

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



Πτυχιακή Εργασία με θέμα:
«Εφαρμογή για κινητά με σκοπό την πρόταση αξιοθέατων»

Μπαλωμένος Σπυρίδων

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:

Έξαρχος Θεμιστοκλής

ΤΡΙΜΕΛΗΣ ΕΠΙΤΡΟΠΗ:

Έξαρχος Θεμιστοκλής

Κερμανίδου Κάτια-Λήδα

Βραχάτης Αριστείδης

ΠΕΡΙΛΗΨΗ

Στα πλαίσια της συγκεκριμένης εργασίας αρχικά πραγματοποιήθηκε βιβλιογραφική ανασκόπηση όσο αφορά τις περιπτώσεις εφαρμογών που έχουν σαν βασικό σκοπό την πρόταση αξιοθέατων. Στην συνέχεια παρουσιάστηκαν τα βασικά συστατικά μιας Android εφαρμογής, καθώς και η βασική λειτουργικότητα του εργαλείου Android Studio. Όσο αφορά το πρακτικό μέρος, υλοποιήθηκε μια Android εφαρμογή σε Android Studio η οποία είχε σαν βασικό αντικείμενο την πρόταση αξιοθέατων όσο αφορά την πόλη της Νέας Υόρκης.

ABSTRACT

In the context of this work, a Bibliographic Review was initially carried out regarding the cases of applications that have as their main purpose the proposal of attractions. Then the basic components of an Android application were presented, as well as the basic functionality of the Android Studio tool. As far as the practical part is concerned, an Android application was implemented in Android Studio which had as its main object the proposal of attractions regarding the city of New York

Πίνακας Περιεχομένων

ΚΕΦΑΛΑΙΟ 1: ΔΗΜΟΦΙΛΕΙΣ ΕΦΑΡΜΟΓΕΣ ΚΙΝΗΤΩΝ ΣΥΣΚΕΥΩΝ ΓΙΑ ΠΡΟΤΑΣΗ

ΑΞΙΟΘΕΑΤΩΝ 10

1.1. TripAdvisor..... 12

1.2. Google Maps 14

1.3. Lonely Planet Guides 16

1.4. Sygic Travel..... 17

1.5. Viator 19

1.6. Culture Trip 21

1.7. Roadtrippers 23

1.8. Rove.me..... 25

1.9. Επιθυμητά χαρακτηριστικά εφαρμογών πρότασης αξιοθέατων..... 27

1.10. Προκλήσεις εφαρμογών πρότασης αξιοθέατων..... 29

ΚΕΦΑΛΑΙΟ 2: ANDROID ΕΦΑΡΜΟΓΕΣ ΚΑΙ ANDROID STUDIO 31

2.1. Βασικά συστατικά Android εφαρμογής..... 31

2.1.1. Activities 31

2.1.2. Services 33

2.1.3. Broadcast receivers..... 35

2.1.4. Content providers	37
2.1.5. Intents	39
2.1.6. Fragments.....	41
2.1.7. Αρχείο Manifest	42
2.2. Δυνατότητες Android Studio.....	44
2.2.1. Σχεδιαστής Διεπαφής Χρήστη.....	44
2.2.2. Επεξεργαστής κώδικα.....	46
2.2.3. Gradle Build System.....	48
2.2.4. Android Emulator	50
2.2.5. SDK Manager	51
ΚΕΦΑΛΑΙΟ 3: ΕΦΑΡΜΟΓΗ ΠΡΟΤΑΣΗΣ ΑΞΙΟΘΕΑΤΩΝ	57
3.1. Βασική Φιλοσοφία Εφαρμογής.....	57
3.2. Κώδικας Εφαρμογής.....	61
3.2.1. Το αρχείο AndroidManifest.xml	61
3.2.2. Main Activity	63
3.2.3. Category Adapter	65
3.2.4. Museums Fragment	68
3.2.5. Parks Fragment	70

3.2.6. Observation Decks Fragment	72
3.2.7. Miscellaneous Fragment	73
3.2.8. Subjects.....	75
3.2.9. Subjects Adapter	78
3.2.10. Strings.xml	81
3.3. Screenshots Εφαρμογής	82
Συμπεράσματα	86

Λίστα Εικόνων

Εικόνα 1: TripAdvisor	13
Εικόνα 2: Google Maps.....	15
Εικόνα 3: Lonely Planet Guides	16
Εικόνα 4: Sygic Travel	18
Εικόνα 5: Viator	20
Εικόνα 6: Culture Trip	22
Εικόνα 7: Roadtrippers	24
Εικόνα 8: Rove.me	26
Εικόνα 9: Κύκλος ζωής δραστηριότητας Android	32
Εικόνα 10: Κύκλος ζωής Service	34
Εικόνα 11: Χρήση broadcast receiver	36
Εικόνα 12: ContentProvider.....	38
Εικόνα 13: Χρήση Intent.....	40
Εικόνα 14: Fragments	41
Εικόνα 15: Παράδειγμα AndroidManifest.xml	43
Εικόνα 16: User Interface Designer.....	45
Εικόνα 17: Χρήση ConstraintLayout	46
Εικόνα 18: Code Editor.....	47

Εικόνα 19: Gradle Build System.....	49
Εικόνα 20: Android Emulator.....	50
Εικόνα 21: SDK Manager	52
Εικόνα 22: Χρήση debugging σε Android Studio	54
Εικόνα 23: Χρήση Android Debug Bridge	55
Εικόνα 24: Android Logcat	56
Εικόνα 25: Περιβάλλον ανάπτυξης εφαρμογής	59
Εικόνα 26: Πληροφορίες μουσείων στο strings.xml.....	81
Εικόνα 27: Πληροφορίες πάρκων στο strings.xml	81
Εικόνα 28: Πληροφορίες σημείων παρατήρησης στο strings.xml	81
Εικόνα 29: Πληροφορίες λοιπών αξιοθέατων στο strings.xml	82
Εικόνα 30: Screenshot 1 - Παρουσίαση πάρκων	82
Εικόνα 31: Screenshot 2 - Παρουσίαση μουσείων.....	83
Εικόνα 32: Screenshot 3 - Παρουσίαση σημείων παρατήρησης	84
Εικόνα 33: Screenshot 4 - Παρουσίαση λοιπών αξιοθέατων	85

Λίστα Πινάκων

Πίνακας 1: Πληροφορίες αξιοθέατων που παρατίθενται στην εφαρμογή.....	57
Πίνακας 2: Λίστα ονομάτων μουσείων	58
Πίνακας 3: Λίστα ονομάτων πάρκων	58
Πίνακας 4: Λίστα ονομάτων σημείων παρατήρησης.....	59
Πίνακας 5: Λίστα ονομάτων λοιπών αξιοθέατων	59
Πίνακας 6: Βασικά Αρχεία Εφαρμογής	60

Εισαγωγή

Η ανάγκη για ενίσχυση της εμπειρίας των τουριστών σε διάφορες περιοχές ενδιαφέροντος έχει οδηγήσει στην αναζήτηση αποτελεσματικών μεθόδων με απώτερο στόχο την προσέλκυσή τους και την εξυπηρέτηση των αναγκών τους. Μια πτυχή της συγκεκριμένης περίπτωσης αποτελεί και η ανάπτυξη εφαρμογών κινητών συσκευών, οι οποίες σαν στόχο έχουν την πρόταση αξιοθέατων προς τους τουρίστες, με όσο το δυνατόν πληρέστερη και πιο διαδραστική πληροφόρηση.

Είναι αδιαμφισβήτητο το γεγονός ότι υπάρχουν αρκετές εφαρμογές οι οποίες κινούνται προς τη συγκεκριμένη κατεύθυνση, με διαφορετικά χαρακτηριστικά, πλεονεκτήματα αλλά και τρωτά σημεία. Κάποιες από αυτές παρουσιάζονται στα πλαίσια της συγκεκριμένης εργασίας.

Στόχος της πτυχιακής εργασίας είναι πέρα από την βιβλιογραφική επισκόπηση και παρουσίαση των πλέον διαδεδομένων εφαρμογών για παρουσίαση αξιοθέατων, και η ανάπτυξη μιας Android εφαρμογής με στόχο την πρόταση αξιοθέατων για την πόλη της Νέας Υόρκης.

Η διάρθρωση της εργασίας έχει ως εξής:

Στο **πρώτο κεφάλαιο** παρουσιάζονται δημοφιλείς εφαρμογές πρότασης αξιοθέατων όπως είναι οι: TripAdvisor, Google Maps και Roadtrippers.

Στο **δεύτερο κεφάλαιο** παρουσιάζεται μια επισκόπηση των τεχνικών βασικών συστατικών μιας Android εφαρμογής, καθώς και των δυνατοτήτων του Android Studio.

Τέλος, στο **τρίτο κεφάλαιο** παρουσιάζεται η εφαρμογή, η οποία υλοποιήθηκε, δείχνοντας τόσο βασικά σημεία του κώδικα, αλλά και screenshots που επιδεικνύουν την λειτουργικότητά της.

ΚΕΦΑΛΑΙΟ 1: ΔΗΜΟΦΙΛΕΙΣ ΕΦΑΡΜΟΓΕΣ ΚΙΝΗΤΩΝ ΣΥΣΚΕΥΩΝ ΓΙΑ ΠΡΟΤΑΣΗ ΑΞΙΟΘΕΑΤΩΝ

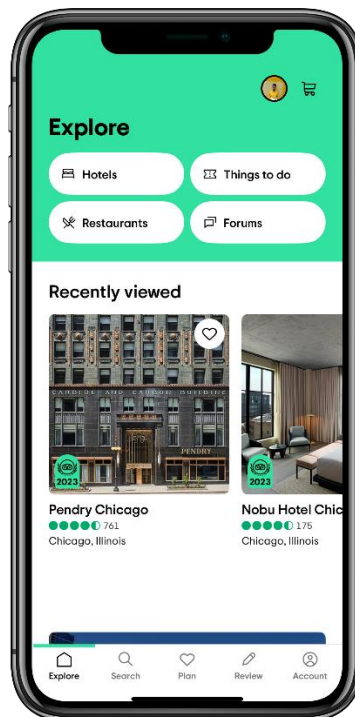
1.1. TripAdvisor

Το TripAdvisor [1] είναι μια ολοκληρωμένη ταξιδιωτική πλατφόρμα που προσφέρει μια πληθώρα χαρακτηριστικών που έχουν σχεδιαστεί για να βοηθούν τους ταξιδιώτες να σχεδιάζουν τα ταξίδια τους και να λαμβάνουν ολοκληρωμένες αποφάσεις. Ένα σημαντικό χαρακτηριστικό του TripAdvisor είναι η εκτεταμένη βάση δεδομένων του με κριτικές και βαθμολογίες που δημιουργούνται από χρήστες για ξενοδοχεία, εστιατόρια, αξιοθέατα και δραστηριότητες σε όλο τον κόσμο. Οι ταξιδιώτες μπορούν να περιηγηθούν σε εκατομμύρια κριτικές για να αποκτήσουν γνώσεις σχετικά με την ποιότητα και την καταλληλότητα διαφόρων εγκαταστάσεων πριν πραγματοποιήσουν τις κρατήσεις τους.

Ένα άλλο βασικό χαρακτηριστικό του TripAdvisor είναι η ισχυρή λειτουργικότητα αναζήτησης, η οποία επιτρέπει στους χρήστες να φιλτράρουν τα αποτελέσματα αναζήτησης με βάση παράγοντες όπως το εύρος τιμών, η τοποθεσία, η βαθμολογία ταξιδιωτών και οι ανέσεις. Αυτή η λειτουργία επιτρέπει στους ταξιδιώτες να βρουν γρήγορα καταλύματα, εστιατόρια και αξιοθέατα που ευθυγραμμίζονται με τις προτιμήσεις τους και τους περιορισμούς του προϋπολογισμού που θέτουν. Επιπλέον, το TripAdvisor παρέχει πληθώρα πληροφοριών σχετικά με προορισμούς, συμπεριλαμβανομένων ταξιδιωτικών οδηγών, άρθρων και φόρουμ όπου οι

χρήστες μπορούν να ζητήσουν συμβουλές, να μοιραστούν εμπειρίες και να ανταλλάξουν συμβουλές με άλλους ταξιδιώτες.

Οι δυνατότητες κράτησης του TripAdvisor είναι ένα άλλο αξιοσημείωτο χαρακτηριστικό, επιτρέποντας στους χρήστες να πραγματοποιούν κράτηση ξενοδοχείων, εστιατορίων, περιηγήσεων και δραστηριοτήτων απευθείας μέσω της πλατφόρμας. Αυτή η λειτουργία εξορθολογίζει τη διαδικασία σχεδιασμού ταξιδιού ενοποιώντας όλες τις απαραίτητες κρατήσεις σε ένα μέρος, εξοικονομώντας χρόνο και προσπάθεια στους ταξιδιώτες. Επιπλέον, το TripAdvisor προσφέρει ένα εργαλείο σύγκρισης τιμών που επιτρέπει στους χρήστες να συγκρίνουν τιμές από πολλούς ιστότοπους κρατήσεων ώστε να διασφαλίσουν ότι λαμβάνουν την καλύτερη δυνατή προσφορά.



Εικόνα 1: TripAdvisor

Τέλος, τα βραβεία "Traveler's Choice" του TripAdvisor αναδεικνύουν τα κορυφαία ξενοδοχεία, προορισμούς, εστιατόρια και αξιοθέατα παγκοσμίως, με βάση την ποιότητα και την ποσότητα των κριτικών και αξιολογήσεων των ταξιδιωτών. Αυτά τα βραβεία χρησιμεύουν ως πολύτιμος πόρος για τους ταξιδιώτες που αναζητούν συστάσεις και έμπνευση για την επόμενη περιπέτειά τους. Συνολικά, το ευρύ φάσμα χαρακτηριστικών του TripAdvisor το καθιστά μια πλατφόρμα μετάβασης για ταξιδιώτες που επιθυμούν να ερευνήσουν, να σχεδιάσουν και να κλείσουν τα ταξίδια τους με αυτοπεποίθηση και ευκολία.

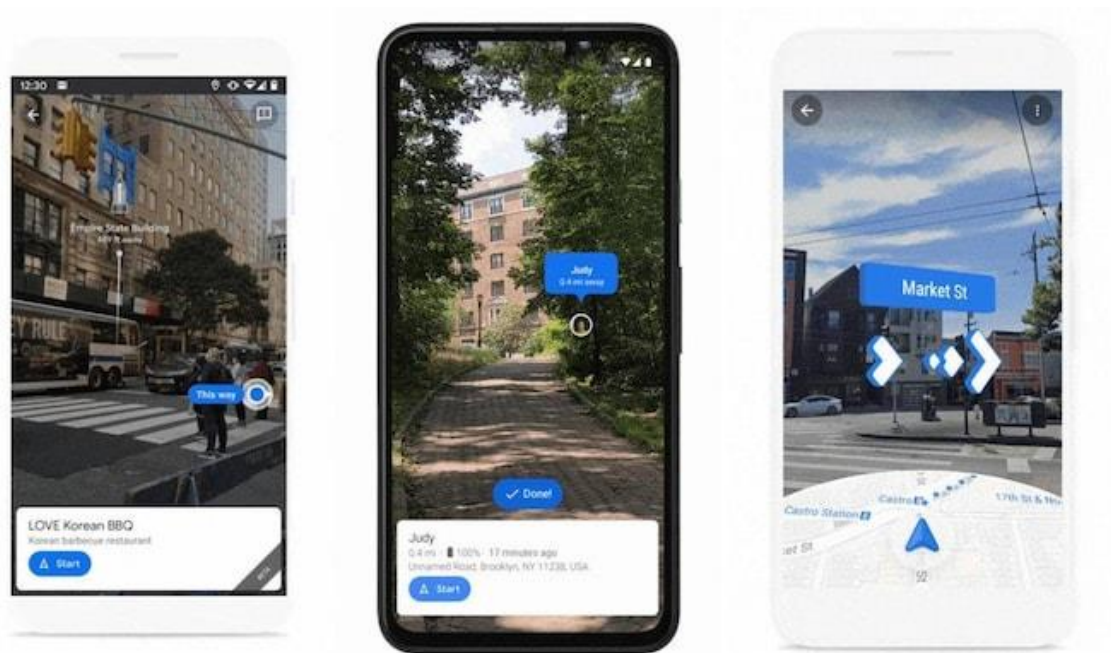
1.2. Google Maps

Η εφαρμογή Google Maps [2] είναι μια εφαρμογή για κινητά που προσφέρει πολύ περισσότερα από απλή πλοήγηση. Αποτελεί ένα ισχυρό εργαλείο για την ανακάλυψη αξιοθέατων. Ένα από τα πιο εντυπωσιακά χαρακτηριστικά του είναι η ικανότητά του να προτείνει κοντινά σημεία ενδιαφέροντος με βάση την τρέχουσα τοποθεσία του χρήστη. Αξιοποιώντας τα δεδομένα τοποθεσίας και τις προτιμήσεις των χρηστών, η εφαρμογή Google Maps μπορεί να προτείνει ένα ευρύ φάσμα αξιοθέατων, συμπεριλαμβανομένων τοποσήμων, μουσείων, πάρκων, εστιατορίων και άλλων, καθιστώντας το ανεκτίμητο οδηγό τόσο για ταξιδιώτες όσο και για ντόπιους.

Επιπλέον, η λειτουργία "Εξερεύνηση" είναι ιδιαίτερα αξιοσημείωτη για την ικανότητά της να παρέχει προσαρμοσμένες συστάσεις για αξιοθέατα. Με ένα απλό πάτημα, οι χρήστες μπορούν να έχουν πρόσβαση σε προσαρμοσμένες λίστες με δημοφιλή μέρη για επίσκεψη σε κοντινή απόσταση. Αυτές οι προτάσεις βασίζονται σε παράγοντες όπως οι αξιολογήσεις χρηστών, οι κριτικές

και οι τάσεις σε πραγματικό χρόνο, διασφαλίζοντας ότι οι χρήστες έχουν πρόσβαση σε ενημερωμένες και σχετικές συστάσεις όπου κι αν βρίσκονται.

