαμβάνουν υπ' όψιν κατά την ανάπτυξη των εφαρμογών τους ώστε αυτές να τρέχουν απροβλημάτιστα και με ομοιόμορφο τρόπο σε όλες τις συσκευές. Αυτό προϋποθέτει επιπλέον δουλειά και πρόνοια από την πλευρά του προγραμματιστή.

Το σημαντικότερο αρνητικό του Android σε σχέση με το iOS είναι το πρόβλημα του κατακερματισμού (Fragmentation). Ο κατακερματισμός στα mobile λειτουργικά συστήματα αφορά τις περιπτώσεις όπου οι χρήστες τρέχουν στις συσκευές τους προηγούμενες εκδόσεις του λειτουργικού συστήματος, ενώ άλλοι τρέχουν νεότερες εκδόσεις. Το χάσμα αυτό δημιουργείται διότι οι κατασκευαστές των συσκευών παύουν να ενημερώνουν τις συσκευές με νεότερες εκδόσεις λόγω περιορισμών στα χαρακτηριστικά των συσκευών ή λόγω της παλαιότητάς τους. Επιπλέον, με την εξέλιξη του λειτουργικού, ορισμένες δυνατότητες δεν μπορεί να εφαρμοστούν σε παλιότερες εκδόσεις με αποτέλεσμα οι προγραμματιστές να μην υποστηρίζουν συσκευές με παλιότερες εκδόσεις. Ενώ στο iOS, η Apple υποστηρίζει για αρκετά χρόνια τις συσκευές της και προβαίνει σε αναβαθμίσεις. Σε αυτό παίζει σημαντικό παράγοντα ο μικρός αριθμός συσκευών που διαθέτει η Apple στην αγορά, συνήθως 2-3 συσκευές το χρόνο.

65% of devices are using iOS 11.



As measured by the App Store on January 18, 2018.



Διάγραμμα 3. Εκδόσεις Λογισμικού iOS και Android μέχρι της 20 Ιανουαρίου 2018(<https://www.apple.com>)

Όπως βλέπουμε στο παραπάνω γράφημα το iOS 11 που κυκλοφόρησε το Σεπτέμβριο του 2017 είναι εγκατεστημένο ήδη στο 65 % των συσκευών που κυκλοφορούν. Από την άλλη πλευρά, η όγδοη και τελευταία έκδοση του Android κυκλοφόρησε τον Αύγουστο το 2017 και μόλις το 0,7 % των συσκευών «τρέχουν» τη νεότερη έκδοση του λειτουργικού συστήματος.

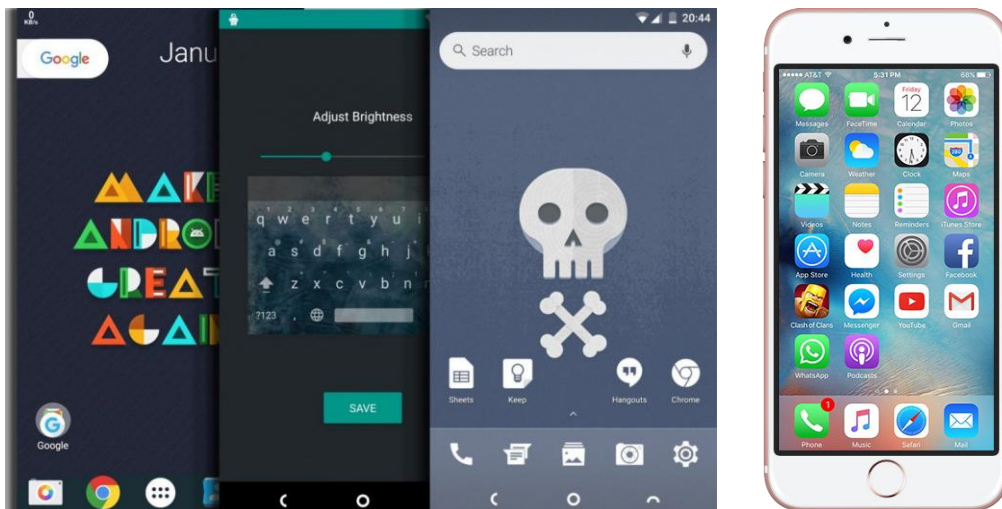
Η Google μέσω του Android 8.0 έχει προσθέσει το “Project Treble” μέσω του οποίου βοηθάει του κατασκευαστές και τους παρόχους να διαθέτουν τις νεότερες εκδόσεις πιο

γρήγορα. Αυτό όμως προϋποθέτει η συσκευή να διαθέτει ήδη Android 8.0 ή να αναβαθμιστεί μελλοντικά.

#### 2.4.2 Αρνητικά του iOS

Το iOS από τη πλευρά του λόγω της κλειστού λογισμικού έχει αρκετούς περιορισμούς που δεν υπάρχουν στο Android. Αρχικά ο κυριότερος περιορισμός βρίσκεται στη συνδεσιμότητα των συσκευών. Για παράδειγμα, ένας χρήστης Android μπορεί να αποστείλει ένα αρχείο μέσω τεχνολογίας Bluetooth σε ένα άλλο χρήστη, στο iOS είναι κλειδωμένη αυτή η δυνατότητα και αναγκαστικά η αποστολή θα γίνει μέσω cloud εφαρμογών. Επιπλέον, το Android υποστηρίζει τη σύνδεση σκληρών δίσκων για τη μεταφορά δεδομένων χωρίς να είναι απαραίτητη η χρήση υπολογιστή. Ο χρήστης για τη μεταφορά αρχείων, φωτογραφιών και οτιδήποτε άλλου αρχείου απαιτείται να χρησιμοποιεί το Apple iTunes για τη μεταφορά.

Ταυτόχρονα, απουσιάζει η παραμετροποίηση του γραφικού περιβάλλοντος από τις iOS συσκευές. Ο χρήστης μπορεί απλά να αλλάξει τη σειρά των εφαρμογών και ταπετσαρία, ενώ στο Android έχει τη δυνατότητα να τροποποιήσει τη συσκευή του όπως ακριβώς επιθυμεί είτε στο υπάρχον γραφικό περιβάλλον είτε μέσω θεμάτων που θα βρει στο Play Store.



Εικόνα 10. Γραφικό Περιβάλλον Android και iOS (<https://www.wikipedia.org/>)

### 3. Εφαρμογές Έξυπνων τηλεφώνων

Τα έξυπνα τηλέφωνα έχουν εισέλθει δυναμικά στην αγορά παρουσιάζοντας μια σταθερά ανοδική πορεία πωλήσεων με εξαιρετικές προοπτικές περαιτέρω ανάπτυξης. Αναπόσπαστο κομμάτι των τηλεφώνων αποτελούν οι εφαρμογές κινητών (mobile applications) οι οποίες μπορούν να καλύψουν μία πληθώρα αναγκών.

Πολλές εταιρίες και πολλοί προγραμματιστές έχουν ήδη αναπτύξει τις δικές τους εφαρμογές για κινητές συσκευές, προσφέροντας υπηρεσίες προστιθέμενης αξίας στους καταναλωτές, καθώς μια κινητή εφαρμογή μπορεί να εξυπηρετήσει προσωπικές, επαγγελματικές και ενδο-επιχειρηματικές ανάγκες.

#### 3.1 Ο κατακερματισμός και τα προβλήματα των προγραμματιστών

Στο προηγούμενο κεφάλαιο αναφερθήκαμε στις διαφορές μεταξύ του Android και του iOS, εντοπίσαμε κάποια σοβαρά προβλήματα που αντιμετωπίζουν οι προγραμματιστές και οι χρήστες. Οι κατηγορίες των προβλημάτων είναι η αμοιβή, το κόστος υλοποίησης και το πιο βασικό είναι η δυσκολία των προγραμματιστών να προσαρμόσουν και να υλοποιήσουν μια εφαρμογή σε πολλές διαφορετικές συσκευές και πλατφόρμες.

Είτε πρόκειται για φορητούς υπολογιστές, ταμπλέτες ή έξυπνα τηλέφωνα, ο αριθμός των νέων φορητών συσκευών που εισέρχονται στην αγορά αυξάνεται με ταχείς ρυθμούς. Οι τομείς των αναγκών της καθημερινότητας των καταναλωτών που αυξάνονται συνεχώς δυσκολεύει περαιτέρω το έργο των προγραμματιστών που θα πρέπει να ξοδέψουν ακόμα περισσότερο χρόνο τελειοποιώντας τις εφαρμογές τους για όσο το δυνατόν μεγαλύτερο μέρος των συσκευών.

Έτσι, κατά την ανάπτυξη μίας εφαρμογής για κινητές συσκευές, προκύπτουν εμπόδια και δυσκολίες λόγω του κατακερματισμού στο υλικό (hardware) και στο λογισμικό (software) των κινητών συσκευών. Οι κινητές συσκευές υποστηρίζονται από λειτουργικά συστήματα τα οποία διακρίνονται από συγκεκριμένα γνωρίσματα. Για να αναπτυχθεί μια εφαρμογή για ένα λειτουργικό σύστημα χρησιμοποιείται συγκεκριμένη γλώσσα προγραμματισμού, στην οποία επένδυσε ο κατασκευαστής, και συγκεκριμένα εργαλεία ανάπτυξης που παρέχονται από τον κατασκευαστή. Το πρόβλημα του κατακερματισμού όμως δεν σταματάει εκεί αφού ακόμη και οι διαφορετικές εκδόσεις του ίδιου λειτουργικού συστήματος ορισμένες φορές παρουσιάζουν μεγάλες διαφορές μεταξύ τους. Ακόμα και συσκευές που μπορεί να βασίζονται στο ίδιο λειτουργικό σύστημα είναι πιθανό να παρουσιάσουν διαφορές μεταξύ τους λόγω των διαφορετικών προδιαγραφών του υλικού (hardware) της συσκευής. Για παράδειγμα το πώς υποστηρίζεται η ανίχνευση αφής, διαφέρει στην εκάστοτε συσκευή λόγω του διαφορετικού αισθητήρα αφής.

Τέλος, τα τελευταία χρόνια έχει δημιουργηθεί η ανάγκη για δημιουργία πλατφόρμας όπου ο προγραμματιστής θα γράφει σε μια συγκεκριμένη γλώσσα προγραμματισμού και θα έχει τη δυνατότητα να εξαγει κώδικα όπου θα τρέχει εφάμιλλα σε πολλά διαφορετικά λειτουργικά συστήματα. Ο παρακάτω πίνακας περιέχει τις διαφορές που παρουσιάζονται στα επικρατέστερα λειτουργικά συστήματα σε χαρακτηριστικά που αφορά την ανάπτυξη μιας εφαρμογής, όπως στις γλώσσες προγραμματισμού, τον τύπο των εκτελέσιμων αρχείων, τα περιβάλλοντα ανάπτυξης, και τα διαδικτυακά καταστήματα για την αγορά εφαρμογών.

	Android	iOS	Windows Phone
Έκδοση λειτουργικού συστήματος	Android 2.3- 8.1	iOS 8 -11	Windows Mobile 10
Περιβάλλον ανάπτυξης	Android Studio	XCode	Visual Studio
Γλώσσες προγραμματισμού	Java & Kotlin	Swift & Objective C	C# & Visual Basic
Ηλεκτρονικά Καταστήματα	Google Play Store & 3rd Party Stores	Apple App Store	Microsoft Marketplace

Πίνακας 1. Χαρακτηριστικά μιας εφαρμογής στο κάθε λειτουργικό σύστημα (<https://www.wikipedia.org/>)

### 3.2 Τύποι εφαρμογών κινητών συσκευών

Υπάρχουν τρία είδη εφαρμογών για κινητές συσκευές, οι εγγενείς εφαρμογές (Native applications), οι διαδικτυακές εφαρμογές (Web applications) και οι υβριδικές εφαρμογές (Hybrid applications). Η επιλογή της κατάλληλης μεθόδου εξαρτάται από το επιχειρηματικούς στόχους που έχει θεσπίσει ο πελάτης και θεωρείται σημαντική επιλογή διότι θα κρίνει σε ένα μέρος την επιτυχία της εφαρμογής. Κάθε ένας από τους τρόπους έχει τα δικά του μειονεκτήματα και πλεονεκτήματα στα οποία θα αναφερθούμε.–

#### 3.2.1 Εγγενείς Εφαρμογές ( Native App):

Οι εγγενείς εφαρμογές είναι ο πιο συνηθισμένος τύπος εφαρμογής. Αναπτύσσονται για συγκεκριμένες πλατφόρμες και είναι γραμμένες σε γλώσσες προγραμματισμού που υποστηρίζει η πλατφόρμα. Τόσο η Apple όσο και η Google παρέχουν στους προγραμματιστές τα δικά τους εργαλεία ανάπτυξης, τα στοιχεία διεπαφής και το SDK. Οι περισσότερες εταιρείες επενδύουν σε ανάπτυξη εγγενών εφαρμογών για κινητά λόγω των πλεονεκτημάτων που προσφέρονται σε σύγκριση με άλλους τύπους εφαρμογών.

#### Πλεονεκτήματα:

- Οι εγγενείς εφαρμογές είναι πολύ γρήγορες και έχουν πολύ καλή απόκριση επειδή είναι κατασκευασμένες για τη συγκεκριμένη πλατφόρμα.
- Έχουν την καλύτερη απόδοση.
- Διανέμονται σε ηλεκτρονικά καταστήματα εφαρμογών.
- Είναι πιο διαδραστικές και τρέχουν πολύ πιο ομαλά .
- Ο εγγενής προγραμματισμός επιτρέπει στους προγραμματιστές να έχουν πρόσβαση στο πλήρες σύνολο των λειτουργιών της πλατφόρμας. Για παράδειγμα πρόσβαση στους αισθητήρες της συσκευής, στην κάμερα και σε πολλά άλλα.
- Δημιουργία περίπλοκων διεπαφών με τα διαθέσιμα γραφικά συστατικά που εμπεριέχονται στο λειτουργικό σύστημα.
- Αξιοποίηση διεπαφών προγραμματισμού εφαρμογών (API) και προσθήκη βιβλιοθηκών από τη κοινότητα της εκάστοτε πλατφόρμας.



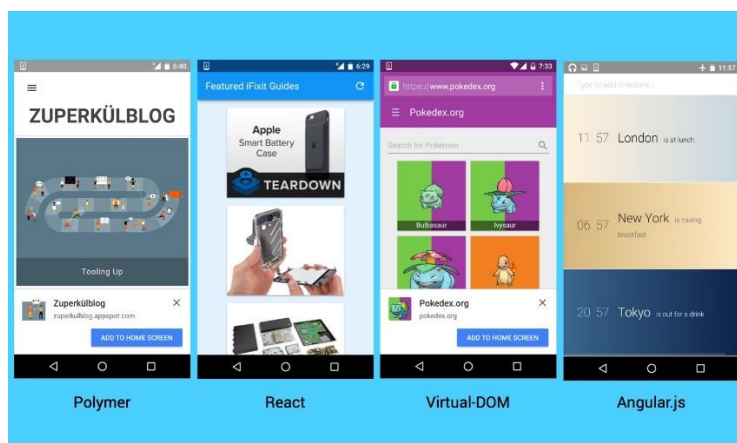
- Δεν απαιτείται συνεχής σύνδεση στο διαδίκτυο, αν και αυτό εξαρτάται από τη λειτουργικότητα.
- Συνολικά καλύτερη εμπειρία χρήσης για τον χρήστη, η ροή είναι πιο φυσική καθώς έχουν συγκεκριμένα πρότυπα διεπαφών για κάθε πλατφόρμα

#### Μειονεκτήματα:

- Δυσκολότερες γλώσσες προγραμματισμού για εκμάθηση που σημαίνει ότι χρειάζονται πιο έμπειροι προγραμματιστές.
- Πιο ακριβείς υλοποιήσεις.
- Ο πηγαίος κώδικας δεν μπορεί να χρησιμοποιηθεί σε άλλη πλατφόρμα.
- Για την ανάπτυξη μιας εφαρμογής σε διαφορετικά λειτουργικά συστήματα χρειάζονται ξεχωριστές ομάδες υλοποίησης το οποίο αυξάνει το κόστος υλοποίησης.
- Σε περίπτωση όπου γίνει ενημέρωση στην εφαρμογή, πρέπει να ανέβει στο κατάστημα και αφού εγκριθεί, ο χρήστης μπορεί να κάνει λήψη, ενώ σε εφαρμογές ιστού οι ενημερώσεις γίνονται άμεσα.

#### 3.2.2 Εφαρμογές Ιστού ( Web App):

Οι εφαρμογές ιστού για έξυπνα τηλέφωνα είναι διαδικτυακές εφαρμογές ειδικά σχεδιασμένες για χρήση σε κινητά τηλέφωνα ή ταμπλέτες και ανοίγουν μέσω του περιηγητή της συσκευής με την επίσκεψή στην αντίστοιχη ηλεκτρονική διεύθυνση (URL) της εφαρμογής. Ωστόσο, οι εφαρμογές ιστού δεν χρειάζονται λήψη από καταστήματα εφαρμογών, όπως οι εγγενείς εφαρμογές. Οι εφαρμογές ιστού φορτώνονται σε προγράμματα περιήγησης όπως ο Chrome, ο Safari ή ο Firefox. Επίσης, μια εφαρμογή ιστού δεν αποθηκεύεται στην συσκευή του χρήστη και δεν δεσμεύει χώρο αποθήκευσης. Οι εφαρμογές ιστού δίνουν λύση στον κατακερματισμό, διότι δημιουργούνται για να προσαρμόζονται σε όλα τα είδη οθονών χάρις σε κάποια πλαίσια (framework) όπως είναι το Bootstrap και τρέχουν σε διαφορετικά λειτουργικά συστήματα, ανεξάρτητα από τα τεχνικά χαρακτηριστικά της εκάστοτε συσκευής.



Εικόνα 11. Framework για δημιουργία εφαρμογών ιστού( [www.wikipedia.com](http://www.wikipedia.com))

Με τη συνεχόμενη ανάπτυξη νέων τεχνολογιών στον ιστό, δημιουργούνται κενά μεταξύ των εγγενών εφαρμογών, του ιστού και των υβριδικών. Σε πολλές περιπτώσεις οι χρήστες

μπορούν να κατεβάσουν μια εφαρμογή από το ηλεκτρονικό κατάστημα της πλατφόρμας που χρησιμοποιούν και τελικά η εφαρμογή λειτουργεί μέσω ενός WebView, το οποίο περιέχει το σύνδεσμο της εφαρμογής του ιστού. Οι εφαρμογές ιστού δημιουργούνται μέσω JavaScript\CSS και HTML5. Το WebView είναι ένα γραφικό στοιχείο μέσω του οποίου εμφανίζουμε ιστοσελίδες.

Επίσης, έχουν δημιουργηθεί εφαρμογές ιστού και εφαρμογές κινητών που ονομάζονται Progressive Web Apps. Αυτές είναι μια υβριδική λύση διότι οι εφαρμογές ιστού δεν έχουν πρόσβαση σε λειτουργίες του λειτουργικού όπως αποστολή ειδοποιήσεων ή λειτουργία χωρίς διαδίκτυο. Μέσω των Progressive Web εφαρμογών μπορούν να αξιοποιούν λειτουργίες παρόμοιες με τις εγγενείς εφαρμογές όπως :

- Αποστολή μηνυμάτων.
- Χρήση gestures και του επιταχυνσιόμετρου της συσκευής.
- Αξιοποίηση κάποιων υλικών της συσκευής όπως της δόνησης.

Παρά τις συνεχείς βελτιώσεις, ένα σημαντικό μειονέκτημα είναι ότι μπορούν να τρέχουν μόνο σε περιηγητή Google Chrome. Επιπλέον, οι εφαρμογές ιστού υστερούν στο κομμάτι της ασφάλειας, διότι οι εγγενείς εφαρμογές πριν διατεθούν στο κοινό, εξετάζεται ο κώδικας από τα ηλεκτρονικά καταστήματα για τυχόν κακόβουλο λογισμικό και για λειτουργίες που μπορεί να εκθέσουν τα προσωπικά δεδομένα των χρηστών.

Οι εφαρμογές ιστών έχουν ορισμένα πλεονεκτήματα κυρίως της εύκολης κατασκευής, μικρότερο κόστος κατασκευής και συντήρησης σε σχέση με τις εγγενείς εφαρμογές. Επίσης δημιουργείται μια εφαρμογή για πολλαπλά λειτουργικά συστήματα, αρκεί η συσκευή να διαθέτει πρόγραμμα περιήγησης.

Από την άλλη, οι εφαρμογές ιστού διαθέτουν μια σειρά μειονεκτημάτων. Απαιτείται η συσκευή να διαθέτει πρόγραμμα περιήγησης εγκατεστημένο για να εκτελεστεί. Οι χρήστες θα πρέπει να πληκτρολογήσουν τη διεύθυνση( URL) της εφαρμογής, πράγμα που ισοδυναμεί με κακή εμπειρία χρήσης. Επιπλέον, οι εφαρμογές ιστού είναι πολύ πιο αργές, λιγότερο διαδραστικές και διαισθητικές συγκριτικά με τις εγγενείς εφαρμογές. Παράλληλα δεν υπάρχει εικονίδιο της εφαρμογής στην επιφάνεια εργασίας του κινητού όπως θα γινόταν αν είχε γίνει λήψη από τα καταστήματα εφαρμογών. Τέλος παρατηρείται η μη αξιοποίηση βοηθητικού υλικού (hardware) των συσκευών και μικρότερη ασφάλεια.

### 3.2.3 Υβριδικές Εφαρμογές ( Native App):

Οι υβριδικές εφαρμογές λειτουργούν σε διάφορες πλατφόρμες και συμπεριφέρονται σαν τις εγγενείς εφαρμογές. Πρόκειται για ένα συνδυασμό των εγγενών και των εφαρμογών ιστού. Οι χρήστες μπορούν να τις εγκαταστήσουν στη συσκευή τους όπως σε μια εγγενή εφαρμογή, αλλά στη πραγματικότητα πρόκειται για μια εφαρμογή ιστού.

Οι συγκεκριμένου τύπου εφαρμογών βασίζονται σε JavaScript, HTML και CSS και τρέχουν σε ένα WebView. Μια υβριδική εφαρμογή αποτελείται από δύο μέρη. Το πρώτο κομμάτι είναι το “back-end” που δημιουργείται χρησιμοποιώντας γλώσσες όπως οι HTML, CSS και JavaScript. Το δεύτερο κομμάτι είναι ένα εγγενές κέλυφος το οποίο γίνεται λήψη και φορτώνει τον κώδικα χρησιμοποιώντας το WebView. Στις υβριδικές εφαρμογές θα

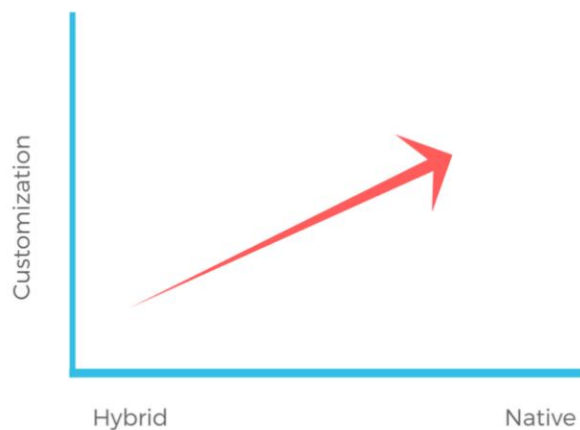
προκύψουν προβλήματα όπως και στις εγγενείς, όμως τα σφάλματα είναι δυσκολότερο να αντιμετωπιστούν.

#### Πλεονεκτήματα:

- Βασίζεται σε τεχνολογίες ιστού (HTML / CSS / JavaScript), που κάνει τη ανάπτυξη ευκολότερη.
- Φθηνότερη υλοποίηση σε σχέση με μια εγγενή εφαρμογή.
- Δημιουργείται μια εφαρμογή για όλες τις πλατφόρμες χρησιμοποιώντας τεχνολογίες όπως η Cordova.
- Δεν απαιτείται πρόγραμμα περιήγησης σε αντίθεση με μια εφαρμογή ιστού.
- Έχουν πρόσβαση στα εσωτερικά API της συσκευής, για παράδειγμα στο χώρο αποθήκευσης, στην κάμερα κ.λπ.
- Ταχύτερη ανάπτυξη από τις εγγενείς εφαρμογές, επειδή υπάρχει μια ενιαία βάση κώδικα.

#### Μειονεκτήματα:

- Πιο αργές από τις εγγενείς εφαρμογές.
- Πιο ακριβές από τις τυπικές εφαρμογές ιστού, επειδή η παραμετροποίηση κοστίζει περισσότερο σε σχέση με τις εγγενείς εφαρμογές.
- Λιγότερο διαδραστικές από τις εγγενείς εφαρμογές
- Μειωμένη δυνατότητα παραμετροποίησης σε σχέση με τις εγγενείς εφαρμογές.



Διάγραμμα 5. Το κόστος παραμετροποίησης ανάμεσα σε εγγενείς και υβριδικές εφαρμογές(<https://www.statista.com>)

### 3.3 Ανάπτυξη εφαρμογών σε διαφορετικά λειτουργικά συστήματα

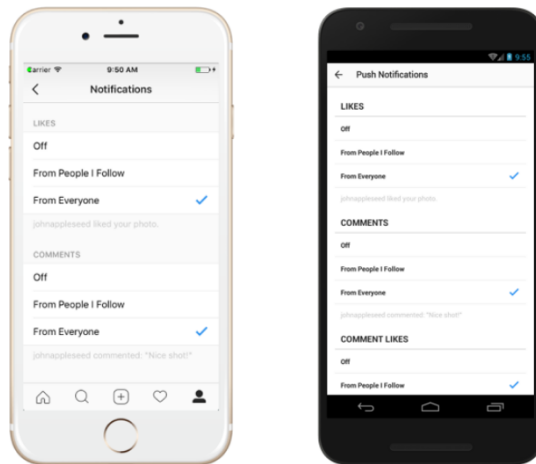
Προηγουμένως αναφερθήκαμε στα τρία είδη ανάπτυξης εφαρμογών. Στο παρόν υποκεφάλαιο θα αναφερθούμε στις εγγενείς εφαρμογές που μπορούν να εκτελεστούν σε διαφορετικά λειτουργικά συστήματα. Αυτό έρχεται να καλύψει ένα από τα σημαντικότερα μειονεκτήματα των εγγενών εφαρμογών που χρειάζονται ξεχωριστή ανάπτυξη εφαρμογών ανάλογα με τη πλατφόρμα.

Τα τελευταία χρόνια έχουν δημιουργηθεί αρκετές εναλλακτικές λύσεις ώστε οι

χρήστες να αποκοιμίσουν οφέλη από την παράλληλη ανάπτυξη χωρίς να θυσιάζεται ή εμπειρία τους ή η πρόσβαση στα εγγενή API. Δύο τέτοιες πλατφόρμες είναι η Xamarin και η React Native.

Η Xamarin δημιουργήθηκε από τη Microsoft και η ανάπτυξη γίνεται σε C#. Η Microsoft παρέχει τα κατάλληλα εργαλεία για την ανάπτυξη, αποσφαλμάτωση και τη διανομή των εφαρμογών σε παράλληλη λειτουργικά συστήματα. Ορισμένες εταιρείες έχουν χτίσει τις εφαρμογές με Xamarin, για παράδειγμα είναι το Slack, Pinterest, Insightly και πολλές άλλες.

Ταυτόχρονα η React Native αναπτύχθηκε από το Facebook. Πρόκειται για μια πλατφόρμα ανοιχτού κώδικα που επιτρέπει τη δημιουργία εγγενών εφαρμογών για iOS και Android με κοινό πηγαίο κώδικα. Η React συνδυάζεται με JavaScript για να επιτευχθεί το τελικό αποτέλεσμα. Σε περίπτωση όπου η React δεν έχει πρόσβαση σε λειτουργίες της συσκευής, μπορεί επίσης να γραφτεί παράλληλα και εγγενής κώδικας εάν χρειαστεί. Αυτή τη στιγμή υπάρχουν μεγάλοι οργανισμοί-εταιρίες που αξιοποιούν αυτή τη πλατφόρμα όπως το Facebook, το Walmart και η Airbnb.



Εικόνα 12. React Native στο Instagram (<https://facebook.github.io/react-native>)

### 3.4 Επιλογή κατάλληλης μεθόδου

Η απόφαση για την επιλογή της κατάλληλης μεθόδου ανάπτυξης βασίζεται στο επιχειρηματικό μοντέλο. Προτού αρχίσει η ανάπτυξη, θα πρέπει να ληφθούν υπόψιν οι ακόλουθοι παράγοντες:

- Το κατά πόσο γρήγορη πρέπει να είναι η εφαρμογή.
- Η ποιότητα εμπειρίας χρήσης (User Experience) που θα έχει η εφαρμογή.
- Η πολυπλοκότητα των χαρακτηριστικών που χρειάζεται.
- Ο προϋπολογισμός.

Οποιαδήποτε μέθοδος και να επιλεγεί, πρώτα από όλα θα πρέπει η εφαρμογή να είναι γρήγορη, άμεση και αξιόπιστη. Καθώς οι χρήστες με το πέρασμα των χρόνων γίνονται ολοένα και περισσότερο απαιτητικοί από τις mobile εφαρμογές. Τέλος, είναι σημαντικό να υποστηρίζουν μελλοντικές αλλαγές εφόσον χρειαστούν για να ανταπεξέλθουν στον ανταγωνισμό.

## 4. Λειτουργικό Σύστημα Android και Τεχνολογίες

Το Android είναι ένα λειτουργικό σύστημα ανοιχτού κώδικα βασισμένο σε μια τροποποιημένη έκδοση του πυρήνα του λειτουργικού Linux, το οποίο αρχικά μπορούσε να υποστηρίξει κινητές συσκευές με οθόνη αφής. Πλέον, έχει διευρυνθεί σε πολλούς τομείς και μπορούμε να το βρούμε σε έξυπνα ρολόγια/tablets/αυτοκίνητα.

Αρχικά το λειτουργικό αναπτύχθηκε από την εταιρεία Android Inc. Το 2007 η Apple παρουσιάζοντας το πρώτο iPhone έδωσε την ώθηση στην Google ώστε να ακολουθήσει τις εξελίξεις της εποχής και προχώρησε στην εξαγορά της Android Inc. Αρχικά αναπτύχθηκε από την Google και αργότερα συνεχίστηκε σε συνεργασία με την Open Handset Alliance η οποία είναι μια κοινοπραξία εταιριών λογισμικού, κατασκευής hardware και τηλεπικοινωνιών. Η πρώτη επίσημη παρουσίαση του λειτουργικού πραγματοποιήθηκε στις 5 Νοεμβρίου 2007, όπου ανακοινώθηκε η ίδρυση του οργανισμού Open Handset Alliance και η Google έδωσε το μεγαλύτερο κομμάτι του κώδικα στους προγραμματιστές.



Εικόνα 13. Open Handset Alliance(<https://developer.android.com/>)

Η πρώτη συσκευή που παρουσιάστηκε και στην οποία έγινε διαθέσιμο το λειτουργικό Android ήταν το HTC Dream στις 22 Οκτωβρίου του 2008. Το λειτουργικό είδε μεγάλη άνθηση ιδιαίτερως λόγω των πολλών κατασκευαστών παγκοσμίως και του ανοιχτού λογισμικού. Δίνοντας τη δυνατότητα σε μικρότερους κατασκευαστές να παράγουν χαμηλού κόστους συσκευές με λογισμικό ανοιχτού κώδικα, εύκολα προσαρμόσιμο στα μέτρα τους και το οποίο ενθάρρυνε αρκετούς προγραμματιστές να στραφούν σε αυτό το λειτουργικό.

Το λειτουργικό αξιοποιεί τη γλώσσα προγραμματισμού Java. Μέσω του IDE Android Studio και του λογισμικού ανάπτυξης ( Android SDK ) ο προγραμματιστής έχει τα απαραίτητα εφόδια για τη δημιουργία σύγχρονων εφαρμογών. Η εξέλιξη του λειτουργικού είναι ραγδαία. Σήμερα είναι στην όγδοη έκδοσή του με σημαντικές βελτιώσεις και προσθήκες λειτουργιών όλα αυτά τα χρόνια.

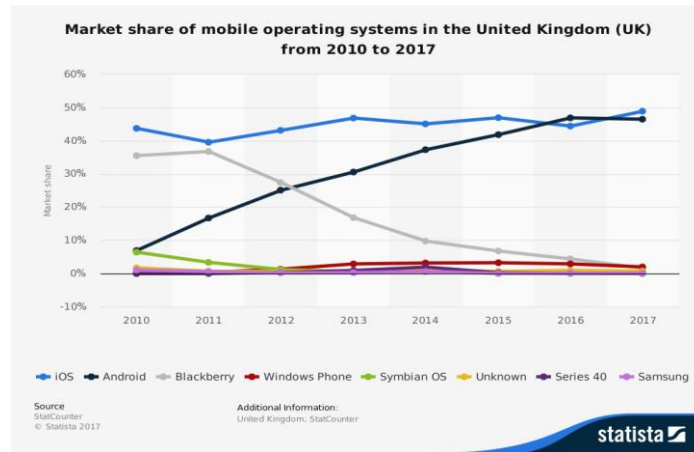
Χαρακτηριστική είναι η έρευνα<sup>2</sup> του Απριλίου-Μαΐου 2013 η οποία δείχνει την εξέλιξη και την εξάπλωση. Στην έρευνα συμμετείχαν προγραμματιστές εφαρμογών κινητών συσκευών και διαπιστώθηκε ότι το 71% των προγραμματιστών δημιουργούν εφαρμογές για το Android. Μια έρευνα<sup>3</sup> του 2015 έδειξε ότι το 44.6 % επαγγελματιών προγραμματιστών

<sup>2</sup> <https://www.statista.com>

<sup>3</sup> <https://insights.stackoverflow.com/survey/2015>

χρησιμοποιούν το Android ως κύρια πλατφόρμα ανάπτυξης λογισμικού, ποσοστό το οποίο είναι συγκρίσιμο με το iOS της Apple που είναι 37%.

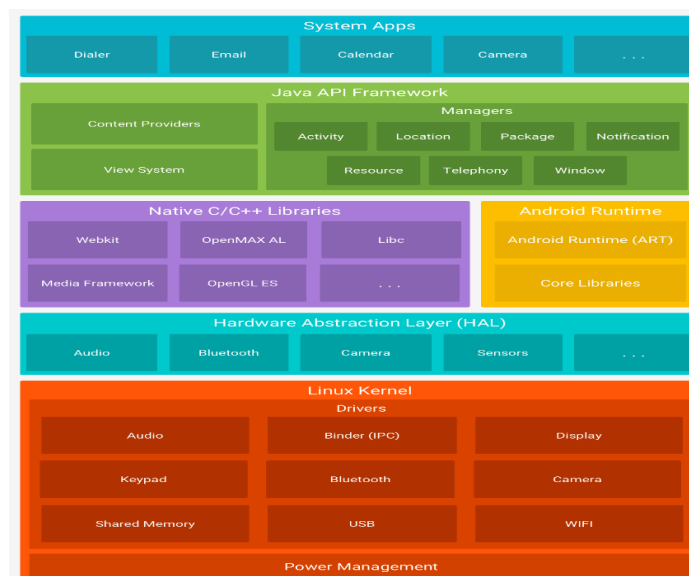
Κατά το Google<sup>4</sup> I / O 2014, η εταιρεία αποκάλυψε ότι υπήρχαν πάνω από ένα δισεκατομμύριο ενεργοί μηνιαίοι χρήστες του Android, από 538 εκατομμύρια τον Ιούνιο του 2013.



Διάγραμμα 6. Μερίδιο Αγοράς Λειτουργικών Συστημάτων από το 2010 έως το 2017. ([www.statista.com](http://www.statista.com))

#### 4.1 Αρχιτεκτονική Συστήματος Android

Το λειτουργικό σύστημα Android χωρίζεται σε ορισμένες συνιστώσες λογισμικού οι οποίες συνθέτουν ένα ολοκληρωμένο σύστημα. Καθένα από αυτά τα επίπεδα μπορεί να αποτελείται από άλλα υποστοιχεία και χρησιμοποιεί τις υπηρεσίες των επιπέδων τα οποία βρίσκονται από κάτω του. Στην εικόνα 14 απεικονίζεται η αρχιτεκτονική του android και φαίνονται ευδιάκριτα τα επίπεδα.



Εικόνα 14. Αρχιτεκτονική του Android. (<https://developer.android.com/>)

<sup>4</sup> <https://www.slashgear.com/one-billion-android-active-users-as-google-io-2014-begins-25335114/>

#### 4.1.1 Linux Kernel

Το Android βασίζεται στον πυρήνα του Linux και βρίσκεται στο χαμηλότερο επίπεδο της αρχιτεκτονικής του λειτουργικού συστήματος. Η Google το έχει τροποποιήσει κατάλληλα ώστε να διαχειρίζεται θέματα ενέργειας, μνήμης, διεργασιών, drivers. Γενικά ελέγχει ότι σχετίζεται μεταξύ software και hardware ώστε να αλληλοεπιδρούν μεταξύ τους.

#### 4.1.2 Hardware Abstraction Layer (HAL)

Στην τελευταία έκδοση του Android προστέθηκε ως ένα νέο επίπεδο μεταξύ του υλικού και του λογισμικού το HAL, μέσω αυτού οι κατασκευαστές υλικού μπορούν να προσθέσουν νέες λειτουργίες χωρίς να επηρεαστούν ή να τροποποιηθούν υψηλότερα επίπεδα του συστήματος. Ένα από τα σημαντικότερα προβλήματα του Android είναι η ενημέρωση των κινητών συσκευών στην τελευταία έκδοση του λογισμικού, διότι υπάρχουν χιλιάδες συσκευές με διαφορετικό λογισμικό και υλικό. Αυτή η διεπαφή θα δώσει τη δυνατότητα στους κατασκευαστές να αναβαθμίσουν ευκολότερα, πιο γρήγορα και χωρίς κόστος τις συσκευές τους στη τελευταία έκδοση του λογισμικού.