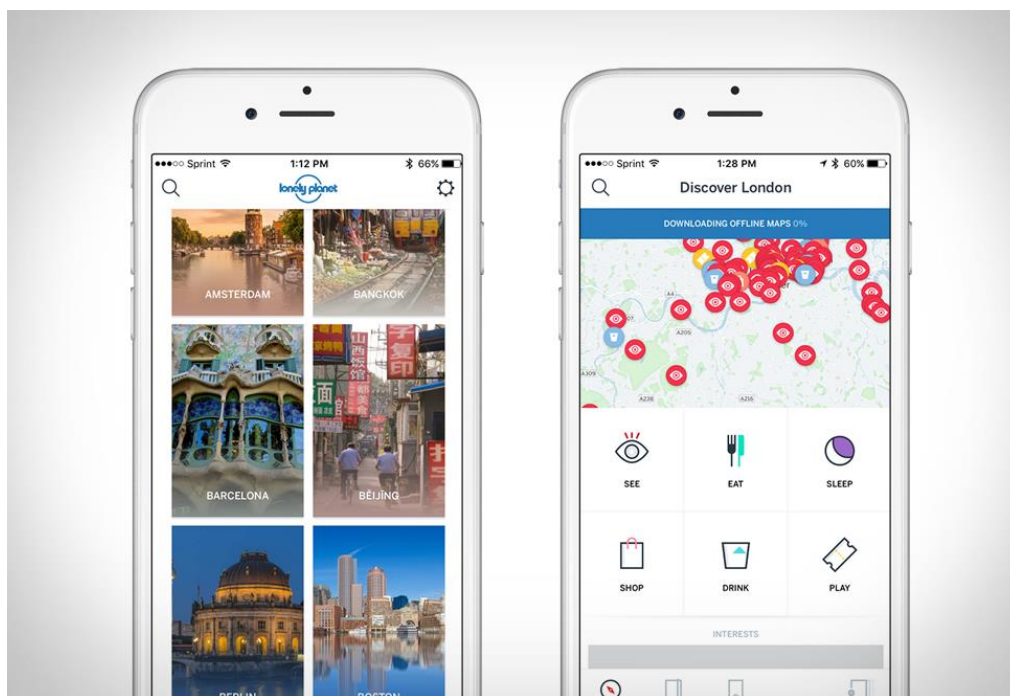
Τέλος, η εφαρμογή Google Maps χρησιμοποιεί προηγμένους αλγόριθμους και τεχνικές μηχανικής μάθησης προκειμένου να εξατομικεύει την εμπειρία περιήγησης στα αξιοθέατα για κάθε χρήστη. Αναλύοντας το ιστορικό αναζήτησης του παρελθόντος, τις αποθηκευμένες τοποθεσίες και τη συμπεριφορά των χρηστών, μπορεί να προσφέρει εξαιρετικά σχετικές προτάσεις που συμβαδίζουν με τα ενδιαφέροντα και τις προτιμήσεις του χρήστη. Αυτό το επίπεδο προσαρμογής διασφαλίζει ότι οι χρήστες λαμβάνουν συστάσεις που αντηχούν με τα ατομικά τους γούστα, ενισχύοντας τη συνολική διαδικασία εξερεύνησης και ανακάλυψης.



Εικόνα 2: Google Maps

1.3. Lonely Planet Guides

Η εφαρμογή Lonely Planet Guides [3] για κινητά ξεχωρίζει ως ένας ολοκληρωμένος ταξιδιωτικός οδηγός, προσφέροντας μια πληθώρα χαρακτηριστικών που έχουν σχεδιαστεί ώστε να βελτιώσουν την ταξιδιωτική εμπειρία για χρήστες σε όλο τον κόσμο. Ένα από τα βασικά χαρακτηριστικά του είναι η εκτεταμένη συλλογή επιμελημένων ταξιδιωτικών οδηγών για πάρα πολλούς προορισμούς. Αυτοί οι οδηγοί ερευνώνται σχολαστικά από την ομάδα ειδικών ταξιδιωτικών συγγραφέων του Lonely Planet και παρέχουν σε βάθος πληροφορίες για αξιοθέατα, καταλύματα, εστιατόρια και δραστηριότητες, διασφαλίζοντας ότι οι χρήστες έχουν πρόσβαση σε αξιόπιστες και ενημερωμένες προτάσεις όπου κι αν πάνε.



Εικόνα 3: Lonely Planet Guides

Ένα άλλο αξιοσημείωτο χαρακτηριστικό της εφαρμογής Lonely Planet Guides είναι η λειτουργικότητά της εκτός σύνδεσης, η οποία επιτρέπει στους χρήστες να έχουν πρόσβαση σε βασικές ταξιδιωτικές πληροφορίες χωρίς να απαιτείται σύνδεση στο Διαδίκτυο. Με τη λήψη συγκεκριμένων οδηγών και χαρτών εκ των προτέρων, οι χρήστες μπορούν να πλοηγηθούν στον προορισμό τους απρόσκοπτα, ακόμη και σε περιοχές με περιορισμένη συνδεσιμότητα. Αυτό το χαρακτηριστικό είναι ιδιαίτερα πολύτιμο για τους ταξιδιώτες που εξερευνούν απομακρυσμένες ή ξένες περιοχές όπου η πρόσβαση στο διαδίκτυο μπορεί να είναι αναξιόπιστη ή μη διαθέσιμη, διασφαλίζοντας ότι μπορούν να έχουν πρόσβαση σε κρίσιμες πληροφορίες όποτε τις χρειάζονται.

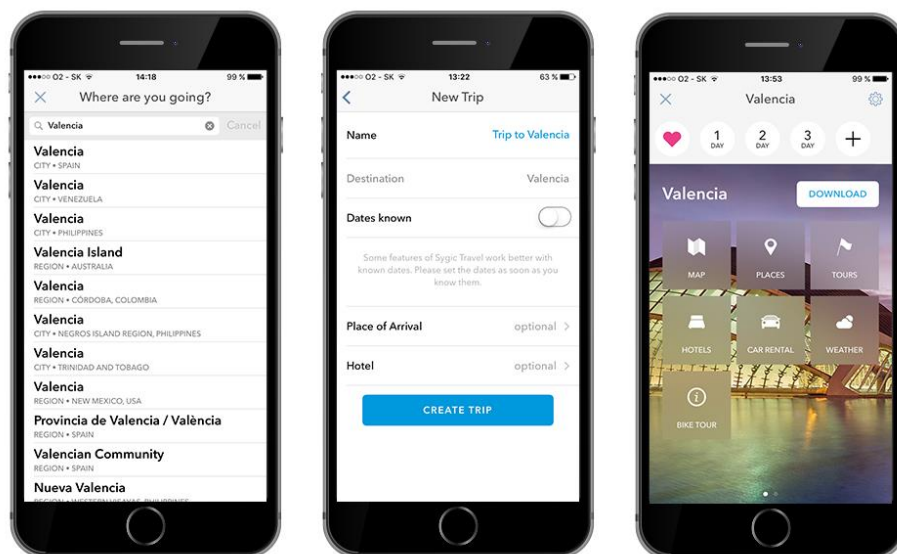
Επιπλέον, η εφαρμογή Lonely Planet Guides προσφέρει εξατομικευμένες προτάσεις και εργαλεία σχεδιασμού ταξιδιού ώστε να βοηθήσει τους χρήστες να δημιουργήσουν αξέχαστες ταξιδιωτικές εμπειρίες. Μέσω της φιλικής διεπαφής του, οι χρήστες μπορούν να δημιουργήσουν προσαρμοσμένα ταξίδια, να προσθέσουν σελιδοδείκτες σε αγαπημένα αξιοθέατα και να ανακαλύψουν προορισμούς που προτείνουν άλλοι ταξιδιώτες. Επιπλέον, η λειτουργία "Ανακαλύψτε" της εφαρμογής επιτρέπει στους χρήστες να εξερευνήσουν λίστες με κορυφαία αξιοθέατα, εμπειρίες και επιλογές φαγητού προσαρμοσμένες στα ενδιαφέροντα και τις προτιμήσεις τους.

1.4. Sygic Travel

Το Sygic Travel [4] είναι μια ισχυρή εφαρμογή για κινητά που έχει σχεδιαστεί ώστε να απλοποιεί τον προγραμματισμό και την εξερεύνηση ταξιδιών για ταξιδιώτες σε όλο τον κόσμο. Ένα από τα ξεχωριστά χαρακτηριστικά του είναι

η ολοκληρωμένη βάση δεδομένων προορισμών, αξιοθέατα, και σημεία ενδιαφέροντος, που εκτείνεται σε χιλιάδες πόλεις και περιοχές. Οι χρήστες μπορούν να περιηγηθούν σε μια τεράστια συλλογή ταξιδιωτικών οδηγών, άρθρων και συστάσεων για να ανακαλύψουν κορυφαία αξιοθέατα.

Ένα άλλο αξιοσημείωτο χαρακτηριστικό του Sygic Travel είναι ο προσαρμόσιμος σχεδιασμός ταξιδιών, ο οποίος επιτρέπει στους χρήστες να δημιουργούν εξατομικευμένα ταξιδιωτικά σχέδια προσαρμοσμένα στα ενδιαφέροντα, τις προτιμήσεις και τα χρονοδιαγράμματα τους. Επιλέγοντας απλώς τον προορισμό και τις ημερομηνίες τους, οι χρήστες μπορούν να εξερευνήσουν προτεινόμενα δρομολόγια ή να δημιουργήσουν τα δικά τους από το μηδέν, προσθέτοντας αξιοθέατα, δραστηριότητες, επιλογές φαγητού, και καταλύματα στην πορεία. Η φιλική διεπαφή μεταφοράς και απόθεσης διευκολύνει την οργάνωση του δρομολογίου, διασφαλίζοντας ότι οι ταξιδιώτες μπορούν να μεγιστοποιήσουν το χρόνο τους και να αξιοποιήσουν στο έπακρο το ταξίδι τους.



Εικόνα 4: Sygic Travel

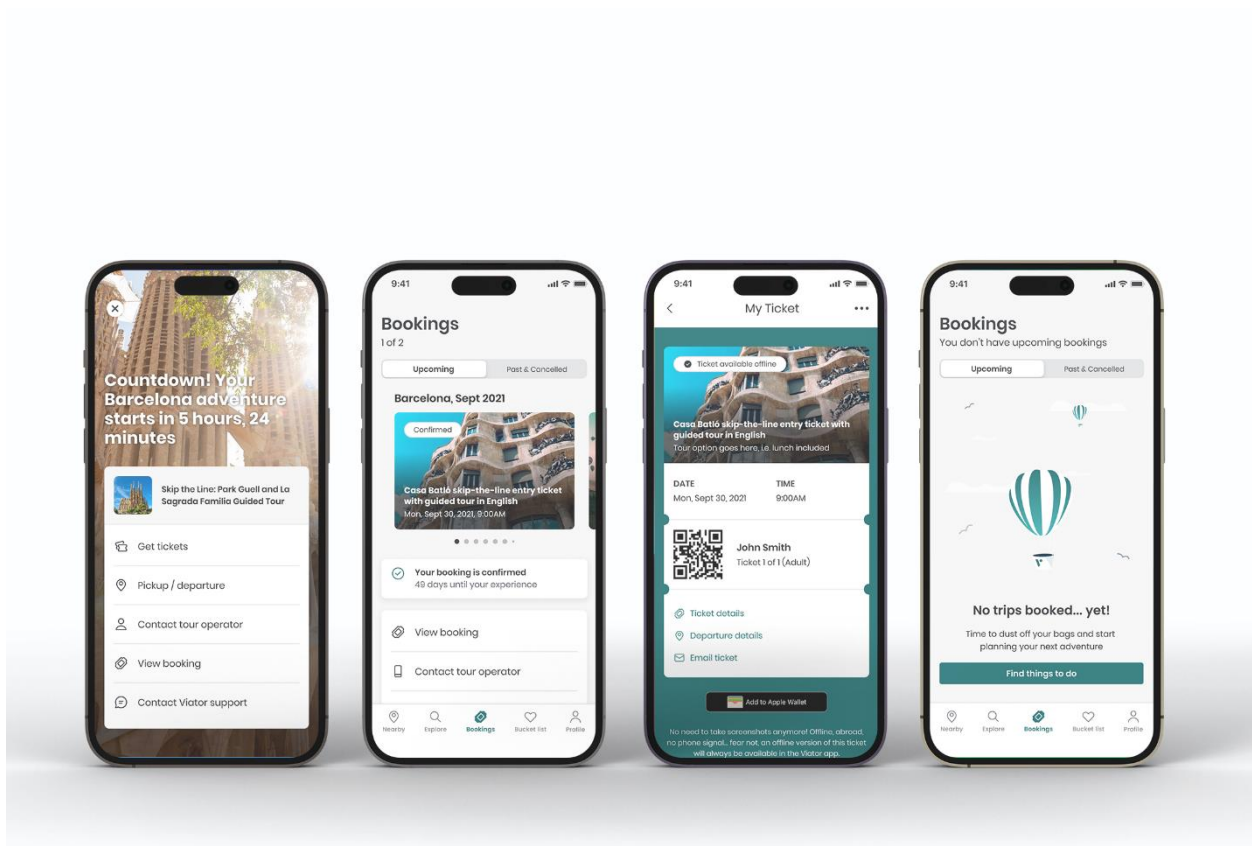
Επιπλέον, το Sygic Travel προσφέρει χάρτες εκτός σύνδεσης και δυνατότητες πλοήγησης, επιτρέποντας στους χρήστες να έχουν πρόσβαση σε βασικές ταξιδιωτικές πληροφορίες και οδηγίες χωρίς να απαιτείται σύνδεση στο Διαδίκτυο. Με τη λήψη χαρτών εκτός σύνδεσης του προορισμού τους, οι χρήστες μπορούν να πλοηγηθούν με σιγουριά, ακόμη και σε περιοχές με περιορισμένη συνδεσιμότητα ή ενώ ταξιδεύουν στο εξωτερικό.

1.5. Viator

Το Viator [5] είναι μια κορυφαία εφαρμογή για κινητά που ειδικεύεται στο να βοηθά τους ταξιδιώτες να ανακαλύψουν και να κλείσουν εκδρομές, δραστηριότητες και επισκέψεις σε αξιοθέατα σε όλο τον κόσμο. Ένα από τα ξεχωριστά χαρακτηριστικά του είναι ο εκτεταμένος κατάλογος εμπειριών του, που κυμαίνονται από ξεναγήσεις σε εμβληματικά τοπόσημα έως περιπέτειες εκτός δρόμου και μοναδικές πολιτιστικές εμπειρίες. Με χιλιάδες επιλογές διαθέσιμες σε προορισμούς παγκοσμίως, οι χρήστες μπορούν εύκολα να περιηγηθούν σε μια ποικιλία δραστηριοτήτων και να βρουν κάτι που ταιριάζει με τα ενδιαφέροντά τους, προτιμήσεις, και στυλ ταξιδιού.

Ένα άλλο αξιοσημείωτο χαρακτηριστικό της εφαρμογής Viator είναι η φιλική προς το χρήστη διεπαφή και η χρηστική λειτουργικότητα αναζήτησης, που διευκολύνουν τους ταξιδιώτες να βρουν και να πραγματοποιήσουν κρατήσεις για εμπειρίες που ταιριάζουν στις ανάγκες τους. Οι χρήστες μπορούν να φιλτράρουν τα αποτελέσματα αναζήτησης με βάση διάφορα κριτήρια όπως η τοποθεσία, η διάρκεια, το εύρος τιμών, ο τύπος δραστηριότητας και οι αξιολογήσεις πελατών, επιτρέποντάς τους να περιορίσουν γρήγορα τις

επιλογές τους και να βρουν την τέλεια δραστηριότητα για το ταξίδι τους. Επιπλέον, το Viator παρέχει λεπτομερείς περιγραφές, φωτογραφίες, κριτικές και βαθμολογίες για κάθε εμπειρία, δίνοντας στους χρήστες τις πληροφορίες που χρειάζονται προκειμένου να λάβουν εμπειριστατωμένες αποφάσεις.



Εικόνα 5: Viator

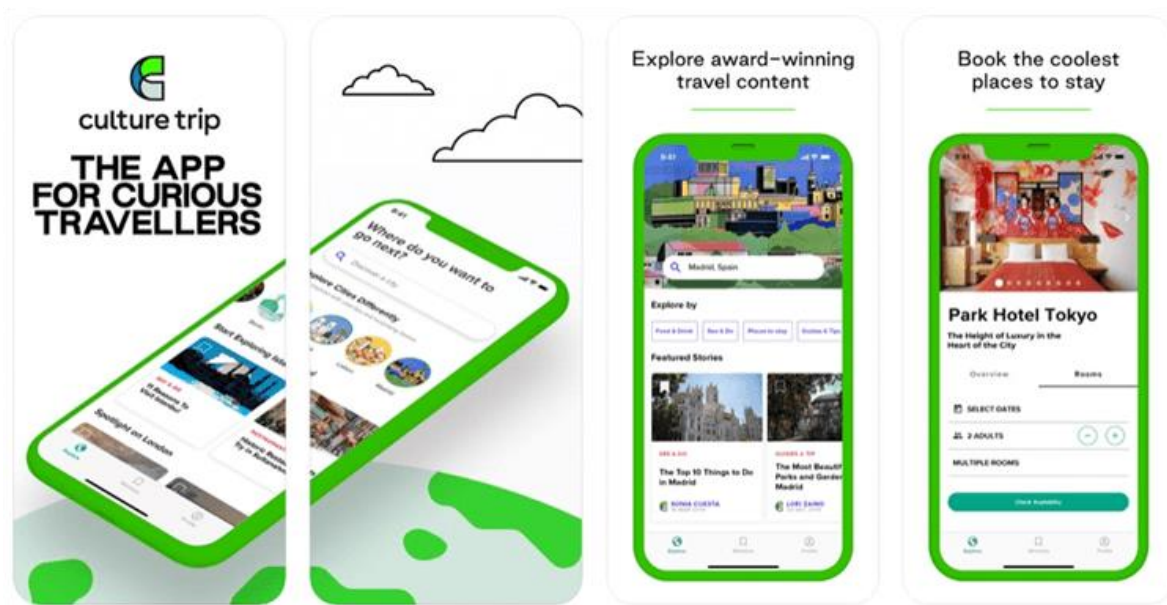
Επιπλέον, το Viator προσφέρει διαδικασίες κράτησης και έκδοσης εισιτηρίων, επιτρέποντας στους χρήστες. Μόλις επιβεβαιωθεί μια κράτηση, οι χρήστες λαμβάνουν άμεση επιβεβαίωση και ηλεκτρονικά εισιτήρια, εξαλείφοντας την ανάγκη για έντυπα εισιτήρια και απλοποιώντας τη διαδικασία σχεδιασμού ταξιδιού. Η Viator παρέχει επίσης υποστήριξη πελατών και βοήθεια ώστε να

εξασφαλίσει μια ομαλή και χωρίς προβλήματα εμπειρία για τους ταξιδιώτες, από τη στιγμή που κάνουν κράτηση για τη δραστηριότητά τους έως τη στιγμή που ξεκινούν την περιπέτειά τους.

1.6. Culture Trip

Η εφαρμογή Culture Trip [6] είναι μια ευέλικτη πλατφόρμα που έχει σχεδιαστεί για να εμπνέει και να εμπλουτίζει τις εμπειρίες των ταξιδιωτών παρέχοντας εξατομικευμένες προτάσεις για αξιοθέατα, δραστηριότητες και πολιτιστικές εκδηλώσεις σε όλο τον κόσμο. Ένα από τα ξεχωριστά χαρακτηριστικά του είναι το επιμελημένο περιεχόμενό του, το οποίο περιλαμβάνει άρθρα, οδηγούς και συστάσεις γραμμένες από τοπικούς εμπειρογνώμονες. Είτε οι χρήστες αναζητούν «κρυμμένους πολύτιμους λίθους», αξιοθέατα ή συμβουλές για δημοφιλείς προορισμούς, και η εφαρμογή Culture Trip προσφέρει πληθώρα διορατικών περιεχομένων για να βοηθήσει τους ταξιδιώτες να ανακαλύψουν μοναδικές και αυθεντικές εμπειρίες όπου κι αν βρίσκονται.

Ένα άλλο αξιοσημείωτο χαρακτηριστικό της εφαρμογής Culture Trip είναι η εξατομικευμένη μηχανή συστάσεων, η οποία αξιοποιεί αλγόριθμους μηχανικής μάθησης για την ανάλυση των προτιμήσεων, των ενδιαφερόντων και της προηγούμενης συμπεριφοράς των χρηστών.



Εικόνα 6: Culture Trip

Κατανοώντας το μοναδικό στυλ ταξιδιού και τις προτιμήσεις κάθε χρήστη, η εφαρμογή παρέχει προσαρμοσμένες προτάσεις. Αυτή η εξατομικευμένη προσέγγιση επιτρέπει στους ταξιδιώτες να εξερευνήσουν νέους προορισμούς με αυτοπεποίθηση και να ανακαλύψουν εμπειρίες που ευθυγραμμίζονται με τα ατομικά πάθη και τις περιέργειές τους.

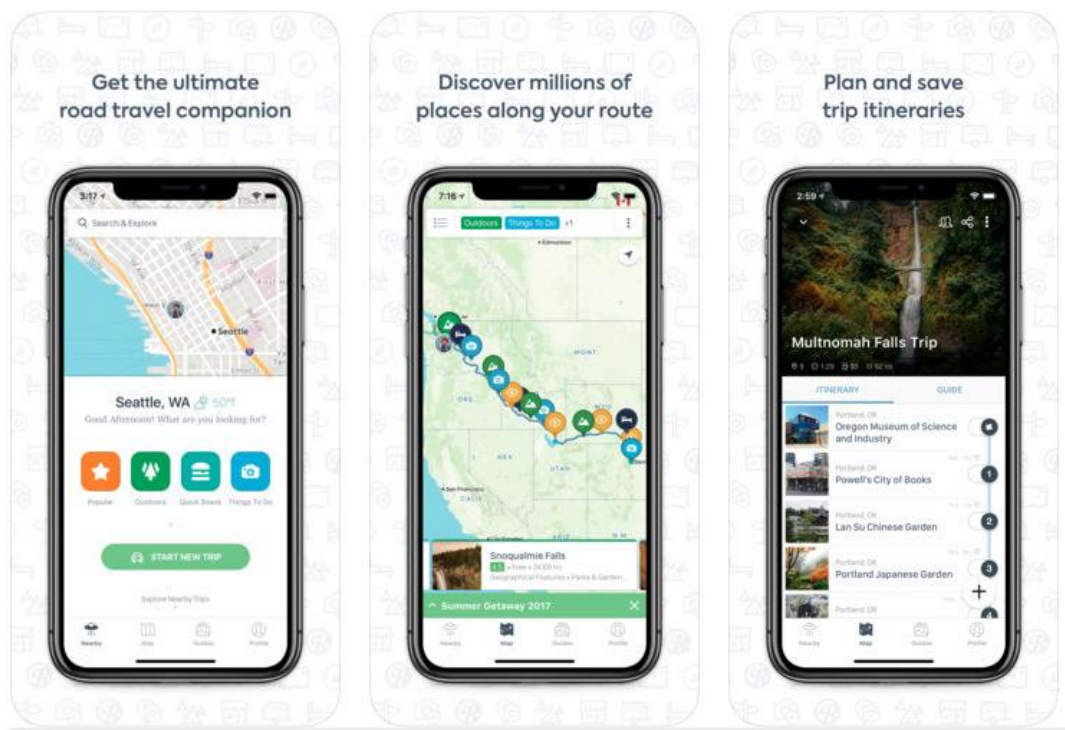
Επιπλέον, η εφαρμογή Culture Trip προσφέρει δυνατότητες κράτησης. Με τη συνεργασία με αξιόπιστους παρόχους τουριστικών υπηρεσιών, η εφαρμογή διασφαλίζει ότι οι χρήστες έχουν πρόσβαση σε υψηλής ποιότητας και αυθεντικές εμπειρίες που παρουσιάζουν τα καλύτερα από κάθε προορισμό. Από συναρπαστικές πολιτιστικές εκδρομές και γαστρονομικές περιπέτειες έως εκθέσεις τέχνης και ζωντανές παραστάσεις, η εφαρμογή Culture Trip παρέχει στους ταξιδιώτες μια ποικιλία επιλογών ώστε να βελτιώσουν τα ταξίδια τους και να δημιουργήσουν μόνιμες αναμνήσεις. Στην ουσία, τα καινοτόμα

χαρακτηριστικά και το περιεχόμενο του Culture Trip δίνουν τη δυνατότητα στους ταξιδιώτες να γνωρίσουν με τρόπο βιωματικό τον τοπικό πολιτισμό, να συνδεθούν με τις κοινότητες και να ανακαλύψουν τον κόσμο με νέους και ουσιαστικούς τρόπους.

1.7. Roadtrippers

Το Roadtrippers [7] είναι μια μοναδική εφαρμογή για κινητά που έχει σχεδιαστεί ώστε να βελτιώνει την εμπειρία του οδικού ταξιδιού για τους ταξιδιώτες παρέχοντας εργαλεία και πόρους για τον προγραμματισμό των εκάστοτε ταξιδιών. Ένα από τα ξεχωριστά χαρακτηριστικά του είναι η διαδραστική διεπαφή σχεδιασμού ταξιδιού, η οποία επιτρέπει στους χρήστες να χαρτογραφούν τις διαδρομές τους, να ανακαλύπτουν σημεία ενδιαφέροντος στην πορεία και να δημιουργούν προσαρμοσμένα δρομολόγια με βάση τα ενδιαφέροντα και τις προτιμήσεις τους.

Ένα άλλο αξιοσημείωτο χαρακτηριστικό του Roadtrippers είναι η ικανότητά του να προτείνει διαδρομές και θεματικά δρομολόγια οδικών ταξιδιών για ταξιδιώτες που αναζητούν έμπνευση και καθοδήγηση. Είτε οι χρήστες ενδιαφέρονται για διαδρομές, γαστρονομικά μονοπάτια, ιστορικές διαδρομές ή υπαίθριες περιπέτειες, η εφαρμογή προσφέρει μια ποικιλία διαδρομών και προτεινόμενων στάσεων που ταιριάζουν σε κάθε γούστο και ενδιαφέρον.



Εικόνα 7: Roadtrippers

Από τα εμβληματικά εθνικά πάρκα και τις γοητευτικές μικρές πόλεις μέχρι τους επισκέπτες του δρόμου και τα αξιοθέατα του δρόμου, η εφαρμογή βοηθά τους ταξιδιώτες να ανακαλύψουν νέους προορισμούς και να δημιουργήσουν αξέχαστες εμπειρίες στην πορεία.

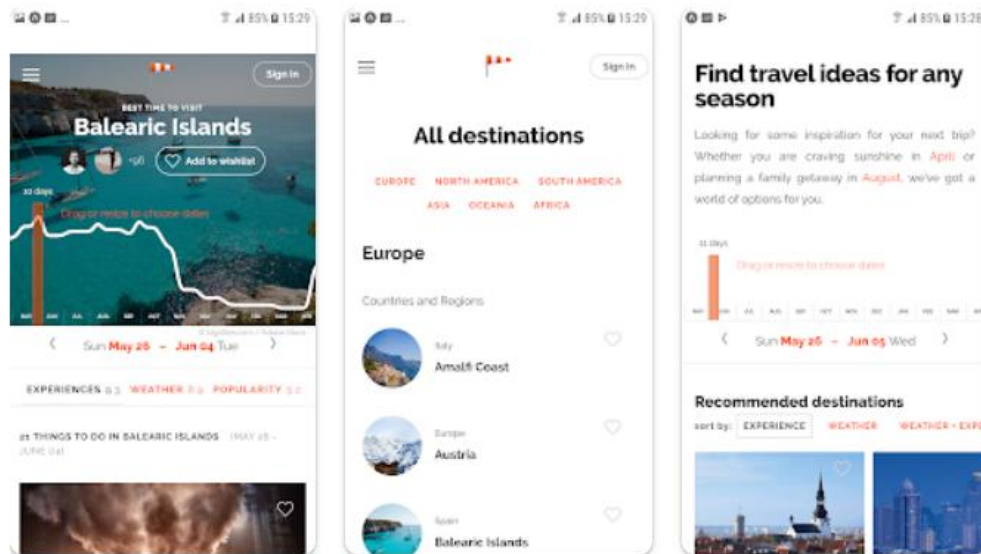
Επιπλέον, το Roadtrippers παρέχει πολύτιμα εργαλεία και πόρους σχεδιασμού ταξιδιών ώστε να βοηθήσει τους χρήστες να οργανώσουν και να βελτιστοποιήσουν τα οδικά ταξίδια τους. Οι χρήστες μπορούν να έχουν πρόσβαση σε πληροφορίες σχετικά με βενζινάδικα, χώρους ανάπαυσης, καταλύματα και επιλογές φαγητού κατά μήκος της διαδρομής τους, καθιστώντας εύκολο τον προγραμματισμό για βασικές ανέσεις και ανάγκες κατά τη διάρκεια του ταξιδιού τους. Επιπλέον, η εφαρμογή προσφέρει

δυνατότητες πλοήγησης σε πραγματικό χρόνο, επιτρέποντας στους χρήστες να περιηγούνται απρόσκοπτα στη διαδρομή τους και να ανακαλύπτουν εύκολα κοντινά αξιοθέατα.

1.8. Rove.me

Η εφαρμογή Rove.me [8] είναι μια ευέλικτη εφαρμογή για κινητά που βοηθά τους ταξιδιώτες να λαμβάνουν αποφάσεις σχετικά με την καλύτερη στιγμή για να επισκεφθούν προορισμούς σε όλο τον κόσμο. Ένα από τα χαρακτηριστικά του είναι ο μοναδικός αλγόριθμός του, ο οποίος αναλύει μια ποικιλία παραγόντων όπως οι καιρικές συνθήκες, τα τοπικά γεγονότα, οι κορυφαίες τουριστικές εποχές και τα φυσικά φαινόμενα ώστε να καθορίσει τον βέλτιστο χρόνο για να γίνει επισκέψιμος ένας προορισμός. Παρέχοντας στους χρήστες πληροφορίες σχετικά με τις πιο ευνοϊκές στιγμές για ταξίδια, η εφαρμογή Rove.me επιτρέπει στους ταξιδιώτες να προγραμματίσουν τα ταξίδια τους στρατηγικά και να μεγιστοποιήσουν τις εμπειρίες τους.

Ένα άλλο αξιοσημείωτο χαρακτηριστικό της Rove.me είναι η ολοκληρωμένη βάση δεδομένων προορισμών και δραστηριοτήτων, η οποία καλύπτει ένα ευρύ φάσμα ενδιαφερόντων και προτιμήσεων. Είτε οι χρήστες ενδιαφέρονται για υπαίθριες περιπέτειες, πολιτιστικά φεστιβάλ, θεάσεις άγριας ζωής, ή εποχιακά αξιοθέατα, η εφαρμογή προσφέρει προτάσεις και πληροφορίες ώστε να βοηθήσει τους ταξιδιώτες να βρουν εμπειρίες που ευθυγραμμίζονται με τα ενδιαφέροντά τους.



Εικόνα 8: Rove.me

Επιπλέον, η εφαρμογή προσφέρει φιλική προς το χρήστη διεπαφή και πλοήγηση, διευκολύνοντας τους ταξιδιώτες να εξερευνήσουν προορισμούς, να ανακαλύψουν επερχόμενες εκδηλώσεις και να έχουν πρόσβαση σε βασικές ταξιδιωτικές πληροφορίες. Η εφαρμογή παρέχει λεπτομερείς περιγραφές, φωτογραφίες και προτάσεις για κάθε προορισμό, επιτρέποντας στους χρήστες να ερευνούν και να σχεδιάζουν τα ταξίδια τους με σιγουριά. Επιπλέον, προσφέρει εξατομικευμένα εργαλεία και δυνατότητες σχεδιασμού ταξιδιού, επιτρέποντας στους χρήστες να δημιουργούν προσαρμοσμένα δρομολόγια, να αποθηκεύουν αγαπημένους προορισμούς και να λαμβάνουν ειδοποιήσεις σχετικά με επερχόμενες εκδηλώσεις και ευκαιρίες εξερεύνησης.

1.9. Επιθυμητά χαρακτηριστικά εφαρμογών πρότασης αξιοθέατων

Εξατομίκευση και προσαρμογή: Η εφαρμογή θα πρέπει να διαθέτει δυνατότητες [9] που επιτρέπουν στους χρήστες να εισάγουν τα ενδιαφέροντα, τις προτιμήσεις και το στυλ ταξιδιού τους. Αυτό θα μπορούσε να περιλαμβάνει επιλογές για την επιλογή συγκεκριμένων κατηγοριών, όπως ιστορικά τοπόσημα, φυσικά τοπία, γαστρονομικές εμπειρίες ή υπαίθριες δραστηριότητες. Η εξατομίκευση διασφαλίζει ότι η εφαρμογή μπορεί να προτείνει αξιοθέατα που ευθυγραμμίζονται με τις μοναδικές προτιμήσεις και ενδιαφέροντα κάθε ταξιδιώτη.

Φιλική διεπαφή και πλοήγηση: Μια φιλική προς το χρήστη διεπαφή και πλοήγηση είναι απαραίτητα για τη βελτίωση της εμπειρίας του χρήστη. Η εφαρμογή θα πρέπει να είναι εύκολη στην πλοήγηση, με σαφείς κατηγορίες, φίλτρα και λειτουργίες αναζήτησης που επιτρέπουν στους χρήστες να εξερευνούν και να ανακαλύπτουν νέα αξιοθέατα χωρίς κόπο. Οι οπτικές ενδείξεις, οι χάρτες και οι διαδραστικές λειτουργίες μπορούν επίσης να βοηθήσουν τους χρήστες να κατανοήσουν την τοποθεσία και το πλαίσιο των προτεινόμενων αξιοθέατων.

Ολοκληρωμένη βάση δεδομένων και συστάσεις: Η εφαρμογή θα πρέπει να διαθέτει μια τεράστια και τακτικά ενημερωμένη βάση δεδομένων με αξιοθέατα, αξιοθέατα και σημεία ενδιαφέροντος παγκοσμίως. Αυτή η βάση δεδομένων θα πρέπει να περιλαμβάνει τόσο δημοφιλείς τουριστικούς προορισμούς όσο και λιγότερο γνωστούς προορισμούς. Η εφαρμογή θα πρέπει να χρησιμοποιεί αλγόριθμους και τεχνικές μηχανικής μάθησης για την ανάλυση δεδομένων

χρήστη και την παροχή σχετικών και εξατομικευμένων συστάσεων με βάση παράγοντες όπως η τοποθεσία, η εποχή του χρόνου και οι προτιμήσεις των χρηστών.

Πλούσιο περιεχόμενο πολυμέσων: Η εφαρμογή θα πρέπει να προσφέρει πλούσιο περιεχόμενο πολυμέσων για κάθε προτεινόμενο θέαμα, συμπεριλαμβανομένων φωτογραφιών, βίντεο, περιγραφών, ιστορικών πληροφοριών και κριτικών χρηστών υψηλής ποιότητας. Αυτό το περιεχόμενο βοηθά τους χρήστες να απεικονίσουν τα αξιοθέατα και να λάβουν τεκμηριωμένες αποφάσεις σχετικά με το ποια θα επισκεφθούν. Το περιεχόμενο που δημιουργείται από χρήστες, όπως κριτικές και προτάσεις, μπορεί επίσης να παρέχει πολύτιμες πληροφορίες και να ενισχύσει την αξιοπιστία των προτάσεων της εφαρμογής.

Πρόσβαση εκτός σύνδεσης: Για να φιλοξενήσει ταξιδιώτες που ενδέχεται να έχουν περιορισμένη ή καθόλου σύνδεση στο διαδίκτυο κατά την εξερεύνηση νέων προορισμών, η εφαρμογή θα πρέπει να προσφέρει πρόσβαση εκτός σύνδεσης σε χάρτες, πληροφορίες και προτάσεις. Οι χρήστες θα πρέπει να μπορούν να κατεβάσουν χάρτες και σχετικό περιεχόμενο για χρήση εκτός σύνδεσης, διασφαλίζοντας ότι μπορούν να έχουν πρόσβαση σε βασικές πληροφορίες ακόμη και σε απομακρυσμένες περιοχές ή ενώ ταξιδεύουν στο εξωτερικό. Επιπλέον, η εφαρμογή θα πρέπει να ενσωματώνεται απρόσκοπτα με άλλες πλατφόρμες και υπηρεσίες που σχετίζονται με ταξίδια, όπως ιστοσελίδες κρατήσεων, εφαρμογές μεταφορών, ώστε να παρέχει μια συνεκτική και εξορθολογισμένη ταξιδιωτική εμπειρία.

1.10. Προκλήσεις εφαρμογών πρότασης αξιοθέατων

Ακρίβεια και αξιοπιστία δεδομένων: Μία από τις σημαντικές προκλήσεις [9] που αντιμετωπίζουν αυτές οι εφαρμογές είναι η διασφάλιση της ακρίβειας και της αξιοπιστίας των δεδομένων που παρέχουν. Τα αξιοθέατα μπορεί να αλλάξουν με την πάροδο του χρόνου λόγω ανακαινίσεων, κλεισίματος ή αλλαγών στις ώρες λειτουργίας. Η διατήρηση ενημερωμένων πληροφοριών απαιτεί συνεχή παρακολούθηση, κάτι που μπορεί να είναι δύσκολο, ειδικά για εφαρμογές με εκτεταμένες βάσεις δεδομένων που καλύπτουν πολλούς προορισμούς παγκοσμίως.

Εξατομίκευση και προτιμήσεις χρηστών: Η παροχή εξατομικευμένων συστάσεων που αντικατοπτρίζουν με ακρίβεια τα ενδιαφέροντα και τις προτιμήσεις κάθε ταξιδιώτη μπορεί να είναι δύσκολη. Οι χρήστες έχουν διαφορετικές προτιμήσεις, καθιστώντας δύσκολη τη δημιουργία αλγορίθμων που προβλέπουν με ακρίβεια ποια αξιοθέατα θα προσελκύσουν συγκεκριμένους χρήστες. Η επίτευξη της σωστής ισορροπίας μεταξύ εξατομικευμένων προτάσεων και γενικών συστάσεων που απευθύνονται σε ένα ευρύτερο κοινό είναι μια συνεχής πρόκληση για αυτές τις εφαρμογές.