#### 4.1.3 Android Runtime

Στο επίπεδο εκτέλεσης περιλαμβάνονται οι βιβλιοθήκες του πυρήνα της Java και του Android ART. Το ART τρέχει σε εκδόσεις μεγαλύτερες του Android 5.0, χρησιμοποιείται για να εκτελέσει εφαρμογές και είναι βελτιστοποιημένο ώστε να λειτουργεί με χαμηλή επεξεργαστική ισχύ και σε συσκευές με μικρή μνήμη RAM. Αναλαμβάνει τη μεταγλώττιση και την δημιουργία εκτελέσιμων αρχείων τύπου .dex τα οποία αναγνωρίζονται και εκτελούνται από το λειτουργικό σύστημα. Η εκτέλεσή τους γίνεται παράλληλα από διαφορετικές μηχανές /νήματα. Σε περίπτωση όπου παρουσιαστεί σφάλμα σε κάποιο, τα υπόλοιπα συνεχίζουν να εκτελούνται κανονικά.

#### 4.1.4 Native Libraries

Είναι το επίπεδο το οποίο επιτρέπει στη συσκευή να διαχειρίζεται διαφορετικούς τύπους δεδομένων. Οι βιβλιοθήκες είναι γραμμένες σε C και C++, η πρόσβαση γίνεται μέσω του κατάλληλου interface, και έχουν τροποποιηθεί κατάλληλα διότι προορίζονται για συσκευές με μικρή μνήμη και χαμηλής ενέργειας επεξεργαστές.

#### 4.1.5 Java API Framework

Το Java Framework περιέχει όλες τις λειτουργίες που απαιτούνται ώστε οι προγραμματιστές να αξιοποιούν πλήρως το υλικό τη συσκευής, να έχουν πρόσβαση σε υπηρεσίες εντοπισμού θέσης, να τρέξουν υπηρεσίες σε νήματα ώστε να μην επηρεάζουν τις επιδόσεις της συσκευής, να εμφανίζουν ειδοποιήσεις, να διαχειρίζονται φωνητικές κλήσεις και πολλά ακόμα. Επιπλέον, δίνεται η δυνατότητα σε κάθε εφαρμογή, εφόσον το επιθυμεί ο δημιουργός της, να χρησιμοποιούνται δυνατότητες από άλλες εφαρμογές.

Οι βασικές οντότητες του framework είναι:

- *View System*: Επιτρέπει την δημιουργία του γραφικού περιβάλλοντος (UI), συμπεριλαμβάνει λίστες, πλαίσια, πεδία κειμένου, κουμπιά, ενσωματωμένο φυλλομετρητή.
- *Notification Manager*: Δίνει τη δυνατότητα στις εφαρμογές να εμφανίζουν ειδοποιήσεις στο χρήστη χωρίς να το διακόπτει στη μπάρα ειδοποιήσεων.

- *Activity Manager*: Διαχειρίζεται τον κύκλο ζωής των εφαρμογών και παρέχει την δυνατότητα μετάβασης σε προηγούμενες καταστάσεις.
- *Content Providers*: Επιτρέπει στις εφαρμογές την πρόσβαση σε δεδομένα άλλων εφαρμογών ή τον διαμοιρασμό των δικών του δεδομένων, όπως οι επαφές.
- *Resource Manager*: Παρέχει πρόσβαση σε πόρους όπως μεταβλητές (strings), γραφικά και αρχεία σχετικά με την διάταξη των στοιχείων του γραφικού περιβάλλοντος

#### 4.1.6 System Apps

Το επίπεδο εφαρμογών είναι το ανώτερο επίπεδο της αρχιτεκτονικής του Android. Σε αυτό βρίσκονται οι εφαρμογές που έρχονται προεγκατεστημένες στη συσκευή όπως οι dialer, εφαρμογή SMS, email client, browser και πολλές άλλες. Ο χρήστης έχει τη δυνατότητα να εγκαταστήσει μέσω του εικονικού καταστήματος της Google εφαρμογές σύμφωνα με τις ανάγκες του.

### 4.2 Δομικά Στοιχεία Μιας Εφαρμογής

Κάθε εφαρμογή χρησιμοποιεί 4 τμήματα κατασκευής ή τουλάχιστον ένα συνδυασμό αυτών. Την απόφαση για το ποια τμήματα θα χρησιμοποιηθούν θα την πάρει ο προγραμματιστής και θα πρέπει να τα ορίσει στο αρχείο AndroidManifest.xml. Είναι ένα αρχείο όπου δηλώνονται οι δραστηριότητες, οι υπηρεσίες, οι προθέσεις και οι πάροχοι περιεχομένου καθώς και ο τρόπος με τον οποίο θα χρησιμοποιηθούν κατά τη διάρκεια ζωής της εφαρμογής. Ακολουθεί η ανάλυση αυτών των τμημάτων:

- **Δραστηριότητα (Activity)**: Αποτελεί το κύριο στοιχείο της εφαρμογής. Είναι μια οθόνη διεπαφής χρήστη, η οποία προβάλλει πληροφορίες. Αποτελείται από εικόνες, κείμενα, κουμπιά και ανταποκρίνεται σε γεγονότα. Υπάρχουν τόσες δραστηριότητες σε μια εφαρμογή, όσες και οι διαφορετικές οθόνες που προβάλλονται στο χρήστη. Οι δραστηριότητες συνεργάζονται μεταξύ τους ώστε να υπάρχει αλληλουχία στη δομή του μενού της εφαρμογής. Ενδεικτικά, η μετάβαση από οθόνη σε οθόνη υλοποιείται με την έναρξη νέας δραστηριότητας, η οποία μπορεί να επιστρέφει κάποια τιμή σε προηγούμενη, όπως για παράδειγμα μια δραστηριότητα που εμφανίζει τις επαφές του τηλεφώνου, και επιστρέφει το προφίλ της επιλεγμένης επαφής στην δραστηριότητα που την κάλεσε.
- **Υπηρεσίες (Services)**: Πρόκειται για λειτουργίες που είναι σχεδιασμένες να επεξεργάζονται δεδομένα στο παρασκήνιο για μεγάλο χρονικό διάστημα χωρίς διεπαφή χρήστη. Μπορούν να εμφανίζουν αποτελέσματα ακόμη και όταν η εφαρμογή δεν είναι στο προσκήνιο. Για παράδειγμα, μια εφαρμογή που ενημερώνει το χρήστη για τα email του, όταν δεν βρίσκεται στο προσκήνιο μέσω μιας υπηρεσίας μπορεί να ελέγξει και να ενημερώσει τον χρήστη αν δέχτηκε κάποιο email. Η διάρκεια και ο κύκλος ζωής μιας υπηρεσίας διαφέρει από αυτού της δραστηριότητας επειδή θα συνεχίσει να τρέχει μέχρι να τη σταματήσει ο χρήστης ή μέχρι να τη σταματήσει το σύστημα για να εξοικονομήσει μνήμη.
- **Προθέσεις (Intents)**: Είναι ο συνδετικός κρίκος μεταξύ των δραστηριοτήτων. Είναι υπεύθυνες για την επικοινωνία και την μετάβαση από μια δραστηριότητα στην άλλη. Επίσης, εξασφαλίζουν την ανταλλαγή δεδομένων είτε μεταξύ των δραστηριοτήτων είτε μεταξύ των εφαρμογών. Δεν μπορούν να προβάλουν μια

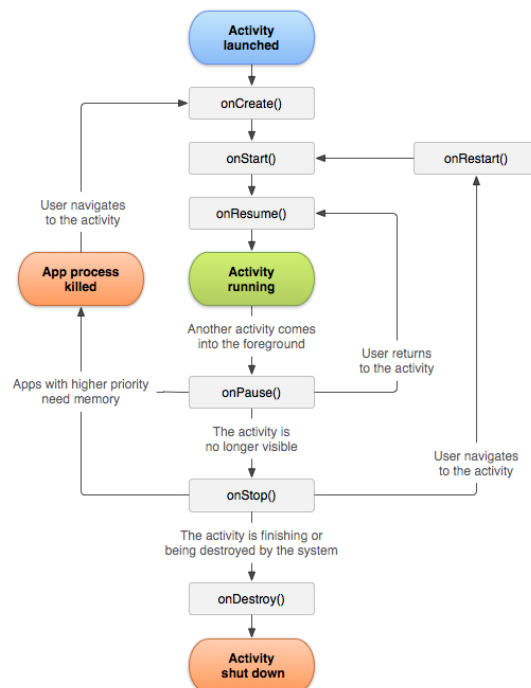


διεπαφή χρήστη, όμως μπορούν να ενημερώσουν τον χρήστη με την προβολή κάποιας ειδοποίησης για κάποιο γεγονός.

- **Πάροχοι Περιεχομένου (Content Providers):** Διαχειρίζονται αποθηκευτικούς χώρους για δεδομένα που μπορεί να χρησιμοποιήσει οποιαδήποτε εφαρμογή. Αποτελούν κλάσεις που υλοποιούν ένα σύνολο μεθόδων και επιτρέπουν στις εφαρμογές να επεξεργαστούν δεδομένα συγκεκριμένου τύπου. Συνηθισμένα δεδομένα που διαμοιράζονται είναι οι ενσωματωμένες βάσεις δεδομένων SQLite.
- **Δέκτες μηνυμάτων Broadcast (Broadcast Receivers):** Πρόκειται για ένα συστατικό του λειτουργικού συστήματος μέσω του οποίου παρέχονται στο χρήστη κατάλληλες πληροφορίες. Για παράδειγμα το επίπεδο της μπαταρίας, τυχόν ειδοποιήσεις από εφαρμογές, ότι έχει ολοκληρωθεί η λήψη ενός αρχείου κτλ. Από την πλευρά του ο προγραμματιστής έχει τη δυνατότητα να παρέχει πληροφορίες στο χρήστη. Επίσης για να μην καταναλώνει πόρους από το κύριο νήμα η εφαρμογή, μέσω των Broadcast, μπορεί να τρέξει μια λειτουργία όπως η καταγραφή στίγματος μέσω GPS σε δευτερεύον νήμα και όταν ολοκληρωθεί να επιστρέψει τις συντεταγμένες στην κύρια δραστηριότητα.

### 4.3 Κύκλος ζωής των δραστηριοτήτων

Η δραστηριότητα (activity) είναι συστατικό μιας εφαρμογής που παρέχει τις κατάλληλες πληροφορίες σε μια οθόνη, μέσω της οποίας ο χρήστης αλληλοεπιδρά και μπορεί να κάνει κλήσεις, να δει την αλληλογραφία του, να έχει πρόσβαση στο διαδίκτυο και πολλά άλλα. Κάθε φορά στην οθόνη της συσκευής εμφανίζεται μία δραστηριότητα. Μια εφαρμογή μπορεί να έχει πολλές δραστηριότητες που είναι συνδεδεμένες μεταξύ τους. Στα προγράμματα που δημιουργούνται συνήθως υπάρχει η κεντρική δραστηριότητα που παρουσιάζεται κατά την εκκίνηση της εφαρμογής και μέσω της οποίας ο χρήστης περιηγείται στις άλλες δραστηριότητες.



Εικόνα 15. Κύκλος ζωής δραστηριότητας(<https://developer.android.com/>)

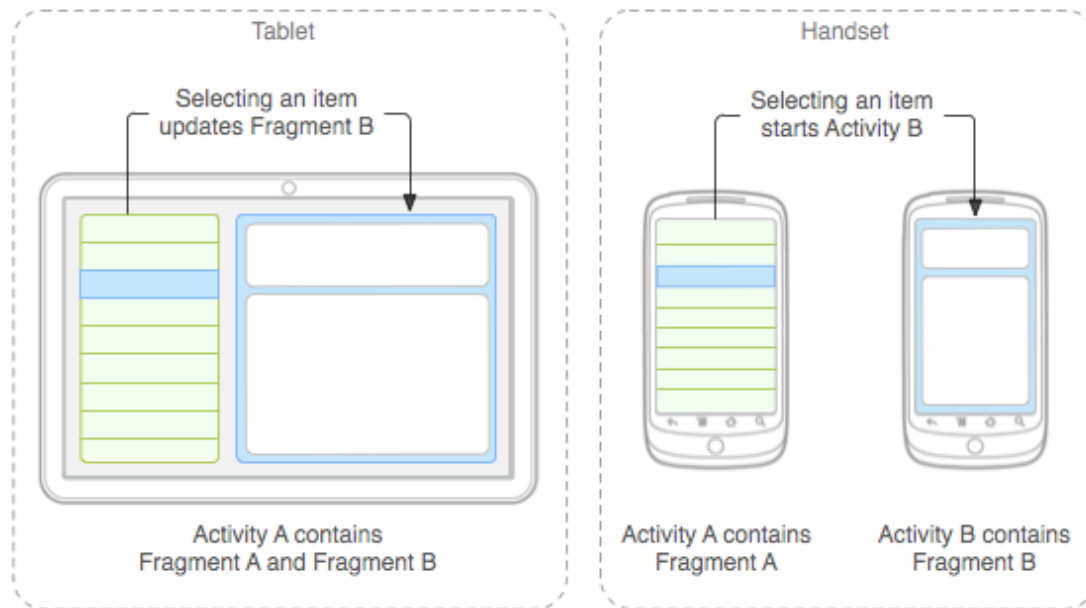
Στην Εικόνα 15 παρουσιάζονται οι μέθοδοι που καλούνται κατά την διάρκεια ζωής μιας δραστηριότητας τα οποία και θα αναλύσουμε.

- **onCreate:** Η μέθοδος καλείται όταν η δραστηριότητα εκτελείται για πρώτη φορά. Σε αυτό το στάδιο αρχικοποιούνται οι βασικές μεταβλητές της δραστηριότητας και οι μέθοδοι που θα κληθούν μια φορά και ό,τι χρειάζεται για να δημιουργηθεί το γραφικό περιβάλλον.
- **onStart:** Αυτή η μέθοδος καλείται πάντα από το σύστημα μετά από την onCreate. Οι δραστηριότητες μπορούν να υπερφορτώσουν αυτήν τη μέθοδο εάν θέλουν να εκτελέσουν συγκεκριμένες εργασίες λίγο πριν γίνει ορατή η δραστηριότητα.
- **onResume:** Σε περίπτωση που η δραστηριότητα «παγώσει» διότι μπορεί να δεχτεί μια εισερχόμενη κλήση, μέσω της onResume όταν τελειώσει η κλήση θα επιστρέψει στη δραστηριότητα και θα επαναφέρει κρίσιμες πληροφορίες που δεν θέλουμε να χάσουμε για παράδειγμα την πρόοδο ενός παιχνιδιού.
- **onPause:** Μέσω της onPause δηλώνουμε στο λειτουργικό σύστημα ότι η δραστηριότητα θα πρέπει να «παγώσει» και θα πρέπει να μπει στο παρασκήνιο ώστε να πάρει τη θέση της εφαρμογής κάποια άλλη και να γίνει αόρατη. Επίσης, μπορούμε να αποθηκεύσουμε στοιχεία ώστε όταν ο χρήστης επιστρέψει αυτά να είναι διαθέσιμα.
- **onStop:** Καλείται όταν η δραστηριότητα δεν είναι πλέον ορατή στον χρήστη. Αυτό μπορεί να συμβεί είτε διότι καταστρέφεται είτε επειδή μια άλλη δραστηριότητα έχει επαναληφθεί και την καλύπτει. Ακολουθεί η μέθοδος onStart () αν η δραστηριότητα επιστρέψει για να αλληλοεπιδράσει με το χρήστη ή ακολουθεί η μέθοδος onDestroy () αν πρόκειται να τερματιστεί.
- **onDestroy:** Αυτή είναι η τελική μέθοδος που καλείται σε ένα κύκλο ζωής μιας δραστηριότητας προτού καταστραφεί και αποδεσμεύσει τη μνήμη. Σε ακραίες καταστάσεις το λειτουργικό μπορεί να σκοτώσει τη διεργασία, η οποία θα οδηγήσει στη μη κλήση της onDestroy.

#### 4.4 Fragment

Στην παρούσα μεταπτυχιακή διατριβή πέρα από τη χρήση των δραστηριοτήτων έχει γίνει εκτεταμένη χρήση fragment για τη δημιουργία πολύπλοκου γραφικού περιβάλλοντος για το προγραμματιστή, φιλικό και εύχρηστο για το χρήστη. Το fragment πρέπει να είναι πάντα ενσωματωμένο μέσα σε μια δραστηριότητα, από την οποία μπορούμε να αλλάζουμε τις οθόνες χωρίς να καλούμε άλλη δραστηριότητα. Τα fragments έχουν το δικό τους κύκλο ζωής όμως επηρεάζονται άμεσα από τον κύκλο ζωής των δραστηριοτήτων.

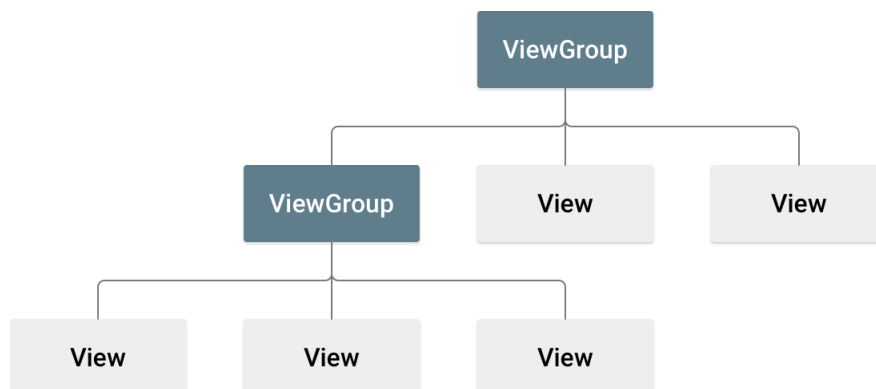
Τα fragments ενσωματώθηκαν στο λειτουργικό σύστημα Android στην 3<sup>η</sup> έκδοση του. Ο βασικός στόχος ήταν η δημιουργία δυναμικών και ευέλικτων διεπαφών με το χρήστη (user interface) κυρίως σε μεγαλύτερες οθόνες.



Εικόνα 16. Ευέλικτη σχεδίαση διεπαφών. (<https://developer.android.com/>)

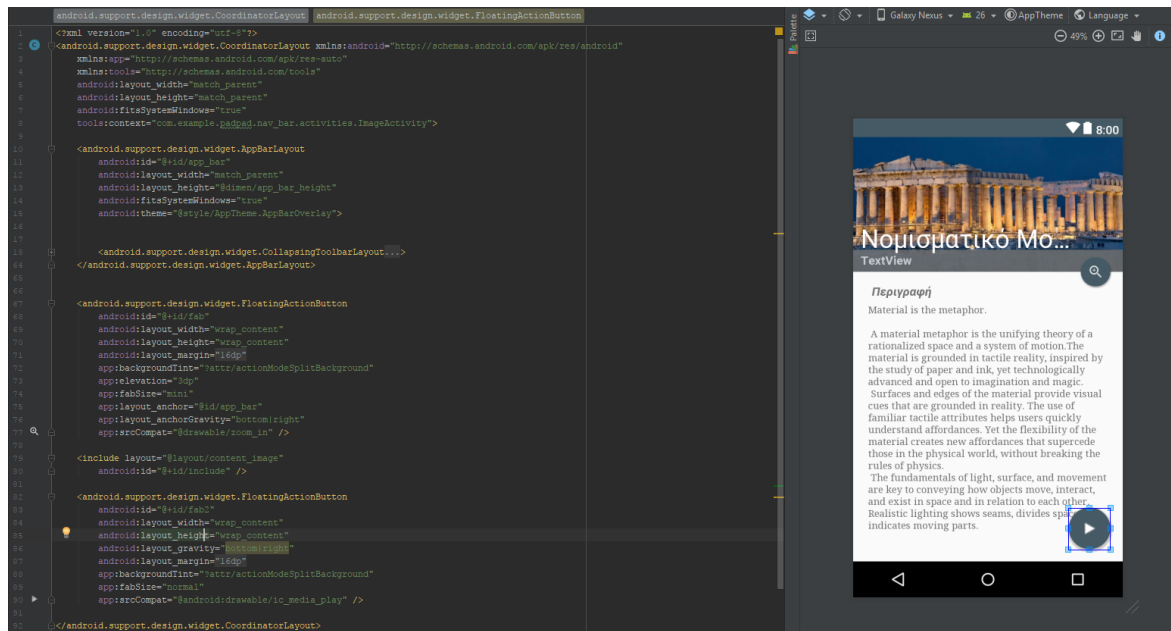
#### 4.5 Διεπαφή χρήστη-User Interface

Η διεπαφή χρήστη περιέχει όλα τα υλικά τα οποία χρησιμοποιούνται από τον προγραμματιστή ώστε να δημιουργήσει το γραφικό περιβάλλον με το οποίο ο χρήστης θα αλληλεπιδρά με την εφαρμογή. Σε αυτό περιέχονται τα αντικείμενα προβολών και οι ομάδες προβολών. Προβολή είναι ένα αντικείμενο το οποίο ζωγραφίζει στοιχεία όπου μπορούμε να αλληλοεπιδρούμε όπως ένα κουμπί. Οι ομάδες προβολής είναι ένα αντικείμενο το οποίο μπορεί να κρατάει μεμονωμένες προβολές ή ακόμα και ομάδες προβολής ώστε να δημιουργηθεί το πλαίσιο που θα περιέχει όλα αυτά τα στοιχεία.



Εικόνα 17. Ιεραρχία διεπαφών. (<https://developer.android.com/>)

Η διεπαφή χρήστη για κάθε στοιχείο της εφαρμογής έχει οριστεί χρησιμοποιώντας μια ιεραρχία από προβολές και ομάδες προβολών, όπως φαίνεται στην εικόνα 17. Κάθε ομάδα προβολής είναι ένα αόρατο δοχείο που οργανώνει τις προβολές. Το δέντρο ιεραρχίας μπορεί να είναι τόσο απλό ή σύνθετο στη σχεδίαση, όμως η απλότητα βελτιώνει τις επιδόσεις της εφαρμογής. Η δημιουργία του γραφικού περιβάλλοντος γίνεται μέσω αρχείων XML.

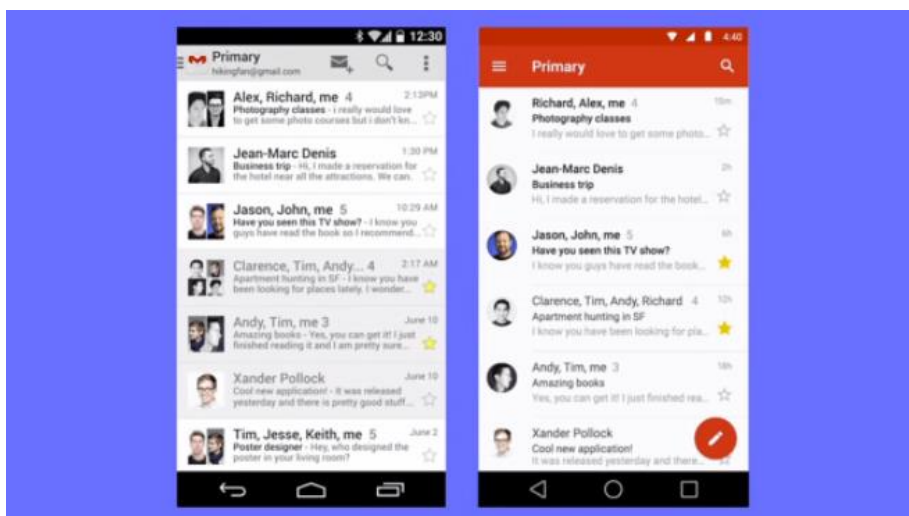


Εικόνα 18. Παράδειγμα XML κώδικα και το οπτικό αποτέλεσμα

## 4.6 Material Design

Στις 25 Ιουνίου του 2014 η Google ανακοίνωσε το Material Design. Πρόκειται για σχεδιαστικούς κανόνες, που θέσπισε η ίδια ώστε οι σχεδιαστές να παρέχουν ένα πιο φιλικό και εύχρηστο περιβάλλον στο τελικό χρήστη, ακολουθώντας συγκεκριμένους οδηγούς. Αποτελεί επέκταση του μοντέλου των "καρτών" που είναι σχεδιασμός με βασικό συστατικό στοιχείο τη διάταξη πίνακα (grid layout).

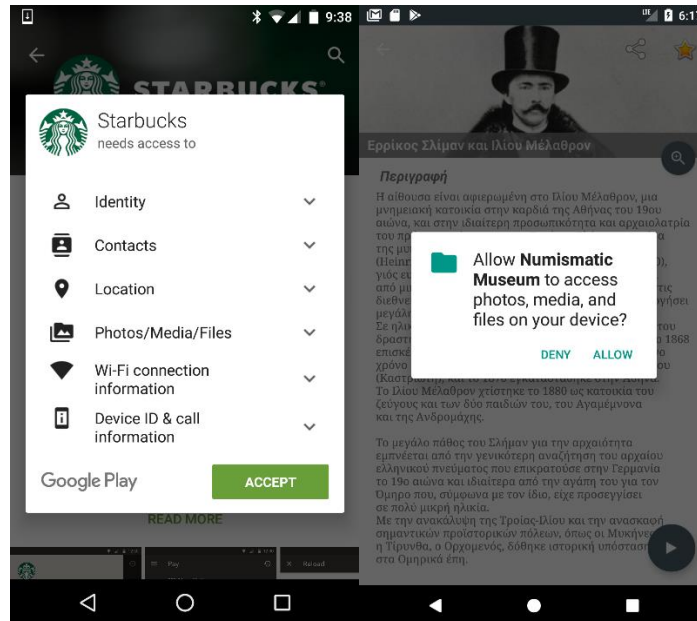
Κύριο χαρακτηριστικό του Material design είναι η ανταπόκριση των γραφικών στις κινήσεις των χρηστών με τη χρήση διάφορων εφέ, όπως αιώρηση αντικειμένων, σκιές, μεταβαλλόμενο φωτισμό, μεταβάσεις γραφικών, αίσθηση βάθους στην οθόνη, έντονα χρώματα, με σκοπό να υπάρχει μια λογική και φυσική αλληλουχία γεγονότων σε τρεις διαστάσεις. Το Material Design πρωτοεμφανίστηκε με την έκδοση Android 5.0.1 (Lollipop) και μέσω της κατάλληλης βελτιστοποίησης στις βιβλιοθήκες είναι διαθέσιμο και σε παλιότερες εκδόσεις.



Εικόνα 19. Gmail App, Πριν και Μετά το Material Design. (<https://developer.android.com/>)

## 4.7 Δικαιώματα

Τον Οκτώβριο του 2015 η Google παρουσίασε την έκτη έκδοση του λειτουργικού συστήματος Android, στο οποίο εισήγαγε ένα νέο μοντέλο εκχώρησης δικαιωμάτων στην εφαρμογή, που επιτρέπει στους χρήστες να κατανοήσουν καλύτερα τους λόγους για τους οποίους μια εφαρμογή θα ζητήσει συγκεκριμένα δικαιώματα. Στις προηγούμενες εκδόσεις του λογισμικού ο χρήστης κατά την εγκατάσταση της εφαρμογής αποδεχόταν όλα τα δικαιώματα διαφορετικά δεν προχωρούσε η εγκατάσταση.



Εικόνα 20. Το μοντέλο δικαιωμάτων πριν και μετά το Android 6.0

Πλέον ο χρήστης όταν πράξει συγκεκριμένες ενέργειες στην εφαρμογή όπως για παράδειγμα θα πρέπει να έχει πρόσβαση στην κάμερα η εφαρμογή, τότε θα εμφανιστεί ένα παράθυρο στο οποίο η εφαρμογή ζητά την άδεια από το χρήστη ώστε να μπορέσει να έχει πρόσβαση στη κάμερα της συσκευής του. Τα δικαιώματα χωρίζονται σε δύο κατηγορίες, στα κανονικά και επικίνδυνα δικαιώματα. Τα κανονικά δικαιώματα καλύπτουν τις περιοχές της εφαρμογής όπου χρειάζεται δεδομένα ή πόρους, των οποίων το ρίσκο είναι μικρό, όπως η ημερομηνία, η κατάσταση του Wi-Fi και πολλά ακόμα.

Τα επικίνδυνα δικαιώματα καλύπτουν τις περιπτώσεις όπου η εφαρμογή χρειάζεται προσωπικά δεδομένα του χρήστη ή που ενδέχεται να επηρεάσουν τα αποθηκευμένα δεδομένα του χρήστη, ακόμα και τη λειτουργία άλλων εφαρμογών.

### Επικίνδυνα Δικαιώματα:

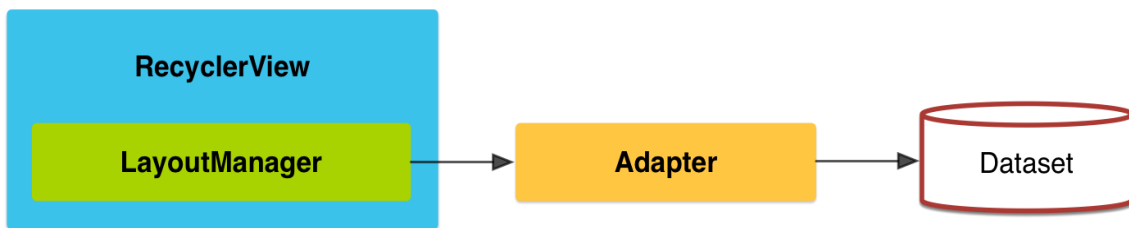
- Ανάγνωση/Εγγραφή στο Ημερολόγιο,
- Πρόσβαση στη Κάμερα.
- Ανάγνωση/Εγγραφή στις Επαφές.
- Πρόσβαση στη Τοποθεσία.
- Εγγραφή ήχων μέσω του μικροφώνου.
- Πρόσβαση στους αισθητήρες.
- Ανάγνωση/Εγγραφή στα Μηνύματα.

- Ανάγνωση/Εγγραφή στον αποθηκευτικό χώρο.

#### 4.8 RecyclerView και CardView

Η ανάγκη για βελτίωση του γραφικού περιβάλλοντος χρήσης του λειτουργικού συστήματος, ώθησε τη Google στη δημιουργία ενός framework, που θα δίνει τη δυνατότητα στους προγραμματιστές να δημιουργούν περίπλοκες λίστες με μοντέρνα σχεδίαση. Αυτό επιτεύχθηκε μέσω του RecyclerView σε συνεργασία με το CardView.

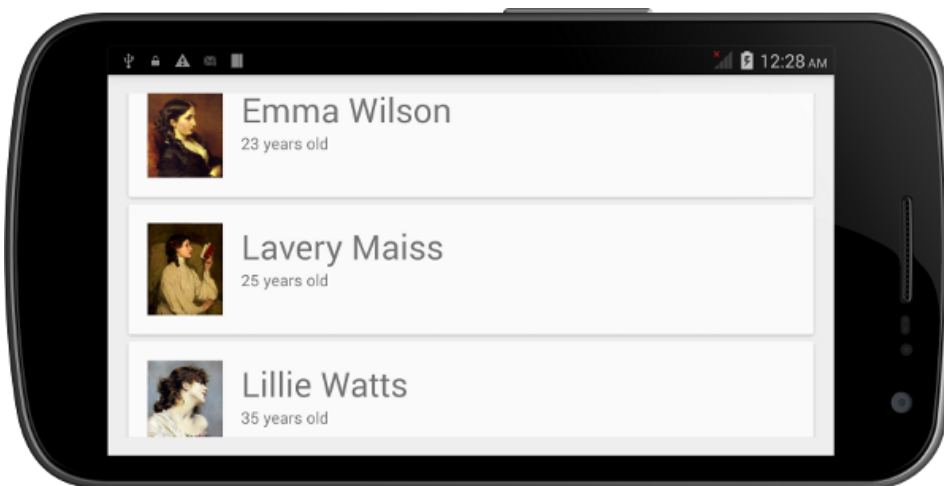
Το RecyclerView είναι μια λίστα με πιο προχωρημένες λειτουργίες και ευελιξία. Η συγκεκριμένη λίστα μπορεί να εμφανίσει ένα μεγάλο σύνολο δεδομένων που η κύλιση γίνεται ομαλά και μπορεί να προστεθούν τα κατάλληλα animations και σκιές. Η χρήση RecyclerView μαζί με CardView έχει αξιοποιηθεί στην παρούσα εφαρμογή όπου χρειάστηκε να δημιουργηθεί μια λίστα ή ένα δυναμικό γραφικό περιβάλλον.



Εικόνα 21. Μοντέλο ενός RecyclerView

Για τη χρήση ενός RecyclerView απαιτούνται:

- RecyclerView.Adapter: Διαχειρίζεται τα δεδομένα και δεσμεύεται να τα εμφανίσει στη λίστα.
- Layout Manager: Βοηθάει στη τοποθέτηση των στοιχείων.
- ItemAnimator: Παρέχει τα animations για κοινές λειτουργίες όπως η προσθήκη ή η αφαίρεση στοιχείου.

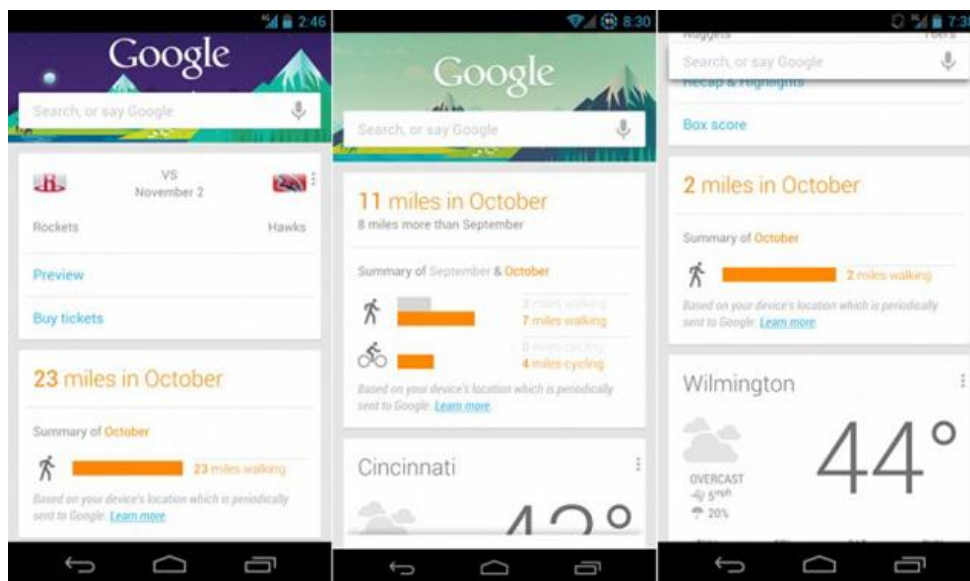


Εικόνα 22. Παράδειγμα Χρήσης Cardview με RecyclerView

#### Cardview

Το CardView προστέθηκε στην Πέμπτη έκδοση του Android μαζί με το Material Design. Πρόκειται για μια κάρτα που περιέχει πιο λεπτομερείς πληροφορίες. Μπορεί να περιέχει

φωτογραφίες, κείμενα και πλήκτρα. Μπορεί να εμφανίζεται περιεχόμενο που περιέχει στοιχεία με διαφορετικό μέγεθος, όπως φωτογραφίες με λεζάντες μεταβλητού μήκους.



Εικόνα 23. Παράδειγμα Χρήσης Cardview στο Google App.

#### 4.9 Αισθητήρες

Οι περισσότερες συσκευές με λειτουργικό σύστημα Android έχουν ενσωματωμένους αισθητήρες που μετρούν την κίνηση, τον προσανατολισμό, και διάφορες περιβαλλοντικές συνθήκες. Αυτοί οι αισθητήρες είναι σε θέση να παρέχουν δεδομένα με μεγάλη ακρίβεια.

Η πλατφόρμα Android υποστηρίζει τρεις πλατφόρμες κατηγοριών αισθητήρων:

- **Αισθητήρες Κίνησης :** Οι συγκεκριμένοι αισθητήρες μετρούν τις δυνάμεις επιτάχυνσης και περιστροφής που δέχεται η συσκευή κατά το μήκος τριών αξόνων. Αυτή η κατηγορία περιέχει επιταχυνσιόμετρα, αισθητήρες βαρύτητας και γυροσκόπια
- **Αισθητήρες Περιβάλλοντος:** Οι αισθητήρες αυτοί μετρούν διάφορες περιβαλλοντικές παραμέτρους, όπως τη θερμοκρασία περιβάλλοντος, την ατμοσφαιρική πίεση, τα επίπεδα φωτεινότητας και την υγρασία. Τέτοιοι αισθητήρες είναι τα βαρόμετρα, τα φωτόμετρα και τα θερμόμετρα.
- **Αισθητήρες Θέσης:** Αυτοί οι αισθητήρες μετρούν τη φυσική θέση μιας συσκευής. Αυτή η κατηγορία περιλαμβάνει αισθητήρες προσανατολισμού (πυξίδα), μαγνητόμετρα και GPS.

Οι αισθητήρες μιας συσκευής είναι προσβάσιμοι στους προγραμματιστές μέσω του Sensor Manager framework του Android, το οποίο παρέχει τις κατάλληλες πληροφορίες του υλικού και τα δεδομένα των αισθητήρων. Σημαντικό πρόβλημα είναι η ορθή χρήση τους διότι είναι αρκετά ενεργοβόρα και εξασθενούν την μπαταρία άμεσα.

## 4.10 Τεχνολογίες Που Χρησιμοποιήθηκαν

### 4.10.1 Βάση Δεδομένων SQLite

Κατά την υλοποίηση της εφαρμογής δημιουργήθηκε η ανάγκη για αποθήκευση δεδομένων και χρήση της εφαρμογής χωρίς τη διαρκή σύνδεση το διαδίκτυο. Τα δεδομένα που θα χρειαζότανε να έχει άμεση πρόσβαση ο χρήστης συσχετίζονται με τα δωμάτια και τα εκθέματα του μουσείου. Για τη διαχείριση των δεδομένων αυτών χρησιμοποιήθηκε η βάση δεδομένων SQLite. Η SQLite είναι μια βάση δεδομένων ανοιχτού κώδικα, αξιοποιεί το πρότυπο σχεσιακής βάσης δεδομένων και ενσωματώνει χαρακτηριστικά, όπως τη σύνταξη SQL, τις συναλλαγές και τις έτοιμες καταστάσεις. Χρησιμοποιώντας μια βάση δεδομένων SQLite στο Android δεν απαιτείται και μια διαδικασία εγκατάστασης ή διαχείριση της βάσης δεδομένων γιατί είναι ενσωματωμένη σε κάθε συσκευή Android. Το μόνο που πρέπει να καθορίζουν οι χρήστες είναι οι δηλώσεις SQL για τη δημιουργία και την ενημέρωση της βάσης δεδομένων. Στη συνέχεια, η βάση δεδομένων διαχειρίζεται αυτόματα από την πλατφόρμα Android.

Η πρόσβαση σε μια βάση δεδομένων SQLite περιλαμβάνει την πρόσβαση στο σύστημα αρχείων. Αυτό μπορεί να είναι αργή σαν διαδικασία. Ως εκ τούτου, συνιστάται να εκτελούνται λειτουργίες σε διαφορετικά νήματα. Η βάση δεδομένων από προεπιλογή αποθηκεύεται στον κατάλογο DATA/data/App\_name/databases/filename. Βέβαια η πρόσβαση στη βάση δεδομένων δεν είναι δυνατή χωρίς ο χρήστης να έχει δικαιώματα διαχειριστή στη συσκευή του.

### 4.10.2 Google Firebase

Το Firebase είναι μια cloud (BaaS) υπηρεσία, μέσω της οποίας οι προγραμματιστές μπορούν προσθέσουν χαρακτηριστικά στις εφαρμογές ανάλογα με τις ανάγκες τους. Η εταιρεία Firebase ιδρύθηκε το 2011 στην Καλιφόρνια από τους Andrew Lee και James Tamplin. Το αρχικό προϊόν ήταν μια βάση δεδομένων πραγματικού χρόνου, που παρείχε ένα API το οποίο επιτρέπει στους προγραμματιστές να συγχρονίζουν δεδομένα μεταξύ πολλαπλών πελατών/συσκευών. Η εταιρεία εξαγοράστηκε από την Google τον Οκτώβριο του 2014 και σήμερα έχει επεκταθεί στην ανάπτυξη μιας πλήρους σουίτας ανάπτυξης εφαρμογών. Στην εφαρμογή που δημιουργήθηκε έχει γίνει εκτεταμένη χρήση των λειτουργιών της Firebase. Κύριο χαρακτηριστικό της είναι πλέον η ενσωμάτωσή της στο λειτουργικό σύστημα Android, οι υπηρεσίες που προσφέρει αλλά και η δωρεάν διάθεσή της μέχρι ένα συγκεκριμένο όριο χρήσης.

Οι λειτουργίες που έχουν προστεθεί στην εφαρμογή θα αναφερθούν αναλυτικά:

#### Firestore Analytics

Πρόκειται για μια δωρεάν εφαρμογή που διατίθεται στη Firebase, που παρέχει μετρήσεις και στατιστικά σχετικά με τη χρησιμοποίηση της εφαρμογής και τους μηνιαία ενεργούς χρήστες ανάλογα με το φύλο, την ηλικία και την περιοχή. Βασίζεται στο Google Analytics, παρέχει απεριόριστες αναφορές και μέχρι πεντακόσια διαφορετικά event που μπορούν να οριστούν μέσω του SDK που παρέχεται. Μέσω αυτών των αναφορών δίνει τη δυνατότητα στο προγραμματιστή να καταλάβει τη συμπεριφορά των χρηστών, επιτρέποντας να παίρνονται αποφάσεις συνειδητά βάσει των χρηστών βελτιστοποιώντας το εκάστοτε μάρκετινγκ και τις επιδόσεις της εφαρμογής. Στην εφαρμογή χρησιμοποιείται ώστε να



καταγράφει τα δωμάτια που δίνουν περισσότερη έμφαση οι επισκέπτες, να καταγράφει τις αγορές της ηχητικής ξενάγησης και τα στατιστικά σχετικά με τους χρήστες όπως ηλικία, φύλο και αρκετές άλλες πληροφορίες.

### Realtime Database

Είναι μία βάση δεδομένων NoSQL αποθηκευμένη στο Cloud σε μορφή JSON. Η υπηρεσία παρέχει στους developers ένα API μέσω αυτού, μπορούν να ενημερώσουν τη βάση δεδομένων σε οποιασδήποτε συσκευή που τρέχει την εκάστοτε εφαρμογή σε πραγματικό χρόνο. Τα δεδομένα παραμένουν αποθηκευμένα ακόμα και εάν η εφαρμογή ξαφνικά δεν έχει πρόσβαση στο διαδίκτυο. Επίσης παρέχει βιβλιοθήκες που επιτρέπουν την ενσωμάτωση σε εφαρμογές Android, iOS, JavaScript, Java, Objective-C, swift και Node.js. Η πρόσβαση στη βάση μπορεί να περιοριστεί από τον προγραμματιστή μέσω κανόνων ασφάλειας. Τα δεδομένα αποθηκεύονται ως αντικείμενα JSON, δεν περιέχει πίνακες και εγγραφές όπως αναφέρθηκε και η διαχείρισή της γίνεται μέσω δένδρων JSON.

```
{
  "users": {
    "alovelace": {
      "name": "Ada Lovelace",
      "contacts": { "ghopper": true },
    },
    "ghopper": { ... },
    "eclarke": { ... }
  }
}
```

Εικόνα 24. Παράδειγμα JSON δέντρου της Firebase

### Cloud Storage

Προσφέρει τη δυνατότητα σε προγραμματιστές όπου χρειάζονται να αποθηκεύσουν και να παρέχουν υλικό στους χρήστες της εφαρμογής. Χρησιμοποιείται για αποθήκευση αρχείων εικόνας, ήχου και βίντεο των χρηστών. Στις περισσότερες εφαρμογές ο χρήστης έχει τη δυνατότητα να ανεβάζει (upload) ή να κατεβάζει (download) αρχεία εικόνας, ήχου ή βίντεο που δεν μπορούν να παρασταθούν σε μορφή κειμένου. Τέτοιου είδους αρχεία αποθηκεύονται στο Cloud Storage. Η ευκολία και ταχύτητα χρήσης του Firebase Storage έγκειται στο γεγονός ότι, αν κατά τη διάρκεια ενός upload ή download χαθεί η σύνδεση δικτύου, ο πελάτης μπορεί να συνεχίσει τη λειτουργία από το σημείο που σταμάτησε μόλις η σύνδεση επανέλθει. Το Storage, όπως και η βάση, διαθέτει κανόνες ασφαλείας που ελέγχουν την πρόσβαση των χρηστών στα αρχεία. Στη υλοποίησή μας αξιοποιήθηκε για την αποθήκευση των αρχείων ήχου της ηχητικής ξενάγησης και της τοπικής βάσης δεδομένων.

### Cloud Messaging

Εργαλείο για αποστολή μηνυμάτων μεταξύ των χρηστών, που διέρχονται από τον server της εφαρμογής. Μέσω του Firebase SDK μπορεί κάποιος να διαχειριστεί σε πραγματικό χρόνο την ανταλλαγή μηνυμάτων και να προωθούνται κατάλληλες ειδοποιήσεις στον χρήστη. Τα

μηνύματα μεταφέρονται και λαμβάνονται ως JSON αντικείμενα και το μέγιστο μέγεθός τους είναι τέσσερα kb. Επίσης, διατίθενται και απομακρυσμένες ειδοποιήσεις (remote notifications) ως απόκριση σε ορισμένα γεγονότα. Οι υπηρεσίες αυτές αξιοποιούνται στο Portal της εφαρμογής, όπου οι χρήστες ανταλλάσσουν μηνύματα ή μπορούν να ανεβάσουν τις δικές τους αναρτήσεις. Τέλος, μέσω των remote notifications ενημερώνεται ο χρήστης για νέα και εκδηλώσεις του μουσείου εφόσον το ενεργοποιήσει.

### Remote Config

Πρόκειται για μια cloud υπηρεσία η οποία δίνει τη δυνατότητα στους προγραμματιστές να αλλάζουν την εμφάνιση της εφαρμογής χωρίς να αναγκάζει τους χρήστες να ενημερώσουν την εφαρμογή τους. Επίσης μπορεί να ενημερώσει προεπιλεγμένες μεταβλητές με νέες σε περίπτωση λάθους ή εάν πρέπει να ενημερωθεί. Η εφαρμογή ενσωματώνει τη λειτουργία αυτή ώστε να μπορεί ο προγραμματιστής απομακρυσμένα να αλλάζει τη τιμή αγοράς της ηχητικής ξενάγησης.

### 4.10.3 Google Maps

Το Google Maps API είναι μια web εφαρμογή υπηρεσιών, η οποία παρέχεται από την Google και προσφέρει οδικούς χάρτες και υπηρεσίες πλοήγησης σε δικτυακούς τόπους ή σε εφαρμογές Android. Οι υπηρεσίες που προσφέρονται είναι οι εξής:

- οδικοί χάρτες
- πλοήγηση διαδρομής (με αυτοκίνητο, με δημόσια μέσα μεταφοράς ή με τα πόδια)
- εντοπισμός επιχειρήσεων σε χώρες σε όλο τον κόσμο.

Το Google Maps API είναι δωρεάν για εμπορική χρήση, υπό τον όρο ότι ο ιστότοπος/εφαρμογή στην οποία χρησιμοποιείται θα είναι προσβάσιμο στο κοινό χωρίς να χρεώνει τον χρήστη του για κάθε πρόσβαση και δεν παράγουν περισσότερα από 25.000 προσβάσεις στο χάρτη ανά ημέρα. Η ερώτηση στο Google Maps API γίνεται με την αποστολή ενός αιτήματος HTTP GET στην Web εφαρμογή, και επιστρέφει την απάντηση σε μορφή XML ή JSON.

Το Google Maps SDK για το λειτουργικό σύστημα παρέχει τα προαναφερθέντα χαρακτηριστικά στον προγραμματιστή, και δίνει τη δυνατότητα για προσθήκη σημείων(markers) στο χάρτη ώστε να ενημερώσει το χρήστη. Επίσης, μπορεί να παρέχει πληροφορίες για τη κίνηση στους δρόμους και να σχεδιάσει κάποια διαδρομή. Επιπλέον, ο προγραμματιστής έχει τη δυνατότητα για ενσωμάτωση ξεχωριστής δραστηριότητας (activity) στην εφαρμογή που δημιούργησε ή να περάσει δεδομένα στην ενσωματωμένη εφαρμογή του λειτουργικού, για παράδειγμα τον προορισμό και το σημείο εκκίνησης και να εμφανίσει τη διαδρομή. Η μεταφορά δεδομένων επιτυγχάνεται μέσω JSON όπως αναφέρθηκε.

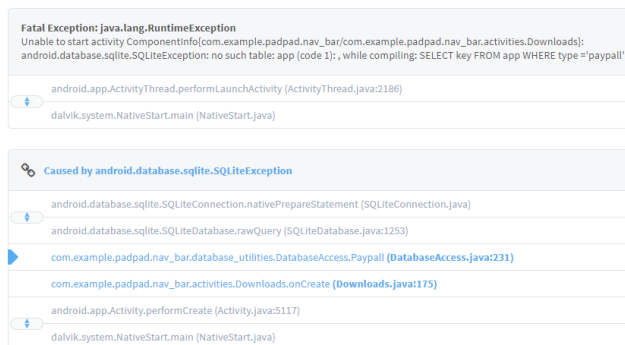
Στην εφαρμογή που αναπτύχθηκε έχει δημιουργηθεί ξεχωριστή δραστηριότητα μέσα στην εφαρμογή αξιοποιώντας το SDK, όπου ο επισκέπτης μέσω της διεπαφής μπορεί να πληκτρολογήσει την διεύθυνση εκκίνησης, να επιλέξει το μέσο μεταφοράς που θα χρησιμοποιήσει, να λάβει τη προτεινόμενη διαδρομή μέσω της Google και το χρονικό διάστημα που θα χρειαστεί για αυτή την απόσταση. Επίσης, του δίνεται η δυνατότητα

αξιοποιώντας το GPS της συσκευής του να βρει τη θέση που βρίσκεται και να τον καθοδηγήσει κατάλληλα.

#### 4.10.4 Fabric Crashlytics

Το crashlytics δημιουργήθηκε το 2011 από τους Wayne Chang και Jeff Seibert. Πρόκειται για ένα SDK το οποίο καταγράφει αστάθειες σε εφαρμογές. Είναι διαθέσιμο σε πλατφόρμες iOS/Android/Unity και μέσω αυτού οι προγραμματιστές έχουν τη δυνατότητα να λαμβάνουν τη σειρά του κώδικα που προκάλεσε την αστάθεια στην εφαρμογή και το σφάλμα.

Τη στιγμή που θα προκύψει η αστάθεια και πριν η εφαρμογή σταματήσει να ανταποκρίνεται, στέλνει στη βάση δεδομένων του Fabric τα απαραίτητα δεδομένα, τα οποία γίνονται διαθέσιμα στο προγραμματιστή στα επόμενα δύο λεπτά ή σε περίπτωση που δεν έχει πρόσβαση στο διαδίκτυο η συσκευή, όταν συνδεθεί στο διαδίκτυο στέλνει τα δεδομένα. Στην αρχή η χρήση του ήταν αποκλειστικά σε φορητές συσκευές, σήμερα έχει χρήση και σε ιστοσελίδες. Το 2017 εξαγοράστηκε από την Google και πλέον βρίσκεται σε περισσότερες από δύο δισεκατομμύρια ενεργές συσκευές .



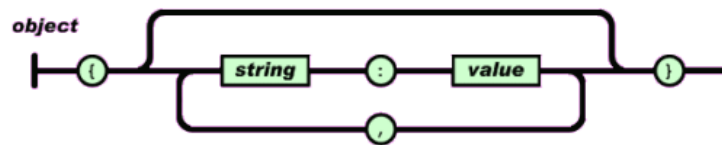
Εικόνα 25. Παράδειγμα ενός σφάλματος σε Android App.

#### 4.10.5 JSON

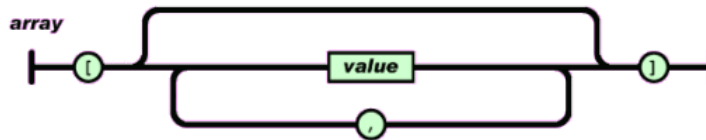
Το JSON (JavaScript Object Notation) είναι μια μορφή αποθήκευσης και ανταλλαγής δεδομένων, που καθιστά εύκολη την ανάγνωση και εγγραφή δεδομένων. Βασίζεται σε ένα υποσύνολο της γλώσσας προγραμματισμού JavaScript. Όμως επειδή τα αρχεία JSON είναι σε μορφή κειμένου είναι δυνατό να διαβαστούν και να γραφούν και από πολλές άλλες γλώσσες προγραμματισμού. Το JSON βασίζεται σε δύο δομές:

- Μία συλλογή από ζεύγη κλειδιού-τιμής, που μπορούν να χαρακτηριστούν ως “Αντικείμενα”, δομές, εγγραφές, λεξικά ή πίνακες κατακερματισμού
- Μία ταξινομημένη λίστα τιμών, που στις περισσότερες γλώσσες υλοποιείται ως πίνακας ή λίστα

Είναι λοιπόν εμφανής η συμβατότητα του JSON με τις περισσότερες γλώσσες προγραμματισμού. Η ουσιαστική χρήση του JSON είναι στη μετάδοση δεδομένων μεταξύ κινητών συσκευών και υπολογιστών αξιοποιώντας τα Web Services SOAP και REST.



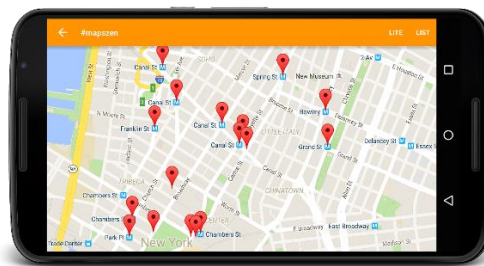
Εικόνα 26. Αλγόριθμος δημιουργίας αντικειμένου σε αρχείο JSON.



Εικόνα 27. Αλγόριθμος δημιουργίας πίνακα σε αρχείο JSON.

#### 4.10.6 Google Geocoder

Μέσω του Geocoder η Google δίνει τη δυνατότητα στους προγραμματιστές να μπορούν να μετατρέπουν διευθύνσεις οδών σε γεωγραφικές συντεταγμένες (γεωγραφικό πλάτος και γεωγραφικό μήκος). Επίσης, μπορούν να κάνουν τη αντίστροφη διαδικασία δίνοντας τις συντεταγμένες που επιθυμούν και να ενημερωθούν για την οδό. Η δυνατότητα επιτρέπει τη προσθήκη δεικτών σε ένα χάρτη ή τη σωστή εμφάνιση της θέσης του χρήστη στο χάρτη.



Εικόνα 28. Χρήση Δεικτών στο Google Map

Η αποστολή του αιτήματος που περιέχει τις συντεταγμένες ή τη διεύθυνση προς τη Google γίνεται μέσω μιας HTTP διεπαφής. Τα δεδομένα αποστέλλονται μέσω JSON ή XML. Πρώτα όμως απαιτείται η έκδοση ενός API Key μέσω της Google ώστε να γίνει ενεργοποίηση του API και η ταύτιση με το προγραμματιστή, ώστε να μην υπερβαίνει το πακέτο χρήσης που αξιοποιεί.

`https://maps.googleapis.com/maps/api/geocode/json?latlng=40.714224,-73.961452&key=YOUR_API_KEY`

Εικόνα 29. Η κλήση προς το διακομιστή της Google

## 4.11 Εργαλεία ανάπτυξης

### 4.11.1 Java

Η προεπιλεγμένη γλώσσα προγραμματισμού που χρησιμοποιείται για τη δημιουργία εφαρμογών για το λειτουργικό σύστημα Android είναι η Java, πλέον από το Μάιο του 2017 έχει προστεθεί και η γλώσσα προγραμματισμού Kotlin. Για την εκπόνηση της διπλωματικής εργασίας αξιοποιήθηκε η Java. Πρόκειται για μια αντικειμενοστραφή γλώσσα προγραμματισμού γενικού σκοπού που αναπτύχθηκε και σχεδιάστηκε από την εταιρεία πληροφορικής Sun Microsystems.

Ένα από τα βασικά πλεονεκτήματα της Java έναντι των περισσότερων γλωσσών είναι ότι τα προγράμματα που είναι γραμμένα σε Java τρέχουν ακριβώς το ίδιο σε Windows, Linux, Unix και Macintosh χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση (compiling) ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα, δηλαδή προσφέρει την ανεξαρτησία λειτουργικού συστήματος και πλατφόρμας.

### 4.11.2 Android SDK

Το Android SDK (Software Development Kit) είναι το επίσημο εργαλείο της Google για την ανάπτυξη εφαρμογών στο λειτουργικό σύστημα Android. Περιέχει μια συλλογή από εργαλεία και βιβλιοθήκες απαραίτητα για τους προγραμματιστές. Πρώτο βήμα στην πορεία της ανάπτυξης είναι η εγκατάσταση και ρύθμισή του. Περιλαμβάνει παραδείγματα εφαρμογών με τον πηγαίο τους κώδικα, βοηθήματα, πληροφορίες και εξομοιωτή για την εκτίμηση της προόδου της εργασίας. Ακόμη, αναλαμβάνει την μεταγλώττιση του κώδικα ώστε να μπορεί να τρέχει στην εικονική μηχανή Dalvik. Οι πλατφόρμες που μπορούν να υποστηρίξουν το Android SDK είναι τα συστήματα που τρέχουν Mac OS X 10.5 ή νεότερο, Windows XP ή νεότερο και οποιαδήποτε σύγχρονη με λειτουργικό Linux.

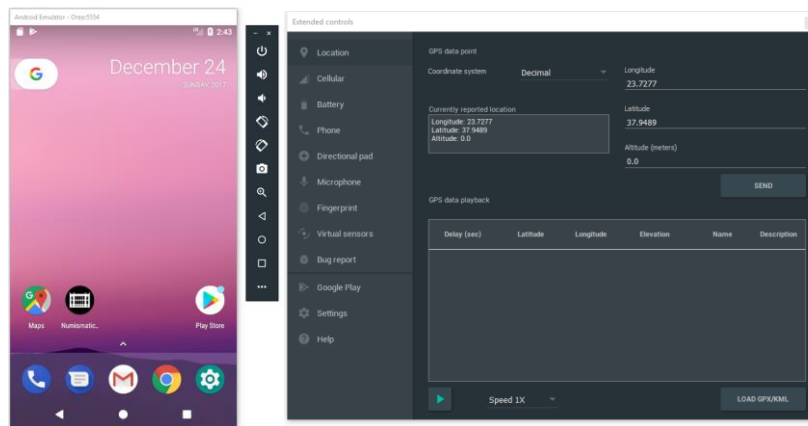
### 4.11.3 Android Studio

Το Android Studio αποτελεί ένα ολοκληρωμένο περιβάλλον ανάπτυξης λογισμικού (IDE) Android εφαρμογών, και βασίζεται στη πλατφόρμα της JetBrains IntelliJ IDEA που επανασχεδιάστηκε ειδικά για το Android λειτουργικό σύστημα. Στις 16 Μαΐου 2013 ανακοινώθηκε μια δοκιμαστική έκδοση, μέχρι τότε το επίσημο πρόγραμμα που χρησιμοποιήσουν ήταν το Eclipse IDE. Η πρώτη επίσημη έκδοση του Android Studio κυκλοφόρησε το Δεκέμβριο 2014, και από τότε έγινε το επίσημο ολοκληρωμένο περιβάλλον ανάπτυξης Android εφαρμογών. Πλέον έχει φτάσει στη τρίτη έκδοση του με πολλές νέες λειτουργίες και προσθήκες.

### 4.11.4 Εξομοιωτής Android

Οι προγραμματιστές προκειμένου να έχουν τη δυνατότητα να παρακολουθεί την πρόοδο και την λειτουργικότητα της εφαρμογής τους σε διαφορετικές συσκευές και εκδόσεις του λειτουργικού συστήματος έκανε αναγκαία τη χρήση εξομοιωτών(emulators). Οι εξομοιωτές παρέχονται από το Android SDK αλλά και από τρίτους κατασκευαστές και μαζί με το Android Studio γίνεται ένα ολοκληρωμένο περιβάλλον προγραμματισμού.

Στον εξομοιωτή έχει προστεθεί η δυνατότητα για προσομοίωση GPS ώστε να μπορεί ο προγραμματιστής να κάνει τις δοκιμές χωρίς να χρησιμοποιεί φυσική συσκευή. Επίσης, δίνεται η δυνατότητα αποστολή μηνυμάτων, πραγματοποίηση κλήσεων και χρήση εικονικών αισθητήρων ώστε να προσομοιωθούν διάφορα σενάρια χρήσης.



Εικόνα 30. Εξομοιωτής Android και οι ρυθμίσεις του.

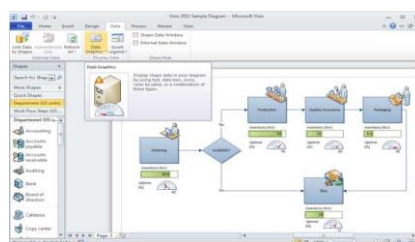
Πλέον στην τελευταία του έκδοση έχει προστεθεί το ηλεκτρονικό κατάστημα Play Store της Google, μέσω του οποίου πλέον ο εξομοιωτής είναι σχεδόν σαν κανονική συσκευή, εισάγοντας ένα ηλεκτρονικό ταχυδρομείο μπορούμε να κάνουμε λήψη εφαρμογών του καταστήματος και να προσομοιώσουμε καταστάσεις που απαιτούνται για την σωστή λειτουργίας της εφαρμογής.

#### 4.11.5 Adobe Photoshop

Για τη δημιουργία και επεξεργασία των εικονιδίων και των φωτογραφιών που χρησιμοποιήθηκαν στην διπλωματική εργασία, αξιοποιήθηκε το πρόγραμμα επεξεργασία γραφικών Adobe Photoshop. Το Photoshop είναι ένα πρόγραμμα επεξεργασίας γραφικών που αναπτύχθηκε και κυκλοφόρησε από την Adobe Systems το 1988 από τους αδερφούς Knoll. Αυτή τη στιγμή αποτελεί τον ηγέτη της αγοράς των προγραμμάτων επεξεργασίας εικόνων και είναι το προϊόν σήμα κατατεθέν της Adobe Systems. Η πιο πρόσφατη έκδοση του είναι η Adobe Photoshop CC 2018(19.0) που κυκλοφόρησε τον Οκτώβριο του 2017.

#### 4.11.6 Microsoft Visio

Πρόκειται για μια εφαρμογή που παρέχει σύγχρονα σχήματα και πρότυπα για μια ποικιλία αναγκών δημιουργίας διαγραμμάτων συμπεριλαμβανομένης της διαχείρισης τεχνολογιών πληροφορικής, της μοντελοποίησης, της δόμησης και της αρχιτεκτονικής, της σχεδίασης UI, της διαχείρισης ανθρωπίνου δυναμικού και άλλων. Δημιουργήθηκε το 1992 από την εταιρεία Shareware Corporation και εξαγοράστηκε το 2000 από τη Microsoft. Στην παρούσα εργασία αξιοποιήθηκε για τη δημιουργία UML διαγραμμάτων ώστε να βοηθήσει στη καλύτερη κατανόηση των λειτουργιών της εφαρμογής και των ρόλων των χρηστών.



Εικόνα 31. Στιγμιότυπο του Microsoft Visio(<https://products.office.com>)

## 5. Ανάλυση των λειτουργιών της Εφαρμογής

Στο κεφάλαιο αρχικά θα αναφερθούμε στην ανάλυση απαιτήσεων, το σχεδιασμό της εφαρμογής και της βάσης δεδομένων. Ολοκληρώνοντας το κεφάλαιο με τον τρόπο υλοποίησης της.

### 5.1 Ανάλυση Απαιτήσεων

Στην παρούσα ενότητα γίνεται περιγραφή της ανάλυσης απαιτήσεων της εφαρμογής που αναπτύχθηκε. Συνοπτικά, μέσω της εφαρμογής ο επισκέπτης θα έχει τη δυνατότητα να περιηγηθεί στους χώρους του μουσείου, να ενημερωθεί για τα δωμάτια και τα εκθέματα που περιέχει, να λάβει τις κατάλληλες πληροφορίες για την πρόσβαση σε αυτό, να δέχεται ειδοποιήσεις από το μουσείο αναφορικά με νέες εκδηλώσεις/ειδήσεις. Επίσης υπάρχει η δυνατότητα για ηχητική ξενάγηση, μέσω της οποίας οι επισκέπτες θα κατεβάζουν τη κατάλληλη ξενάγηση στη γλώσσα που προτιμούν. Παρακάτω θα παραθέσουμε τις απαιτήσεις που συγκεντρώθηκαν.

Οι γενικότερες απαιτήσεις του συστήματος:

- Φιλικό και εύχρηστο γραφικό περιβάλλον στο χρήστη.
- Υποστήριξη διαφορετικών συσκευών ανάλογα με τα τεχνικά χαρακτηριστικά, την έκδοση του λογισμικού και το μέγεθος της οθόνης.
- Δημιουργία της κατάλληλης βάσης δεδομένων που περιέχει όλες τις πληροφορίες και φωτογραφίες αναφορικά με το μουσείο και σε δεύτερη ξένη γλώσσα.
- Δημιουργία λογαριασμού χρήστη και επεξεργασία των στοιχείων.
- Προσθήκη των αγαπημένων εκθεμάτων.
- Χρήση Google Map για την εύκολη πρόσβαση στο μουσείο.
- Λήψη ειδοποιήσεων σχετικά με νέα του μουσείου.
- Δημιουργία portal για την ενημέρωση και την επικοινωνία των επισκεπτών.
- Κατάλληλη ενημέρωση για νέες εκδηλώσεις του μουσείου.
- Δυνατότητα κοινοποίησης των εκδηλώσεων στα κοινωνικά μέσα δικτύωσης.
- Προσθήκη των εκδηλώσεων στο ημερολόγιο της συσκευής ώστε να οριστεί η κατάλληλη υπενθύμιση.
- Ηχητική ξενάγηση και αναπαραγωγή των κατάλληλων αρχείων ήχου.
- Δυνατότητα για αγορά μέσω PayPal.
- Offline χρήση χωρίς να είναι απαραίτητη η συνεχή σύνδεση στο διαδίκτυο.
- Μέσω των κατάλληλων εργαλείων γίνεται προβολή χρονολόγιου.

Επίσης, οι απαιτήσεις που αφορούν τον προγραμματιστή:

- Ασφάλεια των δεδομένων και των συναλλαγών.
- Απομακρυσμένη ενημέρωση της βάσης δεδομένων και συγκεκριμένων μεταβλητών.
- Κατάλληλο optimization για δεδομένα και χρήση του υλικού.
- Δημιουργία analytics σχετικά με τους χρήστες και τις συνήθειες τους.
- Χρήση κατάλληλης βιβλιοθήκης για καλύτερη αποσφαλμάτωση.

- Μέσω κατάλληλου interface ενημερώνει τη βάση δεδομένων και λαμβάνει στοιχεία σχετικά με τυχόν σφάλματα(bugs) και τα analytics.

Η εφαρμογή έχει αναπτυχθεί ώστε να μπορεί εκτελεστεί από συσκευές με έκδοση μεγαλύτερη του λειτουργικού συστήματος Android 4.1. Έχει χρησιμοποιηθεί τοπική βάση δεδομένων ώστε να αποθηκεύονται όλες οι απαραίτητες πληροφορίες και ο χρήστης να μπορεί να λειτουργεί την εφαρμογή και χωρίς διαδίκτυο. Επίσης αξιοποιούνται οι υπηρεσίες του Google Firebase για αποστολή και λήψη πληροφοριών, ενημέρωσης της βάσης δεδομένων, λήψη ειδοποιήσεων και άλλες λειτουργίες. Χρήση του Google map για να μπορεί ο επισκέπτης εύκολα και γρήγορα να λαμβάνει τις κατάλληλες οδηγίες ώστε να φτάσει στο μουσείο. Επίσης αξιοποιείται η τεχνολογία JSON για επικοινωνία με το Google Maps και τη λήψη των κατάλληλων στοιχείων από το Cloud. Τέλος χρησιμοποιείται το Fabric SDK, μέσω αυτού γίνεται έλεγχος για αποσφαλμάτωση, καταγραφεί τυχόν bugs της εφαρμογής και ο προγραμματιστής μέσω ειδικού interface μπορεί να δει οποιαδήποτε στιγμή των λόγο για τον οποίο σταμάτησε να λειτουργεί.

## 5.2 Σχεδιασμός

Με βάση την ανάλυση απαιτήσεων που προαναφέρθηκε σχετικά με την εφαρμογή και τους χρήστες, θα συνεχίσουμε στο σχεδιασμό της εφαρμογής μέσω διαγραμμάτων UML όπου και θα υλοποιηθούν.

### 5.2.1 Δυνατότητες Λειτουργιών των Χρηστών

Όπως αναφέρθηκε και στην ανάλυση απαιτήσεων, στην εφαρμογή υπάρχουν δύο είδη χρηστών, ώστε να υπάρχει διαχωρισμός στις δυνατότητες που έχουν. Οι χρήστες χωρίζονται στους επισκέπτες και στους διαχειριστές. Ο καθένας από τους χρήστες έχει διαφορετικές δυνατότητες πρόσβασης στην εφαρμογή, ο επισκέπτης έχει άμεση επαφή με την εφαρμογή μέσω της συσκευής του, ενώ ο διαχειριστής έχει πρόσβαση σε λειτουργίες της εφαρμογής μέσω ειδικής διεπαφής. Οι ρόλοι θα παρουσιαστούν αναλυτικά και μέσω κατάλληλων διαγραμμάτων περιπτώσεων χρήσης.

Οι λειτουργίες που είναι διαθέσιμες στους επισκέπτες:

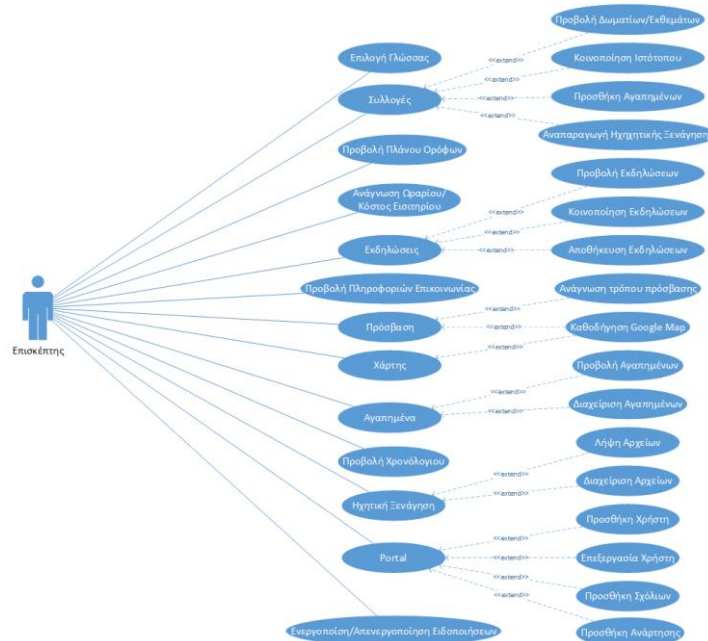
- Επιλογή Γλώσσας: Κατά την είσοδο στην εφαρμογή, δίνεται η δυνατότητα να διαλέξουν τη κατάλληλη γλώσσα ανάμεσα στα Ελληνικά και στα Αγγλικά.
- Προβολή Δωματίων/Εκθεμάτων: Μέσω της κεντρικής διεπαφής ο επισκέπτης μπορεί να επιλέξει να δει αναλυτικά τη λίστα με τα δωμάτια που περιέχονται στο μουσείο αλλά και το περιεχόμενο τους.
- Αναπαραγωγή Ηχητικής Ξενάγησης: Κατά τη περιήγηση του επισκέπτη στα δωμάτια έχει τη δυνατότητα να ακούσει τη κατάλληλη ξενάγηση για το κάθε ένα δωμάτιο ξεχωριστά.
- Προσθήκη Αγαπημένων: Ο επισκέπτης επιπλέον μπορεί να επιλέξει εφόσον το επιθυμεί να προσθέσει κάποιο έκθεμα ή δωμάτιο στα αγαπημένα του, ώστε να μπορεί εύκολα να έχει πρόσβαση σε αυτά, τα αγαπημένα αποθηκεύονται στην βάση δεδομένων της συσκευής.
- Κοινοποίηση Ιστότοπου: Δίνεται η δυνατότητα στον επισκέπτη να μπορεί να κοινοποιήσει στα κοινωνικά δίκτυα τη διεύθυνση του εκθέματος/δωματίου ώστε να



έχουν πρόσβαση και άλλοι χρήστες σε αυτό μέσω του περιηγητή της συσκευής τους.

- Πλάνο Ορόφων: Μέσω αυτού ο επισκέπτης βλέπει τη κάτοψη των δύο ορόφων ξεχωριστά και μπορεί να περιηγηθεί στα δωμάτια που υπάρχουν.
- Εισιτήρια/Ωράριο Λειτουργίας: Παρέχονται οι κατάλληλες πληροφορίες για το κόστος αγοράς των εισιτηρίων ανάλογα στην κατηγορία που βρίσκονται οι επισκέπτες και το ωράριο λειτουργίας του μουσείου.
- Εκδηλώσεις: Ο επισκέπτης μπορεί να έχει πρόσβαση στις απερχόμενες εκδηλώσεις που έχει προγραμματίσει το μουσείο αλλά και σε προηγούμενες. Έχει τη δυνατότητα να αποθηκεύσει τις εκδηλώσεις στην συσκευή του και να λάβει τη κατάλληλη ειδοποίηση πριν πραγματοποιηθεί η εκδήλωση. Επίσης, μπορεί να κοινοποιήσει στα κοινωνικά μέσα πληροφορίες σχετικά με την εκδήλωση.
- Επικοινωνία: Σε αυτή τη διεπαφή βρίσκονται όλες οι πληροφορίες σχετικά με τα στοιχεία επικοινωνίας με το μουσείο, η διεύθυνση του μουσείου και ο ιστότοπος.
- Προβολή τρόπου πρόσβαση στο μουσείο: Περιέχονται όλες οι πληροφορίες ώστε ο επισκέπτης να φτάσει στο μουσείο με τα μέσα μεταφοράς ή με το αυτοκίνητο του.
- Καθοδήγηση Google Map: Μέσω αυτού ο επισκέπτης μπορεί να δει μέσω των χαρτών της Google πως μπορεί να φτάσει στο μουσείο, να επιλέξει το μέσο μεταφοράς που επιθυμεί, να δώσει τη διεύθυνση που βρίσκεται ή να επιλέξει μέσω δορυφόρου να βρει τη θέση του. Παρέχονται ο χρόνος και η απόσταση από το αρχικό σημείο μέχρι τη θέση του μουσείου.
- Προβολή/Διαχείριση Αγαπημένων: Όπως αναφέρθηκε δίνεται η δυνατότητα να προστίθενται εκθέματα και δωμάτια σαν αγαπημένα. Σε αυτή της διεπαφή προβάλλονται οι αγαπημένες επιλογές των επισκεπτών και μπορούν να προβληθούν ή και να αφαιρεθούν.
- Προβολή Χρονολόγιου: Παρέχονται χρονολογικά τα εκθέματα του μουσείου και δίνεται η δυνατότητα για πλοήγηση και προβολή αυτών.
- Ηχητική Ξενάγηση: Αναφέραμε στις προηγούμενες λειτουργίες τη δυνατότητα για αναπαραγωγή ηχητικών αρχείων σχετικά με τη ξενάγηση του μουσείου, σε αυτή τη διεπαφή γίνεται η δυνατότητα λήψης και διαγραφής των αρχείων.
- Portal: Οι επισκέπτες μέσω του Portal μπορούν να γράψουν την εμπειρία τους από την επίσκεψη τους στο μουσείο, να διαβάσουν αναρτήσεις άλλων επισκεπτών ή του διαχειριστή και να σχολιάσουν τις αναρτήσεις. Για την προσθήκη των αναρτήσεων και των σχολίων προϋποθέτει την εγγραφή του χρήστη.
- Εγγραφή Χρήστη: Ο επισκέπτης για την ολοκλήρωση της εγγραφής του, απαιτείται η δήλωση της ηλεκτρονικής του διεύθυνσης, ενός ψευδώνυμου αλλά και του κωδικού του. Σε περίπτωση που υπάρξει λάθος κατά την εγγραφή ενημερώνεται ο χρήστης με κατάλληλο μήνυμα.
- Είσοδος Χρήστη: Για να μπορεί να προσθέτει σχόλια ο επισκέπτης θα πρέπει να κάνει είσοδο στην εφαρμογή.
- Αλλαγή Κωδικού: Από τη στιγμή που θα επιλέξει την αλλαγή κωδικού ή υπενθύμιση του κωδικού λαμβάνει στο ηλεκτρονικό ταχυδρομείο email με νέο κωδικό.