Ενσωμάτωση με υπηρεσίες τρίτων: Πολλές ταξιδιωτικές εφαρμογές βασίζονται στην ενσωμάτωση με υπηρεσίες τρίτων, όπως πλατφόρμες κρατήσεων, υπηρεσίες χαρτογράφησης και ιστότοπους αναθεώρησης. Η διασφάλιση της απρόσκοπτης ολοκλήρωσης και διαλειτουργικότητας με αυτές τις υπηρεσίες μπορεί να είναι πολύπλοκη, ειδικά όταν πρόκειται για διαφορετικές μορφές δεδομένων, API και απαιτήσεις συστήματος. Η διατήρηση

της συμβατότητας και της αξιοπιστίας σε διάφορες πλατφόρμες και συσκευές προσθέτει ένα άλλο επίπεδο πρόκλησης για τους προγραμματιστές.

Δημιουργία εσόδων και βιώσιμα επιχειρηματικά μοντέλα: Ενώ η παροχή αξίας στους χρήστες είναι υψίστης σημασίας, οι εφαρμογές για κινητά στον ταξιδιωτικό κλάδο χρειάζονται επίσης βιώσιμες ροές εσόδων για να ευδοκιμήσουν. Η εξισορρόπηση της ανάγκης δημιουργίας εσόδων μέσω διαφημίσεων, premium χαρακτηριστικών ή προμηθειών από κρατήσεις με την παροχή μιας απρόσκοπτης και ευχάριστης εμπειρίας χρήστη μπορεί να είναι δύσκολη. Επιπλέον, ο ανταγωνισμός μεταξύ των ταξιδιωτικών εφαρμογών είναι έντονος.

Απόρρητο και Ασφάλεια Δεδομένων: Οι ταξιδιωτικές εφαρμογές συλλέγουν σημαντικό αριθμό προσωπικών δεδομένων από τους χρήστες, συμπεριλαμβανομένων πληροφοριών τοποθεσίας, προτιμήσεων ταξιδιού και ιστορικού περιήγησης. Η διασφάλιση του απορρήτου και της ασφάλειας αυτών των δεδομένων είναι ζωτικής σημασίας για τη διατήρηση της εμπιστοσύνης των χρηστών και τη συμμόρφωση με τους κανονισμούς περί απορρήτου.

ΚΕΦΑΛΑΙΟ 2: ANDROID ΕΦΑΡΜΟΓΕΣ ΚΑΙ ANDROID STUDIO

2.1. Βασικά συστατικά Android εφαρμογής

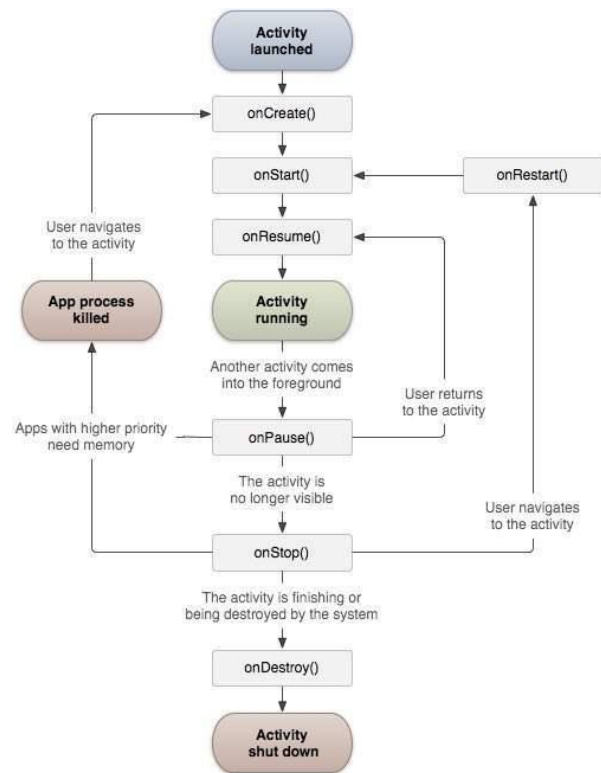
Στα πλαίσια της συγκεκριμένης ενότητας, θα παρουσιαστούν τα βασικά συστατικά μιας Android εφαρμογής, έτσι ώστε να γίνει κατανοητή στην συνέχεια η παράθεση του πρακτικού μέρους, δηλαδή της εφαρμογής για πρόταση αξιοθέατων που υλοποιήθηκε. [10] [11] [12] [13]

2.1.1. Activities

Οι δραστηριότητες (activities) είναι θεμελιώδη συστατικά στοιχεία στην ανάπτυξη μιας Android εφαρμογής, που χρησιμεύουν ως δομικά στοιχεία για τη δημιουργία διεπαφών χρήστη και τη διευκόλυνση των αλληλεπιδράσεων των χρηστών εντός εφαρμογών που αναπτύσσονται χρησιμοποιώντας το Android Studio. Κάθε δραστηριότητα αντιπροσωπεύει μια ενιαία οθόνη με μια διεπαφή χρήστη, με την οποία οι χρήστες μπορούν να αλληλεπιδράσουν για να εκτελέσουν συγκεκριμένες εργασίες.

Ο κύκλος ζωής μιας δραστηριότητας διέπει τη συμπεριφορά της από τη δημιουργία έως την καταστροφή, επιτρέποντας στους προγραμματιστές να διαχειρίζονται διάφορες καταστάσεις και να ανταποκρίνονται αποτελεσματικά στις αλληλεπιδράσεις των χρηστών. Ο κύκλος ζωής αποτελείται από διάφορες βασικές μεθόδους όπως: onCreate(), onStart(), onResume(), onPause(), onStop ()

και `onDestroy()`. Αυτές οι μέθοδοι ονομάζονται ως μεταβάσεις μεταξύ διαφορετικών καταστάσεων, δίνοντας στους προγραμματιστές ευκαιρίες να αρχικοποιήσουν πόρους, να ενημερώσουν το UI, να αποθηκεύσουν πληροφορίες κατάστασης και να απελευθερώσουν πόρους όπως απαιτείται.



Εικόνα 9: Κύκλος ζωής δραστηριότητας Android

Στο Android Studio, οι προγραμματιστές σχεδιάζουν διατάξεις δραστηριότητας χρησιμοποιώντας αρχεία XML στον κατάλογο `res/layout`, καθορίζοντας τη διάταξη και την εμφάνιση στοιχείων UI όπως κουμπιά, προβολές κειμένου, πεδία εισαγωγής και εικόνες. Μέσω του προγράμματος επεξεργασίας διάταξης (layout editor) που παρέχεται από το Android Studio, οι προγραμματιστές μπορούν να σχεδιάσουν οπτικά δραστηριότητες και να τις προβάλουν σε

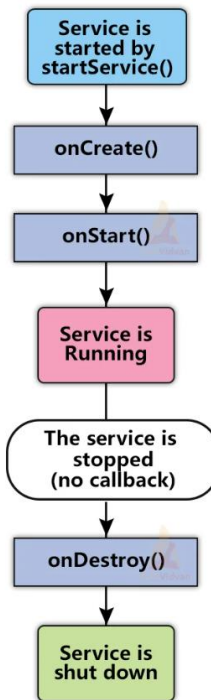
πραγματικό χρόνο, απλοποιώντας τη διαδικασία δημιουργίας ελκυστικών διεπαφών χρήστη.

Επιπλέον, οι δραστηριότητες μπορούν να επικοινωνούν μεταξύ τους και να μεταδίδουν δεδομένα χρησιμοποιώντας Intents, τα οποία είναι αντικείμενα ανταλλαγής μηνυμάτων που χρησιμοποιούνται για την ενεργοποίηση ενεργειών ή τη μεταφορά πληροφοριών μεταξύ διαφορετικών στοιχείων εντός μιας εφαρμογής ή μεταξύ εφαρμογών. Τα Intents διευκολύνουν την απρόσκοπτη πλοήγηση μεταξύ των δραστηριοτήτων. Με την κατανόηση και την αξιοποίηση της δύναμης των δραστηριοτήτων στο Android Studio, οι προγραμματιστές μπορούν να δημιουργήσουν δυναμικές εφαρμογές που ανταποκρίνονται στις ποικίλες ανάγκες και προτιμήσεις των χρηστών Android.

2.1.2. Services

Οι υπηρεσίες (services) είναι ένα κρίσιμο στοιχείο στην ανάπτυξη Android εφαρμογών, επιτρέποντας στις εφαρμογές να εκτελούν μακροχρόνιες λειτουργίες ή εργασίες στο παρασκήνιο χωρίς να απαιτείται αλληλεπίδραση χρήστη. Σε αντίθεση με τις δραστηριότητες, οι υπηρεσίες δεν έχουν διεπαφή χρήστη και συνήθως εκτελούνται στο παρασκήνιο, επιτρέποντας στις εφαρμογές να συνεχίσουν να λειτουργούν ακόμη και όταν ο χρήστης μεταβαίνει σε άλλη εφαρμογή ή η συσκευή βρίσκεται σε διαφορετική κατάσταση. Στις εφαρμογές Android που αναπτύσσονται χρησιμοποιώντας το Android Studio, οι υπηρεσίες υλοποιούνται ως υποκλάσεις της κλάσης Service που παρέχεται από το Android SDK. Μπορούν να εκτελέσουν διάφορες εργασίες, όπως αναπαραγωγή μουσικής, λήψη αρχείων ή λήψη δεδομένων από το Διαδίκτυο.

Υπάρχουν δύο κύριοι τύποι υπηρεσιών στο Android: υπηρεσίες προσκηνίου και υπηρεσίες παρασκηνίου. Οι υπηρεσίες προσκηνίου εμφανίζουν μια ειδοποίηση στον χρήστη, υποδεικνύοντας ότι η υπηρεσία εκτελείται και εκτελεί μια συγκεκριμένη εργασία. Αυτές οι υπηρεσίες θεωρούνται υψηλής προτεραιότητας και θα πρέπει να χρησιμοποιούνται για εργασίες που οι χρήστες γνωρίζουν ενεργά, όπως η αναπαραγωγή μουσικής ή η εκτέλεση λήψης αρχείου. Οι υπηρεσίες παρασκηνίου, από την άλλη πλευρά, δεν εμφανίζουν ειδοποίηση και χρησιμοποιούνται για εργασίες που είναι λιγότερο κρίσιμες ή αισθητές στον χρήστη.



Εικόνα 10: Κύκλος ζωής Service

Ο κύκλος ζωής μιας υπηρεσίας στο Android αποτελείται από διάφορες βασικές μεθόδους, συμπεριλαμβανομένων των `onCreate()`, `onStartCommand()` και `onDestroy()`. Η μέθοδος `onCreate()` καλείται όταν δημιουργείται για πρώτη φορά η υπηρεσία, επιτρέποντας στους προγραμματιστές να εκτελούν εργασίες

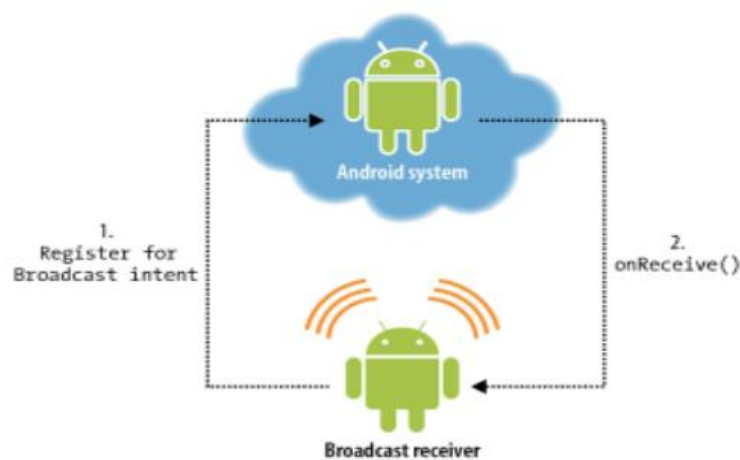
αρχικοποίησης. Η μέθοδος `onStartCommand()` καλείται κάθε φορά που ξεκινά η υπηρεσία και λαμβάνει ένα `Intent` που περιέχει πληροφορίες σχετικά με την ζητούμενη λειτουργία. Τέλος, η μέθοδος `onDestroy()` χρησιμοποιείται όταν η υπηρεσία δεν είναι πλέον απαραίτητη, επιτρέποντας στους προγραμματιστές να απελευθερώσουν πόρους ή να καθαρίσουν τυχόν τρέχουσες εργασίες.

2.1.3. Broadcast receivers

Οι δέκτες μετάδοσης (broadcast receivers) αποτελούν ένα επιπλέον θεμελιώδες στοιχείο της πλατφόρμας Android, επιτρέποντας στις εφαρμογές να λαμβάνουν και να ανταποκρίνονται σε ανακοινώσεις μετάδοσης σε όλο το σύστημα ή σε προσαρμοσμένα μηνύματα μετάδοσης που αποστέλλονται από άλλες εφαρμογές ή το ίδιο το σύστημα. Στην ανάπτυξη Android εφαρμογών, οι δέκτες μετάδοσης χρησιμοποιούνται για να ακούνε και να αντιδρούν σε διάφορα γεγονότα και intents, επιτρέποντας στους προγραμματιστές να ενεργοποιούν συγκεκριμένες ενέργειες ή να ανταποκρίνονται σε αλλαγές στην κατάσταση της συσκευής.

Οι broadcast receivers λειτουργούν ανεξάρτητα από τη διεπαφή χρήστη και συνήθως υλοποιούνται ως υποκλάσεις της κλάσης `BroadcastReceiver` που παρέχεται από το Android SDK. Μπορούν να καταχωρηθούν είτε στατικά στο `AndroidManifest.xml` αρχείο ή δυναμικά κατά το χρόνο εκτέλεσης χρησιμοποιώντας τη μέθοδο `registerReceiver()`. Μόλις εγγραφούν, οι δέκτες μετάδοσης λαμβάνουν intents, επιτρέποντάς τους να ανταποκρίνονται σε συγκεκριμένους τύπους εκπομπών.

Υπάρχουν δύο κύριοι τύποι εκπομπών στο Android: εκπομπές συστήματος και προσαρμοσμένες εκπομπές. Οι εκπομπές συστήματος αποστέλλονται από το σύστημα Android για να ειδοποιούν τις εφαρμογές για συμβάντα συστήματος όπως εκκίνηση συσκευής, αλλαγές συνδεσιμότητας δικτύου, προειδοποιήσεις χαμηλής μπαταρίας και εισερχόμενα μηνύματα SMS. Οι προσαρμοσμένες εκπομπές, από την άλλη πλευρά, είναι εκπομπές που αποστέλλονται από εφαρμογές για επικοινωνία με άλλα στοιχεία εντός της ίδιας εφαρμογής ή με εξωτερικές εφαρμογές εγκατεστημένες στη συσκευή.



Εικόνα 11: Χρήση broadcast receiver

Ο κύκλος ζωής ενός δέκτη εκπομπής αποτελείται από μια ενιαία μέθοδο, την `onReceive()`, η οποία καλείται από το σύστημα Android όταν ο δέκτης ενεργοποιείται από ένα `intent` εισερχόμενης μετάδοσης. Μέσα στη μέθοδο `onReceive()`, οι προγραμματιστές μπορούν να γράψουν κώδικα για να χειριστούν την εκπομπή και να εκτελέσουν τις απαραίτητες ενέργειες ή να ξεκινήσουν περαιτέρω επεξεργασία.

Αξιοποιώντας τους δέκτες μετάδοσης, οι προγραμματιστές μπορούν να δημιουργήσουν ισχυρές και ευέλικτες εφαρμογές που ανταποκρίνονται δυναμικά στις αλλαγές στο περιβάλλον της συσκευής ή σε συμβάντα που ενεργοποιούνται από άλλες εφαρμογές. Οι δέκτες μετάδοσης διαδραματίζουν κρίσιμο ρόλο στην ενεργοποίηση της επικοινωνίας και του συντονισμού μεταξύ εφαρμογών, επιτρέποντας στους προγραμματιστές να δημιουργήσουν ισχυρές και διαδραστικές εφαρμογές Android που ανταποκρίνονται στις ποικίλες ανάγκες των χρηστών. Η κατανόηση του τρόπου αποτελεσματικής χρήσης των δεκτών εκπομπής είναι απαραίτητη για τη δημιουργία εφαρμογών που ανταποκρίνονται και είναι φιλικές προς το χρήστη στο οικοσύστημα Android.

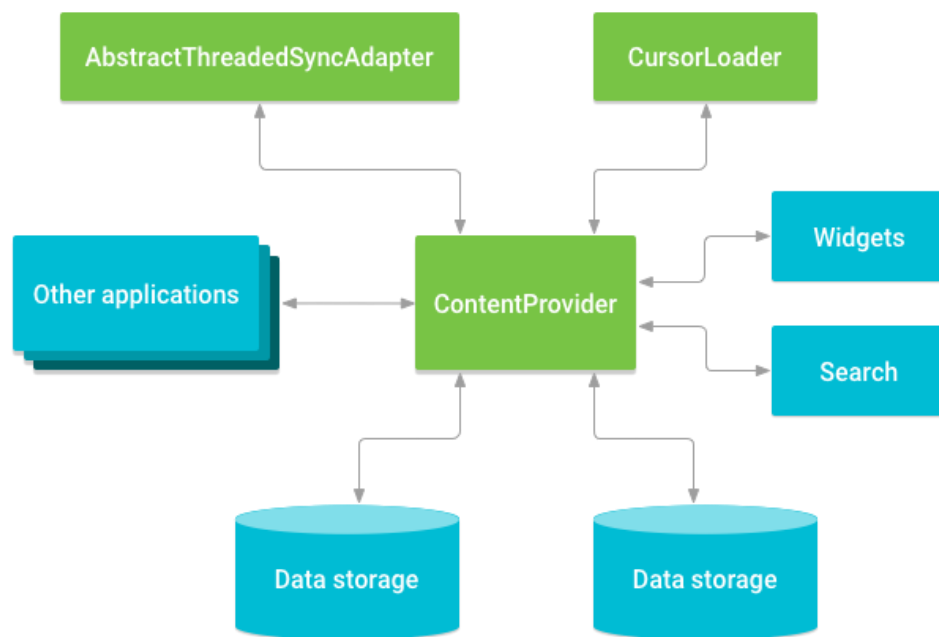
2.1.4. Content providers

Οι πάροχοι περιεχομένου (content providers) αποτελούν ζωτικό στοιχείο του Android framework, διευκολύνοντας την ασφαλή και δομημένη πρόσβαση σε κοινά δεδομένα μεταξύ διαφορετικών εφαρμογών. Λειτουργούν ως διεπαφή για τη διαχείριση και την ανάκτηση δεδομένων που είναι αποθηκευμένα σε ένα κεντρικό αποθετήριο, επιτρέποντας στις εφαρμογές να αλληλεπιδρούν με διάφορους τύπους δεδομένων, συμπεριλαμβανομένων επαφών, αρχείων πολυμέσων, συμβάντων ημερολογίου και προτιμήσεων χρήστη.