- Ειδοποιήσεις: Τέλος, οι επισκέπτες μπορούν να λαμβάνουν εφόσον το επιλέξουν ειδοποιήσεις από το μουσείο, οι ειδοποιήσεις αφορούν περιεχόμενο όπως εκδηλώσεις, νέα και οτιδήποτε επιλέξει το μουσείο να προωθήσει.



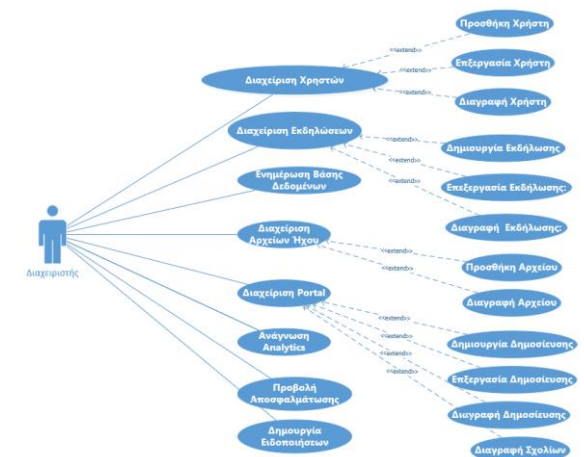
Διάγραμμα 7. Διάγραμμα περίπτωσης χρήσης του επισκέπτη.

Ο διαχειριστής από τη πλευρά του δεν έχει πρόσβαση στην εφαρμογή μέσω της διεπαφής του επισκέπτη, αλλά μέσω περιηγητή στη Firebase και στο Fabric, των οποίων θα αναφέρουμε τις λειτουργίες παρακάτω. Στο Firebase υπάρχει μια βάση δεδομένων στο cloud στην οποία περιέχονται τα στοιχεία των χρηστών, οι εκδηλώσεις, τα σχόλια των επισκεπτών, οι αναρτήσεις τους και οι συναλλαγές του PayPal. Επιπλέον υπάρχει και το Firebase Storage όπου είναι ανεβασμένα τα αρχεία της ηχητικής ξενάγησης.

Οι λειτουργίες που είναι διαθέσιμες στους διαχειριστές:

- Προσθήκη Χρήστη: Ο διαχειριστής έχει τη δυνατότητα να κάνει την εγγραφή νέων χρηστών στη Firebase.
- Επεξεργασία Χρήστη: Να κάνει τυχόν αλλαγές εφόσον το επιθυμεί ο χρήστης στα στοιχεία του με εξαίρεση το κωδικό, που είναι κρυπτογραφημένος στη βάση δεδομένων του Firebase.
- Διαγραφή Χρήστη: Σε περίπτωση που οι διαχειριστές που παρατηρήσουν επισκέπτες που έχουν απρεπή συμπεριφορά μπορούν να παγώσουν ή να διαγράψουν το λογαριασμό τους.
- Δημιουργία Εκδήλωσης: Δίνεται η δυνατότητα για ανάρτηση εκδηλώσεων που αφορούν το μουσείο.
- Επεξεργασία Εκδήλωσης: Εφόσον χρειαστεί να γίνει κάποια αλλαγή στο πρόγραμμα κάποιας εκδήλωσης μπορεί να αλλαχθεί οποιαδήποτε πληροφορία.
- Διαγραφή Εκδήλωσης: Εάν κάποια εκδήλωση ακυρωθεί δίνεται η επιλογή για διαγραφή από τη βάση δεδομένων.

- Προσθήκη/Διαγραφή Ηχητικών Αρχείων: Ο διαχειριστής μπορεί να ανεβάσει στη Firebase τα αρχεία για την ηχητική ξενάγηση και να ανανεώσει εφόσον υπάρχει κάποια νέα έκδοση.
- Δημιουργία Δημοσίευσης: Μέσω της Web διεπαφής δίνεται η δυνατότητα για τη δημιουργία καινούργιας δημοσίευσης αναφορικά με νέα του μουσείου.
- Επεξεργασία Δημοσίευσης: Σε περίπτωση λάθους ή χρειαστεί κάποια αλλαγή, ο διαχειριστής μπορεί να επεξεργαστεί τις δημοσιεύσεις ή ακόμα να αλλάξει δημοσίευση κάποιου επισκέπτη.
- Διαγραφή Δημοσίευσης/Σχολίων: Επίσης μπορεί να διαγραφεί ολοκληρωτικά κάποια δημοσίευση είτε του διαχειριστή είτε του επισκέπτη, ακόμα επίσης και απρεπή σχόλια.
- Δημιουργία Ειδοποιήσεων: Δίνεται η δυνατότητα για δημιουργία Push Notifications ώστε να τις λάβουν όσοι επισκέπτες έχουν ενεργοποιημένη αυτή την επιλογή, μπορεί επίσης να στείλει μεμονωμένες ειδοποιήσεις ανάλογα με την ηλικία, την χώρα και να δημιουργήσει προσαρμοσμένα φίλτρα.
- Ενημέρωση Βάσης Δεδομένων: Μέσω αυτού ο διαχειριστής ανεβάζει στο Firebase Storage την ανανεωμένη βάση δεδομένων και όταν κάποια συσκευή συνδεθεί στο διαδίκτυο και εκκινήσει την εφαρμογή θα λάβει ενημέρωση και θα ξεκινήσει τη λήψη και ενημέρωση της βάσης.
- Αποσφαλμάτωση: Μέσω της διεπαφής του Fabric, έχει τη δυνατότητα να λάβει σε περίπτωση που η εφαρμογή σταματήσει να λειτουργεί λόγω αστοχίας στο κώδικα (bug), να στείλει αναφορά με τους λόγους που έκανε την εφαρμογή να σταματήσει λειτουργεί ορθά.
- Analytics: Τέλος, με το Firebase Analytics ο διαχειριστής λαμβάνει όλες τις πληροφορίες που αφορά τους επισκέπτες σχετικά με την εθνικότητα, την ηλικία, την έκδοση της εφαρμογής, τη συσκευή του χρήστη και analytics σχετικά με το μουσείο όπως σε ποιο δωμάτιο περνάνε περισσότερο χρόνο.



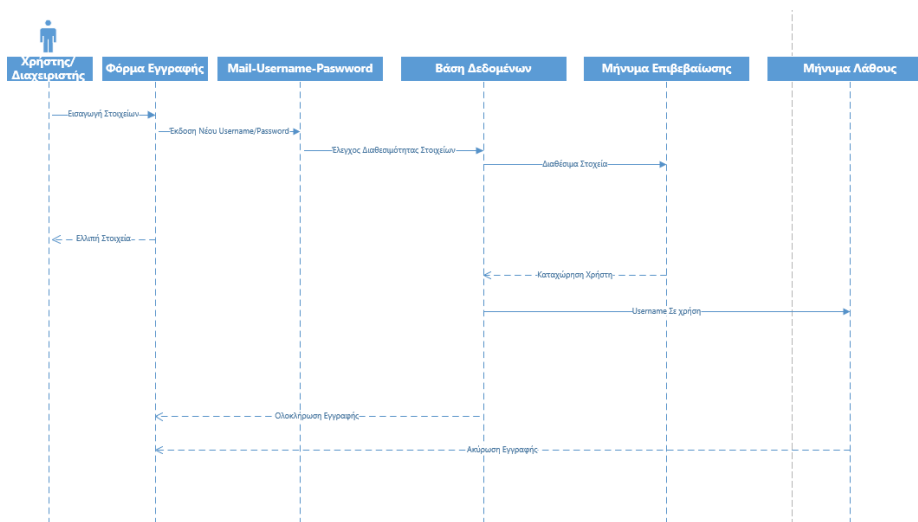
Διάγραμμα 8. Διάγραμμα περίπτωσης χρήσης του διαχειριστή.

### 5.2.2 Σχεδιασμός της Διαδικασίας Σύνδεσης και Εγγραφής του Χρήστη

Στη ανάλυση απαιτήσεων είδαμε την αναγκαιότητα για εγγραφή και σύνδεση του επισκέπτη στην εφαρμογή, είναι σκόπιμο να δοθεί προσοχή στον σχεδιασμό της

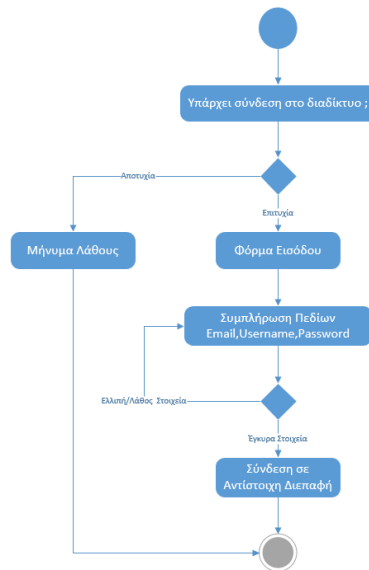
διαδικασίας που θα ακολουθείται κατά τη σύνδεση του επισκέπτη ώστε να αποφευχθούν τυχόν λάθη.

Η εγγραφή του χρήστη μπορεί να γίνει είτε μέσω της εφαρμογής είτε από τον διαχειριστή μέσω του Firebase. Τα στοιχεία που θα χρειαστεί να καταχωρήσει είναι το ηλεκτρονικό ταχυδρομείο, ένα ψευδώνυμο και το κωδικός ασφαλείας. Όλα τα στοιχεία αποθηκεύονται στην cloud βάση δεδομένων, ο κωδικός είναι κρυπτογραφημένος ώστε να τον γνωρίζει ο χρήστης και μόνο. Στην εγγραφή ζητούνται τα παραπάνω στοιχεία, σε περίπτωση όπου ο χρήστης δεν έχει καταχωρήσει κάποιο πεδίο, ο χρήστης λαμβάνει κατάλληλο μήνυμα ώστε να συμπληρώσει το κενό πεδίο και να προχωρήσει την διαδικασία. Κατά τη διάρκεια της εγγραφής γίνεται έλεγχος εάν το ηλεκτρονικό ταχυδρομείο έχει τη σωστή μορφή και δεν το χρησιμοποιεί κάποιος άλλος χρήστης. Ειδάλλως εμφανίζεται ένα μήνυμα λάθους και τον ενημερώνει να αλλάξει τα στοιχεία με διαφορετικά ώστε να ολοκληρωθεί η εγγραφή. Στο παρακάτω διάγραμμα ακολουθίας περιγράφεται η διαδικασία εγγραφής.



Διάγραμμα 9. Διάγραμμα Ακολουθίας Εγγραφής Χρήστη.

Με το πέρας της εγγραφής ο επισκέπτης έχει τη δυνατότητα να κάνει είσοδο μέσω της κατάλληλης διεπαφής συμπληρώνοντας τα στοιχεία του στα σωστά πεδία. Εάν σε περίπτωση που δεν έχει συμπληρωθεί κάποιο πεδίο λαμβάνει και εδώ το κατάλληλο μήνυμα ώστε να προβεί στη συμπλήρωσή τους. Εφόσον τα πεδία είναι σωστά, η εφαρμογή ελέγχει την εγκυρότητα στην cloud βάση δεδομένων ώστε να πιστοποιήσει τα στοιχεία του επισκέπτη και να ολοκληρωθεί η σύνδεση. Σε περίπτωση που δεν ταυτοποιηθούν, λαμβάνει κατάλληλο μήνυμα λάθους. Όλα τα παραπάνω προϋποθέτουν τη σύνδεση της συσκευής στο διαδίκτυο, για να μπορέσει να γίνει ο έλεγχος πρώτα ελέγχεται η συνδεσιμότητα και μετά προχωράει η ταυτοποίηση των στοιχείων.



Διάγραμμα 10. Διάγραμμα Δραστηριοτήτων Εισόδου Χρήστη.

### 5.2.3 Σχεδιασμός της Διαδικασίας Αγοράς Ηχητικής Ξενάγησης

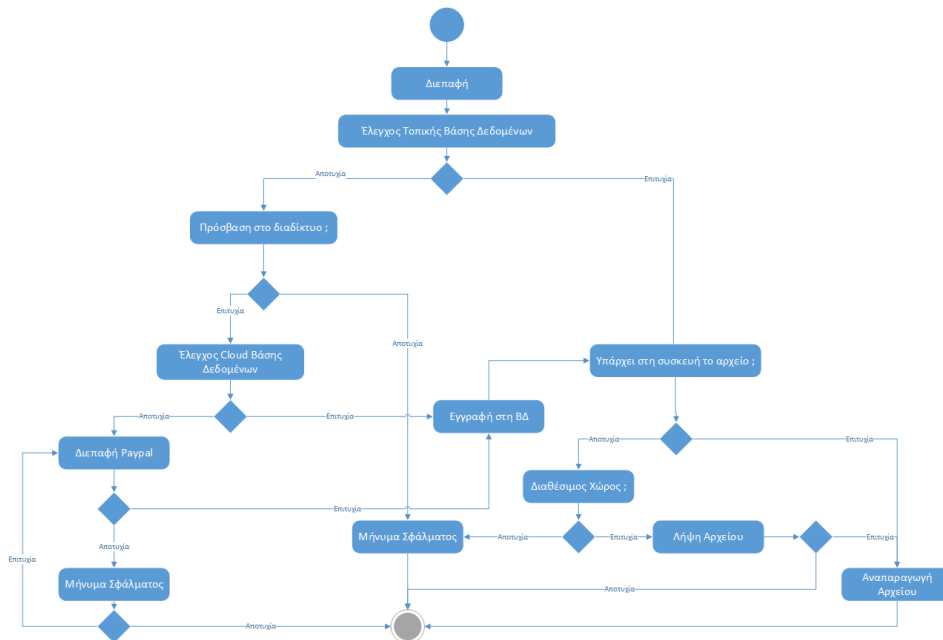
Κατά την περιήγηση του επισκέπτη στη διεπαφή των δωματίων, υπάρχει η δυνατότητα για αναπαραγωγή των αντίστοιχων αρχείων ήχου ώστε να μπορέσει να ξεναγηθεί στο μουσείο χωρίς τη φυσική παρουσία ξεναγού. Ο επισκέπτης κατά την είσοδο του στην εφαρμογή του ζητείται να επιλέξει την γλώσσα που επιθυμεί, ώστε για να εμφανίζονται τα κείμενα στη γλώσσα που επέλεξε αλλά και σε περίπτωση αγοράς της ηχητικής ξενάγησης, τα αρχεία θα ληφθούν σύμφωνα με την επιλογή που έκανε. Εάν θελήσει να αλλάξει τη γλώσσα έχει τη δυνατότητα να κατεβάσει τα αρχεία και στα Αγγλικά, δεν υπάρχει κάποιος περιορισμός

Έχοντας ο επισκέπτης επιλέξει ότι θα προβεί στην αγορά της ξενάγησης ή για να αναπαράγει κάποια από τις ξεναγήσεις, ο πρώτος έλεγχος πραγματοποιείται στη τοπική βάση της συσκευής διότι υπάρχει περίπτωση ο επισκέπτης να μην έχει πρόσβαση στο διαδίκτυο ή να έχει ήδη αγοράσει το πακέτο. Εάν δεν υπάρχει κάποια εγγραφή στη τοπική βάση προχωράει σε έλεγχο της cloud βάσης δεδομένων που βρίσκεται στη Firebase, το ερώτημα γίνεται με το μοναδικό αναγνωριστικό της κάθε συσκευής, προτού όμως γίνει ο έλεγχος στη Firebase, έχει ελεγχθεί εάν η συσκευή είναι συνδεδεμένη στο διαδίκτυο, σε περίπτωση που δεν είναι έχει πρόσβαση στο διαδίκτυο εμφανίζεται ανάλογο μήνυμα λάθους και προτρέπει τον επισκέπτη να ενεργοποιήσει τα δεδομένα ή το ασύρματο δίκτυο ώστε να προχωρήσει τη διαδικασία. Κατά τον έλεγχο στη βάση της Firebase ελέγχεται εάν έχει αγοράσει κάποια στιγμή ο επισκέπτης τη ξενάγηση από τη συγκεκριμένη συσκευή, εάν δεν βρεθεί εγγραφή, για να μπορέσει να αποκτήσει πρόσβαση θα πρέπει να καταβάλει το αντίστοιχο αντίτιμο που ορίζει το μουσείο. Η διαδικασία αγοράς της ηχητικής ξενάγησης πραγματοποιείται μέσω μιας κατάλληλης διεπαφής και χρήση του PayPal, στο οποίο ζητούνται τα στοιχεία του PayPal και το ποσό που θα χρεωθεί στο λογαριασμό του.

Εάν κατά την αγορά προκύψει οπουδήποτε πρόβλημα, για παράδειγμα δεν υπάρχει λογαριασμός στο PayPal ή είναι λάθος τα στοιχεία εισόδου, ο χρήστης θα λάβει το κατάλληλο μήνυμα ώστε να το διορθώσει. Έχοντας πλέον ολοκληρωθεί η αγορά, το PayPal εκδίδει ένα μοναδικό αναγνωριστικό της συναλλαγής, το οποίο αποθηκεύεται τοπικά στη συσκευή και στην cloud βάση δεδομένων, ώστε σε περίπτωση που ξανά εγκατασταθεί η

εφαρμογή στην ίδια συσκευή να μην χρειαστεί να αγοράσει ξανά το πακέτο.

Σε συνέχεια της περίπτωσης όπου ο χρήστης έχει ήδη αγοράσει την ηχητική ξενάγηση αλλά έχει εγκαταστήσει ξανά την εφαρμογή στη συσκευή του, έχοντας εγκριθεί από τη Firebase για την εγκυρότητα, γίνεται εγγραφή του κωδικού συναλλαγής στη τοπική βάση για χρήση της συσκευής χωρίς διαδίκτυο και στο επόμενο βήμα ελέγχεται εάν υπάρχει το ηχητικό αρχείο που επέλεξε ο χρήστης για ακρόαση στη συσκευή. Εάν το έχει κατεβασμένο στη συσκευή του γίνεται αναπαραγωγή, όμως εάν δεν βρεθεί προχωράει σε λήψη του αρχείου, προτού όμως ελέγχεται ο διαθέσιμος αποθηκευτικός χώρος της συσκευής. Ο χρήστης ενημερώνεται ότι δεν θα μπορέσει να γίνει λήψη εάν διαπιστωθεί ότι δεν υπάρχει αρκετός διαθέσιμος χώρος, σε αντίθετη περίπτωση ξεκινάει η λήψη και μετά από μερικά λεπτά ανάλογα με τη ταχύτητα της σύνδεσης έχει ολοκληρωθεί η λήψη του αρχείου. Τα αρχεία ήχου μόλις ληφθούν επιτυχώς αποθηκεύονται τοπικά στον εσωτερικό χώρο της εφαρμογής, σε αυτό το χώρο δεν έχει πρόσβαση ο χρήστης μέσω κάποιου file manager προγράμματος και εάν γίνει απεγκατάσταση της εφαρμογής διαγράφονται μαζί.



Διάγραμμα 11. Διάγραμμα Δραστηριοτήτων Αγοράς και Αναπαραγωγής Ηχητικής Ξενάγησης.

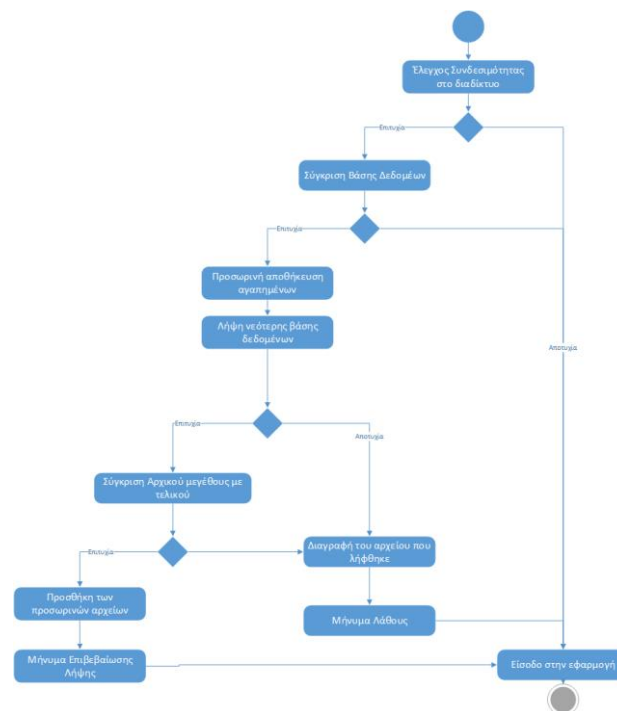
#### 5.2.4 Σχεδιασμός της Διαδικασίας Ενημέρωσης της Τοπικής Βάσης Δεδομένων

Μια επιπλέον λειτουργία της εφαρμογής είναι ότι μπορεί να ενημερώνει την τοπική βάση που βρίσκεται αποθηκευμένη στις συσκευές των επισκεπτών χωρίς να χρειάζεται αναβάθμιση ολόκληρης της εφαρμογής. Δίνει τη δυνατότητα στο μουσείο να κάνει αλλαγές «on the fly», σε περιπτώσεις όπου έχει προστεθεί κάποια λάθος πληροφορία ή να προστεθεί κάποια καινούργια, ο χρήστης ανοίγοντας την εφαρμογή θα λάβει ενημέρωση ότι υπάρχει νεότερη ενημέρωση και ότι προχωράει σε αναβάθμιση της βάσης δεδομένων.

Κατά τον αρχικό έλεγχο, η εφαρμογή ελέγχει πρώτα εάν η συσκευή έχει πρόσβαση στο διαδίκτυο, σε περίπτωση όπου δεν έχει πρόσβαση συνεχίζει στην εφαρμογή. Συνεχίζοντας όπου έχει σύνδεση στο διαδίκτυο η συσκευή, η εφαρμογή συγκρίνει την ημερομηνία όπου ανέβηκε η βάση δεδομένων στο cloud σε σχέση με την ημερομηνία δημιουργία της τοπικής βάσης. Εάν προκύψει ότι η βάση δεδομένων που βρίσκεται

αποθηκευμένη στη Google Firebase έχει νεότερη ημερομηνία ξεκινά η λήψη και μέσω της αντίστοιχης διεπαφής ενημερώνεται ο χρήστης ότι γίνεται η λήψη, σε αντίθετη περίπτωση θα συνεχίσει τη πλοήγηση του στην εφαρμογή. Πριν την έναρξη λήψης της βάσης δεδομένων ελέγχεται εάν ο επισκέπτης έχει προσθέσει κάποιο δωμάτιο ή έκθεμα σαν αγαπημένο ή έχει ολοκληρώσει αγορά της ηχητικής ξενάγησης, εάν βρεθεί κάποια εγγραφή στα προαναφερθέντα, αποθηκεύονται προσωρινά ώστε να μην χαθούν οι πληροφορίες και μόλις ολοκληρωθεί η λήψη προστίθενται στη νέα τοπική βάση.

Έχοντας πλέον έχει ολοκληρωθεί η λήψη της βάσης, γίνεται ένας τελευταίος έλεγχος ώστε να διαπιστωθεί τυχόν απώλειες στο αρχείο που έλαβε, συγκρίνοντας το αρχικό μέγεθος του αρχείου που έγινε λήψη με το τελικό μέγεθος. Επιβεβαιώνοντας ότι ολοκληρώθηκε η διαδικασία της λήψης προχωράει στην αρχική οθόνη της εφαρμογής και να επιλέξει τη γλώσσα που επιθυμεί. Σε περιπτώσεις όπου δεν ολοκληρωθεί η λήψη ο χρήστης ενημερώνεται με κατάλληλο μήνυμα λάθους και διαγράφεται το κατεστραμμένο αρχείο που έγινε λήψη, ο χρήστης μπορεί να ξαναδοκιμάσει να κάνει λήψη της ενημέρωσης όταν γίνει επανεκκίνηση της εφαρμογής.



Διάγραμμα 12. Διάγραμμα Δραστηριοτήτων Ενημέρωσης Τοπικής Βάσης Δεδομένων

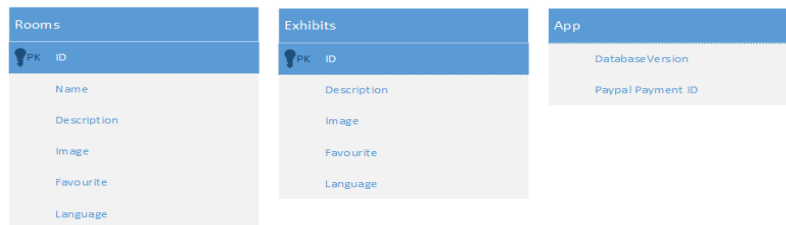
### 5.2.5 Σχεδιασμός της Τοπικής και της Firebase Βάσης Δεδομένων

Προκειμένου να επιτευχθεί η χρήση της εφαρμογής χωρίς τη συνεχή σύνδεση στο διαδίκτυο χρησιμοποιήθηκε τοπική βάση δεδομένων στην εφαρμογή όπου θα περιέχει όλες τις πληροφορίες που απαιτούνται για την ορθή χρήση της εφαρμογής. Επιπλέον λόγω των λειτουργιών της ηχητικής ξενάγησης του portal και των εκδηλώσεων υπάρχει η ανάγκη για βάση δεδομένων που θα βρίσκεται σε κάποιο εξυπηρετητή ώστε να καλύπτονται οι ανάγκες της εφαρμογής. Ως εξυπηρετητή έχει χρησιμοποιηθεί η Firebase Realtime Database όπου και βρίσκεται η βάση δεδομένων. Επιλέχτηκε η Firebase για να φιλοξενήσει τη βάση δεδομένων διότι πρόκειται για real time βάση, δίνοντας τη δυνατότητα για άμεσες και αποτελεσματικές αλλαγές, διαθέτει αρκετές μεθόδους υποβολής ερωτημάτων (querying),

μεγάλη αξιοπιστία και ασφάλεια των δεδομένων με την υποστήριξη της Google.

Στη τοπική βάση έχουν δημιουργηθεί οι τρεις πίνακες:

- **Rooms:** Στο πίνακα Rooms περιέχονται όλα τα στοιχεία που αφορούν το εκάστοτε δωμάτιο. Το πρωτεύων κλειδί είναι το "ID" όπου περιγράφει τον αριθμό του δωματίου. Στο πεδίο Name έχει οριστεί το όνομα του δωματίου. Επιπλέον υπάρχει η περιγραφή σχετικά με το δωμάτιο, μια εγγραφή τύπου blob όπου περιέχει τη φωτογραφία εξωφύλλου του δωματίου, τέλος υπάρχουν τα πεδία Favourite και Language που είναι τύπου Boolean και String αντίστοιχα. Στο Favourite γνωρίζουμε εάν ο επισκέπτης το έχει ορίσει σαν αγαπημένο ώστε να εμφανιστεί στη κατάλληλη διεπαφή και στο πεδίο Language γνωρίζουμε τη γλώσσα της εγγραφής.
- **Exhibits:** Ο συγκεκριμένος πίνακας περιέχει τις αντίστοιχες πληροφορίες σχετικά με τα εκθέματα, η χρησιμότητα των πεδίων παραμένει ίδια με του πίνακα Rooms.
- **App:** Ο τελευταίος πίνακας περιέχει την ημερομηνία έκδοσης της βάσης δεδομένων ώστε να γνωρίζει εάν πρόκειται για παλιότερη ή νεότερη έκδοση σε σχέση με τη βάση δεδομένων που βρίσκεται στη Firebase. Και η Paypal Payment ID όπου σε περίπτωση που ο επισκέπτης αγοράσει την ηχητική ξενάγηση θα αποθηκευτεί ο κωδικός συναλλαγής.



Διάγραμμα 13. Μοντέλο της Τοπικής Βάσης Δεδομένων

Η βάση δεδομένων που βρίσκεται στη Firebase είναι τύπου NoSQL αποθηκεύει τα δεδομένα ως JSON αντικείμενα. Σε σχέση με μια SQL βάση δεν περιέχει πίνακες και εγγραφές, τα δεδομένα προστίθενται σε JSON δέντρα και αποθηκεύονται στις υπάρχουσες JSON δομές με ένα συσχετισμένο κλειδί. Αναλυτική περιγραφή της Firebase Database υπάρχει στο επόμενο κεφάλαιο. Οι δομές που έχουν δημιουργηθεί χωρίζονται σε τέσσερις κατηγορίες.

Στη δομή Chat περιέχονται οι αναρτήσεις που γίνονται από τους χρήστες και το διαχειριστή στη διεπαφή Portal της εφαρμογής. Χωρίζονται σε δύο δομές ανάλογα με τη γλώσσα του χρήστη. Το μοντέλο του αντικειμένου είναι κοινό όπου εμπεριέχει το ID ως πρωτεύον κλειδί για να ξεχωρίζουν οι δομές μεταξύ τους, την ημερομηνία της ανάρτησης, το τίτλο που έχει επιλέξει ο χρήστης και το ψευδώνυμο του.

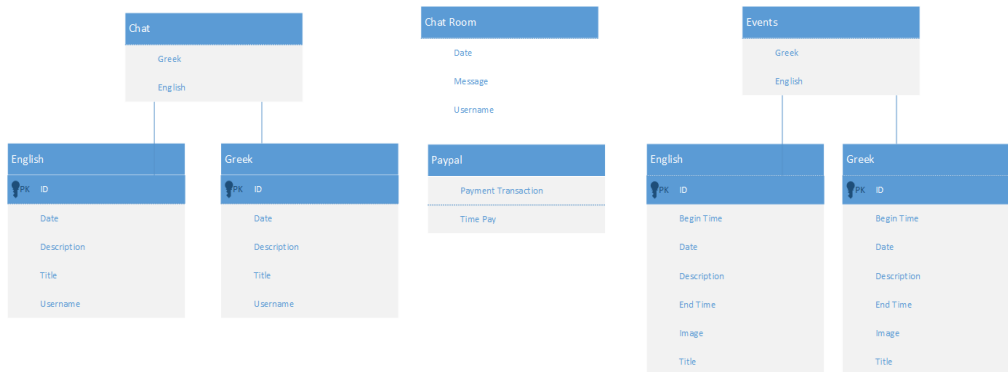
Η Chat Room περιέχει όλα τα σχόλια που έχουν προσθέσει στις αναρτήσεις οι χρήστες, πριν ανέβουν τα σχόλια η εφαρμογή γνωρίζει το παιδί της Chat Room από τη δομή Chat ώστε το σχόλιο να προστεθεί σωστά, καθώς ανεβαίνει το σχόλιο η Firebase δημιουργεί



ένα μοναδικό αναγνωριστικό ID ώστε να ξεχωρίζουν μεταξύ τους. Το μοντέλο περιέχει την ώρα που έγινε το σχόλιο ώστε να εμφανίζονται ταξινομημένα ανάλογα με την ώρα που ανέβηκαν, το ψευδώνυμο του χρήστη και τέλος το μήνυμα που ανεβάζει το οποίο περιορίζεται στους ογδόντα χαρακτήρες ώστε να μην επιβαρυνθεί ο εξυπηρετητής.

Με τη σειρά της η δομή PayPal περιέχει όλα τα αναγνωριστικά των συναλλαγών που έχουν κάνει οι χρήστες ώστε να αγοράσουν την ηχητική ξενάγηση. Όπως αναφέρθηκε τα κλειδιά αποθηκεύονται ώστε σε περίπτωση που εγκαταστήσει ξανά την εφαρμογή ο χρήστης να μην χρειάζεται να προβεί εκ νέου σε αγορά. Το μοντέλο περιέχει το κωδικό συναλλαγής και την ημερομηνία.

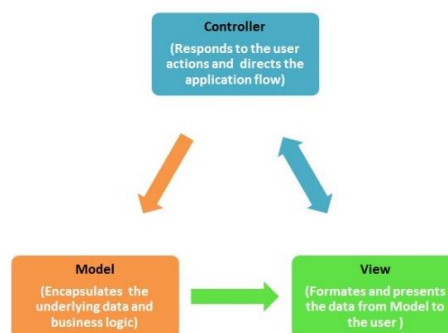
Τέλος, η Events περιέχει δύο δομές ανάλογα με τη γλώσσα του χρήστη και εδώ αναρτώνται όλες οι εκδηλώσεις του μουσείου. Στο μοντέλο έχουν προστεθεί το αναγνωριστικό ID ώστε να ξεχωρίζουν οι εκδηλώσεις μεταξύ τους, η ημερομηνία έναρξης και λήξης της εκδήλωσης, η περιγραφή της, η Image περιέχει το σύνδεσμο της εικόνας που εμφανίζεται και τέλος ο τίτλος της εκδήλωσης.



Διάγραμμα 14. Cloud Βάση Δεδομένων στη Firebase

### 5.2.6 Η MVC Αρχιτεκτονική

Η Model-View-Controller (MVC) είναι ένα αρχιτεκτονικό μοντέλο σχεδίασης, το οποίο χρησιμοποιείται ευρέως στο διαδίκτυο και σε άλλα περιβάλλοντα αλληλεπίδρασης χρήστη και συστήματος. Το MVC αποσκοπεί στο να διαχωρίσει τον τρόπο που αλληλοεπιδρά ο χρήστης με την εφαρμογή, επιδιώκοντας να διευκολύνει τους προγραμματιστές να προβαίνουν σε αλλαγές τμημάτων λογισμικού χωρίς να επηρεάζονται τα άλλα.



Εικόνα 32. Η MVC Αρχιτεκτονική

Στην αρχιτεκτονική MVC το μοντέλο (model) αναπαριστά την πληροφορία ή τα δεδομένα ενσωματώνοντας τους κανόνες και τους περιορισμούς που τα διέπουν, η όψη (view) εμπεριέχει γραφικά στοιχεία που αθροιστικά αποτελούν τη διεπαφή της εφαρμογής, με την οποία ο χρήστης αλληλεπιδρά και λαμβάνει τις κατάλληλες πληροφορίες και, τέλος, ο ελεγκτής (controller) διαχειρίζεται την επικοινωνία ανάμεσα στο μοντέλο και την όψη.

Οι τρεις ρόλοι που ορίζονται από το MVC περιγράφονται παρακάτω:

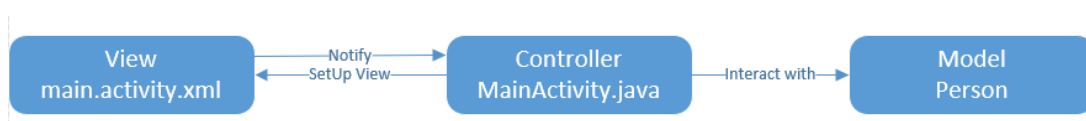
- **Μοντέλα (Models):** Το μοντέλο αναπαριστά τα δεδομένα της εφαρμογής και περιέχει τη λογική για την πρόσβαση και τον χειρισμό των δεδομένων. Τα δεδομένα για την καλύτερη διαχείρισή τους μετατρέπονται σε μοντέλα. Συνήθως τα μοντέλα σε μια εφαρμογή είναι όσα και οι πίνακες της βάσης δεδομένων, δεδομένα τα οποία λαμβάνουμε μέσω κάποιου server και πρέπει να τα αναπαραστήσουμε στο τελικό χρήστη. Έτσι το μοντέλο δύναται να εκπροσωπεί έναν πίνακα ή μία άλλου τύπου οντότητα, όπως κάποια ανεξάρτητη φόρμα η οποία επιτελεί έναν περιφερειακό σκοπό του συστήματος.
- **Όψεις (Views):** Οι όψεις αναλαμβάνουν τη δημιουργία των διεπαφών και την απεικόνιση των μοντέλων σε αυτές. Επικοινωνούν με τον ελεγκτή όταν ο χρήστης αλληλεπιδρά με την εφαρμογή ώστε να πράξει τις κατάλληλες ενέργειες.
- **Ελεγκτές (Controllers):** Οι ελεγκτές ελέγχουν τη ροή και τη λογική της εφαρμογής. Κάθε αίτηση μέσω των επιλογών των χρηστών κατευθύνεται σε ένα συγκεκριμένο ελεγκτή όπου αποδέχεται τα δεδομένα που εισήγαγε ο χρήστης. Μέσω των επιλογών του χρήστη ο ελεγκτής είτε δίνει τα μοντέλα στις όψεις ώστε να εμφανιστούν στη διεπαφή είτε πράττει τις ανάλογες κινήσεις που επέλεξε ο χρήστης από τις όψεις. Οι ελεγκτές αποτελούν το επίπεδο λήψης αποφάσεων της εφαρμογής.

### Η MVC αρχιτεκτονική και το Android

Το λειτουργικό σύστημα Android δεν υλοποιεί ακριβώς την αρχιτεκτονική του MVC, αλλά ακολουθεί μια παρεμφερή προσέγγιση. Στη παρούσα μεταπτυχιακή εργασία η εφαρμογή έχει υλοποιηθεί σε αρχιτεκτονική MVC βάσει των παρακάτω κανόνων.

- Μοντέλο: Περιέχει τα μοντέλα της βάσης δεδομένων, και τα δεδομένα που λαμβάνει και στέλνει από και προς τους servers που χρησιμοποιεί.
- Όψη: Περιέχει το γραφικό περιβάλλον μέσω των αρχείων XML .
- Ελεγκτής: Διαχειρίζεται τις ενέργειές του χρήστη και των δεδομένων μέσω αρχείων Java .

### Παράδειγμα:



Εικόνα 33. Η MVC Αρχιτεκτονική στο Android

## 6. Παρουσίαση της Εφαρμογής

Σε αυτό το κεφάλαιο θα παρουσιάσουμε την λειτουργικότητα της εφαρμογής και θα δούμε στην πράξη τις δυνατότητες που έχει ο επισκέπτης και ο διαχειριστής. Βοηθητικά θα παρουσιαστούν στιγμιότυπα (screenshots) για μεγαλύτερη ευκολία στην κατανόηση, η παρουσίαση αυτή χρησιμεύει και ως εγχειρίδιο χρήστη (user manual).

Σκοπός της μεταπτυχιακής εργασίας είναι παροχή μιας εφαρμογής που θα βελτιώσει την ποιότητα ξενάγησης στο μουσείο, θα κάνει ευκολότερη την αναζήτηση πληροφοριών, και θα δώσει τη δυνατότητα στον επισκέπτη να ξεναγηθεί χωρίς την ύπαρξη φυσικού προσώπου. Μέσω της ηχητικής ξενάγησης ο επισκέπτης θα επιλέξει τη γλώσσα που επιθυμεί να έχει μια ευχάριστη περιήγηση στους χώρους του μουσείου. Επίσης, χάρις στα κοινωνικά δίκτυα, στην εφαρμογή έχει προστεθεί ένας ψηφιακός χώρος όπου οι επισκέπτες θα μπορούν να ανταλλάξουν άμεσα ιδέες και απόψεις σχετικά με το μουσείο και να επικοινωνήσουν με προσωπικό του μουσείου έγκαιρα.

Όλα τα παραπάνω, πλέον, υλοποιούνται σε μια εφαρμογή χωρίς ο χρήστης να χρειάζεται να έχει πολλές διαφορετικές εφαρμογές εγκατεστημένες στη συσκευή του. Αυτό μειώνει τη πολυπλοκότητα. Μέσω μιας μοντέρνας και με απλή σχεδίαση εφαρμογής επιτυγχάνουμε την πλέον ευχάριστη και εύχρηστη εμπειρία στο τελικό χρήστη.

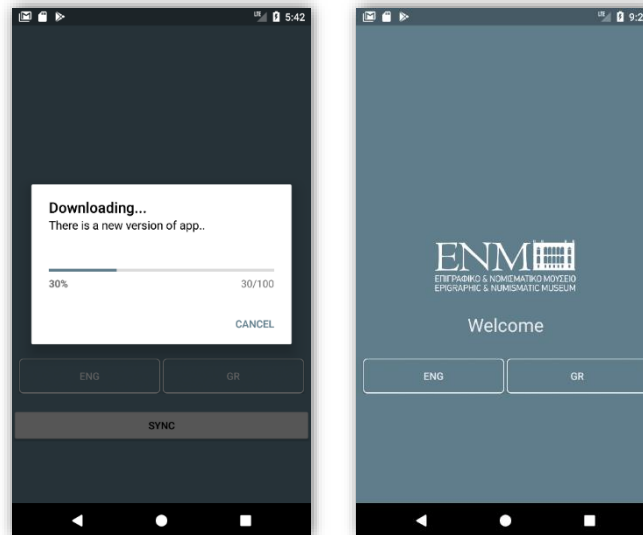
### 6.1 Η εφαρμογή στις κινητές συσκευές

Η εφαρμογή αναπτύχθηκε για το λειτουργικό σύστημα Android και είναι συμβατή με τις εκδόσεις του λογισμικού μεγαλύτερες από την έκδοση 4.1 και μέχρι τη νεότερη έκδοση 8.0. Υποστηρίζει όλες τις διαστάσεις οθονών ανεξαρτήτως κατασκευαστή. Όπως έχουμε αναφέρει για τη διαχείριση των δεδομένων και τη χρήση της εφαρμογής χωρίς τη διαρκή σύνδεση στο διαδίκτυο έχει δημιουργηθεί τοπικά μια βάση δεδομένων SQLite.

#### 6.1.1 Αρχική Διεπαφή

Κατά την είσοδο του χρήστη στην εφαρμογή του, εμφανίζεται η διεπαφή Splash όπου δίνεται η δυνατότητα για να επιλέξει τη γλώσσα που επιθυμεί ανάμεσα στην Ελληνική και στην Αγγλική γλώσσα. Πρωτού όμως προβεί στην επιλογή της γλώσσας, η εφαρμογή εφόσον υπάρχει σύνδεση στο διαδίκτυο, ελέγχει στη Firebase Storage όπου βρίσκεται η νεότερη έκδοση της βάσης δεδομένων της εφαρμογής, εάν η τοπική βάση της συσκευής είναι νεότερη ή παλιότερη σε σχέση με τη βάση δεδομένων που βρίσκεται στη Firebase.

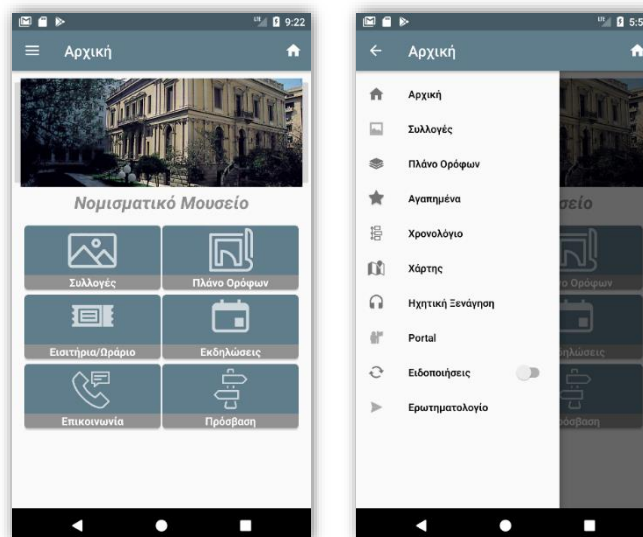
Εάν αντιληφθεί ότι υπάρχει νεότερη βάση στη Firebase τότε ενημερώνει τον χρήστη μέσω κατάλληλου μηνύματος και προχωρά στη λήψη της βάσης δεδομένων όπως φαίνεται στη εικόνα 32. Ο χρήστης έχει τη δυνατότητα να ακυρώσει τη λήψη και να προχωρήσει στη κεντρική διεπαφή. Κατά την έναρξη της λήψης, αποθηκεύονται προσωρινά όσα εκθέματα/δωμάτια έχει προσθέσει σαν αγαπημένα ο χρήστης. Τέλος, έχοντας ολοκληρωθεί η λήψη, στη νέα βάση προστίθενται τα αγαπημένα και ο χρήστης μπορεί να επιλέξει τη γλώσσα που επιθυμεί ώστε να προχωρήσει στην εφαρμογή.



Εικόνα 34. Αρχική Διεπαφή

### 6.1.2 Κεντρική Διεπαφή

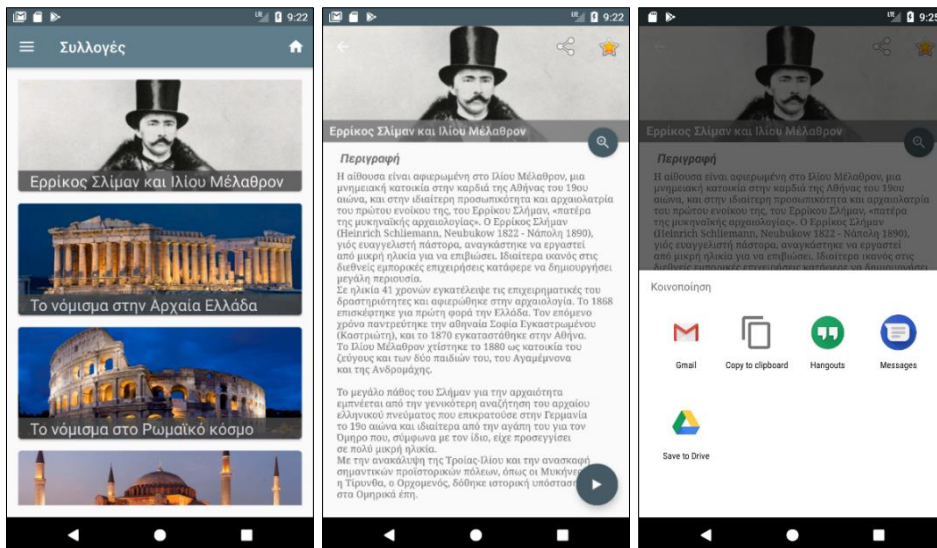
Έχοντας ο χρήστης προχωρήσει από την αρχική διεπαφή, πλέον έχει τη δυνατότητα να πλοηγηθεί στο μενού της εφαρμογής μέσα από τα έξι κουμπιά που βρίσκονται στην κεντρική οθόνη. Στη γραμμή εργαλείων(toolbar) βρίσκεται ο τίτλος της οθόνης, το κουμπί εμφάνισης της λίστας με τις επιπλέον διεπαφές της εφαρμογής και στα δεξιά το home button μέσω αυτού ο χρήστης επιστρέφει στη κεντρική διεπαφή όταν βρίσκεται σε άλλη διεπαφή. Μέσω αυτής της λίστας ο χρήστης μπορεί να πλοηγηθεί άμεσα και γρήγορα στις διεπαφές που επιθυμεί.



Εικόνα 35. Κεντρική Διεπαφή

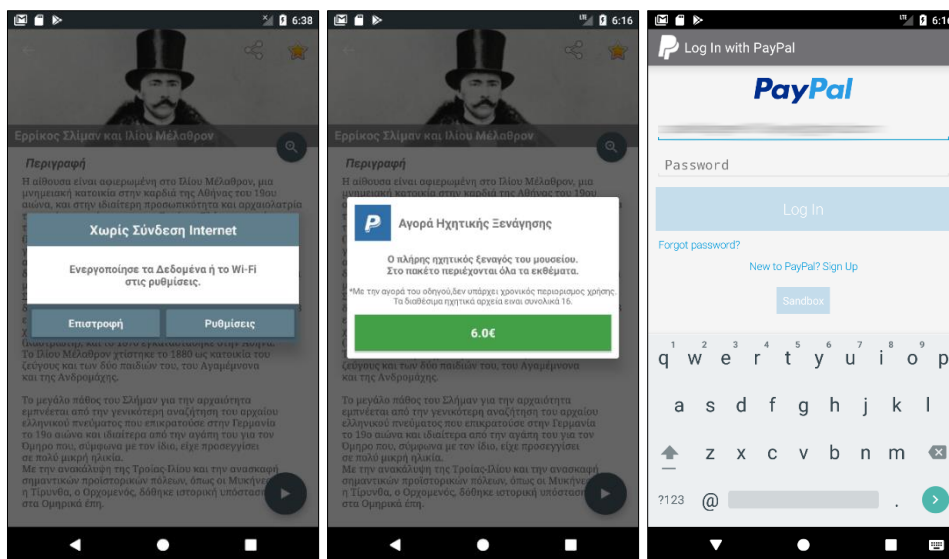
### 6.1.3 Διεπαφή Συλλογών

Επιλέγοντας ο χρήστης το κουμπί <Συλλογές> εμφανίζεται μια λίστα όπου εμφανίζονται οι οχτώ θεματικές ενότητες του Νομισματικού Μουσείου, από τις οποίες ο χρήστης μπορεί να επιλέξει να δει την κατηγορία που τον ενδιαφέρει.



Εικόνα 36. Λίστα της Συλλογής και η διεπαφή ανάλογα με τη κατηγορία

Όπως παρατηρούμε στην διεπαφή που εμφανίζεται όταν επιλέξει ο χρήστης, πέρα από την ανάγνωση του κειμένου, ο χρήστης έχει τη δυνατότητα να μεγεθύνει το κείμενο σε μεγαλύτερη γραμματοσειρά, να επιλέξει τη κατηγορία/έκθεμα ως αγαπημένο ώστε να μπορεί μέσω της διεπαφής των αγαπημένων να πλοηγηθεί άμεσα. Επιπλέον, μπορεί να αγοράσει την ηχητική ξενάγηση και να περιηγηθεί στο εκάστοτε δωμάτιο. Επιλέγοντας το κουμπί της αναπαραγωγής, θα εμφανιστεί το κατάλληλο παράθυρο όπου ενημερώνεται για το κόστος αγοράς και τους όρους. Εάν δεν υπάρχει σύνδεση στο διαδίκτυο θα ενημερωθεί με κατάλληλο παράθυρο ώστε να το ενεργοποιήσει. Εφόσον επιλέξει την αγορά, εμφανίζεται η διεπαφή του PayPal όπου ο χρήστης καταχωρεί το username και το password του PayPal του και προχωρά στην αγορά. Εάν δεν συμπληρωθούν σωστά ή υπάρχουν ελλιπή στοιχεία ο χρήστης θα ενημερωθεί μέσω κατάλληλου μηνύματος.

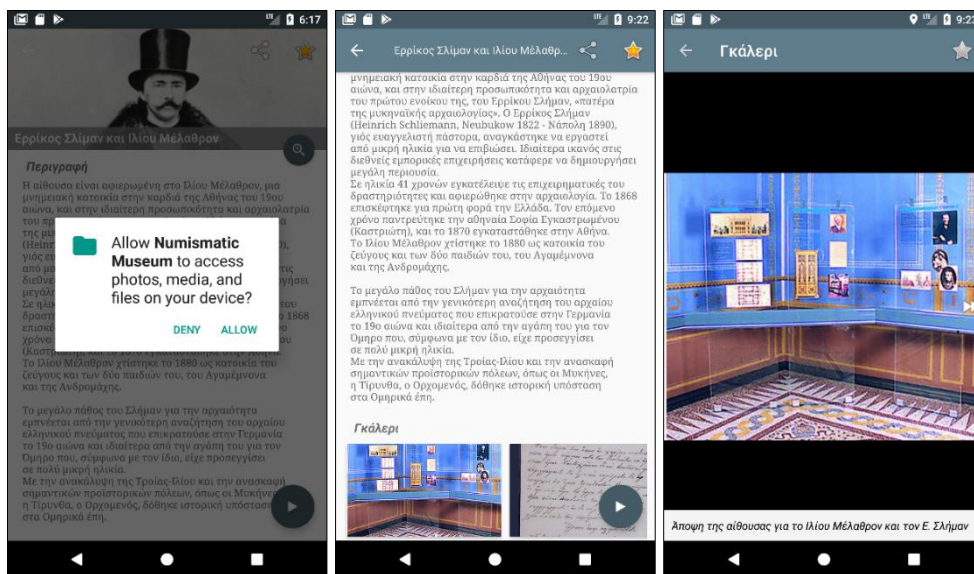


Εικόνα 37. Παράθυρα ενημέρωσης του χρήστη και διεπαφή PayPal

Εφόσον ολοκληρωθεί η διαδικασία αγοράς, πλέον ο χρήστης έχει τη δυνατότητα να λάβει το αρχείο ήχου ανάλογα με τη γλώσσα που έχει επιλέξει. Το PayPal εκδίδει τον κωδικό συναλλαγής, ο οποίος αποθηκεύεται στη τοπική βάση δεδομένων ώστε να μην χρειάζεται συνεχής σύνδεση στο διαδίκτυο για να επαληθευτεί εάν ο χρήστης έχει αγοράσει την ηχητική ξενάγηση. Ο κωδικός συναλλαγής αποθηκεύεται επίσης και στη βάση δεδομένων της Firebase, ώστε σε περίπτωση όπου ο χρήστης θελήσει να εγκαταστήσει ξανά την εφαρμογή να μην χρειαστεί να αγοράσει εκ νέου την ξενάγηση.

Πατώντας ξανά το κουμπί της αναπαραγωγής ο χρήστης λαμβάνει μήνυμα εάν επιθυμεί να δώσει την άδεια στην εφαρμογή ώστε να διαβάσει από τη μνήμη της συσκευής τα αρχεία ήχου. Αυτό συμβαίνει σε συσκευές με έκδοση του λειτουργικού συστήματος μεγαλύτερη της έκτης έκδοσης. Σε παλαιότερες εκδόσεις δεν γίνεται αυτός ο έλεγχος και η εφαρμογή προχωρά στον έλεγχο διαθεσιμότητας του αρχείου ήχου στη συσκευή. Εάν υπάρχει, γίνεται αναπαραγωγή του αρχείου αλλιώς εμφανίζεται η διεπαφή της ηχητικής ξενάγησης, για την οποία θα γίνει αναφορά στη συνέχεια.

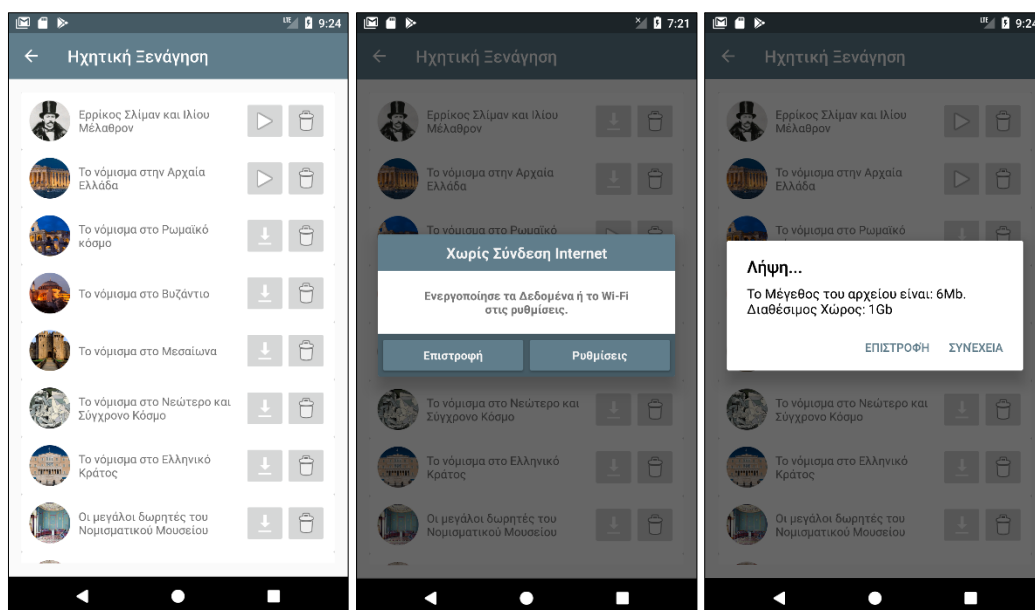
Στο τέλος της διεπαφής της Συλλογής υπάρχει ένα οριζόντιο γκάλερι στο οποίο υπάρχουν φωτογραφίες και περιγραφές για τα εκθέματα που βρίσκονται στο εκάστοτε δωμάτιο και κατηγορία. Ο χρήστης μπορεί να περιηγείται ανάμεσα στις φωτογραφίες και να τις αποθηκεύει ως αγαπημένες. Έχει προστεθεί και η δυνατότητα gestures για μεγέθυνση στις φωτογραφίες. Τέλος, στο toolbar της διεπαφής υπάρχει και το κουμπί κοινοποίησης όπου δίνεται η δυνατότητα για κοινοποίηση του περιεχομένου στα κοινωνικά δίκτυα.



Εικόνα 38. Διεπαφή φωτογραφίας και οριζόντιο γκάλερι

#### 6.1.4 Διεπαφή Ηχητικής Ξενάγησης

Όπως αναφέρθηκε ο χρήστης έχει τη δυνατότητα για αναπαραγωγή της ηχητικής ξενάγησης ανάλογα με το δωμάτιο ή τη θεματική ενότητα. Οι λειτουργίες που παρέχονται σε αυτή τη διεπαφή είναι η αγορά της ξενάγησης και η λήψη των αρχείων επειδή η διαδικασία αγοράς παραμένει ίδια, θα επικεντρωθούμε στη λήψη των αρχείων. Τα αρχεία ήχου βρίσκονται



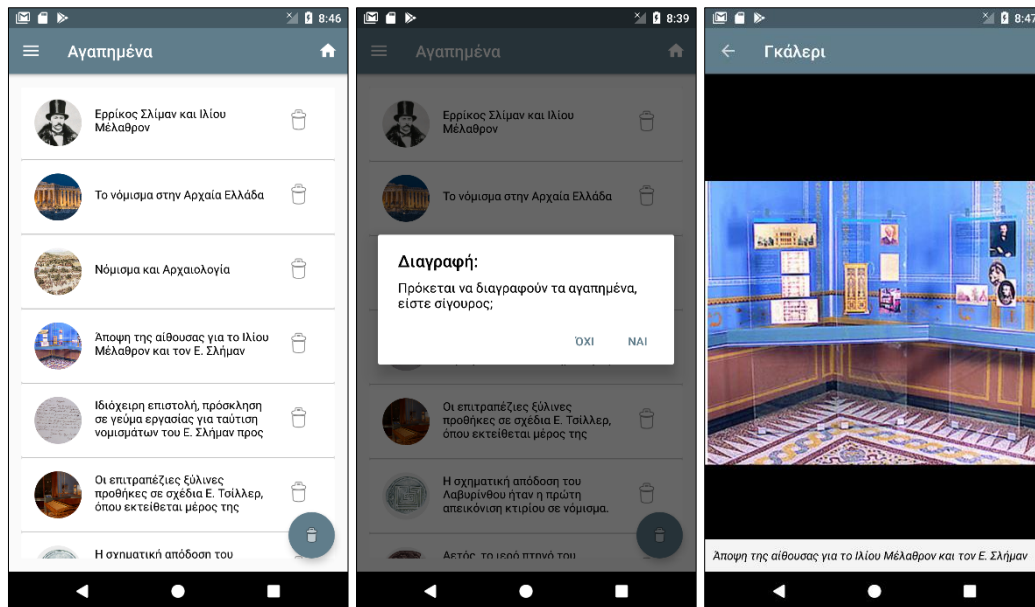
Εικόνα 39. Διεπαφή φωτογραφίας και οριζόντιο γκάλερι

αποθηκευμένα στη Firebase από όπου γίνεται η λήψη. Εφόσον έχει ολοκληρωθεί η αγορά μέσω του PayPal και ο χρήστης επιλέξει κάποιο από τα δεκαπέντε διαθέσιμα αρχεία για λήψη, γίνεται έλεγχος μήπως υπάρχει στη συσκευή, διαφορετικά ξεκινά η λήψη. Η εφαρμογή ελέγχει εάν υπάρχει διαθέσιμος χώρος στη συσκευή. Εάν δεν υπάρχει ενημερώνει το χρήστη με κατάλληλο μήνυμα και σταματάει η λήψη. Επιπλέον, εάν ο χώρος είναι λιγότερος από 500Mb ενημερώνεται με μήνυμα ο χρήστης αλλά η λήψη συνεχίζεται.

Η διαχείριση της διαδικασίας της λήψης γίνεται μέσω του Firebase SDK. Σε περίπτωση όπου η λήψη διακοπεί είτε επειδή ο χρήστης έχει αλλάξει εφαρμογή είτε λάβει μια εισερχόμενη κλήση, η λήψη παγώνει και καταχωρούμε σε μια μεταβλητή το σημείο αναφοράς της λήψης ώστε όταν επιστρέψει στην εφαρμογή ο χρήστης να συνεχίσει η λήψη από εκεί που σταμάτησε. Εάν για κάποιο λόγο η λήψη σταματήσει απροόσμενα μέσω μηνύματος ενημερώνουμε το χρήστη ότι δεν ολοκληρώθηκε η λήψη και να δοκιμάσει να κατεβάσει το αρχείο ξανά. Τέλος, ο χρήστης έχει τη δυνατότητα για αναπαγωγή των αρχείων εφόσον έχουν κατέβει στη συσκευή ή και να διαγράψει μεμονωμένα τα αρχεία ήχου.

### 6.1.5 Διεπαφή Αγαπημένων

Στην διεπαφή των Συλλογών αναφέρθηκε η δυνατότητα για προσθήκη αγαπημένων δωματίων/κατηγοριών και εκθεμάτων. Όταν ο χρήστης επιλέξει το έκθεμα σαν αγαπημένο, αυτό αποθηκεύεται στη βάση δεδομένων. Πηγαίνοντας στη διεπαφή των αγαπημένων, γίνεται έλεγχος ποια εκθέματα/δωμάτια έχουν προστεθεί και εμφανίζονται σαν λίστα στο χρήστη.

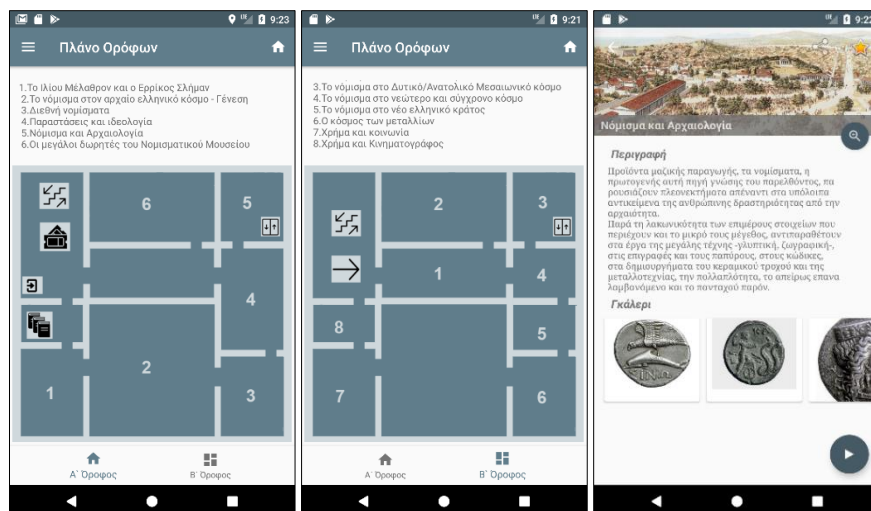


Εικόνα 40. Στιγμιότυπα Διεπαφής Αγαπημένων

Μέσω της λίστας ο χρήστης έχει τη δυνατότητα για προβολή του εκθέματος που έχει επιλέξει και μπορεί να το διαγράψει από τη λίστα του. Τέλος, έχει προστεθεί ένα κουμπί για τη διαγραφή όλων των αγαπημένων εφόσον το επιλέξει ο χρήστης.

#### 6.1.6 Διεπαφή Πλάνου Ορόφων

Στη συγκεκριμένη διεπαφή ο χρήστης έχει την επιλογή να δει τις κατόψεις των δύο ορόφων του μουσείου. Η εναλλαγή γίνεται από τα δύο κουμπιά στο τέλος της διεπαφής. Οι δύο εικόνες είναι διαδραστικές. Ο χρήστης επιλέγοντας κάποιο από το δωμάτιο ανοίγει νέα διεπαφή και έχει πρόσβαση στις πληροφορίες του δωματίου που επέλεξε. Στο πάνω μέρος της διεπαφής όπως βλέπουμε στην εικόνα 39 και 40 υπάρχει υπόμνημα σχετικά με τα δωμάτια.

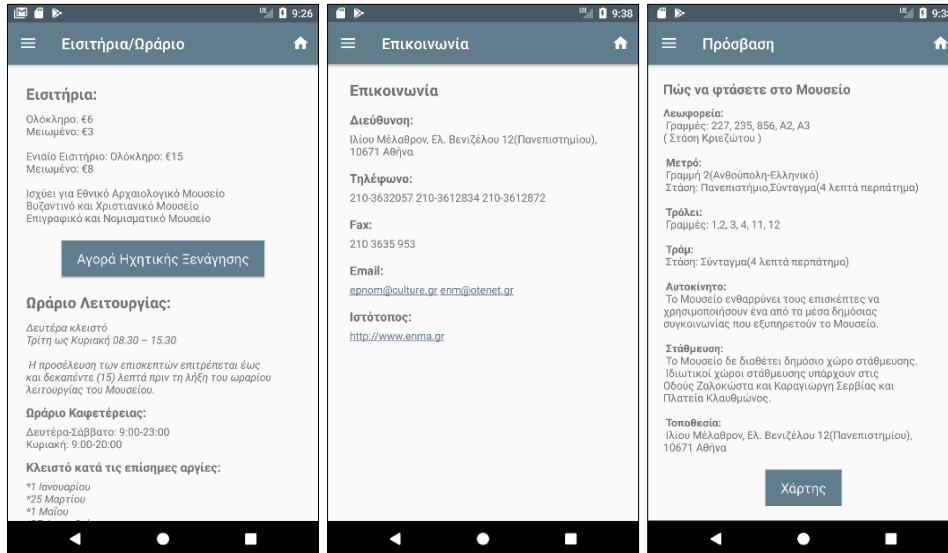


Εικόνα 41. Στιγμιότυπα Διεπαφής Πλάνου Ορόφων



### 6.1.7 Διεπαφή Εισιτήρια/Πρόσβαση

Μέσω αυτής της διεπαφής ο χρήστης ενημερώνεται για το κόστος αγοράς των εισιτηρίων στο μουσείο, και έχει τη δυνατότητα μέσα από το κουμπί <Αγορά Ηχητικής Ξενάγησης> να μεταφερθεί στη συγκεκριμένη διεπαφή άμεσα. Επιπλέον, ενημερώνεται για το ωράριο λειτουργίας του μουσείου, και της καφετέριας που βρίσκεται στο μουσείο και για τις ημέρες που το μουσείο παραμένει κλειστό.



Εικόνα 42. Στιγμιότυπα Διεπαφών

### 6.1.8 Διεπαφή Επικοινωνίας και Πρόσβασης

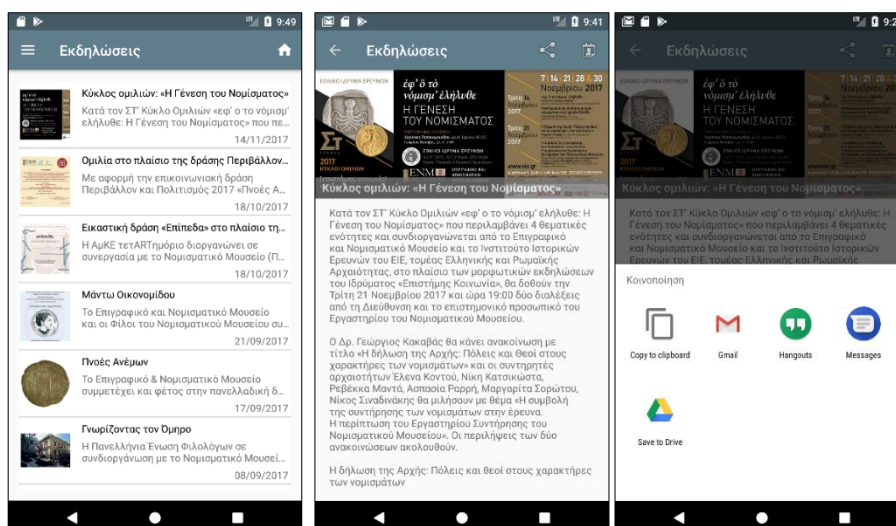
Μέσω των επόμενων δύο διεπαφών ο χρήστης έχει τη δυνατότητα να ενημερωθεί για τα στοιχεία επικοινωνίας με το Νομισματικό Μουσείο και για τους τρόπους με τους οποίους μπορεί να κατευθυνθεί στο μουσείο. Είτε πρόκειται να χρησιμοποιηθεί το αυτοκίνητο είτε τα μέσα μαζικής μεταφοράς. Στο τέλος της διεπαφής της Πρόσβασης υπάρχει μια συντόμηση ώστε να ανοίξει η διεπαφή του Google Map εσωτερικά στην εφαρμογή ώστε να μπορέσει να λάβει τις κατάλληλες πληροφορίες που επιθυμεί. Η δυνατότητα αυτή θα παρουσιαστεί αναλυτικότερα σε επόμενη παράγραφο.

### 6.1.9 Διεπαφή Εκδηλώσεων

Ο χρήστης μέσω της συγκεκριμένης δραστηριότητας μπορεί να διαβάσει τις επερχόμενες εκδηλώσεις που διοργανώνει το Νομισματικό Μουσείο, και να ενημερωθεί για περισσότερες λεπτομέρειες σχετικά με την εκδήλωση και την ημερομηνία διεξαγωγής.

Επιλέγοντας από τη λίστα κάποια εκδήλωση, στη νέα διεπαφή που ανοίγει ο χρήστης έχει τις επιλογές για κοινοποίηση της εκδήλωσης στα κοινωνικά μέσα δικτύωσης, μέσω ηλεκτρονικού ταχυδρομείου και μέσω μηνύματος μέσω του κατάλληλου πλήκτρου.

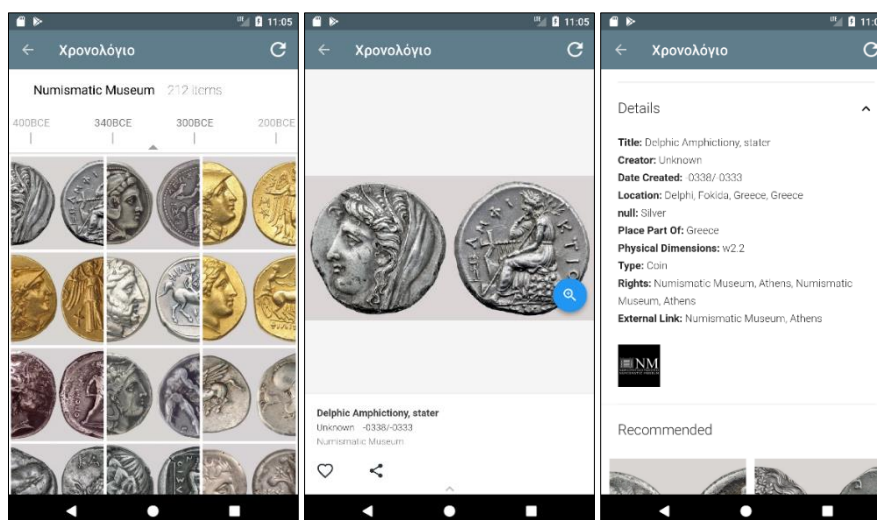
Επιπλέον, μπορεί να αποθηκεύσει την εκδήλωση στην ενσωματωμένη εφαρμογή ημερολογίου της συσκευής και να λάβει την κατάλληλη ειδοποίηση όταν κοντεύει η ημερομηνία διεξαγωγής της εκδήλωσης.



Εικόνα 43. Στιγμιότυπα Διεπαφής Εκδήλωσης

### 6.1.10 Διεπαφή Χρονολογίου

Ο χρήστης μέσω της διεπαφής του χρονολογίου έχει τη δυνατότητα για πλοήγηση ανάμεσα σε διακόσια δώδεκα εκθέματα που βρίσκονται το Νομισματικό Μουσείο. Η συγκεκριμένη διεπαφή απαιτεί σύνδεση στο διαδίκτυο. Πρόκειται για ένα WebView που παρέχεται στο Android SDK, μέσω του οποίου μπορούμε να προβάλλουμε μια ιστοσελίδα εσωτερικά στην εφαρμογή χωρίς να απαιτείται από το χρήστη να ανοίξει τον περιηγητή της συσκευής. Το περιεχόμενο του χρονολογίου έχει δημιουργηθεί μέσω της συνεργασίας της Google και του Νομισματικού Μουσείου προκειμένου να ανέβει στο Google Art Project.

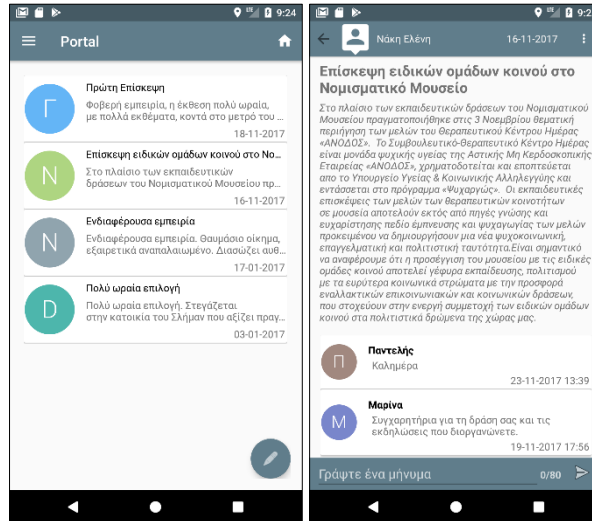


Εικόνα 44. Στιγμιότυπα Διεπαφής Χρονολογίου

Ο χρήστης μέσω της οριζόντιας μπάρας που βρίσκεται στο πάνω μέρος της διεπαφής μπορεί να πλοηγηθεί ανάμεσα σε εκθέματα ανά χρονολογία, από κάτω εμφανίζονται σε πλέγμα τα εκθέματα, ο χρήστης επιλέγοντας κάποιο από αυτά έχει πρόσβαση στη φωτογραφία σε μεγαλύτερη ανάλυση και στις πληροφορίες που παρέχονται. Τέλος έχει προστεθεί η δυνατότητα για ανανέωσης της σελίδας σε περίπτωση κακής σύνδεσης στο διαδίκτυο, ώστε να φορτώσει ξανά το περιεχόμενο μέσω του κουμπιού που βρίσκεται στα δεξιά της γραμμής εργαλείων.

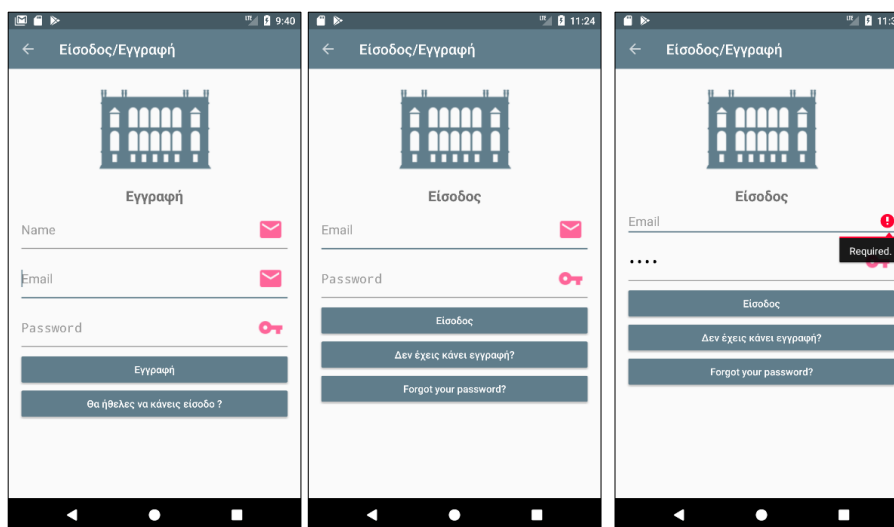
### 6.1.11 Διεπαφή Portal

Στην εφαρμογή έχει προστεθεί η δυνατότητα στους χρήστες και τους διαχειριστές για ανταλλαγή μηνυμάτων. Μέσω του Portal ο χρήστης μπορεί να κάνει σχόλια σε κάποια από τις υπάρχουσες αναρτήσεις ή μπορεί να δημιουργήσει ο ίδιος κάποια ανάρτηση. Για την ανάγνωση των σχόλιων δεν χρειάζεται η δημιουργία λογαριασμού, σε περίπτωση όπου ο χρήστης είτε θελήσει να σχολιάσει είτε να ανεβάσει μια ανάρτηση απαιτείται η δημιουργία λογαριασμού. Το περιεχόμενο της διεπαφής αποθηκεύονται στη Firebase Database.