Στην ανάπτυξη εφαρμογών Android, οι πάροχοι περιεχομένου υλοποιούνται ως υποκλάσεις της κλάσης `ContentProvider` που παρέχεται από το Android SDK. Ενσωματώνουν τα υποκείμενα δεδομένα και τα εκθέτουν σε άλλες εφαρμογές μέσω ενός συνόλου τυπικών λειτουργιών CRUD (δημιουργία, ανάγνωση, ενημέρωση, διαγραφή). Οι πάροχοι περιεχομένου επιβάλλουν δικαιώματα πρόσβασης σε δεδομένα και παρέχουν έναν τυποποιημένο τρόπο για τις

εφαρμογές να αναζητούν, να εισάγουν, να ενημερώνουν και να διαγράφουν δεδομένα χρησιμοποιώντας URI περιεχομένου.

Οι πάροχοι περιεχομένου χρησιμοποιούν URI (Uniform Resource Identifier) περιεχομένου για τον μοναδικό προσδιορισμό δεδομένων εντός του παρόχου και για τον προσδιορισμό του τύπου της λειτουργίας που εκτελείται. Κάθε URI περιεχομένου αποτελείται από μια αρχή περιεχομένου (content authority) και μια διαδρομή που προσδιορίζει μοναδικά ένα συγκεκριμένο σύνολο δεδομένων. Τα URI περιεχομένου συμβάλλουν στη διασφάλιση της ακεραιότητας και της ασφάλειας των δεδομένων, επιτρέποντας στους παρόχους περιεχομένου να ελέγχουν την πρόσβαση σε συγκεκριμένα σύνολα δεδομένων και να επιβάλλουν δικαιώματα βάσει της αιτούσας εφαρμογής.



Εικόνα 12: ContentProvider

Το σύστημα Android περιλαμβάνει αρκετούς ενσωματωμένους παρόχους περιεχομένου για πρόσβαση σε κοινούς τύπους δεδομένων, όπως επαφές, αρχεία πολυμέσων και συμβάντα ημερολογίου. Επιπλέον, οι προγραμματιστές μπορούν να δημιουργήσουν προσαρμοσμένους παρόχους περιεχομένου για να εκθέσουν δεδομένα ειδικά για εφαρμογές σε άλλα στοιχεία ή εξωτερικές εφαρμογές.

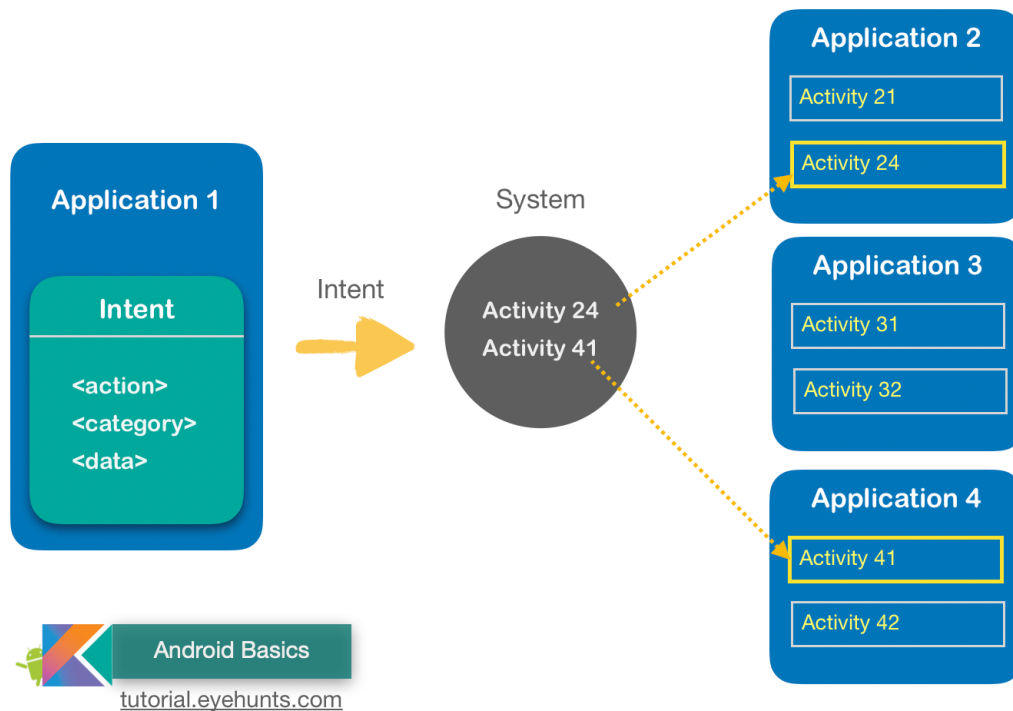
Αξιοποιώντας τους παρόχους περιεχομένου, οι προγραμματιστές μπορούν να δημιουργήσουν αρθρωτές και διαλειτουργικές εφαρμογές που μοιράζονται με ασφάλεια δεδομένα και παρέχουν μια συνεπή εμπειρία χρήστη σε διαφορετικές συσκευές και εφαρμογές Android.

2.1.5. Intents

Τα intents είναι μια επιπλέον θεμελιώδης έννοια στην ανάπτυξη Android εφαρμογών, που χρησιμεύει ως μηχανισμός ανταλλαγής μηνυμάτων που επιτρέπει την επικοινωνία μεταξύ διαφορετικών στοιχείων εντός μιας εφαρμογής ή μεταξύ διαφορετικών εφαρμογών στην πλατφόρμα Android. Τα intents διευκολύνουν την ανταλλαγή δεδομένων και ενεργοποιούν διάφορες ενέργειες, επιτρέποντας στους προγραμματιστές να ξεκινούν δραστηριότητες, να ξεκινούν υπηρεσίες, να μεταδίδουν μηνύματα και να εκτελούν επικοινωνία μεταξύ εφαρμογών.

Στις εφαρμογές Android που αναπτύσσονται χρησιμοποιώντας το Android Studio, τα intents αντιπροσωπεύονται από την κλάση Intent που παρέχεται από το Android SDK. Ένα intent είναι μια αφηρημένη περιγραφή μιας λειτουργίας που πρέπει να εκτελεστεί, ενσωματώνοντας το intent να εκτελέσει μια

συγκεκριμένη ενέργεια, όπως η έναρξη μιας δραστηριότητας, η έναρξη μιας υπηρεσίας ή η μετάδοση ενός μηνύματος.

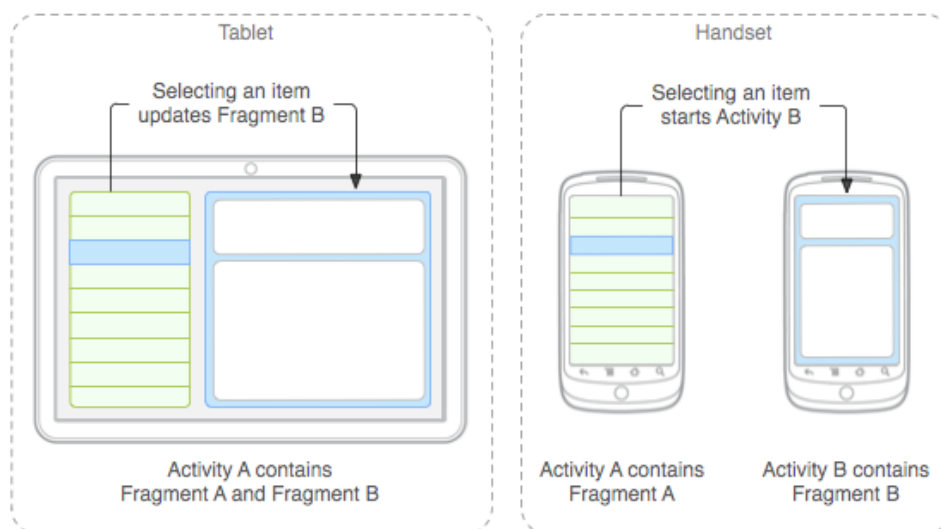


Εικόνα 13: Χρήση Intent

Υπάρχουν δύο κύριοι τύποι intents στο Android: explicit intents και implicit intents. Τα explicit intents καθορίζουν το στοιχείο-στόχο ονομάζοντας ρητά την κλάση συνιστωσών που θα κληθεί. Συνήθως χρησιμοποιούνται για την εκκίνηση στοιχείων εντός της ίδιας εφαρμογής, όπως η έναρξη μιας συγκεκριμένης δραστηριότητας ή υπηρεσίας. Τα σιωπηρά intents, από την άλλη πλευρά, δεν καθορίζουν ρητά ένα στοιχείο-στόχο. Τα intents μπορούν επίσης να μεταφέρουν δεδομένα μαζί τους χρησιμοποιώντας πρόσθετα, τα οποία είναι ζεύγη κλειδιών-τιμών που αποθηκεύονται εντός του intent.

2.1.6. Fragments

Τα fragments προσφέρουν μια ευέλικτη και αρθρωτή προσέγγιση για τη δημιουργία διεπαφών χρήστη που μπορούν να προσαρμοστούν σε διαφορετικά μεγέθη και προσανατολισμούς οθόνης. Ουσιαστικά, τα fragments αντιπροσωπεύουν επαναχρησιμοποιήσιμα τμήματα της διεπαφής χρήστη μέσα σε μια δραστηριότητα, επιτρέποντας στους προγραμματιστές να δημιουργούν δυναμικές και ανταποκρινόμενες διατάξεις που μπορούν να προσαρμοστούν και να αναδιαταχθούν με βάση τις αλληλεπιδράσεις των χρηστών και τις διαμορφώσεις των συσκευών.



Εικόνα 14: Fragments

Στο Android Studio, τα fragments υλοποιούνται ως υποκατηγορίες της κλάσης Fragment που παρέχονται από το Android SDK. Ενσωματώνουν τα στοιχεία και τη λογική του UI, επιτρέποντας στους προγραμματιστές να κατασκευάσουν αρθρωτά και επαναχρησιμοποιήσιμα στοιχεία που μπορούν να συνδυαστούν και να διαχειριστούν μέσα σε δραστηριότητες για τη δημιουργία σύνθετων διεπαφών χρήστη. Τα fragments έχουν τον δικό τους κύκλο ζωής και μπορούν να

περιέχουν τα δικά τους στοιχεία διεπαφής χρήστη, χειριστές συμβάντων και λογική διαχείρισης δεδομένων.

Επιπροσθέτως, τα fragments προσφέρουν πολλά πλεονεκτήματα σε σχέση με τις παραδοσιακές δραστηριότητες, συμπεριλαμβανομένης της βελτιωμένης οργάνωσης κώδικα, της βελτιωμένης επαναχρησιμοποίησης και της καλύτερης υποστήριξης για τη δημιουργία ανταποκρινόμενων διαμορφώσεων. Διαιρώντας τη διεπαφή χρήστη σε μικρότερα, αυτόνομα fragments, οι προγραμματιστές μπορούν να ενσωματώσουν τη λειτουργικότητα και τη συμπεριφορά πιο αποτελεσματικά, διευκολύνοντας τη διαχείριση και τη συντήρηση σύνθετων εφαρμογών. Επιπλέον, τα fragments επιτρέπουν στους προγραμματιστές να δημιουργούν διατάξεις πολλαπλών παραθύρων για tablet και συσκευές μεγαλύτερης οθόνης, επιτρέποντας στους χρήστες να αλληλεπιδρούν με πολλά fragments ταυτόχρονα.

2.1.7. Αρχείο Manifest

Το αρχείο AndroidManifest.xml χρησιμεύει ως σχέδιο που παρέχει βασικές πληροφορίες σχετικά με την εφαρμογή στο λειτουργικό σύστημα Android. Λειτουργεί ως αρχείο ρυθμίσεων που καθορίζει τη δομή, τα στοιχεία, τα δικαιώματα και τις δυνατότητες της εφαρμογής, διασφαλίζοντας τη σωστή ενσωμάτωση με την πλατφόρμα Android και επιτρέποντας στο σύστημα να αλληλεπιδρά αποτελεσματικά με την εφαρμογή. Επιπλέον, το αρχείο αυτό βρίσκεται στον ριζικό κατάλογο του έργου και περιέχει μεταδεδομένα σχετικά με την εφαρμογή, συμπεριλαμβανομένου του ονόματος πακέτου, του αριθμού έκδοσης, των ελάχιστων και στοχευμένων εκδόσεων SDK και των δικαιωμάτων που απαιτούνται για την πρόσβαση σε πόρους συστήματος και την εκτέλεση

συγκεκριμένων λειτουργιών. Λειτουργεί ως κεντρικός κόμβος που οργανώνει και συντονίζει τα διάφορα στοιχεία και χαρακτηριστικά της εφαρμογής, επιτρέποντας στο σύστημα να διαχειρίζεται ανάλογα τον κύκλο ζωής και τη συμπεριφορά του.

```
1<?xml version="1.0" encoding="utf-8"?>
2<manifest xmlns:android="http://schemas.android.com/apk/res/android"
3    package="com.androidapp.basicElements"
4    android:versionCode="1"
5    android:versionName="1.0">
6    <application android:icon="@drawable/icon" android:label="@string/app_name">
7        <activity android:name=".BasicElements"
8            android:label="@string/app_name">
9            <intent-filter>
10                <action android:name="android.intent.action.MAIN" />
11                <category android:name="android.intent.category.LAUNCHER" />
12            </intent-filter>
13        </activity>
14    </application>
15    <uses-sdk android:minSdkVersion="2" />
16
17
18</manifest>
```

Εικόνα 15: Παράδειγμα AndroidManifest.xml

Επιπροσθέτως, ορίζει τα σημεία εισόδου και τις ρυθμίσεις διαμόρφωσης για τα στοιχεία της εφαρμογής, συμπεριλαμβανομένων των δραστηριοτήτων, των υπηρεσιών, των δεκτών εκπομπής και των παρόχων περιεχομένου. Επιπλέον, το αρχείο αυτό χρησιμεύει ως μηχανισμός ασφαλείας που ελέγχει την πρόσβαση σε ευαίσθητους πόρους και διασφαλίζει ότι μόνο οι εξουσιοδοτημένες εφαρμογές μπορούν να αλληλεπιδρούν με ορισμένες λειτουργίες ή δεδομένα στη συσκευή.

2.2. Δυνατότητες Android Studio

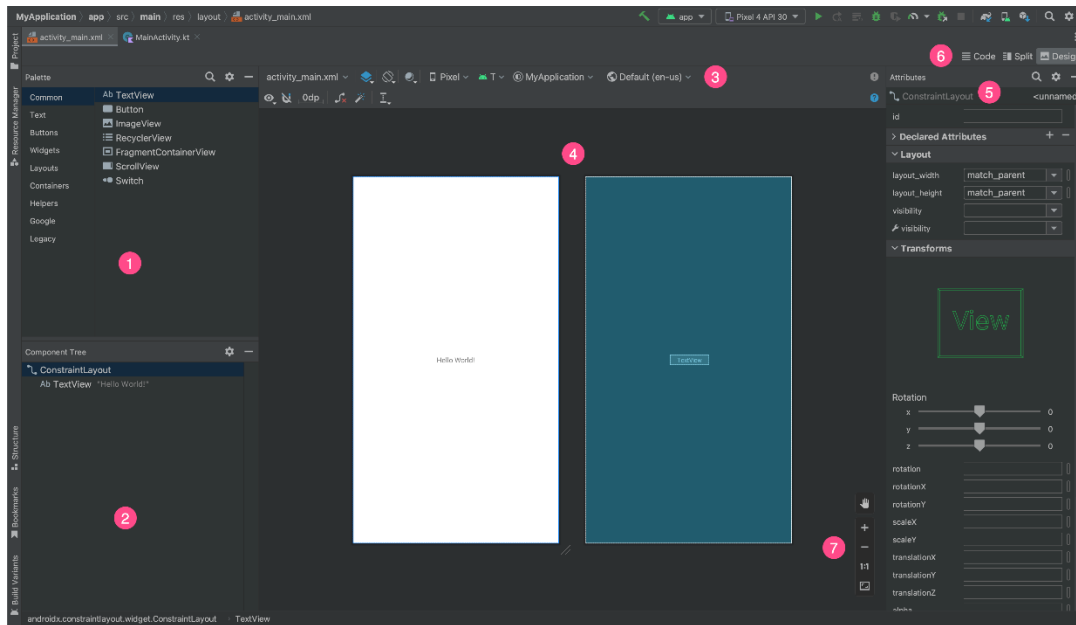
Στα πλαίσια της συγκεκριμένης ενότητας, θα παρουσιαστούν οι βασικές δυνατότητες του εργαλείου Android Studio [14] [15] [16] [17], το οποίο χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής στα πλαίσια της συγκεκριμένης εργασίας:

2.2.1. Σχεδιαστής Διεπαφής Χρήστη

Ο σχεδιαστής διεπαφής χρήστη, που συχνά αναφέρεται ως επεξεργαστής διάταξης υποβοηθά τους προγραμματιστές να κατασκευάσουν οπτικά τη διεπαφή χρήστη των εφαρμογών τους. Προσφέρει μια διεπαφή μεταφοράς και απόθεσης (drag and drop), επιτρέποντας στους προγραμματιστές να προσθέτουν, να τοποθετούν και να διαμορφώνουν απρόσκοπτα στοιχεία διεπαφής χρήστη, όπως κουμπιά, πεδία κειμένου, εικόνες και κοντέινερ διάταξης. Αυτή η προσέγγιση εξαλείφει την ανάγκη για προγραμματιστές να γράφουν χειροκίνητα κώδικα XML, καθιστώντας τη διαδικασία σχεδιασμού UI πιο προσιτή και αποτελεσματική. Πρόκειται για μια διαδικασία η οποία ακολουθήθηκε και στα πλαίσια της συγκεκριμένης εργασίας.

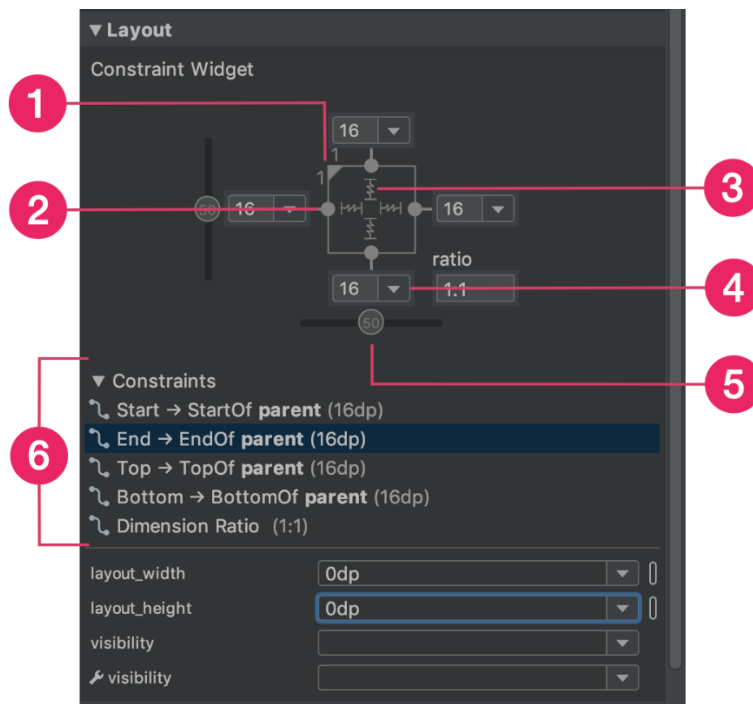
Ένα από τα βασικά χαρακτηριστικά του σχεδιαστή UI είναι η υποστήριξη του για το ConstraintLayout, έναν ισχυρό διαχειριστή διάταξης που επιτρέπει στους προγραμματιστές να δημιουργούν δυναμικά user interfaces. Το ConstraintLayout επιτρέπει στους προγραμματιστές να ορίζουν σχέσεις (περιορισμούς) μεταξύ των στοιχείων UI, διασφαλίζοντας ότι η διάταξη προσαρμόζεται αποτελεσματικά σε διαφορετικά μεγέθη οθόνης και προσανατολισμούς. Μέσω του Attributes Panel, οι προγραμματιστές μπορούν να προσαρμόσουν διάφορες

ιδιότητες των στοιχείων του UI, συμπεριλαμβανομένου του μεγέθους, των περιθωρίων, της εμφάνισης κειμένου και του χρώματος φόντου, παρέχοντας λεπτομερή έλεγχο των οπτικών πτυχών του user interface της εφαρμογής.