Εικόνα 45. Στιγμιότυπα Διεπαφής Portal

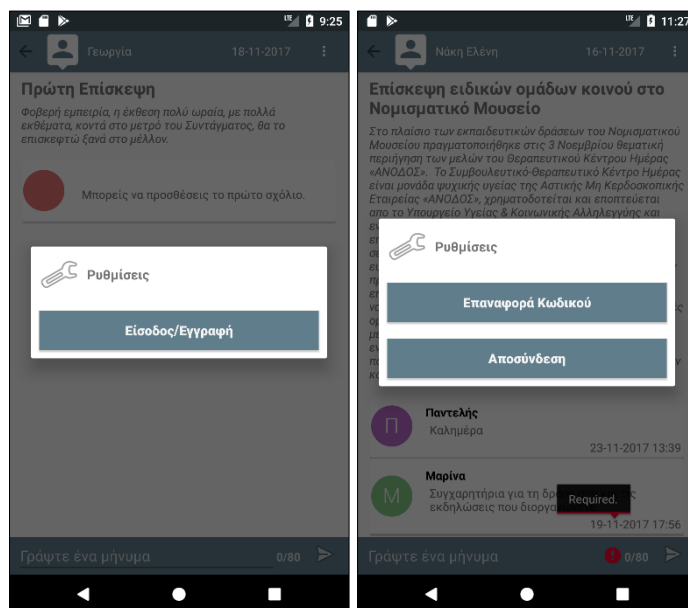
Κατά την είσοδο στη διεπαφή, ο χρήστης βλέπει σε μορφή λίστας τις αναρτήσεις που υπάρχουν διαθέσιμες για ανάγνωση, επιλέγοντας μια από τις αναρτήσεις, ανοίγει καινούργια διεπαφή. Ο χρήστης παρατηρεί στη γραμμή εργασιών το ψευδώνυμο του χρήστη που ανέβασε την ανάρτηση, την ημερομηνία ανάρτησης και το πλήκτρο μενού όπου δίνονται διαφορετικές επιλογές ανάλογα τη κατάσταση του χρήστη. Εάν έχει κάνει είσοδο του δίνεται η δυνατότητα για αποσύνδεση και αλλαγή του κωδικού πρόσβασης, διαφορετικά προτρέπεται να κάνει είσοδο ή εγγραφή.



Εικόνα 46. Στιγμιότυπα Διεπαφής Portal

Ο χρήστης επιλέγοντας την επιλογή Εισόδου/Εγγραφής πλέον έχει τη δυνατότητα για δημιουργία νέου λογαριασμού ή να κάνει είσοδο με τα στοιχεία που είχε ήδη

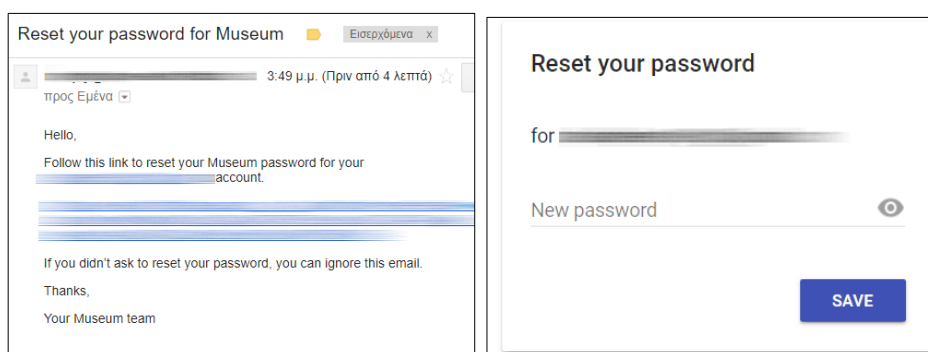
δημιουργήσει. Κατά την εγγραφή είναι απαραίτητα προς συμπλήρωση το όνομα του, το ηλεκτρονικό ταχυδρομείο και ο κωδικός ασφαλείας. Σε περίπτωση όπου δεν συμπληρώσει κάποιο από τα πεδία και πατήσει είτε να κάνει είσοδο είτε εγγραφή ενημερώνεται ότι δεν



Εικόνα 47. Στιγμιότυπα Διεπαφής Portal Μενού

έχει συμπληρώσει όλα τα στοιχεία. Αν ο χρήστης έχει κάνει ξανά εγγραφή με το ίδιο ηλεκτρονικό ταχυδρομείο γίνεται έλεγχος και ενημερώνεται με κατάλληλο μήνυμα λάθους και στη περίπτωση όπου έχει πληκτρολογήσει λάθος το κωδικό επίσης ενημερώνεται.

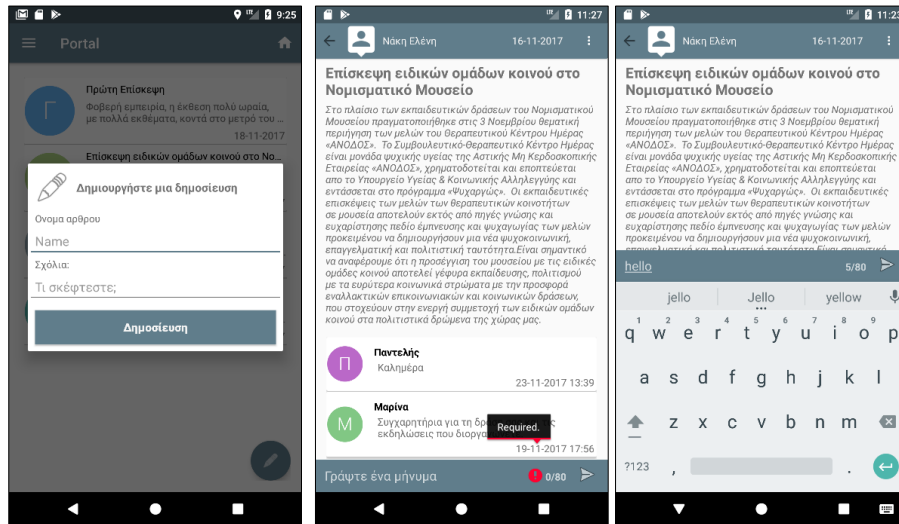
Επιπλέον, στη διεπαφή εισόδου υπάρχει δυνατότητα για ανάκτηση του κωδικού ασφαλείας. Ο χρήστης συμπληρώνει το ηλεκτρονικό ταχυδρομείο που είχε κάνει την εγγραφή στη φόρμα και στο ηλεκτρονικό ταχυδρομείο του λαμβάνει ένα email που περιέχει το σύνδεσμο από όπου μπορεί να κάνει την αλλαγή του κωδικού με καινούργιο. Πατώντας στο σύνδεσμο του ζητείται ο νέος κωδικός και πληκτρολογώντας το νέο κωδικό έχει ολοκληρωθεί η αλλαγή.



Εικόνα 48. Στιγμιότυπα Αλλαγής Κωδικού

Έχοντας ολοκληρώσει την εγγραφή πλέον έχει τη δυνατότητα να πραγματοποιήσει νέα ανάρτηση. Όπως βλέπουμε στη παρακάτω εικόνα, ζητείται από το χρήστη ο τίτλος της ανάρτησης και η περιγραφή. Σε αυτό το σημείο εάν παραλείψει κάποιο από τα δύο πεδία κατά τη συμπλήρωση θα ενημερωθεί να συμπληρώσει το κενό πεδίο. Στην προσθήκη των σχολίων ο χρήστης μπορεί να ανεβάσει μέχρι ογδόντα χαρακτήρες. Έχει θεσπιστεί το όριο αυτό ώστε να μην καταπονηθεί η βάση δεδομένων της Firebase με πολλά στοιχεία,

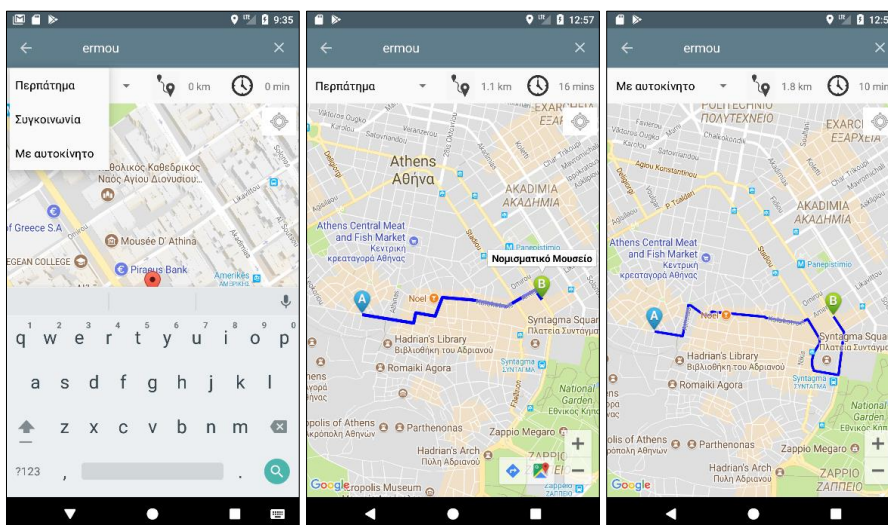
δεδομένου ότι χρησιμοποιείται η δωρεάν έκδοσή της Firebase, όπου παρέχει μέγιστο αποθηκευτικό χώρο 1GB, μέγιστο αριθμό χρηστών ταυτόχρονα 100 χρήστες και 10GB δεδομένων που μπορούν να γίνουν λήψη.



Εικόνα 49. Δημιουργία ανάρτησης και σχόλιων

### 6.1.12 Διεπαφή Χάρτη

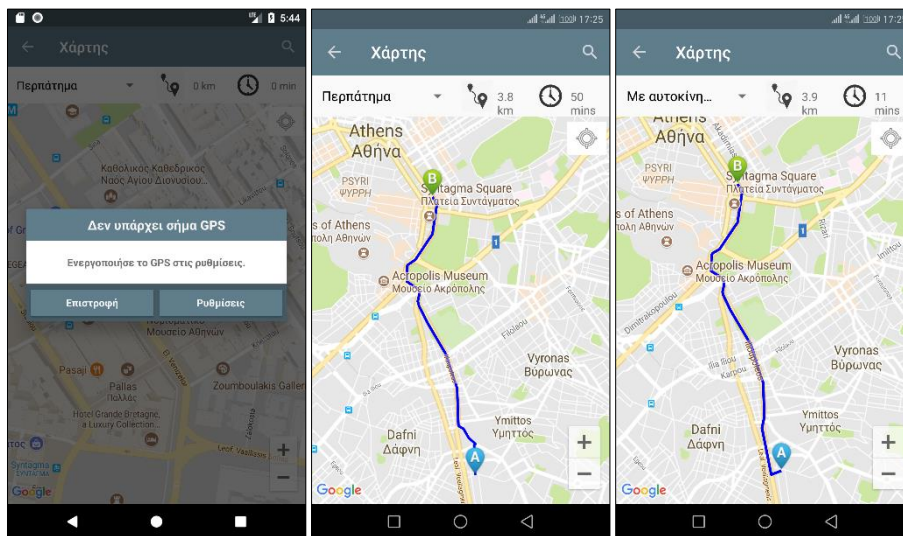
Στην εφαρμογή έχει προστεθεί Google Map προκειμένου ο χρήστης να μπορεί να βρίσκει την τοποθεσία του μουσείου και να λαμβάνει τις κατάλληλες οδηγίες ώστε να κατευθυνθεί προς το μουσείο. Στη γραμμή εργαλείων μέσω του πλήκτρου της αναζήτησης, ο χρήστης μπορεί να δώσει τη διεύθυνση που επιθυμεί σαν αρχικό σημείο. Τα δεδομένα που δίνει ο χρήστης μέσω του Google Geocoder μεταφράζονται σε συνταγμένες, ώστε να τοποθετηθεί η πινέζα στο χάρτη. Μέσω του Google Maps Directions API επιστρέφεται το δρομολόγιο που θα ακολουθήσει. Πριν στείλουμε το αίτημα με το προορισμό και το αρχικό σημείο, από προ-επιλογή η διαδρομή υπολογίζεται ότι χρήστης περπατάει προς το μουσείο. Μέσω ενός drop down list ο χρήστης μπορεί να επιλέξει ανάμεσα στις επιλογές που παρέχονται για τη διαδρομή. Μπορεί να επιλέξει εάν θα πάει με τα πόδια, μέσω συγκοινωνίας ή με αυτοκίνητο.



Εικόνα 50. Αναζήτηση μέσω διεύθυνσης

Επίσης, ο χρήστης πατώντας το πλήκτρο του GPS που βρίσκεται στην διεπαφή. Εάν δεν έχει ενεργοποιημένο το GPS θα εμφανιστεί το κατάλληλο μήνυμα λάθους, όπου ενημερώνεται ο χρήστης ότι δεν έχει ενεργοποιημένο το GPS στη συσκευή του και τον προτρέπει να πάει στις ρυθμίσεις της συσκευής του να το ενεργοποιήσει, εφόσον είναι ενεργοποιημένο το GPS πλέον υπολογίζει τη τοποθεσία του χρήστη. Ο υπολογισμός της θέσης πραγματοποιείται με δύο τρόπους, μέσω δορυφόρων και μέσω των δεδομένων της κινητής τηλεφωνίας. Οι συντεταγμένες λαμβάνονται βάσει της μεθόδου που ολοκληρώθηκε πρώτη ή εάν δεν έχει ενεργοποιημένη τη σύνδεση στο διαδίκτυο θα την υπολογίσει βάσει του στίγματος που θα πάρει από τους δορυφόρους όταν κλειδώσουν.

Πλέον με το GPS να έχει λάβει τις συντεταγμένες, το Geocoder έχει μεταφράσει τη διεύθυνση εφόσον έχει πληκτρολογηθεί. Υπολογίζεται η κατάλληλη διαδρομή μέσω του Google Map API και η σχεδίαση της διαδρομής ολοκληρώνεται από την εφαρμογή. Ταυτόχρονα εμφανίζονται στο πάνω μέρος της οθόνης η απόσταση σε χιλιόμετρα που χρειάζεται ώστε να φτάσει στο Νομισματικό Μουσείο και η ώρα που θα χρειαστεί. Οι πληροφορίες αυτές παρέχονται από το Google API. Με τη διαδρομή ολοκληρωμένη ο χρήστης μπορεί να αλλάξει το τρόπο που θα κατευθυνθεί προς το μουσείο και να επιλέξει για παράδειγμα την επιλογή του αυτοκίνητου, χωρίς να χρειαστεί να δώσει ξανά το αρχικό σημείο. Τέλος, για λόγους εξοικονόμησης της μπαταρίας των συσκευών το GPS είναι σε λειτουργία μέχρι να βρει το στίγμα της συσκευής.



Εικόνα 51. Αναζήτηση μέσω GPS

## 6.2 Η διεπαφή του Google Firebase

Προκειμένου να παρέχονται υπηρεσίες και λειτουργίες Cloud στην διπλωματική εργασία κρίθηκε απαραίτητο να αξιοποιηθεί η πλατφόρμα Firebase, της οποίας θα κάνουμε μια μικρή παρουσίαση παρακάτω και τις λειτουργίες που παρέχει στο διαχειριστή. Η διαχείριση των λειτουργιών γίνεται μέσω περιηγητή και κατάλληλων διεπαφών που παρέχει η Firebase.

### 6.2.1 Authentication

Η δημιουργία λογαριασμού που αναφέρθηκε στη διεπαφή Portal διαχειρίζεται μέσω της Firebase Authentication. Ο προγραμματιστής μέσω της διεπαφής έχει τη δυνατότητα να επιλέξει τους τρόπους με τους οποίους θα μπορεί ο χρήστης να δημιουργήσει λογαριασμό

όπως φαίνεται και στην εικόνα 50. Στην παρούσα μεταπτυχιακή έχει οριστεί να γίνεται μέσω Email/Password, η προσθήκη χρηστών γίνεται είτε μέσω της εφαρμογής είτε

Provider	Status
Email/Password	Enabled
Phone	Disabled
Google	Disabled
Facebook	Disabled
Twitter	Disabled
GitHub	Disabled
Anonymous	Disabled

Εικόνα 52. Επιλογές Δημιουργίας Λογαριασμού

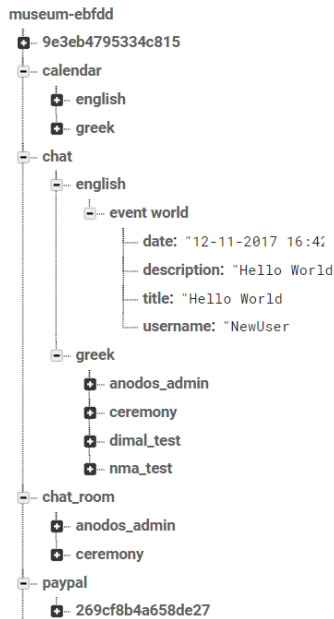
μέσω του διαχειριστή από τη διεπαφή της Firebase. Μέσω της διεπαφής αυτής μπορεί να διαχειριστεί τους λογαριασμούς των χρηστών ως προς τη διαγραφή/απενεργοποίηση λογαριασμού και την επαναφορά του κωδικού πρόσβασης αποστέλλοντας το κατάλληλο email. Επίσης ενημερώνει το διαχειριστή για την χρήση που γίνεται στο διακομιστή ώστε να μην υπερβεί το όριο χρήσης ανάλογα με το πλάνο που χρησιμοποιείται.

Identifier	Providers	Created	Signed In	User UID ↑
ntina@gmail.com	✉	10 Nov 2017	10 Nov 2017	C8luROZomoU2s01SxtK7LJZXZ452
	✉	12 Nov 2017	30 Dec 2017	
stel@gmail.com	✉	10 Nov 2017	10 Nov 2017	XmhjqrFFgQOWX868o44YKx0cif72
pan@gmail.com	✉	10 Nov 2017	10 Nov 2017	efiZwdMVjxMAFGe0alf6Mx6QALu1
geo@gmail.com	✉	9 Nov 2017	9 Nov 2017	jmvIEUzgtxez00JgcJJ1Zh8svbn1
pad@gmail.com	✉	9 Nov 2017	30 Dec 2017	zDsLx40YCjcu627J50ah3cnN04j1

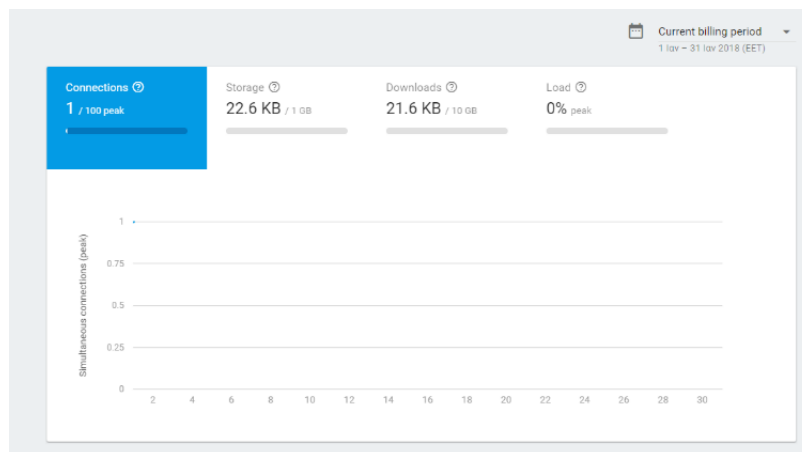
Εικόνα 53. Διαχείριση Λογαριασμών Χρηστών

## 6.2.2 Database

Η Cloud βάση δεδομένων της Firebase διαχειρίζεται τα δεδομένα που είναι απαραίτητα να βρίσκονται σε απομακρυσμένη βάση δεδομένων και δίνει τη δυνατότητα στο προγραμματιστή να προσθέτει επιπλέον λειτουργίες στην εφαρμογή μειώνοντας τη πολυπλοκότητα. Ο διαχειριστής από τη πλευρά του μπορεί να προσθέσει νέες εκδηλώσεις και αναρτήσεις στο Portal σε πραγματικό χρόνο, να τροποποιήσει ήδη υπάρχουσες εκδηλώσεις και απρεπή σχόλια που έχουν αναρτηθεί. Επίσης έχει δυνατότητα εφόσον αναβαθμίσει το πακέτο χρήσης για αυτόματη δημιουργία αντιγράφων ασφαλείας της βάσης δεδομένων ώστε σε περίπτωση απώλειας δεδομένων, να επαναφέρουν την προηγούμενη έκδοση της. Επίσης λαμβάνει ενημέρωση για το όριο χρήσης της βάσης δεδομένων.



Εικόνα 54. Βάση Δεδομένων Firebase



Εικόνα 55. Μηνιαίο Όριο Χρήσης της Βάσης Δεδομένων.

### 6.2.3 Storage

Μέσω του Storage η Firebase παρέχει τη δυνατότητα για αποστολή και λήψη αρχείων στις εφαρμογές. Στη παρούσα διπλωματική διατριβή το storage φιλοξενεί τα αρχεία ήχου της ηχητικής ξενάγησης και τη νεότερη έκδοση της τοπικής βάσης δεδομένων. Ο διαχειριστής μπορεί να ανεβάσει ή να κατεβάσει οποιοδήποτε αρχείο απαιτείται και να διαχειριστεί ορθά το διαθέσιμο χώρο που παρέχεται στο δωρεάν πακέτο χρήσης.

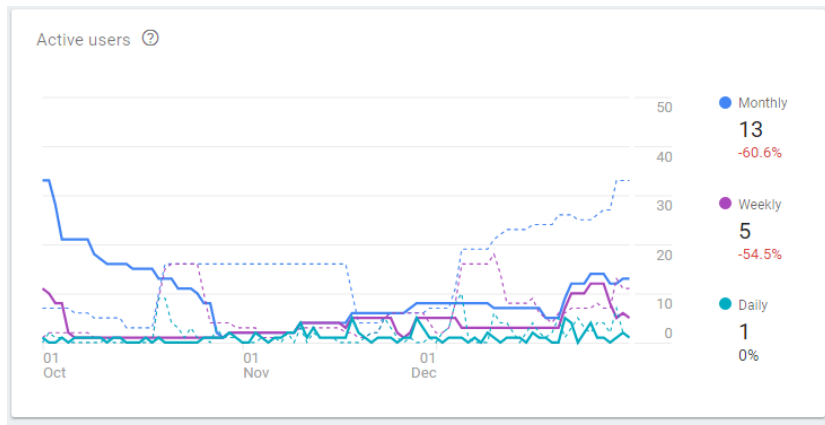
	Name	Size	Type	Last modified
<input type="checkbox"/>	database/	–	Folder	–
<input type="checkbox"/>	music_files/	–	Folder	–
<input type="checkbox"/>	music_files_gr/	–	Folder	–

Εικόνα 56. Firebase Storage



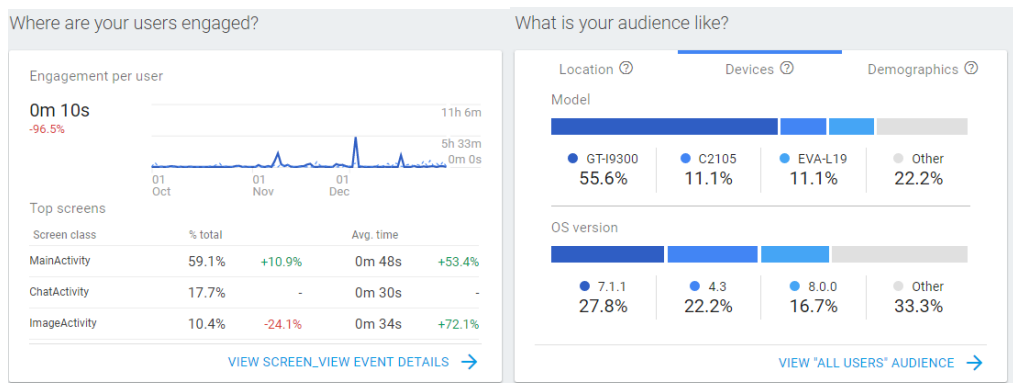
### 6.2.4 Analytics

Επιπλέον, ενσωματώνοντας τη Firebase σε μια εφαρμογή από προεπιλογή το SDK της καταγραφεί στοιχεία και ενέργειες των χρηστών ως προς τη χρήση της εφαρμογής. Όπως φαίνεται και στις παρακάτω εικόνες ο διαχειριστής ενημερώνεται για τους ενεργούς χρήστες, το χρόνο που αφιερώνουν περισσότερο σε μια διεπαφή, την έκδοση λογισμικού της συσκευής του και άλλες πληροφορίες. Μέσω αυτών ο διαχειριστής έχει την επιλογή για εξαγωγή των δεδομένων σε υπολογιστικά φύλλα ( excel ) ώστε μέσω των κατάλληλων



Εικόνα 57. Ενεργοί Χρήστες το διάστημα 1/10-31/12

ανθρώπων να αξιοποιηθούν με το κατάλληλο δυνατό τρόπο, γνωρίζοντας καλύτερα της συνήθειες των χρηστών. Επιπλέον ενημερώνει για τυχόν προβλήματα που προέκυψαν στην εφαρμογή(bugs). Δεν έχει χρησιμοποιηθεί αυτή η επιλογή στη διατριβή αυτή. Για τη αποσφαλμάτωση έχει αξιοποιηθεί το Fabric Crashlytics.

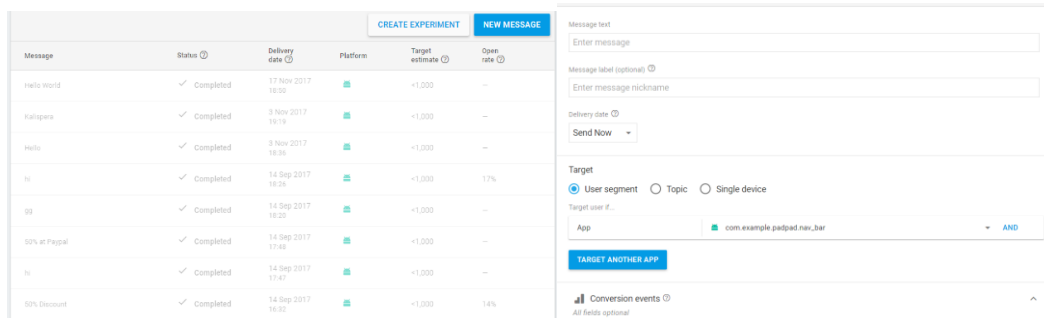


Εικόνα 58. Πληροφορίες σχετικά με την εφαρμογή.

### 6.2.5 Ειδοποιήσεις και Remote Config

Κλείνοντας την αναφορά στη Firebase, θα αναφερθούμε στις δύο εναπομείνασες λειτουργίες που αξιοποιήθηκαν στην εφαρμογή. Μέσω του Firebase Notification δίνεται η δυνατότητα για αποστολή ειδοποιήσεων στους χρήστες της εφαρμογής που το έχουν επιλέξει. Αυτό μπορεί να αφορά κάποια προσφορά στην τιμή του εισιτηρίου, ακόμα και κάποια εκδήλωση που θα γίνει στο επόμενο διάστημα.

Μέσω της διεπαφής αυτής ο διαχειριστής μπορεί να δημιουργήσει μια νέα ειδοποίηση. Κατά τη δημιουργία απαιτείται ο τίτλος και η περιγραφή της. Επιπλέον, οι ειδοποιήσεις μπορεί να γίνουν στοχευμένα ως προς τη συμπεριφορά των χρηστών και τα δημογραφικά χαρακτηριστικά τους. Για παράδειγμα μια μικρή μείωση στη τιμή μπορεί να γίνει γνωστό τους Έλληνες επισκέπτες τις εορταστικές ημέρες των Χριστουγέννων ώστε να βελτιωθεί η επισκεψιμότητα του μουσείου.



Message	Status	Delivery date	Platform	Target estimate	Open rate
Hello world	Completed	17 Nov 2017 18:00		+1,000	--
Kaligara	Completed	9 Nov 2017 18:19		+1,000	--
Hello	Completed	9 Nov 2017 18:36		+1,000	--
Hi	Completed	14 Sep 2017 18:24		+1,000	17%
Hi	Completed	14 Sep 2017 18:25		+1,000	--
50% et Paypal	Completed	14 Sep 2017 17:44		+1,000	--
Hi	Completed	14 Sep 2017 18:47		+1,000	--
50% Discount	Completed	14 Sep 2017 18:01		+1,000	14%

CREATE EXPERIMENT NEW MESSAGE

Message text

Enter message

Message label (optional)

Enter message nickname

Delivery date

Send Now

Target

User segment Topic Single device

Target user if...

App com.example.padpad.nav\_bar AND

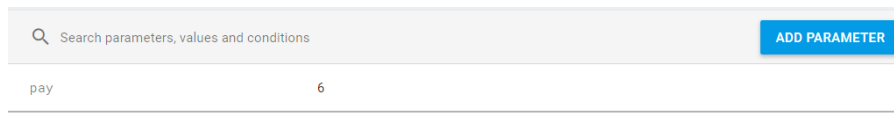
TARGET ANOTHER APP

Conversion events

All fields optional

Εικόνα 59. Δημιουργία και Διαχείριση Notification

Τέλος, μέσω της διεπαφής Remote Config ο διαχειριστής μπορεί να ενημερώσει τη τιμή αγοράς της ηχητικής ξενάγησης στην εφαρμογή άμεσα εφόσον το επιθυμεί το Νομισματικό Μουσείο.



Search parameters, values and conditions

ADD PARAMETER

pay 6

Εικόνα 60. Προσθήκη Παραμέτρων στο Remote Config

## 7. Αξιολόγηση της Εφαρμογής

Στα πλαίσια της παρούσας μεταπτυχιακής διατριβής δημιουργήθηκε ένα ερωτηματολόγιο, μέσω του οποίου θα πραγματοποιήσουμε την αξιολόγηση της εφαρμογής, για να διαπιστώσουμε κατά πόσο καλύπτει τις ανάγκες των χρηστών από πλευράς ευχρηστίας και λειτουργικότητας. Για την αξιολόγηση της εργασίας, έγινε χρήση ερωτηματολογίων χρηστικότητας (usability questionnaire), τα οποία στάλθηκαν στους χρήστες μέσω ηλεκτρονικού ταχυδρομείου και αξιοποιήθηκε η πλατφόρμα Google Forms, όπου αναρτήθηκε το ερωτηματολόγιο.

Στη διαδικασία αξιολόγησης χρηστικότητας συμμετείχαν εικοσιπέντε χρήστες με ηλικίες από 18 έως 35 χρόνων, από διαφορετικούς επαγγελματικούς χώρους, οι οποίοι έχουν διαφορετική εξοικείωση με ψηφιακές εφαρμογές και ήταν τουλάχιστον απόφοιτοι λυκείου. Προτού αποσταλεί το ερωτηματολόγιο, οι χρήστες κατέβασαν την εφαρμογή στις συσκευές τους ώστε να διαπιστώσουμε και τη σταθερότητα της εφαρμογής. Μέσω του Fabric Crashlytics θα εξάγουμε τα κατάλληλα συμπεράσματα. Το ερωτηματολόγιο περιλαμβάνει 26 ερωτήσεις ανοιχτού και κλειστού τύπου, χωρισμένες σε τέσσερις ενότητες. Οι κατηγορίες είναι οι ακόλουθες:

- Δημογραφικές πληροφορίες σχετικά με τους χρήστες που πήραν μέρος στην αξιολόγηση.
- Ερωτήσεις σχετικά με τις συνήθειες των χρηστών και το βαθμό εξοικείωσης τους με τα έξυπνα τηλέφωνα.
- Μέτρηση της ευχρηστίας της εφαρμογής.
- Ερωτήσεις που αφορούν τη λειτουργικότητα της εφαρμογής.

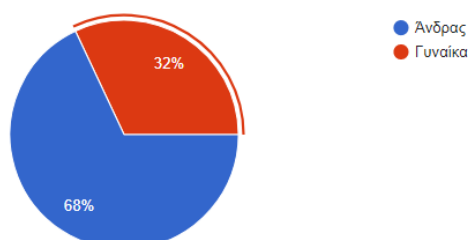
Οι ενότητες αυτές πέραν από την μέτρηση της ικανοποίησης των χρηστών από τις λειτουργίες της εφαρμογής, περιέχουν ερωτήσεις, οι οποίες έχουν σκοπό να ανακαλύψουν τυχόν πρόσθετες ανάγκες των χρηστών, οι οποίες θα ήταν καλό να προστεθούν σε επόμενες εκδόσεις της εφαρμογής.

### 7.1 Αποτελέσματα ερωτηματολογίου

Στο πλαίσιο της αξιολόγησης, τα ερωτηματολόγια στάλθηκαν μέσω του ηλεκτρονικού ταχυδρομείου όπως αναφέραμε και απαντήθηκαν από 25 χρήστες, στο διάστημα μεταξύ 1 Δεκεμβρίου 2017 και 10 Ιανουαρίου του 2018.

#### 1. Φύλο

25 responses



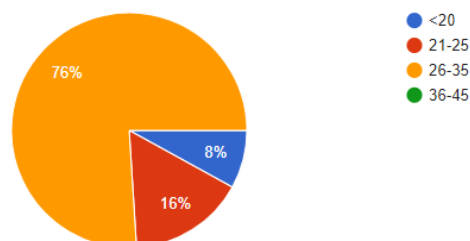
Διάγραμμα 15. Κατανομή δείγματος βάση το φύλο

Από το σύνολο των ατόμων που απάντησαν το 68% ήταν άντρες και το 32% γυναίκες.

Οι ηλικίες κυμάνθηκαν μεταξύ 18 και 35 χρονών. Όπως παρατηρούμε στο παρακάτω γράφημα το μεγαλύτερο ποσοστό του δείγματος ήταν μεταξύ 26 και 35 χρονών σε ποσοστό 76%, το 16% μεταξύ 21-25 και μόλις το 8% μικρότερο των 20 ετών.

## 2. Ηλικία

25 responses

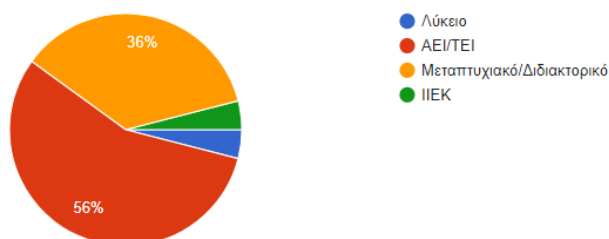


Διάγραμμα 16. Κατανομή δείγματος με βάση την ηλικία.

Όσον αφορά το μορφωτικό επίπεδο η πλειοψηφία των ατόμων έχουν τελειώσει την τριτοβάθμια εκπαίδευση. Συγκεκριμένα δύο άτομα έχει τελειώσει Ιδιωτικό Εκπαιδευτήριο Κατάρτισης και Γενικό Λύκειο (ποσοστό 8%), 9 άτομα είναι κάτοχοι μεταπτυχιακού τίτλου (ποσοστό 36%) και 14 άτομα είναι πτυχιούχοι ή σπουδαστές ανώτατων σχολών (ΑΕΙ – ΤΕΙ) (ποσοστό 56%).

## 3.Μορφωτικό επίπεδο

25 responses

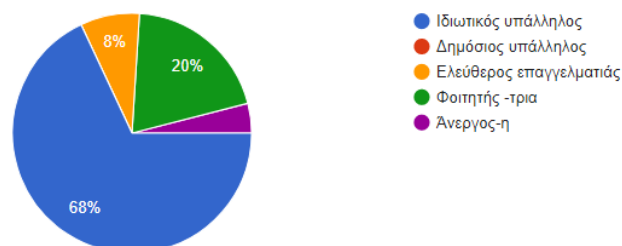


Διάγραμμα 17. Μορφωτικό επίπεδο.

Επίσης από το σύνολο των ατόμων σχετικά με την θέση εργασίας όπου απασχολούνται, η πλειοψηφία των ατόμων είναι ιδιωτικοί υπάλληλοι. Συγκεκριμένα 1 άτομο είναι άνεργο (ποσοστό 1%), 17 άτομα είναι ιδιωτικοί υπάλληλοι (ποσοστό 68%), 2 άτομα είναι ελεύθεροι επαγγελματίες (ποσοστό 8%) και 5 άτομα είναι φοιτητές (ποσοστό 20%).

## 4.Επάγγελμα

25 responses



Διάγραμμα 18. Εργασιακή απασχόληση.

Στα επόμενα δύο γραφήματα παρατηρούμε οι χρήστες διαθέτουν έξυπνο τηλέφωνο και γνωρίζουν σχετικά με τις εφαρμογές που έχουν εγκατεστημένες στις συσκευές τους.

Γνωρίζετε τις εφαρμογές (application) που υποστηρίζουν τα έξυπνα β. Διαθέτετε έξυπνο κινητό (smartphone);  
κινητά (smartphone);

25 responses

25 responses

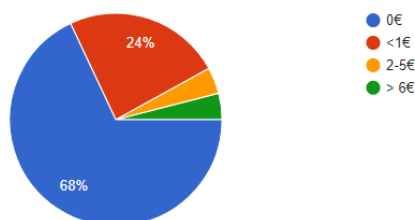


Διάγραμμα 19. Ερωτήσεις 5 και 6.

Στην επόμενη ερώτηση, οι ερωτηθέντες απάντησαν σχετικά με το ποσό που ξοδεύουν μηνιαία για αγορά υπηρεσιών που χρησιμοποιούν στις εφαρμογές τους. Η πλειοψηφία απάντησε ότι δεν ξοδεύει χρήματα για αγορά υπηρεσιών (ποσοστό 68%), 6 άτομα απάντησαν ότι ξοδεύουν 1€ μηνιαίως (ποσοστό 24%). Τέλος 1 χρήστης ξοδεύει 2-5€ και ένας ακόμα περισσότερα από 6€ (συνολικό ποσοστό 8%). Επιπλέον παρουσιάζουμε τις συνήθειες των ερωτηθέντων σχετικά με τη αξιοποίηση των συσκευών τους.

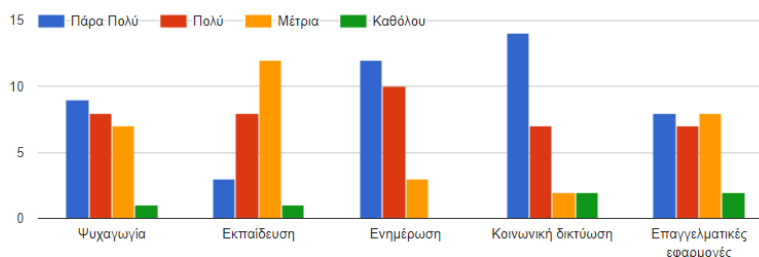
7. Τι ποσό πληρώνετε κατά μέσο όρο το μήνα για τις υπηρεσίες των εφαρμογών σας;

25 responses



Διάγραμμα 20. Μηνιαίες αγορές των χρηστών.

8. Στο σύνολο αξιολογήστε ποιές εργασίες κάνετε κατά βάση περισσότερο με τις φορητές συσκευές;



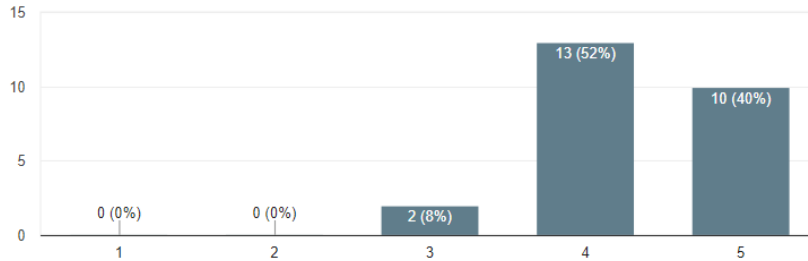
Διάγραμμα 21. Συνήθειες των χρηστών.

Στο παραπάνω γράφημα παρατηρούμε ότι οι χρήστες αξιοποιούν τις έξυπνες συσκευές τους περισσότερο για τα κοινωνικά μέσα, στη συνέχεια για ενημερωτικές εφαρμογές, και για επαγγελματικούς σκοπούς και, τέλος, έρχεται η εκπαίδευση. Κλείνοντας τη δεύτερη ενότητα του ερωτηματολογίου παρατηρούμε ότι το 92% έχει μεγάλη έως τέλεια εξοικείωση με το λειτουργικό σύστημα Android, ενώ μόλις το 8% δηλώνει ότι έχει μέτρια τριβή. Αυτό

Θεωρείται σημαντικό για την έρευνά μας διότι θα μπορέσουν να κατανοήσουν περισσότερο τις λειτουργίες της εφαρμογής.

### 9. Βαθμός Εξοικείωσης με το λειτουργικό σύστημα Android

25 responses

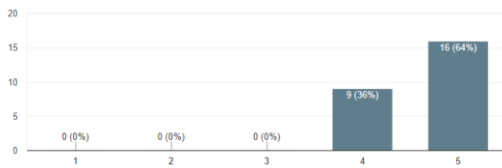


Διάγραμμα 22. Βαθμός εξοικείωσης με το Android.

Στη τρίτη ενότητα οι χρήστες θα ερωτηθούν σχετικά με τις ευχρηστία της εφαρμογής. Μέσω αυτών προσπαθήσαμε να εξάγουμε συμπεράσματα ως προς τη φιλικότητα (UI) και τη χρηστικότητα (UX) της εφαρμογής. Όπως παρατηρούμε στο επόμενο γράφημα οι χρήστες βρήκαν εύκολη τη περιήγηση στην εφαρμογή το 64% δεν αντιμετώπισε κάποια δυσκολία. Όπως και το περιεχόμενο της εφαρμογής είναι κατανοητό στο 60% των χρηστών και μόλις 2 άτομα (ποσοστό 8%) ήταν ουδέτεροι ως προς το περιεχόμενο.

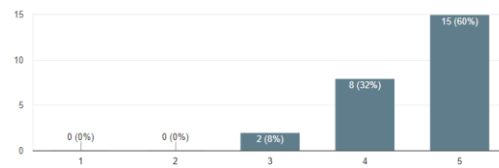
### 10. Πόσο εύκολη βρήκατε την πλοήγηση μέσα στην εφαρμογή;

25 responses



### 11. Είναι κατανοητό το περιεχόμενο της εφαρμογής;

25 responses

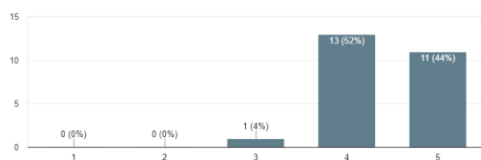


Διάγραμμα 23. Πλοήγηση και Περιεχόμενο της εφαρμογής.

Στις επόμενες ερωτήσεις οι χρήστες παρατηρούν συνέπεια στους όρους και στα σύμβολα που χρησιμοποιούνται στην εφαρμογή σε ποσοστό 96% και μόλις το 4% είναι ουδέτερο. Επίσης βρήκε χρήσιμο το περιεχόμενο το 92% και το 8% είτε δεν βρήκαν πρόβλημα είτε κάτι δε τους άρεσε.

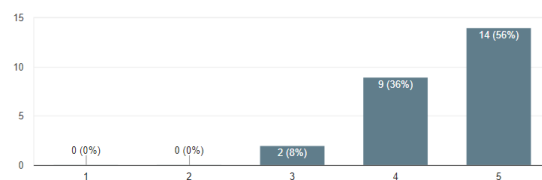
### 12. Υπάρχει συνέπεια στους όρους και τα σύμβολα που χρησιμοποιούνται;

25 responses



### 13. Πόσο χρήσιμο βρήκατε το περιεχόμενό;

25 responses

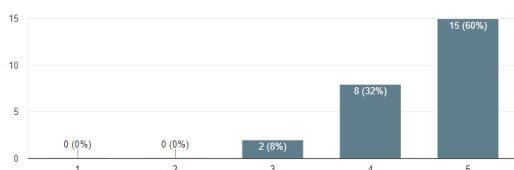


Διάγραμμα 24. Ερωτήσεις 12 και 13

Στα παρακάτω δύο γραφήματα παρατηρούμε ότι το σύνολο των ερωτηθέντων έκριναν ότι βρήκαν ικανοποιητική την ευκρίνεια των κειμένων σε ποσοστό 94% ενώ μόλις το 8% είναι ουδέτερο. Για την αισθητική εικόνα της εφαρμογής, τα σχόλια ήταν θετικά με το 96% να έχει πολύ καλή εικόνα για τη σχεδίαση της εφαρμογής και το 4% ουδέτερη.

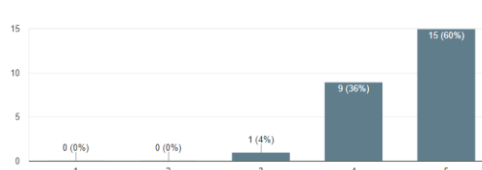
14. Είναι ικανοποιητική η ευκρίνεια των κειμένων και των εικόνων;

25 responses



15. Ποια είναι η γνώμη σας σχετικά με τη γενικότερη αισθητική της εφαρμογής;

25 responses

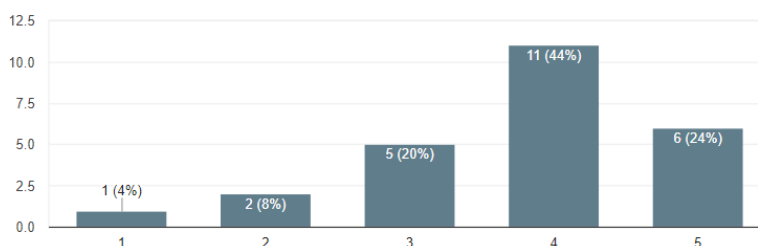


Διάγραμμα 25. Ερωτήσεις 14 και 15

Η ερώτηση αναφορικά με τη χρήση για παράδειγμα έντονων χαρακτήρων ώστε να τονίσουμε κάποιο κείμενο ή ένα κουμπί έχει ανάμεικτα αποτελέσματα. Συγκεκριμένα 6 άτομα πιστεύουν ότι χρειάζονται απόλυτα (ποσοστό 24%), 11 άτομα συμφωνούν απλά σε αυτή τη μέθοδο (ποσοστό 44%), 5 άτομα είναι ουδέτεροι (ποσοστό 20%), 2 άτομα διαφωνούν με αυτή τη τακτική (ποσοστό 8%) και ένα άτομο διαφωνεί απόλυτα (ποσοστό 4%) ώστε να δίνεται έμφαση σε σημαντικά σημεία-τμήματα του κειμένου..

16. Δίνεται έμφαση σε σημαντικά σημεία-τμήματα του κειμένου (π.χ. μέσω χρήσης έντονων χαρακτήρων);

25 responses

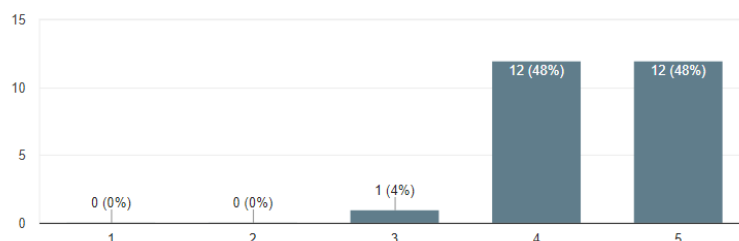


Διάγραμμα 26. Έμφαση στο κείμενο

Στην ευκολία αναζήτησης πληροφορίας στην εφαρμογή βλέπουμε ότι 12 άτομα (ποσοστό 48%) δεν αντιμετώπισε δυσκολία στην εύρεση ότι χρειαζόταν, 12 άτομα βρήκαν ικανοποιητική (ποσοστό 48%) την ευκολία χρήσης, τέλος μόλις 1 άτομο (ποσοστό 4%) είχε ουδέτερη στάση.

17. Πόσο εύκολα και γρήγορα μπορείτε να βρείτε στην εφαρμογή μία συγκεκριμένη πληροφορία που ζητάτε;

25 responses



Διάγραμμα 27. Ευκολία στην αναζήτηση πληροφορίας.

Συνοψίζοντας για την ευχρηστία της εφαρμογής, το σύνολο των χρηστών δεν αντιμετώπισε πρόβλημα με την εφαρμογή, την βρήκε ιδιαίτερα φιλική, ευχάριστη και ευέλικτη και τα ποσοστά από τις απαντήσεις το αποδεικνύουν. Προκειμένου να επιτευχθούν οι θετικές απαντήσεις των χρηστών σημαντικό ρόλο έπαιξε η φιλική σχεδίαση της εφαρμογής ως προς

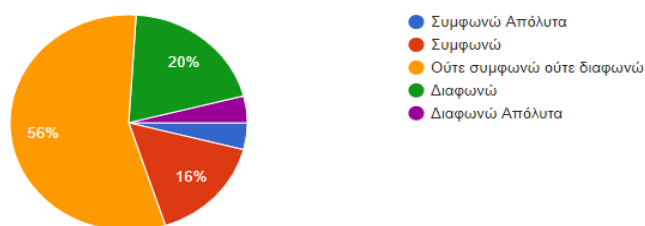
το χρήστη και μέσω αυτών ένας από τους σημαντικότερους σκοπούς κατά την ανάπτυξη εφαρμογών επετεύχθη.

Στην τέταρτη και τελευταία ενότητα, λαμβάνουμε ανάδραση από τους χρήστες ως προς τη λειτουργικότητα της εφαρμογής, τυχόν προβλήματα που αντιμετώπισαν, λειτουργίες που θα ήθελαν να προστεθούν στο μέλλον και κλείνοντας ερωτώνται εάν έχουν επισκεφτεί το νομισματικό μουσείο, ώστε να διαπιστωθεί το κατά πόσο θα βοηθήσει να εξάγουμε τα κατάλληλα συμπεράσματα.

Κατά τη σχεδίαση της εφαρμογής θεωρήθηκε απαραίτητη η λειτουργία της εφαρμογής χωρίς να απαιτείται η διαρκής σύνδεση στο διαδίκτυο (offline). Για αυτό το λόγο δημιουργήθηκε τοπική βάση στη συσκευή του χρήστη ώστε η εφαρμογή να λειτουργεί στο μουσείο ή οπουδήποτε αλλού χωρίς να είναι συνεχώς συνδεδεμένη στο διαδίκτυο. Όμως αυτό λόγω των πολλών εκθεμάτων απαιτεί μεγαλύτερο αποθηκευτικό χώρο στη συσκευή. Στο παραπάνω ερώτημα η πλειοψηφία των ατόμων ήταν ουδέτεροι (ποσοστό 56%), 5 άτομα διαφώνησαν για τη συνεχή σύνδεση στο διαδίκτυο (14 άτομα, ποσοστό 20%), 4 άτομα συμφωνούν με τη συνεχή σύνδεση (ποσοστό 16%). Τέλος το 4% συμφωνεί απόλυτα ενώ το εναπομείναν 4% διαφωνεί απόλυτα ως προς τη συνεχή συνδεσιμότητα στο ίντερνετ.

18. Θεωρείται απαραίτητη τη συνεχή σύνδεση στο διαδίκτυο για τη χρήση της εφαρμογής ;

25 responses

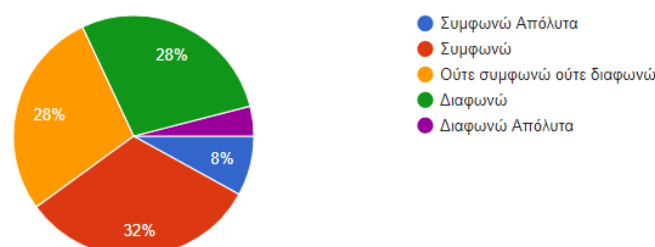


Διάγραμμα 28. Ανάγκη για διαρκή σύνδεση στο διαδίκτυο.

Στο ερώτημα εάν χρειάζονται περισσότερες λειτουργίες η εφαρμογή, 10 άτομα συμφωνούν (ποσοστό 40%) ότι χρειάζονται προσθήκες, 7 άτομα κρατούν ουδέτερη στάση (ποσοστό 28%) και 8 άτομα θεωρούν ότι δεν χρειάζονται περαιτέρω λειτουργίες (ποσοστό 32%).

19. Θεωρείται ότι υπάρχει ανάγκη για περαιτέρω λειτουργίες;

25 responses



Διάγραμμα 29. Ανάγκη για περαιτέρω λειτουργίες.

Στο portal της εφαρμογής όπου ο χρήστης μπορεί να ανταλλάξει απόψεις, να προσθέσει την εμπειρία του από τη ξενάγηση στο Νομισματικό και να συζητήσει θέματα που μπορούν να προσθέσουν οι διαχειριστές, απαιτείται πρώτα ο χρήστης να έχει κάνει εγγραφή.

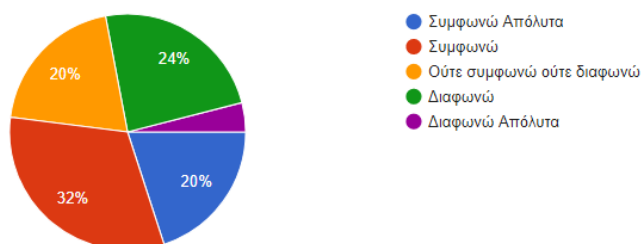


Στη παρακάτω πίττα ζητάμε την άποψη των χρηστών, ώστε να διαπιστώσουμε εάν θεωρούν απαραίτητο να βρίσκεται στην φόρμα που κάνουν είσοδο στην εφαρμογή, η δυνατότητα να κάνουν είσοδο μέσω ενός λογαριασμού που έχουν ήδη, για παράδειγμα να κάνουν είσοδο με το λογαριασμό που έχουν στο Facebook ή στο Google Mail τους.

Όπως παρατηρούμε 13 άτομα (ποσοστό 52%) θα χρησιμοποιούσαν τους ήδη υπάρχοντες λογαριασμούς να κάνουν είσοδο από το να δημιουργήσουν νέο λογαριασμό στην εφαρμογή, 5 άτομα είναι ουδέτεροι ως προς αυτή τη δυνατότητα (ποσοστό 20%) και οι τελευταίοι 7 ερωτηθέντες κρίνουν ότι δεν ενδιαφέρονται (ποσοστό 25%).

20. Θα θέλατε να κάνετε είσοδο με social account;

25 responses

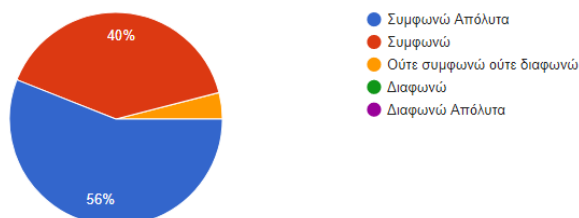


Διάγραμμα 30. Ανάγκη για είσοδο με social λογαριασμό.

Στο επόμενο γράφημα παρατηρούμε ότι οι ερωτηθέντες σε ποσοστό 96% θα χρησιμοποιούσαν την εφαρμογή στη ξενάγησή τους στο Νομισματικό μουσείο και 1 άτομο έχει ουδέτερη στάση(ποσοστό 4%).

21. Θα χρησιμοποιούσατε την εφαρμογή για την ξενάγησή σας στο μουσείο;

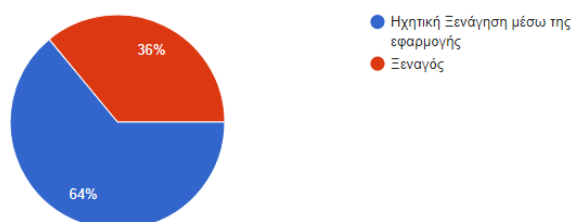
25 responses



Διάγραμμα 31. Χρήση της εφαρμογής στο Νομισματικό Μουσείο.

22. Για τη ξενάγησή στο μουσείο θα αγοράζατε την ηχητική ξενάγηση ή θα προτιμούσατε ξεναγό;

25 responses



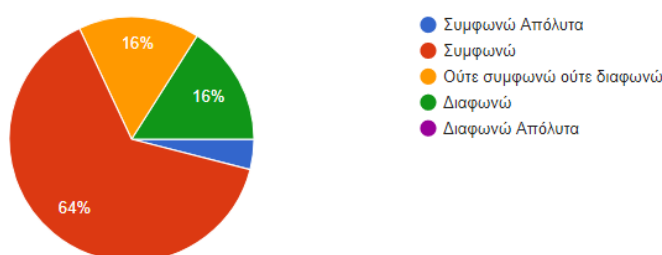
Διάγραμμα 32. Χρήση της ηχητικής ξενάγησης.

Στην ερώτηση εάν θα προτιμούσαν τη χρήση της έξυπνης συσκευής τους για την ηχητική ξενάγηση στο μουσείο ή την αξιοποίηση του ξεναγού, το 64% των ερωτηθέντων απάντησε θετικά, ενώ το 36% θα προτιμούσε τον ξεναγό.

Όπως αναφέρθηκε στην παρουσίαση της εφαρμογής, ο χρήστης διαθέτει στην εφαρμογή το Portal όπου μπορεί να ανταλλάξει μηνύματα με άλλους χρήστες. Μέσω της επόμενης ερώτησης θα προσπαθήσουμε να βγάλουμε συμπέρασμα εάν οι χρήστες θεωρούν το Portal απαραίτητο και εάν θα το αξιοποιούσαν.

23 Θα αξιοποιούσατε το portal για να συζητήσετε με άλλους επισκέπτες και να ρωτήσετε απορίες;

25 responses



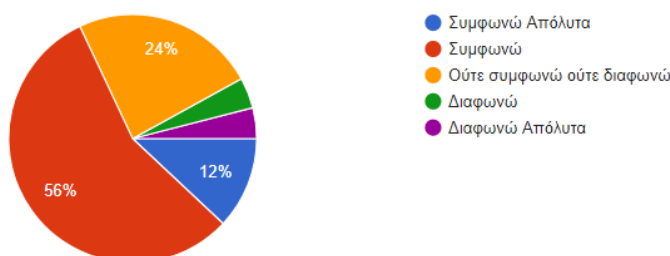
Διάγραμμα 33. Χρήση του Portal.

Όπως παρατηρούμε το 68% του συνόλου απάντησε θετικά στη χρήση του Portal, ενώ το 16% κράτησε ουδέτερη στάση και το 20% δεν θα το χρησιμοποιούσε. Δεδομένου του μικρού ποσοστού των αρνητικών απαντήσεων, μπορούμε να συμπεράνουμε ότι αυτή η λειτουργία αξιολογείται ως χρήσιμη.

Στην συνέχεια ερωτήθηκαν εάν η λήψη ειδοποιήσεων στη συσκευή τους θα τους ενδιέφερε. Κρίνοντας από τα αποτελέσματα το 68% των ατόμων έχει θετική εικόνα ως προς τη λειτουργία αυτή. Ενώ μόλις το 16% δεν θα το αξιοποιούσε. Τέλος, το 28% έχει ουδέτερη στάση.

24. Θα θέλατε να λαμβάνετε ειδοποιήσεις στη συσκευή σας σχετικά με νέες εκδηλώσεις του μουσείου;

25 responses

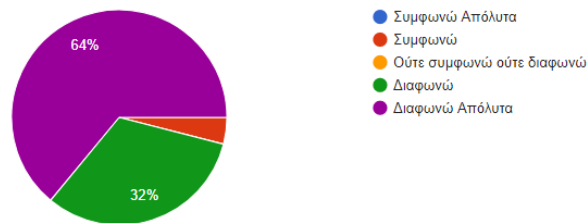


Διάγραμμα 34. Χρήση των ειδοποιήσεων.

Επίσης οι χρήστες ρωτήθηκαν εάν διαπίστωναν κάποιο πρόβλημα κατά την περιήγησή τους στην εφαρμογή. Όπως βλέπουμε στο παρακάτω διάγραμμα το 96% των χρηστών δεν διαπίστωσε κάποιο σημαντικό πρόβλημα στη χρήση της εφαρμογής, ενώ μόλις 1 χρήστης διαπίστωσε κάποια δυσλειτουργία.

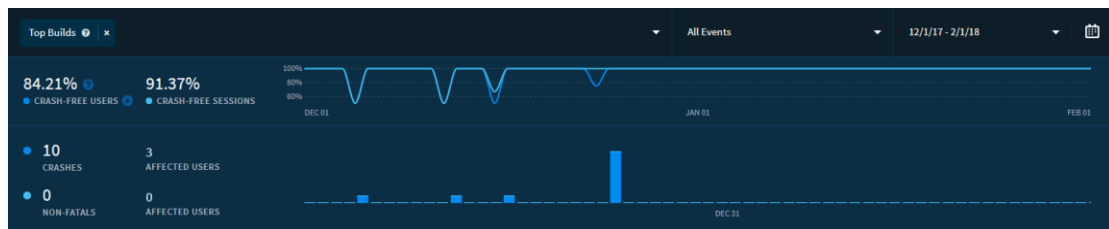
25. Παρουσιάστηκε κάποιο πρόβλημα κατά τη περιήγηση σας στην εφαρμογή;

25 responses



Διάγραμμα 35. Προβλήματα κατά την περιήγηση.

Όπως παρατηρούμε και στο Fabric Crashlytics που καταγράφει τα bugs και τις δυσλειτουργίες της εφαρμογής σε περιπτώσεις που σταματήσει να λειτουργεί ορθά, σε τρεις χρήστες παρουσιάστηκαν αστάθειες κατά τη περιήγηση τους στην εφαρμογή, επομένως καταλήγουμε στο γεγονός ότι ταυτίζεται το Fabric με τα αποτελέσματα των χρηστών από το ερωτηματολόγιο.

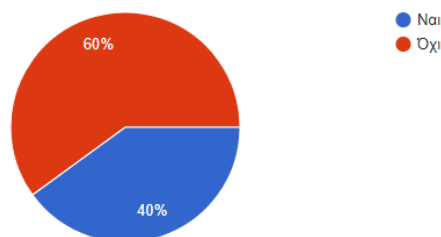


Διάγραμμα 36. Καταγραφές ασταθειών μέσω Fabric.

Κλείνοντας το ερωτηματολόγιο οι χρήστες απαντήσαν εάν έχουν επισκεφτεί το Νομισματικό Μουσείο Αθηνών ώστε τα αποτελέσματα της έρευνας να εγκυρότερα. Καθώς παρατηρούμε στο γράφημα ότι τα 15 άτομα της έρευνας(ποσοστό 60%) έχουν επισκεφτεί το μουσείο και τα 10 άτομα(40% ποσοστό) δεν το έχουν επισκεφτεί ακόμα.

26. Έχετε επισκεφτεί το νομισματικό μουσείο;

25 responses



Διάγραμμα 37. Επίσκεψη στο Νομισματικό Μουσείο.

## 8. Συμπεράσματα - Μελλοντικές Προτάσεις

Στο συγκεκριμένο κεφάλαιο θα παρουσιαστούν τα συμπεράσματα που προέκυψαν από την υλοποίηση της παρούσας μεταπτυχιακής εργασίας. Θα γίνει αναφορά για την απόκτηση της τεχνογνωσίας που αποκτήσαμε και θα προταθούν μελλοντικές επεκτάσεις για την υπάρχουσα εφαρμογή.

### 8.1 Συμπεράσματα

Η τεχνολογία εξελίσσεται με ραγδαίους ρυθμούς, και αυτό δίνει τη δυνατότητα σε ολοένα και περισσότερα ενσωματωμένα συστήματα να υλοποιούνται για την κάλυψη των αυξημένων αναγκών στην καθημερινότητα του ανθρώπου.

Ταυτόχρονα τα έξυπνα κινητά (smartphones) τα οποία έχουν γνωρίσει ταχεία ανάπτυξη τα τελευταία χρόνια και έχουν γίνει απαραίτητα προϊόντα στην καθημερινότητα του ανθρώπου. Τα έξυπνα κινητά προσφέρουν πάρα πολλές υπηρεσίες ώστε να ικανοποιούν τις ανάγκες μας, να διευκολύνουν την καθημερινότητα μας και να μας παρέχουν γνώσεις που είναι χρήσιμες και θα ήταν δύσκολο να βρεθούν στο παρελθόν.

Στη παρούσα μεταπτυχιακή εργασία σχεδιάστηκε και υλοποιήθηκε μια Android εφαρμογή για έξυπνα τηλέφωνα που αφορά το Νομισματικό Μουσείο Αθηνών. Δίνοντας τη δυνατότητα στους επισκέπτες να γνωρίσουν καλύτερα το μουσείο με τις πλέον ενδεδειγμένες τεχνολογίες που αξιοποιούν τα έξυπνα τηλέφωνα, να πετύχουμε τη βελτίωση της ξενάγησης μέσω των λειτουργιών που προσφέρει η εφαρμογή, να δώσουμε τη δυνατότητα στον επισκέπτη να είναι ενήμερος για νέα του μουσείου και οτιδήποτε αφορά το μουσείο σχετικά με το χώρο, τη ξενάγηση αλλά και γενικότερες πληροφορίες.

Το σημαντικότερο κομμάτι μιας εφαρμογής είναι η εμπειρία χρήσης του χρήστη (UX) και η σχεδίαση της (UI). Ακολουθώντας το Material Design της Google, το οπτικό αποτέλεσμα που δημιουργήθηκε είναι φιλικό προς το χρήστη, εύχρηστο, μέσω της απλότητας ο χρήστης δεν κουράζεται από περίπλοκα γραφικά και η ροή μεταξύ των διεπαφών είναι ξεκάθαρη. Οι διεπαφές έχουν όμορφη σχεδίαση και η εφαρμογή τρέχει εξίσου καλά σε συσκευές με χαμηλότερη επεξεργαστική ισχύ και δυνατότητες, δημιουργώντας ένα άμεσο περιβάλλον χρήσης.

Οι στόχοι που τέθηκαν στην αρχή της εργασίας, φαίνεται να επιτεύχθηκαν μέσω των δοκιμών και της έρευνας που διεξήχθη, από πλευράς λειτουργικότητας και σχεδίασης τα σχόλια ήταν θετικά στο μεγαλύτερο μέρος, η εφαρμογή δεν παρουσίασε αστάθειες στους χρήστες το οποίο επιβεβαιώνεται από το ερωτηματολόγιο και το Fabric.

### 8.2 Μελλοντικές Επεκτάσεις

Στην εποχή μας που ανάπτυξη των έξυπνων συσκευών και των υπηρεσιών που προσφέρουν είναι ραγδαία και συνεχής κάθε εφαρμογή πρέπει να αναβαθμίζεται και να βελτιώνεται. Έτσι και οι δυνατότητες της εφαρμογής που υλοποιήσαμε μπορούν να επεκταθούν μελλοντικά προς αρκετές κατευθύνσεις. Μερικές ιδέες για μελλοντική επέκταση της εφαρμογής παρουσιάζονται παρακάτω:

- Βελτιστοποίηση των διεπαφών ως προς το γραφικό περιβάλλον (UI) και την εμπειρία χρήσης (UX).
- Προσθήκη Swipe Gestures στο γκάλερι ώστε οι χρήστες να έχουν τη δυνατότητα να αλλάζουν τις εικόνες των εκθεμάτων μέσω των κατάλληλων κινήσεων στην οθόνη, αφαιρώντας τα εικονικά πλήκτρα πλοήγησης στην οθόνη.

- Προσθήκη ηχητικών αποσπασμάτων ανά έκθεμα, ώστε να ενημερώνεται πιο ολοκληρωμένα ο χρήστης.
- Δημιουργία τρισδιάστατης απεικόνισης του πλάνου ορόφων όπως στα μεγάλα μουσεία του εξωτερικού ή σε συνεργασία του Νομισματικού Μουσείου και της Google θα μπορούσε να δημιουργηθεί 3D απεικόνιση, στέλνοντας φωτογραφίες 360 μοιρών και προστίθενται στο Google Map, ακόμα μέσω του Indoor Map της Google στέλνοντας απλά τη κάτοψη του χώρου και των ορόφων, δημιουργούν τη 3D απεικόνιση και ανεβάζει στο Google Map. Οι δύο τελευταίες δυνατότητες έχουν αποτέλεσμα την αξιοποίηση εγγενών μεθόδων του λειτουργικού συστήματος το οποίο είναι διαθέσιμο στο μεγαλύτερο ποσοστό χρηστών Android.
- Δυνατότητα για αγορά των εισιτηρίων πρόσβασης στο μουσείο μέσω του PayPal και προσθήκη επιπρόσθετου τραπεζικού SDK για αγορές μέσω πιστωτικών/χρεωστικών καρτών.
- Στη διεπαφή των χαρτών ο χρήστης να έχει δυνατότητα να προσθέσει πινέζες στο χάρτη ώστε να δει τη διαδρομή από το σημείο που επέλεξε χωρίς να πληκτρολογεί τη διεύθυνση.
- Επίσης, στην ίδια διεπαφή, θα μπορούσε να προστεθεί η προβολή περισσότερων αποτελεσμάτων κατά τη πληκτρολόγηση των οδών αλλά και κατά τη πληκτρολόγηση να εμφανίζονται αποτελέσματα (AutoComplete).
- Στην ηχητική ξενάγηση, προσθήκη δυνατότητας για αλλαγή της γλώσσας εσωτερικά στη διεπαφή και προσθήκη πλήκτρων για να μπορεί ο χρήστης να προχωρά μπροστά ή πίσω στη ξενάγηση, όπως στις μοντέρνες εφαρμογές αναπαραγωγής εικόνας και ήχου.
- Επιπλέον, καθώς αποδείχτηκε μέσω του ερωτηματολογίου, οι χρήστες προτιμούν τη δυνατότητα για είσοδο σε μια πλατφόρμα μέσω ενός λογαριασμού ηλεκτρονικού ταχυδρομείου ή ενός λογαριασμού κοινωνικών δικτύων, θα μπορούσε να προστεθεί η δυνατότητα για εγγραφή μέσω του Google Mail και του Facebook/Twitter.
- Δυνατότητα για αλλαγή του κωδικού πρόσβασης εσωτερικά στην εφαρμογή.
- Θα μπορούσε σε συνεργασία με το Νομισματικό Μουσείο, να προστεθούν στα δωμάτια και στα εκθέματα Barcodes ώστε μέσω της εφαρμογής ο επισκέπτης να αναζητά άμεσα το έκθεμα/δωμάτιο και έχει πρόσβαση στις εκάστοτε πληροφορίες.
- Εναλλακτική μέθοδος του Barcode είναι η προσθήκη beacons. Ο επισκέπτης έχοντας ανοιχτή την εφαρμογή και το Bluetooth όταν πηγαίνει κοντά σε ένα έκθεμα, η εφαρμογή αναγνωρίζει το beacon και μπορεί να εμφανίσει στο χρήστη τη περιγραφή του εκθέματος και μπορεί να υπάρχει δυνατότητα για αναπαραγωγή της ξενάγησης.

## Βιβλιογραφία

1. *Android 6 for Programmers An App-Driven Approach 3rd Edition*, Alexander Wald
2. *Android Application Development Cookbook Second Edition*, Rick Boyer Kyle Mew
3. *The Busy Coder's Guide to Android Development*, Mark L. Murphy
4. *Android UI Design*, Francesco Azzola
5. *Android Programming for Beginners [Horton 2016-01-06]*, John Horton
6. *Creating Dynamic UI Android Fragments*, Jim Wilson
7. *Android Programming The Big Nerd Ranch Guide 2nd Edition*, Bill Phillips,Chris Stewart, Brian Hardy & Kristin Marsicano
8. *Creating an evaluation system for a mobile application design to enhance usability and aesthetic 2012*, Jiyoung Choi, Iowa State Univeristy
9. <https://developer.apple.com/support/app-store/>
10. <https://developer.android.com/about/dashboards/index.html>
11. <https://developer.android.com/index.html>
12. <https://firebase.google.com/docs/android/setup>
13. <https://developers.google.com/maps/documentation/android-api/start>
14. <https://docs.fabric.io/android/fabric/overview.html>
15. <https://developer.paypal.com/docs/api/rest-sdks/>
16. <https://medium.com/google-developers/building-a-responsive-ui-in-android-7dc7e4efcbb3>
17. <https://medium.com/upday-devs/android-architecture-patterns-part-1-model-view-controller-3baecef5f2b6>
18. <https://www.androidhive.info/2016/06/android-getting-started-firebase-simple-login-registration-auth/>
19. [https://en.wikipedia.org/wiki/Microsoft\\_Visio](https://en.wikipedia.org/wiki/Microsoft_Visio)
20. [https://en.wikipedia.org/wiki/Java\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))
21. <https://en.wikipedia.org/wiki/iOS>
22. [https://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))
23. <https://en.wikipedia.org/wiki/Smartphone>
24. [https://www.diffen.com/difference/Android\\_vs\\_iOS](https://www.diffen.com/difference/Android_vs_iOS)
25. <https://www.mobiloud.com/blog/native-web-or-hybrid-apps/>
26. <https://clearbridgemobile.com/mobile-app-development-native-vs-web-vs-hybrid/>
27. <https://code.facebook.com/posts/1189117404435352/react-native-for-android-how-we-built-the-first-cross-platform-react-native-app/>
28. <https://www.statista.com/statistics/266211/distribution-of-free-and-paid-android-apps/>
29. <http://www.javahelps.com/2015/10/android-60-runtime-permission-model.html>
30. <https://www.newsbeast.gr/technology/arthro/437660/sto-art-project-to-nomismatiko-mouseio-athinon>
31. <https://www.google.com/culturalinstitute/beta/partner/numismatic-museum>
32. <https://github.com/jgilfelt/android-sqlite-asset-helper>
33. <https://developers.google.com/beacons/get-started>
34. <http://www.enma.gr/>

## ΠΑΡΑΡΤΗΜΑ

### Επεξήγηση κώδικα

Στο παρόν κεφάλαιο θα αναφερθούμε στα πιο σημαντικά σημεία του κώδικα της εφαρμογής.

#### 1. Δημιουργία Base Activity και Fragment

Προκειμένου να εκμεταλλευτούμε την κληρονομικότητα που μας προσφέρει η Java ώστε ο κώδικας να είναι όσο γίνεται πιο καθαρός και ευανάγνωστος, δημιουργήθηκαν δύο αρχεία Base Java, ένα για τα activities και ένα για τα Fragments.