Εικόνα 16: User Interface Designer

Επιπλέον, ο σχεδιαστής UI του Android Studio περιλαμβάνει λειτουργίες όπως προεπισκόπηση σε πραγματικό χρόνο σε διαφορετικά μεγέθη οθόνης και προσανατολισμούς, επιτρέποντας στους προγραμματιστές να απεικονίσουν πώς θα εμφανίζονται οι διατάξεις τους σε διάφορες συσκευές. Ο επεξεργαστής θεμάτων παρέχει μια βολική διεπαφή για την προσαρμογή θεμάτων και στυλ οπτικά, με αλλαγές που αντικατοπτρίζονται άμεσα στον σχεδιαστή διεπαφής χρήστη. Επιπλέον, ο έλεγχος προσβασιμότητας βοηθά τους προγραμματιστές να εντοπίζουν και να αντιμετωπίζουν προβλήματα προσβασιμότητας στο περιβάλλον της εφαρμογής τους, διασφαλίζοντας ότι η εφαρμογή λειτουργεί χωρίς αποκλεισμούς και μπορεί να χρησιμοποιηθεί για όλους τους χρήστες.



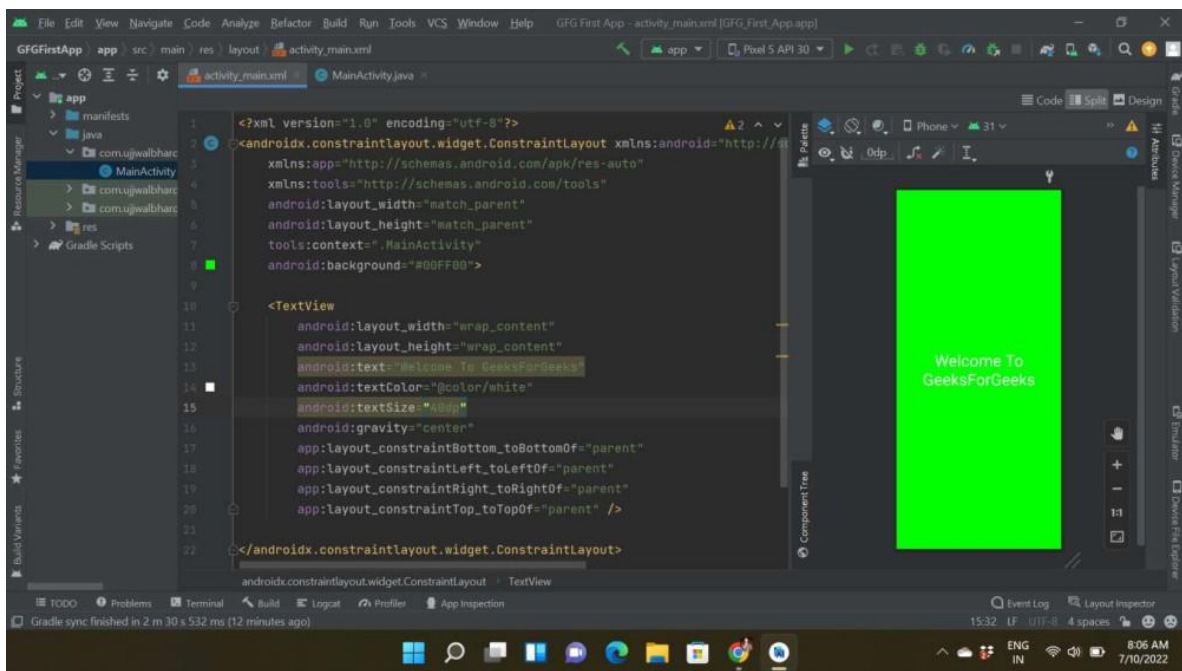
Εικόνα 17: Χρήση ConstraintLayout

2.2.2. Επεξεργαστής κώδικα

Ο επεξεργαστής κώδικα (code editor) στο Android Studio χρησιμεύει ως ο κύριος χώρος εργασίας για τους προγραμματιστές ώστε να γράφουν, να επεξεργάζονται και να διαχειρίζονται τον πηγαίο κώδικα των εφαρμογών τους. Παρέχει ένα πλούσιο σε χαρακτηριστικά περιβάλλον που υποστηρίζει πολλαπλές γλώσσες προγραμματισμού, συμπεριλαμβανομένων των Java και Kotlin, οι οποίες είναι οι δύο κύριες γλώσσες που χρησιμοποιούνται για την ανάπτυξη εφαρμογών Android. Ο επεξεργαστής κώδικα προσφέρει βασικές λειτουργίες όπως επισήμανση σύνταξης, ολοκλήρωση κώδικα και πλοήγηση κώδικα, οι οποίες ενισχύουν την παραγωγικότητα των προγραμματιστών

παρέχοντας προτάσεις που γνωρίζουν το πλαίσιο και επισημαίνοντας σφάλματα σύνταξης σε πραγματικό χρόνο.

Ένα από τα ξεχωριστά χαρακτηριστικά του επεξεργαστή κώδικα είναι η υποστήριξή του για έξυπνη ολοκλήρωση κώδικα, η οποία βοηθά τους προγραμματιστές να γράφουν κώδικα πιο γρήγορα και με λιγότερα σφάλματα. Ο επεξεργαστής κώδικα αναλύει το πλαίσιο του κώδικα που γράφεται και προτείνει σχετικά αποσπάσματα κώδικα, ονόματα μεταβλητών, υπογραφές μεθόδων και δηλώσεις εισαγωγής.



Εικόνα 18: Code Editor

Επιπλέον, ο επεξεργαστής κώδικα στο Android Studio περιλαμβάνει ισχυρά εργαλεία refactoring που επιτρέπουν στους προγραμματιστές να αναδιαρθρώσουν και να βελτιστοποιήσουν αποτελεσματικά τον κώδικα τους. Οι λειτουργίες αναδιαμόρφωσης όπως η αλλαγή ονομασίας μεταβλητών, οι

μέθοδοι εξαγωγής και η αναδιάταξη των μπλοκ κώδικα μπορούν να εκτελεστούν με ευκολία, χωρίς να διακυβεύεται η ακεραιότητα του κώδικα. Επιπλέον, ο επεξεργαστής κώδικα ενσωματώνεται απρόσκοπτα με συστήματα ελέγχου εκδόσεων όπως το Git, επιτρέποντας στους προγραμματιστές να παρακολουθούν τις αλλαγές, να ελέγχουν το ιστορικό κώδικα και να συνεργάζονται αποτελεσματικά με τα μέλη της ομάδας.

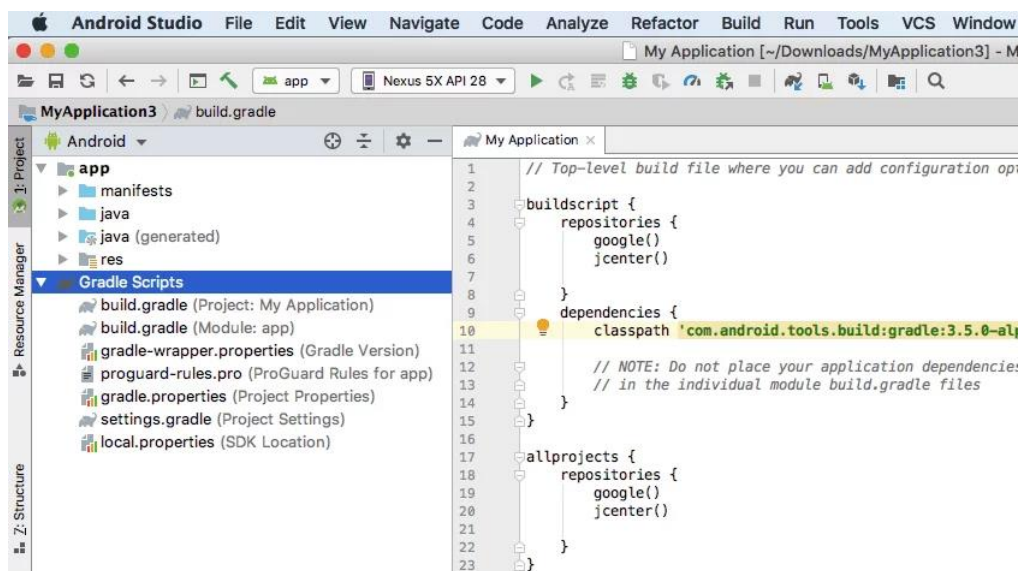
Συνολικά, ο επεξεργαστής κώδικα στο Android Studio παρέχει ένα ευέλικτο και παραγωγικό περιβάλλον για τους προγραμματιστές ώστε να γράφουν κώδικα υψηλής ποιότητας, να εξορθολογίζουν τις ροές εργασίας και να δημιουργούν ισχυρές εφαρμογές Android.

2.2.3. Gradle Build System

Το Gradle Build System, αναπόσπαστο μέρος του Android Studio, είναι ένα ευέλικτο και ισχυρό εργαλείο αυτοματισμού που χρησιμοποιείται για τη διαχείριση της διαδικασίας δημιουργίας εφαρμογών Android και άλλων έργων λογισμικού. Χρησιμοποιεί ένα Groovy-based DSL (Domain Specific Language) που επιτρέπει στους προγραμματιστές να ορίζουν διαμορφώσεις έργων, εξαρτήσεις και εργασίες σε μια συνοπτική και ευανάγνωστη μορφή. Το Gradle παρέχει ένα ευέλικτο και επεκτάσιμο πλαίσιο κατασκευής που μπορεί να φιλοξενήσει διαφορετικές δομές και απαιτήσεις έργου, καθιστώντας το κατάλληλο τόσο για απλά όσο και για σύνθετα σενάρια ανάπτυξης.

Ένα από τα βασικά πλεονεκτήματα του συστήματος Gradle Build είναι η υποστήριξή του για τη διαχείριση εξάρτησης, η οποία απλοποιεί την ενσωμάτωση βιβλιοθηκών σε έργα Android. Οι προγραμματιστές μπορούν να

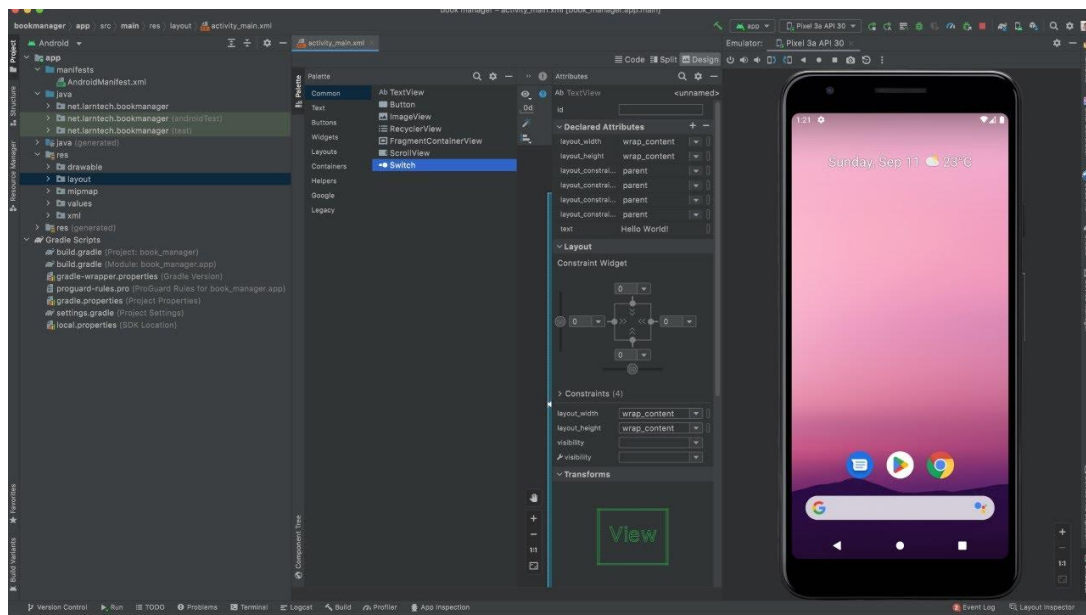
δηλώσουν εξαρτήσεις στο αρχείο ρυθμίσεων του έργου και το Gradle κατεβάζει αυτόματα και περιλαμβάνει τις απαιτούμενες βιβλιοθήκες κατά τη διαδικασία δημιουργίας. Αυτή η προσέγγιση εξορθολογίζει τη ρύθμιση του έργου, μειώνει τη χειρωνακτική προσπάθεια και βοηθά να διασφαλιστεί ότι οι εξαρτήσεις διαχειρίζονται με συνέπεια σε διαφορετικά περιβάλλοντα ανάπτυξης.



Επιπλέον, το Gradle προσφέρει ισχυρές επιλογές προσαρμογής και επεκτασιμότητας μέσω προσηθικών, επιτρέποντας στους προγραμματιστές να προσαρμόσουν τη διαδικασία δημιουργίας ώστε να ταιριάζει σε συγκεκριμένες απαιτήσεις έργου. Το Gradle ενσωματώνεται επίσης απρόσκοπτα με άλλα εργαλεία και πλατφόρμες, όπως διακομιστές συνεχούς ολοκλήρωσης και υπηρεσίες δημιουργίας που βασίζονται σε χρήση cloud, επιτρέποντας αυτοματοποιημένους και κλιμακωτούς αγωγούς κατασκευής για ανάπτυξη.

2.2.4. Android Emulator

Ο εξομοιωτής Android χρησιμεύει ως εικονική συσκευή που μιμείται τη συμπεριφορά και τη λειτουργικότητα των πραγματικών συσκευών Android. Επιτρέπει στους προγραμματιστές να δοκιμάσουν τις εφαρμογές τους σε μια ποικιλία εικονικών συσκευών με διαφορετικά μεγέθη οθόνης, αναλύσεις, διαμορφώσεις υλικού και εκδόσεις Android. Ο εξομοιωτής Android παρέχει έναν βολικό και οικονομικά αποδοτικό τρόπο για τους προγραμματιστές να διασφαλίσουν ότι οι εφαρμογές τους αποδίδουν βέλτιστα σε ένα ευρύ φάσμα συσκευών προορισμού χωρίς να απαιτείται φυσικό υλικό.



Εικόνα 20: Android Emulator

Ένα από τα σημαντικά πλεονεκτήματα του εξομοιωτή Android είναι η υποστήριξή του για επιτάχυνση υλικού, η οποία αξιοποιεί τους πόρους υλικού

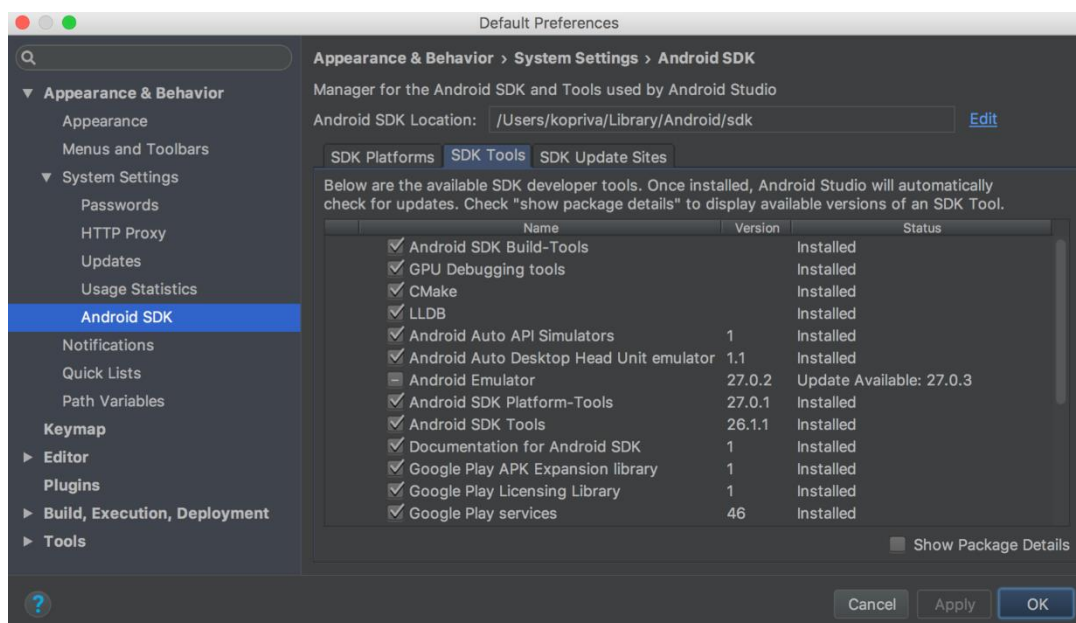
του κεντρικού υπολογιστή ώστε να βελτιώσει την απόδοση και την απόκριση. Χρησιμοποιώντας τεχνολογίες όπως το Intel HAXM (Hardware Accelerated Execution Manager) ή το Hypervisor της AMD, ο εξομοιωτής Android μπορεί να εκτελεί εφαρμογές Android και εικόνες συστήματος πιο αποτελεσματικά, με αποτέλεσμα ταχύτερους χρόνους εκκίνησης εφαρμογών, ομαλότερες κινούμενες εικόνες και βελτιωμένη συνολική εμπειρία χρήστη κατά τη διάρκεια δοκιμών και ανάπτυξης.