Η BaseActivity είναι μια abstract κλάση που έχει επεκταθεί με την AppCompatActivity, η οποία εμπεριέχει όλες τις απαραίτητες βιβλιοθήκες που απαιτούνται. Στη BaseActivity έχουμε προσθέσει μια abstract μέθοδο την initLayout όπου είναι απαραίτητη για κάθε activity όπου αρχικοποιούμε μεθόδους ή ότι στιδήποτε κρίνεται απαραίτητο στην εκάστοτε δραστηριότητα, και μια void μέθοδο την initToolbar που δέχεται σαν παράμετρο το Toolbar layout. Μέσω αυτής της μεθόδου αρχικοποιούμε το toolbar στο activity και ενεργοποιούμε τις λειτουργίες που επιθυμούμε.

```
public abstract class BaseActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

    public abstract void initLayout();

    protected void initToolbar(Toolbar toolbar) {
        if (toolbar == null) {
            return;
        }
        setSupportActionBar(toolbar);
        if (getSupportActionBar() != null) {
            getSupportActionBar().setDisplayHomeAsUpEnabled(isEnabled);
            getSupportActionBar().setHomeButtonEnabled(isEnabled);
            // getSupportActionBar().setHomeAsUpIndicator(R.mipmap.ic_arrow_back);
            getSupportActionBar().setDisplayHomeAsUpEnabled(false);
            getSupportActionBar().setIcon(null);
        }
    }

    protected void loadLanguage() {
        String lang = getSharedPreferences("prefs", MODE_PRIVATE)
            .getString(getString(R.string.language_pref), getResources().getString(R.string.language_pref));

        if (TextUtils.isEmpty(lang)) {
            lang = "gr";
        }
        Locale locale1 = new Locale(lang);
        Locale.setDefault(locale1);
        Configuration config = new Configuration();
        config.locale = locale1;
        getResources().updateConfiguration(config, getResources().getDisplayMetrics());
    }

    protected boolean checkNetwork() {
        boolean isOnline;
        ConnectivityManager = (ConnectivityManager) this.getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo = connectivityManager.getActiveNetworkInfo();
        if (networkInfo != null) {
            isOnline = networkInfo.isConnected();
        } else {
            isOnline = false;
        }
        return isOnline;
    }

    protected String loadLanguage() {
        String lang = getSharedPreferences("prefs", MODE_PRIVATE)
            .getString(getString(R.string.language_pref), getResources().getString(R.string.language_pref));

        if (TextUtils.isEmpty(lang)) {
            lang = "gr";
        }
        Locale locale1 = new Locale(lang);
        Locale.setDefault(locale1);
        Configuration config = new Configuration();
        config.locale = locale1;
        getResources().updateConfiguration(config, getResources().getDisplayMetrics());
        return lang;
    }

    protected void hideSoftKeyboard(View view) {
        if (view.requestFocus()) {
            InputMethodManager imm = (InputMethodManager)
                getSystemService(Context.INPUT_METHOD_SERVICE);
            if (imm != null) {
                imm.hideSoftInputFromWindow(view.getWindowToken(), 0);
            }
        }
    }
}
```

Επίσης, υπάρχουν δύο μέθοδοι σχετικά με τη γλώσσα που χρησιμοποιείται στην εφαρμογή. Η πρώτη μέθοδος είναι τύπου void ώστε να μας εμφανίσει τη γλώσσα που επιλέγει ο χρήστης και η δεύτερη επιστρέφει ποια γλώσσα έχει επιλέξει ο χρήστης ώστε να πραγματοποιούμε τους κατάλληλους ελέγχους. Μέσω της CheckNetwork() η οποία είναι τύπου Boolean και ελέγχουμε εάν έχει πρόσβαση στο διαδίκτυο η συσκευή. Τέλος μέσω της hideSoftKeyboard() κλείνουμε το πληκτρολόγιο της συσκευής όπου κρίνεται απαραίτητο.

Το BaseFragment ακολουθεί την ίδια λογική όπως το BaseActivity με τη διαφορά ότι επεκτείνεται με το Fragment, περιέχει σχεδόν τις ίδιες μεθόδους, οι δύο αυτές κλάσεις θα μπορούσαν να θεωρηθούν περιττές διότι εμπεριέχουν τις ίδιες μεθόδους, όμως ένα fragment και ένα activity διαφέρουν ώστε να είχαμε μια Base κλάση.

```
public class BaseFragment extends Fragment {
    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

    protected void loadLanguage() {
        String lang = getActivity().getSharedPreferences("prefs", MODE_PRIVATE)
            .getString(getString(R.string.language_pref), getResources().getString(R.string.language_pref));

        if (TextUtils.isEmpty(lang)) {
            lang = "gr";
        }
        Locale locale1 = new Locale(lang);
        Locale.setDefault(locale1);
        Configuration config = new Configuration();
        config.locale = locale1;
        getResources().updateConfiguration(config, getResources().getDisplayMetrics());
    }

    protected String loadLanguage() {
        String lang = getActivity().getSharedPreferences("prefs", MODE_PRIVATE)
            .getString(getString(R.string.language_pref), getResources().getString(R.string.language_pref));

        if (TextUtils.isEmpty(lang)) {
            lang = "gr";
        }
        Locale locale1 = new Locale(lang);
        Locale.setDefault(locale1);
        Configuration config = new Configuration();
        config.locale = locale1;
        getResources().updateConfiguration(config, getResources().getDisplayMetrics());
        return lang;
    }

    protected boolean checkNetwork() {
        boolean isOnline;
        ConnectivityManager = (ConnectivityManager) getActivity().getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo = connectivityManager.getActiveNetworkInfo();
        if (networkInfo != null) {
            isOnline = networkInfo.isConnected();
        } else {
            isOnline = false;
        }
        return isOnline;
    }
}
```

## 2. Έλεγχος Νεότερης Έκδοσης της Βάσης Δεδομένων.

Κατά την είσοδο του χρήστη στην εφαρμογή, εφόσον είναι συνδεδεμένη η συσκευή του στο διαδίκτυο, η εφαρμογή ελέγχει εάν υπάρχει νεότερη έκδοση της βάσης δεδομένων ώστε να γίνει ενημέρωση εφόσον υπάρχει. Αρχικά διαβάζουμε από τη τοπική βάση δεδομένων της συσκευής ένα timestamp που περιέχει την ημερομηνία έκδοσης της βάσης, στη συνέχεια μέσω κατάλληλου listener έχουμε πρόσβαση στα αρχεία της Firebase, πριν προβούμε στον έλεγχο, ελέγχουμε εάν υπάρχει στη συσκευή η βάση, εφόσον το αρχείο υπάρχει, κρατάμε το timestamp της βάσης δεδομένων που βρίσκεται στη Firebase. Έχοντας πλέον τις δύο ημερομηνίες να συγκρίνουμε, προχωράμε στον έλεγχο μέσω της συνάρτησης Compare ανάμεσα στις ημερομηνίες. Εάν το αποτέλεσμα είναι αρνητικό σημαίνει ότι η τοπική βάση είναι παλαιότερη και θα συνεχίσει στη λήψη της βάσης μέσω της μεθόδου DownloadDB(), πριν τη λήψη, αποθηκεύουμε προσωρινά τα αγαπημένα δωμάτια/εκθέματα του χρήστη και το κωδικό συναλλαγής εφόσον έχει προβεί σε αγορά της ηχητικής ξενάγησης, προκειμένου όταν ολοκληρωθεί η νέα βάση δεδομένων να περαστούν εκ νέου τα προσωπικά δεδομένα.



```

mStorageRef = FirebaseStorage.getInstance().getReferenceFromUrl("gs://museum-abfdd.appspot.com");
databaseRef = mStorageRef.child("database/" + "museumEN.db");
// progressDialog.setProgress(0);

databaseAccess.open();
version = databaseAccess.VersionApp();
databaseAccess.close();

databaseRef.getMetadata().addOnSuccessListener(new OnSuccessListener<StorageMetadata>() {
    @Override
    public void onSuccess(StorageMetadata storageMetadata) {
        //get the time that the file has uploaded to cloud
        time = storageMetadata.getCreationTimeMillis();
        SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy HH:mm:ss");
        cloud_time = sdf.format(new Date(time));

        //if there is an old database ,check for new update
        if (file_final.exists()) {
            Long temp2 = file_final.lastModified();
            SimpleDateFormat test = new SimpleDateFormat("dd-MM-yyyy HH:mm:ss");
            local_file = test.format(new Date(temp2));

            //compare the two times to find if there is any new update
            int compare = cloud_time.compareTo(version);

            //there is update in cloud
            if (compare < 0) {
                //copy the favourites from old database
                databaseAccess.open();
                rooms = databaseAccess.copyFavourites("rooms");
                exhibits = databaseAccess.copyFavourites("exhibits");
                paypal_id = databaseAccess.Paypal("read", null);
                databaseAccess.close();
                downloadDb();
                //there is not any new update
            }
        }
    }
});

```

Στη συνέχεια εμφανίζεται ένα μήνυμα (dialog) ότι ξεκινάει η λήψη του αρχείου, με το χρήστη να έχει τη δυνατότητα να το ακυρώσει και ενημερώνεται για το ποσοστό που απομένει ώστε να ολοκληρωθεί η λήψη. Όπως βλέπουμε στο κώδικα παρακάτω, η διαδικασία λήψης περιλαμβάνει τρεις μεθόδους, την onSuccess στην οποία εφόσον ολοκληρωθεί η λήψη, συνεχίζει η ροή του κώδικα, σε αυτήν με την επιτυχή λήψη του αρχείου, προσθέτουμε τα αγαπημένα στην νέα βάση και κλείνουμε το μήνυμα(dialog) και προχωράει στην εφαρμογή. Στην onFailure διαχειριζόμαστε σε περίπτωση αποτυχίας, διαγράφουμε το προσωρινό αρχείο που έχει ληφθεί και ενημερώνουμε το χρήστη με κατάλληλο μήνυμα. Τέλος, μέσω της onProgress ενημερώνουμε το dialog για το ποσοστό του αρχείου που έχουμε λάβει.

```

public void downloadDb() {
    //initialize progress dialog
    progressDialog.setProgress(0);
    progressDialog.setCancelable(false);
    progressDialog.setButton(DialogInterface.BUTTON_NEGATIVE, "Cancel", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            deleteDatabase("museumEN_temp.db");
            databaseRef.getFile(file_temp).cancel();
            progressDialog.dismiss();
        }
    });
    progressDialog.show();

    //if the download is completed
    databaseRef.getFile(file_temp).addOnSuccessListener(new OnSuccessListener<FileDownloadTask.TaskSnapshot>() {
        @Override
        public void onSuccess(FileDownloadTask.TaskSnapshot taskSnapshot) {
            //check if the file that downloaded has the same size as the file in cloud
            if (file_temp.length() == size) {
                //rename the temp file to overwrite the old file with the new content
                if (file_temp.renameTo(file_final)) {
                    //paste the favourites from the old database in the new database
                    if (rooms != null) {
                        databaseAccess.open();
                        databaseAccess.pasteFavourites("rooms", rooms);
                        databaseAccess.close();
                        Log.d("rooms:" + rooms, "");
                    }
                    if (exhibits != null) {
                        databaseAccess.open();
                        databaseAccess.pasteFavourites("exhibits", exhibits);
                        databaseAccess.close();
                    }
                    if (paypal_id != null) {
                        databaseAccess.open();
                        databaseAccess.Paypal("write", paypal_id);
                        databaseAccess.close();
                    }
                }
            }
        }
    });
}

```

```

    }
    progressDialog.dismiss();
    Toast.makeText(getApplicationContext(), "Downloaded", Toast.LENGTH_LONG).show();
}
} else {
    //if downloaded file is not correct, delete it
    file_temp.delete();
    progressDialog.dismiss();
    Toast.makeText(SplashActivity.this, "Problem with Download", Toast.LENGTH_SHORT).show();
}
}
})
.addOnFailureListener(new OnFailureListener() {
    @Override
    public void onFailure(@NonNull Exception exception) {
        //if download failed, dismiss the progress dialog and delete the temp file
        deleteDatabase("museumEN_temp.db");
        progressDialog.dismiss();
        int errorCode = ((StorageException) exception).getErrorCode();
        String errorMessage = exception.getMessage();
        Log.d("errorCode:", "" + errorCode);
        Log.d("errorMessage:", errorMessage);
        // Handle any errors
        Toast.makeText(SplashActivity.this, "Failed to download. Try Again", Toast.LENGTH_LONG).show();
    }
})
.addOnProgressListener(new OnProgressListener<FileDownloadTask.TaskSnapshot>() {
    @Override
    public void onProgress(FileDownloadTask.TaskSnapshot taskSnapshot) {
        size = taskSnapshot.getTotalByteCount();
        long temp = (100 * taskSnapshot.getBytesTransferred()) / (taskSnapshot.getTotalByteCount());
        int temp2 = (int) temp;
        //progressDialog.setMax((int)taskSnapshot.getTotalByteCount()/(100000));
        progressDialog.setMax(100);
        progressDialog.setProgress(temp2);
        Log.d("bytes:" + taskSnapshot.getBytesTransferred(), "kb");
        checkNetwork();
    }
});
}
}
}

```

Σε περίπτωση όπου κατά τη διάρκεια λήψης της βάσης δεδομένων ο χρήστης αλλάξει εφαρμογή για παράδειγμα λάβει κλήση ή κάποιο μήνυμα, η λήψη παγώνει, κρατάμε το σημείο που μέχρι έχουμε λάβει μέσω της `onSaveInstanceState` και εάν επανέλθει στην εφαρμογή ο χρήστης, συνεχίζεται η λήψη του αρχείου μέσω της `onRestoreInstanceState`.

```

@Override
protected void onSaveInstanceState(Bundle outState) {
    // If there's a download in progress, save the reference so you can query it later
    if (mStorageRef != null) {
        outState.putString("reference", mStorageRef.toString());
    }
    if (exhibits != null) {
        outState.putStringArrayList("exhibits", new ArrayList<>(exhibits));
    }
    if (rooms != null) {
        outState.putStringArrayList("rooms", new ArrayList<>(rooms));
    }
    if (paypal_id != null) {
        outState.putString("paypal", paypal_id);
    }
    super.onSaveInstanceState(outState);
}
@Override
protected void onRestoreInstanceState(Bundle savedInstanceState) {
    super.onRestoreInstanceState(savedInstanceState);
    // If there was a download in progress, get its reference and create a new StorageReference
    final String stringRef = savedInstanceState.getString("reference");
    if (stringRef == null) {
        return;
    }
    mStorageRef = FirebaseStorage.getInstance().getReferenceFromUrl(stringRef);

    // Find all DownloadTasks under this StorageReference (in this example, there should be one)
    List<FileDownloadTask> tasks = mStorageRef.getActiveDownloadTasks();
    if (tasks.size() > 0) {
        // Get the task monitoring the download
        FileDownloadTask task = tasks.get(0);

        // Add new listeners to the task using an Activity scope
        task.addOnSuccessListener(this, new OnSuccessListener<FileDownloadTask.TaskSnapshot>() {
            @Override
            public void onSuccess(FileDownloadTask.TaskSnapshot state) {
                checkFile(); //call a user defined function to handle the event.
            }
        });
    }
    exhibits = savedInstanceState.getStringArrayList("exhibits");
    rooms = savedInstanceState.getStringArrayList("rooms");
    paypal_id = savedInstanceState.getString("paypal");
}
}
}
}

```

### 3. Αγορά της ηχητικής ξενάγησης

Εφόσον ο χρήστης προχωρήσει σε αναπαραγωγή της ηχητικής ξενάγησης ή στην αγορά της, στην εφαρμογή ελέγχουμε πρώτα στη τοπική βάση δεδομένων διότι μπορεί ο χρήστης να μην έχει πρόσβαση στο διαδίκτυο. Εάν υπάρχει κωδικός συναλλαγής και στη περίπτωση που αποτύχει ο έλεγχος και η πρόσβαση στο διαδίκτυο, ο χρήστης προχωράει σε έλεγχο στη βάση δεδομένων της Firebase όπου καταγράφουμε τις αγορές που γίνονται στο PayPal ανά χρήστη, εάν βρεθεί εγγραφή στη βάση δεδομένων της Firebase, την αποθηκεύουμε στην τοπική βάση δεδομένων και ο χρήστης προχωρά στη λήψη των αρχείων.

Εάν δεν βρεθεί εγγραφή, εμφανίζεται ένα dialog σχετικά με την αγορά της ηχητικής ξενάγησης. Εάν προχωρήσει ο χρήστης στην αγορά, εμφανίζεται η διεπαφή του PayPal ζητώντας από το χρήστη να εισάγει τα στοιχεία του στο PayPal για να προχωρήσει η αγορά. Σε όλες τις περιπτώσεις ελέγχουμε μέσω της Boolean μεθόδου CheckNetwork() εάν υπάρχει σύνδεση στο διαδίκτυο και ενημερώνουμε το χρήστη κατάλληλα.

```
final SharedPreferences prefs = getApplicationContext().getSharedPreferences("prefs", MODE_PRIVATE);
String paypal_price = prefs.getString(getString(R.string.paypal), "5");
paypal = Double.parseDouble(paypal_price);

//check if user has paypal receipt in database
if (hasPaypall == null) {
    final DatabaseAccess = DatabaseAccess.getInstance(this);
    databaseAccess.open();
    hasPaypall = databaseAccess.Paypall("read", null);
    databaseAccess.close();
}
//if has not record in database ,we will check in cloud if there is any receipt
if (hasPaypall.contains("null")) {
    progress = new ProgressDialog(CategoryActivity.this);
    progress.setMessage(CategoryActivity.this.getResources().getString(R.string.sync));
    progress.show();
    if (checkNetwork()) {
        fbref.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                //check from cloud if find the id from the receipt
                if (progress != null) {
                    progress.dismiss();
                }
                payPall_var = dataSnapshot.getValue(PayPall_Var.class);
                hasPaypall = payPall_var.getId();
                //we write in database the id from the receipt
                final DatabaseAccess = DatabaseAccess.getInstance(CategoryActivity.this);
                databaseAccess.open();
                databaseAccess.Paypall("write", hasPaypall);
                databaseAccess.close();

                //if it is not find any kind , open up paypall activity
                if (payPall_var == null) {
                    mBuilder = new AlertDialog.Builder(CategoryActivity.this);
                    //mBuilder.setView(null);
                    final Button next_button = (Button) paypall_dialog.findViewById(R.id.buy);

                    mBuilder.setView(paypall_dialog);
                    final AlertDialog dialog = mBuilder.create();
                    dialog.show();
                    next_button.setText(String.valueOf(paypal) + "€");
                    next_button.setOnClickListener(new View.OnClickListener() {
                        @Override
                        public void onClick(View v) {
                            progress.dismiss();
                            hasPaypall = null;
                            startService(m_service);
                            PayPalPayment payment = new PayPalPayment(new BigDecimal(paypal), "EUR", "Nomismatic
Museum Audio Guide", PayPalPayment.PAYMENT_INTENT_SALE);
                            Intent = new Intent(CategoryActivity.this, PaymentActivity.class);
                            intent.putExtra(PayPalService.EXTRA_PAYPAL_CONFIGURATION, m_Configuration);
                            intent.putExtra(PaymentActivity.EXTRA_PAYMENT, payment);
                            startActivityForResult(intent, m_paypallRequestCode);
                            dialog.dismiss();
                        }
                    });
                }
                progress.dismiss();
            } else {
                checkAudioFile(path);
            }
        });
    }
}
```

### 4. Λήψη των αρχείων της ηχητικής ξενάγησης

Έχοντας πλέον αγοράσει ο χρήστης την ηχητική ξενάγηση, του εμφανίζεται η διεπαφή όπου μπορεί να κάνει λήψη τα ηχητικά αρχεία που τον ενδιαφέρουν. Στη λίστα που θα εμφανιστεί στο χρήστη, μέσω του κατάλληλου πλήκτρου μπορεί να αρχίσει τη λήψη, μέσω

της μεθόδου LoadLanguage() ελέγχουμε ποια γλώσσα επέλεξε ο χρήστης κατά την είσοδο στην εφαρμογή, ελέγχουμε εάν υπάρχει ο κατάλληλος φάκελος στη συσκευή, διαφορετικά τον δημιουργούμε μέσω της mkdir και προχωράμε στο listener όπου θα αρχίσει η λήψη.

```
public void audio_download(int path_photo) {
    checkNetwork();
    mStorageRef = FirebaseStorage.getInstance().getReferenceFromUrl("gs://museum-ebfdd.appspot.com");
    final_path = path_photo + ".mp3";
    mountainsRef = mStorageRef.child("music_files/" + final_path);

    //Set up firebase based on language preferred
    if (load_language().contains("gr")) {
        mountainsRef = mStorageRef.child("music_files_gr/" + final_path);
        play_path = new File("/data/data/com.example.padpad.nav_bar/files/audio_downloadGr/" + path_photo +
        ".mp3");
        file = file_gr;
    } else {
        // mountainsRef = mStorageRef.child("music_files/" + calculateRoute);
        file = file_en;
        play_path = new File("/data/data/com.example.padpad.nav_bar/files/audio_downloadEn/" + path_photo +
        ".mp3");
    }
    localFile = new File(file, final_path);

    //check if folder is existed otherwise is making a new folder
    if (!file.exists()) {
        Boolean ff = file.mkdir();
        if (ff) {
            Toast.makeText(mcontext, "Folder is done", Toast.LENGTH_LONG).show();
        } else {
            Toast.makeText(mcontext, "Failed", Toast.LENGTH_LONG).show();
        }
    }

    //start progress bar
    progress.setMessage(mcontext.getResources().getString(R.string.wait));
    progress.setTitle(mcontext.getResources().getString(R.string.connection));
    progress.setCancelable(false);
    progress.show();

    mountainsRef.getMetadata().addOnSuccessListener(new OnSuccessListener<StorageMetadata>() {
        @Override
        public void onSuccess(StorageMetadata storageMetadata) {

            //get file size from firebase
            size = storageMetadata.getSizeBytes();
            progress.dismiss();
            Long temp;
            String temp_size;

            //check device for free space
            if (file.getFreeSpace() / (1024 * 1024) > 1000) {
                temp = file.getFreeSpace() / (1024 * 1024 * 1024);
                temp_size = temp + "Gb";
            } else {
                temp = file.getFreeSpace() / (1024 * 1024);
                temp_size = temp + "Mb";
            }
            //storage warning
            if (temp < 100) {
                Toast.makeText(mcontext, mcontext.getResources().getString(R.string.space_error) + temp_size,
                Toast.LENGTH_SHORT).show();
            }
            //if device have not enough space ,user get a Toast Warning
            if (file.getFreeSpace() / (1024 * 1024) < 60) {
                Toast.makeText(mcontext, "Not Enough Storage Space", Toast.LENGTH_SHORT).show();
            }
        }
    });
}
```

Πριν τη λήψη όμως πραγματοποιείται ένας τελευταίος έλεγχος για να διαπιστώσουμε εάν ο χρήστης έχει επαρκή διαθέσιμο χώρο στη συσκευή του. Εάν διαθέτει περισσότερα από 60MB διαθέσιμα προχωράμε στη λήψη αλλιώς τον ενημερώνουμε ότι δεν επαρκεί ο χώρος. Επίσης, ενημερώνεται σε περίπτωση όπου έχει διαθέσιμο χώρο μικρότερο των 100MB. Εφόσον περάσει τον έλεγχο του διαθέσιμου χώρου και ότι έχει πρόσβαση στο διαδίκτυο,

```
if (isOnline) {
    progress.setMessage(mcontext.getResources().getString(R.string.wait));
    progress.setTitle(mcontext.getResources().getString(R.string.download));
    progress.setCancelable(false);
    progress.setProgress(50);
    progress.show();
    //Download file in device storage
    mountainsRef.getFile(localFile).addOnSuccessListener(new OnSuccessListener<FileDownloadTask.TaskSnapshot>() {
        @Override
        public void onSuccess(FileDownloadTask.TaskSnapshot taskSnapshot) {
            checkNetwork();
            progress.dismiss();
            Toast.makeText(mcontext, "Downloaded", Toast.LENGTH_LONG).show();
            notifyDataSetChanged();
        }
    });
}
```

```

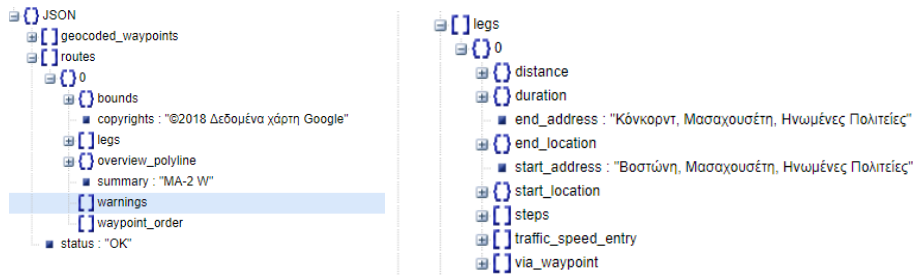
    }).addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception exception) {
            // Handle any errors
            int errorCode = ((StorageException) exception).getErrorCode();
            String errorMessage = exception.getMessage();
            Log.d("errorCode:", "" + errorCode);
            Log.d("errorMessage:", errorMessage);
            play_path.delete();
            progress.dismiss();
            notifyDataSetChanged();
            Toast.makeText(mcontext, "Failed to download. Try Again", Toast.LENGTH_LONG).show();
        }
    });
} else {
    Toast.makeText(mcontext, "Check Your Network Connection", Toast.LENGTH_SHORT).show();
}
}

```

αρχίζει η λήψη του αρχείου, μόλις ολοκληρωθεί ενημερώνεται με κατάλληλο μήνυμα και εξαφανίζεται το μήνυμα λήψης, σε αντίθετη περίπτωση ενημερώνεται ότι απέτυχε η λήψη μέσω της `onFailure`.

## 5. Αποστολή και λήψη πληροφοριών με Google Map

Ο χάρτης που έχει προστεθεί χρησιμοποιεί το Google Map, μέσω του οποίου ο χρήστης λαμβάνει οδηγίες ώστε να κατευθυνθεί προς το μουσείο και να δει τη τοποθεσία του. Για να λειτουργήσει το Google Map στην εφαρμογή θα πρέπει να ενεργοποιηθεί το API μέσω του <https://developers.google.com/maps> ώστε να λάβουμε το κατάλληλο API Key για να λειτουργήσει. Για να μπορέσουμε να απεικονίσουμε δεδομένα κατεύθυνσης, τα δεδομένα που ανταλλάσσονται με τους διακομιστές της Google είναι τύπου JSON, παραθέτουμε το response ενός HTTP αιτήματος.



Η πληροφορία που χρειαζόμαστε είναι στο JSON Array `routes`, για το οποίο θα χρειαστούμε τα κατάλληλα αντικείμενα για να λάβουμε τη πληροφορία. Η διαδικασία λήψης των πληροφοριών από τους server της Google επειδή επιβαρύνει τις συσκευές θα χρησιμοποιήσουμε τη κλάση `AsyncTask` του Android ώστε να αξιοποιήσουμε τα νήματα παρασκήνιου (background threads) και το κυρίως νήμα να είναι ελεύθερο. Στο Android η κάθε εφαρμογή διαθέτει το δικό της κυρίως νήμα, το οποίο ονομάζεται `UI thread`. Το κυρίως νήμα περιέχει όλα τα οπτικά στοιχεία (UI objects) που διαθέτει μια εφαρμογή ώστε να δίνει ομαλή περιήγηση του χρήστη στις διεπαφές. Εάν πραγματοποιήσουμε τη λήψη των δεδομένων στο κυρίως νήμα, ο χρήστης παρατηρήσει καθυστερήσεις στις διεπαφές.

```

public class DirectionFinder {
    private DirectionFinderListener listener;
    private String origin;
    private String destination;
    private String mode;

    public DirectionFinder(DirectionFinderListener listener, String origin, String destination, String mode) {
        this.listener = listener;
        this.origin = origin;
        this.destination = destination;
        this.mode = mode;
    }
}

```

```

public void execute() throws UnsupportedOperationException {
    listener.onDirectionFinderStart();
    new DownloadRawData().execute(createUrl());
}

private String createUrl() throws UnsupportedOperationException {
    String urlOrigin = URLEncoder.encode(origin, "utf-8");
    String urlDestination = URLEncoder.encode(destination, "utf-8");
    String urlMode = URLEncoder.encode(mode, "utf-8");

    return DIRECTION_URL_API + "origin=" + urlOrigin + "&destination=" + urlDestination + "&mode=" + urlMode +
"&key=" + GOOGLE_API_KEY;
}

private class DownloadRawData extends AsyncTask<String, Void, String> {

    @Override
    protected String doInBackground(String... params) {
        String link = params[0];
        try {
            URL url = new URL(link);
            InputStream is = url.openConnection().getInputStream();
            StringBuffer buffer = new StringBuffer();
            BufferedReader reader = new BufferedReader(new InputStreamReader(is));
            String line;
            while ((line = reader.readLine()) != null) {
                buffer.append(line + "\n");
            }
            return buffer.toString();
        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
        return null;
    }

    @Override
    protected void onPostExecute(String res) {
        try {
            parseJSON(res);
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
}

private void parseJSON(String data) throws JSONException {
    if (data == null)
        return;

    List<Route> routes = new ArrayList<Route>();
    JSONObject jsonData = new JSONObject(data);
    JSONArray jsonRoutes = jsonData.getJSONArray("routes");
    for (int i = 0; i < jsonRoutes.length(); i++) {
        JSONObject jsonRoute = jsonRoutes.getJSONObject(i);
        Route = new Route();

        JSONObject overview_polylineJson = jsonRoute.getJSONObject("overview_polyline");
        JSONArray jsonLegs = jsonRoute.getJSONArray("legs");
        JSONObject jsonLeg = jsonLegs.getJSONObject(0);
        JSONObject jsonDistance = jsonLeg.getJSONObject("distance");
        JSONObject jsonDuration = jsonLeg.getJSONObject("duration");
        JSONObject jsonEndLocation = jsonLeg.getJSONObject("end_location");
        JSONObject jsonStartLocation = jsonLeg.getJSONObject("start_location");

        route.distance = new Distance(jsonDistance.getString("text"), jsonDistance.getInt("value"));
        route.duration = new Duration(jsonDuration.getString("text"), jsonDuration.getInt("value"));
        route.endAddress = jsonLeg.getString("end_address");
        route.startAddress = jsonLeg.getString("start_address");
        route.startLocation = new LatLng(jsonStartLocation.getDouble("lat"),
jsonStartLocation.getDouble("lng"));
        route.endLocation = new LatLng(jsonEndLocation.getDouble("lat"), jsonEndLocation.getDouble("lng"));
        route.points = decodePolyLine(overview_polylineJson.getString("points"));

        routes.add(route);
    }

    listener.onDirectionFinderSuccess(routes);
}

private List<LatLng> decodePolyLine(final String poly) {
    int len = poly.length();
    int index = 0;
    List<LatLng> decoded = new ArrayList<LatLng>();
    int lat = 0;
    int lng = 0;

    while (index < len) {
        int b;
        int shift = 0;
        int result = 0;
        do {
            b = poly.charAt(index++) - 63;
            result |= (b & 0x1f) << shift;
            shift += 5;
        } while (b >= 0x20);
        int dlat = ((result & 1) != 0 ? ~(result >> 1) : (result >> 1));
        lat += dlat;

        shift = 0;
        result = 0;
    }
}

```

```

        do {
            b = poly.charAt(index++) - 63;
            result |= (b & 0x1f) << shift;
            shift += 5;
        } while (b >= 0x20);
        int dlng = ((result & 1) != 0 ? ~(result >> 1) : (result >> 1));
        lng += dlng;

        decoded.add(new LatLng(
            lat / 100000d, lng / 100000d
        ));
    }
    return decoded;
}
}

```

Στο διακομιστή στέλνουμε το αρχικό σημείο, το τελικό σημείο και το τρόπο με τον οποίο θα πάει στο προορισμό. Στη `doInBackground()` αρχίζει η λήψη των δεδομένων. Μόλις ολοκληρωθεί η λήψη, επιστρέφουμε στο κυρίως νήμα μέσω της `onPostExecute()`. Μέσω της `parseJson()` μεθόδου ολοκληρώνουμε το Parsing των δεδομένων και μέσω κατάλληλου listener επιστρέφουμε τη πληροφορία στη δραστηριότητα που διαχειρίζεται το Google Map. Τέλος, μέσω της `decodePolyline` αποκωδικοποιούμε την πληροφορία που μας δίνει η Google για τη διαδρομή που θα ακολουθήσει ώστε να την απεικονίσουμε στη δραστηριότητα.

```

@Override
public void onDirectionFinderSuccess(List<Route> routes) {
    progressDialog.dismiss();
    polylinePaths = new ArrayList<>();
    originMarkers = new ArrayList<>();
    destinationMarkers = new ArrayList<>();
    //clear markers from the map
    mMap.clear();

    for (Route ; routes) {
        ((TextView) findViewById(R.id.tvDuration)).setText(route.duration.text);
        ((TextView) findViewById(R.id.tvDistance)).setText(route.distance.text);

        int distance = route.distance.value;
        if (distance < 4000) {
            mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(route.endLocation, 14));
        } else if (distance < 8000) {
            mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(route.endLocation, 13));
        } else {
            mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(route.endLocation, 12));
        }

        originMarkers.add(mMap.addMarker(new MarkerOptions()
            .icon(BitmapDescriptorFactory.fromResource(R.drawable.start_blue))
            .title(route.startAddress)
            .position(route.startLocation)));
        destinationMarkers.add(mMap.addMarker(new MarkerOptions()
            .icon(BitmapDescriptorFactory.fromResource(R.drawable.point_end))
            .title(getResources().getString(R.string.app_name))
            .position(route.endLocation)));

        PolylineOptions = new PolylineOptions().
            geodesic(true).
            color(Color.BLUE).
            width(10);

        for (int i = 0; i < route.points.size(); i++)
            polylineOptions.add(route.points.get(i));

        polylinePaths.add(mMap.addPolyline(polylineOptions));
    }
}

```

Τέλος, μέσω της παραπάνω μεθόδου, έχουμε επιστρέψει μέσω του Callback στη δραστηριότητα του χάρτη, εμφανίζονται οι πληροφορίες της απόστασης και της διάρκειας μέσω των `setText()`. Μέσω της απόστασης που θα λάβουμε ανανεώνουμε κατάλληλα το zoom που γίνεται στο χάρτη ώστε να φαίνονται και τα δύο σημεία, προστεθούμε τις πινέζες στο χάρτη και ζωγραφίζουμε τη διαδρομή μέσω της κατάλληλης επανάληψης `for`.

## 6. Εκδηλώσεις

Τα δεδομένα που εμφανίζουμε στις εκδηλώσεις βρίσκονται στη βάση δεδομένων της Firebase. Τα δεδομένα όπως έχουμε αναφέρει είναι τύπου JSON για το οποίο έχει δημιουργηθεί το αντικείμενο Calendar. Κατά την είσοδο στην οθόνη εμφανίζεται ένας loader με κατάλληλο μήνυμα ώστε να ενημερώνουμε το χρήστη ότι γίνεται λήψη πληροφοριών. Μόλις ολοκληρωθεί η λήψη αυτών εξαφανίζεται το μήνυμα λήψης και εμφανίζονται τα δεδομένα στη αντίστοιχη λίστα της διεπαφής.

```
private void getData() {
    fbref.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            for (DataSnapshot data : dataSnapshot.getChildren()) {
                if (dialog != null) {
                    dialog.dismiss();
                }
                Calendar = data.getValue(Calendar.class);
                calendar.setTitle(calendar.getTitle());
                calendar.setDate(calendar.getDate());
                calendar.setImage_cal(calendar.getImage_cal());
                calendar.setDescription(calendar.getDescription());
                calendar.setBegin_time(calendar.getBegin_time());
                calendar.setEnd_time(calendar.getEnd_time());
                calendar.setUrl(calendar.getUrl());
                result.add(calendar);
            }
            //sort by last
            sortList(result);
            mAdapter = new EventRecyclerViewAdapter(result, getActivity());
            mRecyclerView.setAdapter(mAdapter);
            if (progress != null) {
                progress.dismiss();
            }
        }
        @Override
        public void onCancelled(DatabaseError databaseError) {

        }
    });
}
```

Στο παραπάνω κώδικα μέσω της addValueEventListener πραγματοποιείται η λήψη των δεδομένων, η οποία σε περίπτωση όπου έχουμε λάβει τα δεδομένα ,εάν αντιληφθεί αλλαγή στη βάση δεδομένων σε πραγματικό χρόνο ανανεώνει τα δεδομένα στον αντάπτορα. Όπως παρατηρούμε λαμβάνουμε τα δεδομένα, πραγματοποιούμε ταξινόμηση βάσει της τελευταίας χρονικά εκδήλωσης ώστε να είναι στη κορυφή και στέλνουμε τη λίστα στον κατάλληλο αντάπτορα, μέσω του οποίου θα εμφανίσουμε τα δεδομένα στη διεπαφή.

```
public class EventRecyclerViewAdapter extends RecyclerView.Adapter<EventRecyclerViewAdapter.ViewHolder> {
    private Context mContext;
    private List<Calendar> List;

    public EventRecyclerViewAdapter(List<Calendar> events, Context context) {
        this.List = events;
        this.mContext = context;
    }

    @Override
    public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View v = LayoutInflater.from(parent.getContext())
            .inflate(R.layout.row_events, parent, false);

        return new EventRecyclerViewAdapter.ViewHolder(v);
    }

    @Override
    public void onBindViewHolder(final ViewHolder holder, int position) {

        final Calendar _ = List.get(position);
        holder.mTitle.setText(calendar_.title);
        holder.mDescr.setText(calendar_.description);
        holder.mdate.setText(calendar_.date);
        Glide.with(holder.imageView.getContext()).load(calendar_.image_cal).asBitmap().centerCrop().override(200, 200).diskCacheStrategy(DiskCacheStrategy.ALL).error(R.drawable.museum).into(holder.imageView);

        holder.cardView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent = new Intent(mContext, EventsActivity.class);
            }
        });
    }
}
```



```
        intent.putExtra("event", calendar_);
        mContext.startActivity(intent);
    });
}

@Override
public int getItemCount() {
    return List.size();
}
```

Πλέον στον αντάπτορα έχοντας τα δεδομένα, εμφανίζουμε τις πληροφορίες, η φωτογραφία της κάθε εκδήλωσης δεν είναι αποθηκευμένη στη βάση δεδομένων της Firebase και θα πρέπει να γίνει λήψη αυτής, στη Firebase έχουμε καταχωρημένο το σύνδεσμο της φωτογραφίας. Μέσω της βιβλιοθήκης Glide γίνεται λήψη της φωτογραφίας, μόλις ολοκληρωθεί η λήψη, κάνει τους απαραίτητες αλλαγές και εμφανίζει την εικόνα στη διεπαφή. Εάν ο χρήστης πατήσει σε κάποια από τις εκδηλώσεις μέσω του `setOnClickListener()` θα ανοίξει η δραστηριότητα όπου περνάμε όλες τις πληροφορίες της εκάστοτε εκδήλωσης για να εμφανιστεί στην επόμενη οθόνη.