Επιπλέον, ο εξομοιωτής Android προσφέρει ένα πλούσιο σύνολο λειτουργιών που διευκολύνουν τον ολοκληρωμένο έλεγχο εφαρμογών και τον εντοπισμό σφαλμάτων. Οι προγραμματιστές μπορούν να προσομοιώσουν διάφορες διαμορφώσεις συσκευών, συμπεριλαμβανομένων των συνθηκών δικτύου, της θέσης GPS, του επιπέδου μπαταρίας και των εισόδων αισθητήρων, για να αναπαράγουν σενάρια χρήσης πραγματικού κόσμου και να εντοπίζουν πιθανά ζητήματα νωρίς στη διαδικασία ανάπτυξης. Ο εξομοιωτής ενσωματώνεται επίσης απρόσκοπτα με τα εργαλεία εντοπισμού σφαλμάτων του Android Studio, επιτρέποντας στους προγραμματιστές να επιθεωρούν τη συμπεριφορά της εφαρμογής, να παρακολουθούν τα αρχεία καταγραφής του συστήματος και να αναλύουν μετρήσεις απόδοσης απευθείας στο περιβάλλον ανάπτυξης.

2.2.5. SDK Manager

Ο SDK Manager, ένα βασικό εργαλείο που συνδυάζεται με το Android Studio, είναι υπεύθυνος για τη διαχείριση των διαφόρων στοιχείων και πόρων που απαιτούνται για την ανάπτυξη εφαρμογών Android. Χρησιμεύει ως κεντρικός κόμβος όπου οι προγραμματιστές μπορούν να κατεβάσουν, να εγκαταστήσουν, να ενημερώσουν και να διαμορφώσουν τα στοιχεία του Android Software

Development Kit (SDK), τα εργαλεία πλατφόρμας, τις εικόνες συστήματος και τους πρόσθετους πόρους που απαιτούνται για την κατασκευή και δοκιμή των εφαρμογών Android. Ο διαχειριστής SDK διασφαλίζει ότι οι προγραμματιστές έχουν πρόσβαση στα πιο πρόσφατα API, βιβλιοθήκες και εργαλεία Android, επιτρέποντάς τους να αξιοποιήσουν τις πλήρεις δυνατότητες της πλατφόρμας Android.



Εικόνα 21: SDK Manager

Μία από τις κύριες λειτουργίες του SDK Manager είναι η διαχείριση της εγκατάστασης και της ενημέρωσης των στοιχείων Android SDK και των εργαλείων πλατφόρμας. Οι προγραμματιστές μπορούν να χρησιμοποιήσουν τον SDK Manager για να εγκαταστήσουν συγκεκριμένες εκδόσεις SDK, εικόνες συστήματος για διαφορετικές εκδόσεις Android και διαμορφώσεις συσκευών και εργαλεία πλατφόρμας όπως ADB (Android Debug Bridge) και Fastboot. Επιπλέον, ο διαχειριστής SDK παρέχει έναν μηχανισμό για τους

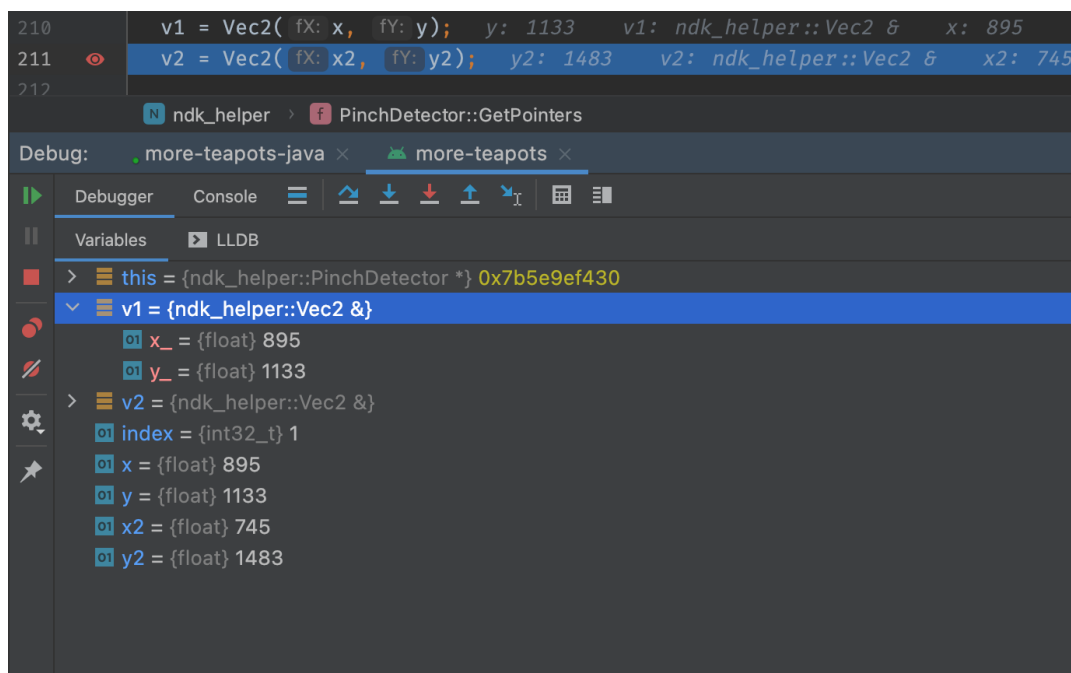
προγραμματιστές ώστε να ενημερώνουν τα εγκατεστημένα στοιχεία σε νεότερες εκδόσεις, διασφαλίζοντας τη συμβατότητα με τις πιο πρόσφατες λειτουργίες.

Επιπλέον, ο διαχειριστής SDK προσφέρει υποστήριξη για τη Διαχείριση πρόσθετων στοιχείων και πόρων, όπως υπηρεσίες Google Play, βιβλιοθήκη υποστήριξης Android και εικόνες συστήματος εξομοιωτή Android. Οι προγραμματιστές μπορούν να εγκαταστήσουν ή να απεγκαταστήσουν επιλεκτικά αυτά τα στοιχεία με βάση τις απαιτήσεις του έργου τους, βελτιστοποιώντας έτσι το χώρο στο δίσκο και μειώνοντας την ακαταστασία. Ο διαχειριστής SDK παρέχει επίσης επιλογές διαμόρφωσης για τη διαχείριση διαδρομών SDK, ρυθμίσεων διακομιστή μεσολάβησης και πηγών αποθετηρίου, επιτρέποντας στους προγραμματιστές να προσαρμόζουν το περιβάλλον ανάπτυξης σύμφωνα με τις προτιμήσεις και τις διαμορφώσεις δικτύου τους.

2.2.6. Εργαλεία εντοπισμού σφαλμάτων

Τα εργαλεία εντοπισμού σφαλμάτων στο Android Studio είναι απαραίτητα για τον εντοπισμό και την επίλυση προβλημάτων σε εφαρμογές Android κατά τη διάρκεια της διαδικασίας ανάπτυξης. Το Android Studio προσφέρει μια ολοκληρωμένη σουίτα εργαλείων εντοπισμού σφαλμάτων που βοηθούν τους προγραμματιστές να διαγνώσουν προβλήματα, να παρακολουθήσουν τη συμπεριφορά της εφαρμογής και να βελτιστοποιήσουν την απόδοση. Ένα από τα κύρια χαρακτηριστικά αποσφαλμάτωσης είναι η δυνατότητα να ορίζονται σημεία διακοπής (breakpoints) στον κώδικα, το οποίο επιτρέπει στους προγραμματιστές να διακόψουν την εκτέλεση της εφαρμογής σε συγκεκριμένα

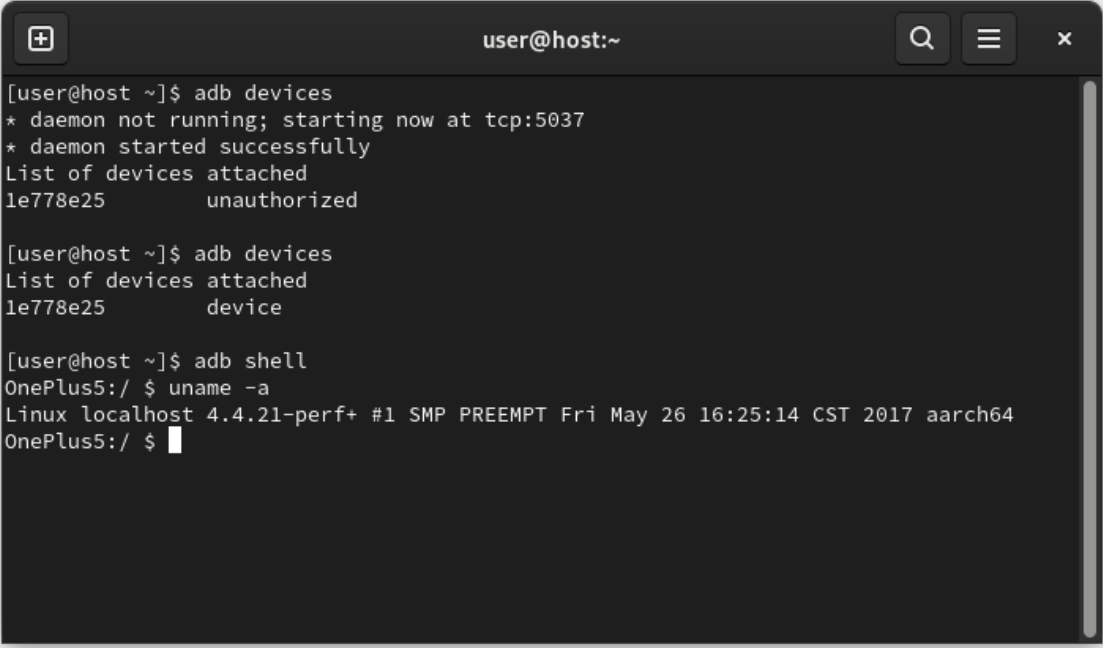
σημεία και να επιθεωρήσουν την κατάσταση του προγράμματος, τις μεταβλητές και τη στοίβα κλήσεων. Τα σημεία διακοπής μπορούν να οριστούν σε κώδικα Java ή Kotlin, καθώς και σε αρχεία διάταξης XML και αρχεία πόρων, παρέχοντας στους προγραμματιστές λεπτομερή έλεγχο της διαδικασίας εντοπισμού σφαλμάτων.



Εικόνα 22: Χρήση debugging σε Android Studio

Ένα άλλο βασικό εργαλείο εντοπισμού σφαλμάτων στο Android Studio είναι το Android Debug Bridge (ADB), ένα ευέλικτο εργαλείο γραμμής εντολών που επιτρέπει την επικοινωνία μεταξύ του μηχανήματος ανάπτυξης και συσκευών ή εξομοιωτών Android. Το ADB προσφέρει διάφορες λειτουργίες για τον εντοπισμό σφαλμάτων, συμπεριλαμβανομένης της εγκατάστασης και απεγκατάστασης εφαρμογών, της μεταφοράς αρχείων μεταξύ της συσκευής και

του υπολογιστή, της λήψης στιγμιότυπων οθόνης της συσκευής και της πρόσβασης σε αρχεία καταγραφής συστήματος και πληροφορίες εντοπισμού σφαλμάτων. Οι προγραμματιστές μπορούν να χρησιμοποιήσουν το ADB για την αντιμετώπιση προβλημάτων συνδεσιμότητας, την ανάλυση μετρήσεων απόδοσης συσκευών και τη διάγνωση σφαλμάτων και σφαλμάτων εφαρμογών απευθείας από τη γραμμή εντολών.

A screenshot of a terminal window titled "user@host:~". The terminal shows the following commands and output:

```
[user@host ~]$ adb devices
* daemon not running; starting now at tcp:5037
* daemon started successfully
List of devices attached
1e778e25    unauthorized

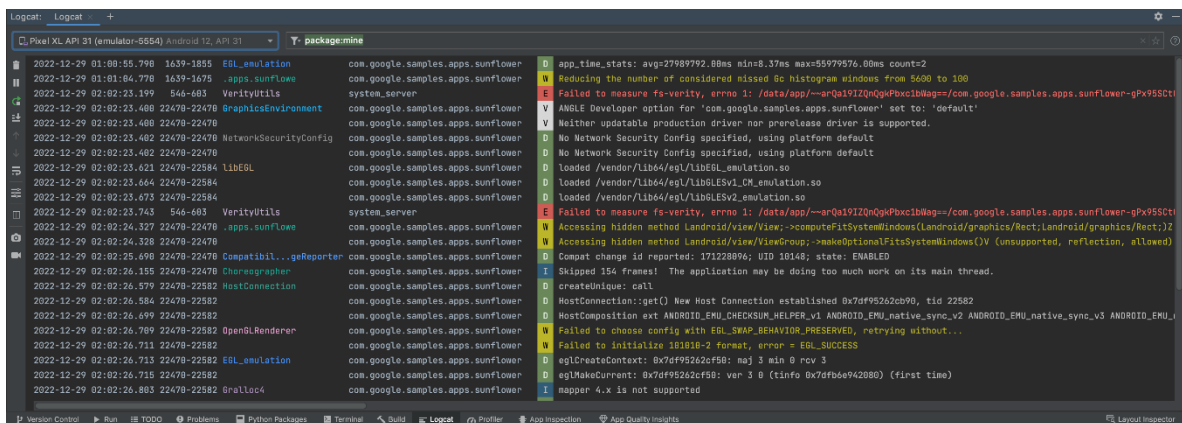
[user@host ~]$ adb devices
List of devices attached
1e778e25    device

[user@host ~]$ adb shell
OnePlus5:/ $ uname -a
Linux localhost 4.4.21-perf+ #1 SMP PREEMPT Fri May 26 16:25:14 CST 2017 aarch64
OnePlus5:/ $
```

Εικόνα 23: Χρήση Android Debug Bridge

Επιπλέον, το Android Studio παρέχει λειτουργίες εντοπισμού σφαλμάτων σε πραγματικό χρόνο που επιτρέπουν στους προγραμματιστές να παρακολουθούν και να αναλύουν δυναμικά τη συμπεριφορά της εφαρμογής ενώ εκτελείται η εφαρμογή. Το παράθυρο εντοπισμού σφαλμάτων στο Android Studio εμφανίζει πληροφορίες σε πραγματικό χρόνο σχετικά με μεταβλητές, κλήσεις μεθόδων

και συμβάντα συστήματος, παρέχοντας στους προγραμματιστές πληροφορίες σχετικά με τη συμπεριφορά εκτέλεσης της εφαρμογής. Επιπλέον, το ενσωματωμένο πρόγραμμα προβολής Logcat του Android Studio επιτρέπει στους προγραμματιστές να βλέπουν και να φιλτράρουν μηνύματα καταγραφής που δημιουργούνται από το σύστημα Android και την ίδια την εφαρμογή, βοηθώντας στη διάγνωση ζητημάτων που σχετίζονται με τη λογική της εφαρμογής, τη φόρτωση πόρων και τις αλληλεπιδράσεις του συστήματος.



Εικόνα 24: Android Logcat

ΚΕΦΑΛΑΙΟ 3: ΕΦΑΡΜΟΓΗ ΠΡΟΤΑΣΗΣ ΑΞΙΟΘΕΑΤΩΝ

3.1. Βασική Φιλοσοφία Εφαρμογής

Βασικός στόχος του πρακτικού μέρους της εργασίας είναι η υλοποίησης μιας Android εφαρμογής με σκοπό την πρόταση αξιοθέατων για την πόλη της Νέας Υόρκης. Πιο συγκεκριμένα, πραγματοποιήθηκε πρόταση αξιοθέατων για τις παρακάτω κατηγορίες:

- Πάρκα
- Μουσεία
- Σημεία Παρατήρησης
- Λοιπά αξιοθέατα (που δεν καλύπτονται από τις παραπάνω κατηγορίες)

Επιπλέον, οι πληροφορίες που παρατίθενται για κάθε αξιοθέατο, συνοψίζονται στον παρακάτω πίνακα:

Όνομα
Περιγραφή
Διεύθυνση
Ώρες Λειτουργίας

Πίνακας 1: Πληροφορίες αξιοθέατων που παρατίθενται στην εφαρμογή

Όσο αφορά τα μουσεία, παρατίθενται αυτά τα οποία συμπεριλήφθηκαν (σε μελλοντική επέκταση εννοείται ότι μπορούν να προστεθούν πολλά

περισσότερα, καθώς ο χαρακτήρας της εφαρμογής είναι καθαρά εκπαιδευτικός):

Όνομα Μουσείου
American Museum of Natural History
Solomon R. Guggenheim Museum
Ground Zero Museum Workshop

Πίνακας 2: Λίστα ονομάτων μουσείων

Ακολουθεί η λίστα των πάρκων, τα οποία έχουν συμπεριληφθεί στην εφαρμογή:

Όνομα Πάρκου
Central Park
Bryant Park
Brooklyn Bridge Park

Πίνακας 3: Λίστα ονομάτων πάρκων

Επιπλέον, ο πίνακας των σημείων παρατήρησης:

Σημείο Παρατήρησης
One World Observatory
Top of the Rock

Empire State Building Observatory

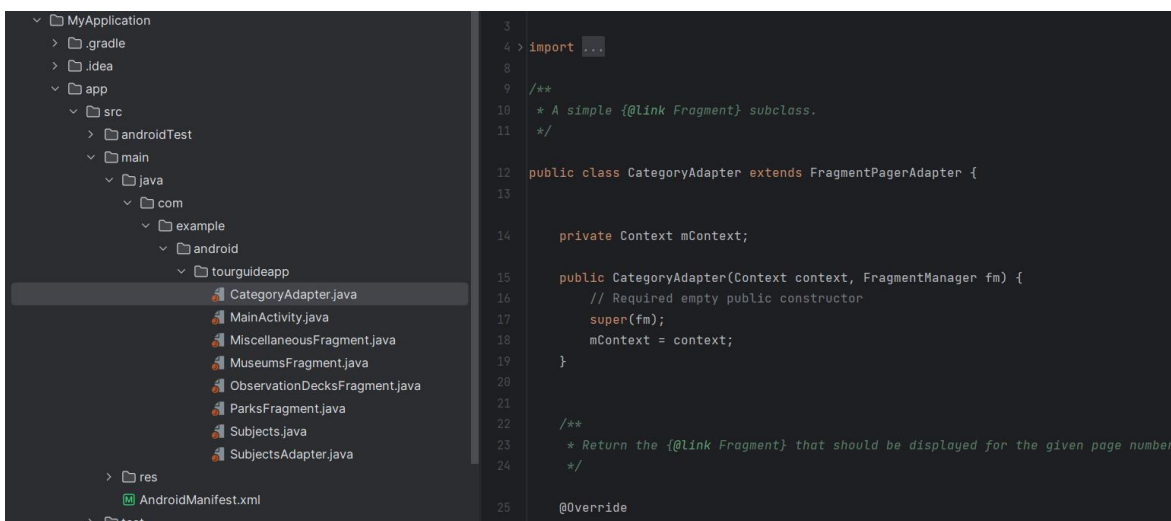
Πίνακας 4: Λίστα ονομάτων σημείων παρατήρησης

Τέλος, ο πίνακας με τα αξιοθέατα τα οποία ανήκουν στην κατηγορία «Λοιπά»:

Όνομα Αξιοθέατου
The Highline
The Cathedral of St. John the Divine
Chelsea Market

Πίνακας 5: Λίστα ονομάτων λοιπών αξιοθέατων

Επιπλέον αξίζει να αναφερθεί ότι η υλοποίηση πραγματοποιήθηκε σε περιβάλλον Android Studio (έκδοση Hedgehog) και ένα screenshot από το περιβάλλον ανάπτυξης που δείχνει και την δομή του project παρουσιάζεται στην συνέχεια:



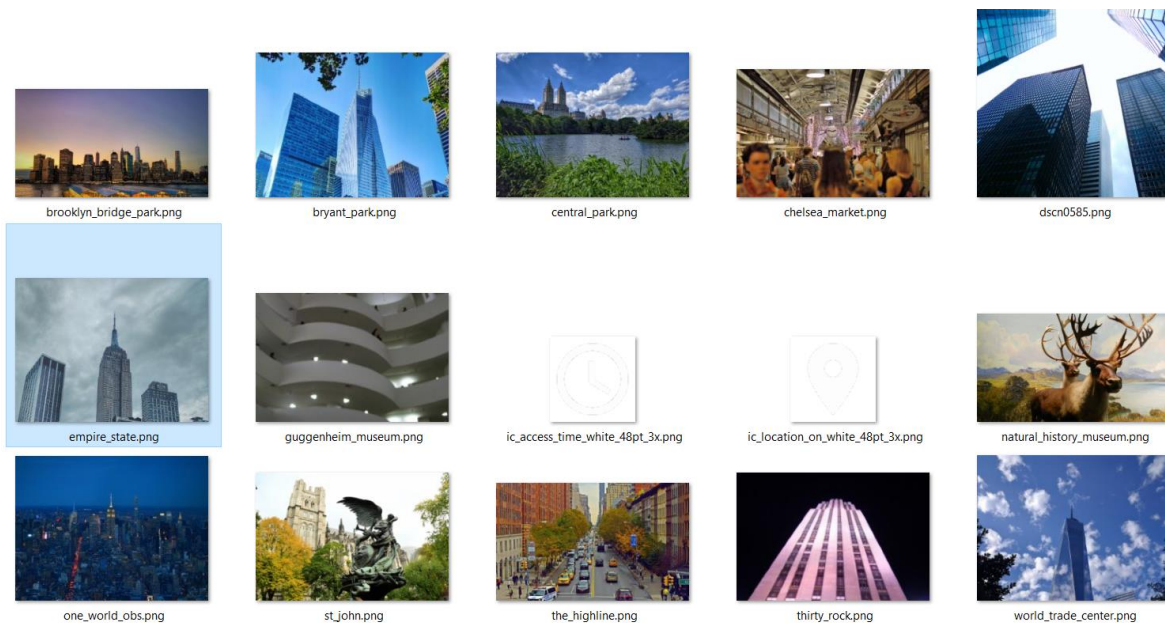
Εικόνα 25: Περιβάλλον ανάπτυξης εφαρμογής

Όπως γίνεται εμφανές από την παραπάνω εικόνα έχουν υλοποιηθεί τα παρακάτω αρχεία, ο κώδικας των οποίων θα παρατεθεί και θα σχολιαστεί στην συνέχεια:

Όνομα Αρχείου
CategoryAdapter.java
MainActivity.java
MiscellaneousFragment.java
MuseumsFragment.java
ObservationDecksFragment.java
ParksFragment.java
Subjects.java
SubjectsAdapter.java

Πίνακας 6: Βασικά Αρχεία Εφαρμογής

Τέλος, για λόγους πληρότητας, παρουσιάζονται οι εικόνες που θα χρησιμοποιηθούν για την προβολή των αξιοθέατων:



3.2. Κώδικας Εφαρμογής

Στα πλαίσια της συγκεκριμένης ενότητας, θα παρουσιαστούν κώδικες των βασικών αρχείων της εφαρμογής, ξεκινώντας από το αρχείο AndroidManifest.xml.

3.2.1. Το αρχείο AndroidManifest.xml

Αρχικά, παρατίθεται ο κώδικας για το συγκεκριμένο αρχείο:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.android.tourguideapp">

    <application
```

```
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity android:name=".MainActivity">

        <intent-filter>

            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>

</manifest>
```

Στο αρχείο αυτό, η εφαρμογή έχει ρυθμιστεί με χαρακτηριστικά όπως:

- η δυνατότητα δημιουργίας αντιγράφων ασφαλείας
- ο ορισμός εικονιδίου
- η ρύθμιση της ετικέτας και του θέματος της εφαρμογής
- ο καθορισμός της κύριας δραστηριότητας, η οποία είναι η MainActivity.

Το στοιχείο < intent-filter> εντός της ετικέτας <activity > δηλώνει ότι το MainActivity είναι το κύριο σημείο εισόδου της εφαρμογής, καθώς ορίζει τη Αυτό σημαίνει ότι η δραστηριότητα MainActivity πρέπει να ξεκινήσει κατά την εκκίνηση της εφαρμογής.

3.2.2. Main Activity

Ακολουθεί ο κώδικας της δραστηριότητας MainActivity:

```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        // Set the content of the activity to use the activity_main.xml layout file
        setContentView(R.layout.activity_main);

        final Toolbar toolbar = findViewById(R.id.htab_toolbar);
        setSupportActionBar(toolbar);
        if (getSupportActionBar() != null)
            getSupportActionBar().setTitle("New York City Tour Guide");

        Objects.requireNonNull(getSupportActionBar()).setDisplayHomeAsUpEnabled(
false);
```

```
// Find the view pager that will allow the user to swipe between fragments
ViewPager viewPager = findViewById(R.id.viewpager);

// Create an adapter that knows which fragment should be shown on each
page
CategoryAdapter adapter = new CategoryAdapter(this,
getSupportFragmentManager());

// Set the adapter onto the view pager
viewPager.setAdapter(adapter);

// Find the tab layout that shows the tabs
TabLayout tabLayout = findViewById(R.id.tabs);

tabLayout.setupWithViewPager(viewPager);

}
}
```

Ο παραπάνω κώδικας ορίζει την κλάση MainActivity, η οποία χρησιμεύει ως σημείο εισόδου για την εφαρμογή Android. Η MainActivity επεκτείνει την κλάση AppCompatActivity. Μέσα στη μέθοδο onCreate(), η διάταξη της δραστηριότητας ορίζεται χρησιμοποιώντας την setContentView(R.layout.activity_main), η οποία αρχικοποιεί το activity_main.xml

αρχείο για την εμφάνιση των στοιχείων UI. Στη συνέχεια, ο κώδικας αρχικοποιεί μια γραμμή εργαλείων (toolbar), ορίζει τον τίτλο της και την ρυθμίζει για εμφάνιση. Έπειτα, βρίσκει ένα ViewPager, το οποίο επιτρέπει στους χρήστες να πραγματοποιούν εναλλαγή μεταξύ fragments και δημιουργεί έναν Category Adapter, ο οποίος καθορίζει τα fragments που θα εμφανίζονται στο ViewPager. Στη συνέχεια, ο ViewPager συνδέεται με τον Adapter. Τέλος, ο κώδικας βρίσκει ένα TabLayout και το ρυθμίζει ώστε να λειτουργεί με τον ViewPager, επιτρέποντας στους χρήστες να εναλλάσσονται μεταξύ καρτελών που αντιστοιχούν σε διαφορετικά fragments.

3.2.3. Category Adapter

Ο κώδικας για το συγκεκριμένο αρχείο είναι ο παρακάτω:

```
public class CategoryAdapter extends FragmentPagerAdapter {

    private Context mContext;

    public CategoryAdapter(Context context, FragmentManager fm) {
        // Required empty public constructor
        super(fm);
        mContext = context;
    }

    /**
     * Return the {@link Fragment} that should be displayed for the given page
     number.
```

```
*/  
  
@Override  
public Fragment getItem(int position) {  
    if (position == 0) {  
        return new ParksFragment();  
    } else if (position == 1) {  
        return new MuseumsFragment();  
    } else if (position == 2) {  
        return new ObservationDecksFragment();  
    } else {  
        return new MiscellaneousFragment();  
    }  
}
```

```
/**  
 * Return the total number of pages.  
 */
```

```
@Override  
public int getCount() {  
    return 4;  
}
```

```
@Override  
public CharSequence getPageTitle(int position) {  
    if (position == 0) {
```

```
        return mContext.getString(R.string.parks_fragment);
    } else if (position == 1) {
        return mContext.getString(R.string.museums_fragment);
    } else if (position == 2) {
        return mContext.getString(R.string.observation_decks_fragment);
    } else {
        return mContext.getString(R.string.miscellaneous_fragment);
    }
}
}
```

Ο παραπάνω κώδικας ορίζει μια κλάση CategoryAdapter, η οποία επεκτείνει την FragmentPagerAdapter. Ο Category Adapter είναι υπεύθυνος για τη διαχείριση των fragments που εμφανίζονται σε ένα ViewPager στην κύρια δραστηριότητα της εφαρμογής. Ο κατασκευαστής αρχικοποιεί τον Adapter με ένα πλαίσιο και ένα FragmentManager.

Επιπλέον, η μέθοδος getItem () καθορίζει ποιο fragment πρέπει να εμφανίζεται με βάση την παρεχόμενη θέση. Επιστρέφει διαφορετικά fragments ανάλογα με τη θέση:

- Parks Fragment από τη θέση 0
- Museums Fragment από τη θέση 1
- ObservationDecks Fragment για τη θέση 2
- Miscellaneous Fragment για οποιαδήποτε άλλη θέση

Επιπροσθέτως, η μέθοδος `getCount ()` επιστρέφει τον συνολικό αριθμό σελίδων, που είναι 4 σε αυτήν την περίπτωση, και η μέθοδος `getPageTitle ()` επιστρέφει τον τίτλο για κάθε καρτέλα που αντιστοιχεί στο fragment στη δεδομένη θέση, χρησιμοποιώντας συμβολοσειρές, οι οποίες ορίζονται στο αρχείο `strings.xml`.

3.2.4. Museums Fragment

Ο κώδικας για το συγκεκριμένο αρχείο:

```
public class MuseumsFragment extends Fragment {

    public MuseumsFragment() {
        // Required empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        View rootView = inflater.inflate(R.layout.list_layout, container, false);

        //Create an ArrayList for Place, Description, Location, Hours, and Image
        final ArrayList<Subjects> words = new ArrayList<>();
        words.add(new Subjects(getString(R.string.natural_history_museum),
                                getString(R.string.natural_history_museum_des),
                                getString(R.string.natural_history_museum_loc),
```

```
        getString(R.string.natural_history_museum_hours),
        R.drawable.natural_history_museum));

words.add(new Subjects(getString(R.string.guggenheim_museum),
        getString(R.string.guggenheim_museum_des),
        getString(R.string.guggenheim_museum_loc),
        getString(R.string.guggenheim_museum_hours),
        R.drawable.guggenheim_museum));

words.add(new Subjects(getString(R.string.world_trade_center),
        getString(R.string.world_trade_center_des),
        getString(R.string.world_trade_center_loc),
        getString(R.string.world_trade_center_hours),
        R.drawable.world_trade_center));

SubjectsAdapter adapter = new SubjectsAdapter(getActivity(), words);
ListView listView = rootView.findViewById(R.id.list);

//Allows the CollapsingToolbar to scroll up with the ViewPager
listView.setNestedScrollingEnabled(true);
listView.setAdapter(adapter);

return rootView;
}
}
```

Ο σκοπός της παραπάνω κλάσης είναι να δημιουργήσει εκείνο το fragment, το οποίο περιέχει πληροφορίες σχετικά με τα μουσεία. Στην συνέχεια θα ακολουθήσουν και οι κώδικες για τα υπόλοιπα fragments.

3.2.5. Parks Fragment

Ο αντίστοιχος κώδικας της κλάσης που δημιουργεί το fragment για την περίπτωση των πάρκων:

```
public class ParksFragment extends Fragment {

    public ParksFragment() {
        // Required empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {

        View rootView = inflater.inflate(R.layout.list_layout, container, false);

        //Create an ArrayList for Place, Description, Location, Hours, and Image
        final ArrayList<Subjects> words = new ArrayList<>();
        words.add(new Subjects(getString(R.string.central_park),
                                getString(R.string.central_park_des),
                                getString(R.string.central_park_loc),
                                getString(R.string.central_park_hours),
```

```
        R.drawable.central_park));

words.add(new Subjects(getString(R.string.bryant_park),
        getString(R.string.bryant_park_des),
        getString(R.string.bryant_park_loc),
        getString(R.string.bryant_park_hours),
        R.drawable.bryant_park));

words.add(new Subjects(getString(R.string.brooklyn_bridge_park),
        getString(R.string.brooklyn_bridge_des),
        getString(R.string.brooklyn_bridge_loc),
        getString(R.string.brooklyn_bridge_hours),
        R.drawable.brooklyn_bridge_park));

SubjectsAdapter adapter = new SubjectsAdapter(getActivity(), words);
ListView listView = rootView.findViewById(R.id.list);

//Allows the CollapsingToolbar to scroll up with the ViewPager
listView.setNestedScrollingEnabled(true);
listView.setAdapter(adapter);

return rootView;
    }
}
```

3.2.6. Observation Decks Fragment

Ο αντίστοιχος κώδικας της κλάσης που δημιουργεί το fragment για την περίπτωση των σημείων παρατήρησης:

```
public class ObservationDecksFragment extends Fragment {

    public ObservationDecksFragment() {
        // Required empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {

        View rootView = inflater.inflate(R.layout.list_layout, container, false);

        //Create an ArrayList for Place, Description, Location, Hours, and Image
        final ArrayList<Subjects> words = new ArrayList<>();
        words.add(new Subjects(getString(R.string.one_world_obs),
            getString(R.string.one_world_obs_des),
            getString(R.string.one_world_obs_loc),
            getString(R.string.one_world_obs_hours),
            R.drawable.one_world_obs));
        words.add(new Subjects(getString(R.string.thirty_rock),
            getString(R.string.thirty_rock_des),
            getString(R.string.thirty_rock_loc),
```

```
        getString(R.string.thirty_rock_hours),
        R.drawable.thirty_rock));

words.add(new Subjects(getString(R.string.empire_state),
        getString(R.string.empire_state_des),
        getString(R.string.empire_state_loc),
        getString(R.string.empire_state_hours),
        R.drawable.empire_state));

SubjectsAdapter adapter = new SubjectsAdapter(getActivity(), words);
ListView listView = rootView.findViewById(R.id.list);

//Allows the CollapsingToolbar to scroll up with the ViewPager
listView.setNestedScrollingEnabled(true);
listView.setAdapter(adapter);

return rootView;
}

}
```

3.2.7. Miscellaneous Fragment

Ο αντίστοιχος κώδικας της κλάσης που δημιουργεί το fragment για την περίπτωση των λοιπών αξιοθέατων:

```
public class MiscellaneousFragment extends Fragment {

    public MiscellaneousFragment() {
        // Required empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        View rootView = inflater.inflate(R.layout.list_layout, container, false);
        //Create an ArrayList for Place, Description, Location, Hours, and Image
        final ArrayList<Subjects> words = new ArrayList<Subjects>();
        words.add(new Subjects(getString(R.string.the_highline),
                                getString(R.string.the_highline_des),
                                getString(R.string.the_highline_loc),
                                getString(R.string.the_highline_hours),
                                R.drawable.the_highline));
        words.add(new Subjects(getString(R.string.st_john),
                                getString(R.string.st_john_des),
                                getString(R.string.st_john_loc),
                                getString(R.string.st_john_hours),
                                R.drawable.st_john));
        words.add(new Subjects(getString(R.string.chelsea_market),
                                getString(R.string.chelsea_market_des),
                                getString(R.string.chelsea_market_loc),
                                getString(R.string.chelsea_market_hours),
```

```
R.drawable.chelsea_market));

SubjectsAdapter adapter = new SubjectsAdapter(getActivity(), words);

ListView listView = rootView.findViewById(R.id.list);

//Allows the CollapsingToolbar to scroll up with the ViewPager
listView.setNestedScrollingEnabled(true);
listView.setAdapter(adapter);

return rootView;
}
}
```

3.2.8. Subjects

Παρατίθεται ο κώδικας για το συγκεκριμένο αρχείο:

```
public class Subjects {
    private String mPlace;
    private String mDescription;
    private String mLocation;
    private String mHours;
    private int mImageResourceId;
```

```
public Subjects(String place, String description, String location, String hours, int
imageResourceId) {
    mPlace = place;
    mDescription = description;
    mLocation = location;
    mHours = hours;
    mImageResourceId = imageResourceId;
}

public String getPlace() {
    return mPlace;
}

public String getDescription(){
    return mDescription;
}

public String getLocation() {
    return mLocation;
}

public String getHours() {
    return mHours;
}
```

```
public int getImageResourceId() {  
    return mImageResourceId;  
}  
  
@Override  
public String toString() {  
    String output = getPlace() + "\n" +  
        getDescription() + "\n" +  
        getLocation() + "\n" +  
        getHours() + "\n" +  
        getImageResourceId();  
    return output;  
}  
}
```

Στον παραπάνω κώδικα ορίζεται η κλάση για τον ορισμό ενός αξιοθέατου που περιέχει τα παρακάτω χαρακτηριστικά:

- Όνομα
- Περιγραφή
- Τοποθεσία
- Ώρες Λειτουργίας
- Id εικόνας αξιοθέατου

Επιπλέον, έχουν οριστεί και αντίστοιχες μέθοδοι, οι οποίες επιστρέφουν τις τιμές των συγκεκριμένων χαρακτηριστικών ενός αξιοθέατου.

3.2.9. Subjects Adapter

Ο κώδικας για το συγκεκριμένο αρχείο:

```
public class SubjectsAdapter extends ArrayAdapter<Subjects> {
    private Context mContext;

    public SubjectsAdapter(Context context, ArrayList<Subjects> words) {
        super(context, 0, words);
        this.mContext = context;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        // Check if the existing view is being reused, otherwise inflate the view
        View listItemView = convertView;
        if (listItemView == null) {
            listItemView = LayoutInflater.from(getContext()).inflate(
                R.layout.list_item, parent, false);
        }

        // Get the object position in the list
        Subjects currentSubject = getItem(position);

        // Find the TextView in the list_item.xml layout with the ID
        TextView placeTextView =
listItemView.findViewById(R.id.place_text_view);
```

```

placeTextView.setText(currentSubject.getPlace());

// Find the TextView in the list_item.xml layout with the ID
TextView                descriptionTextView                =
listItemView.findViewById(R.id.description_text_view);
descriptionTextView.setText(currentSubject.getDescription());

// Find the TextView in the list_item.xml layout with the ID
TextView                locationTextView                =
listItemView.findViewById(R.id.location_text_view);
locationTextView.setText(currentSubject.getLocation());

RelativeLayout          locationLayout                =
listItemView.findViewById(R.id.location_layout);
locationLayout.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String                location                =                ((TextView)
view.findViewById(R.id.location_text_view)).getText().toString();
        Uri gmmIntentUri = Uri.parse("geo:0,0?q=" + "+" + location + ", NYC");
        Intent mapIntent = new Intent(Intent.ACTION_VIEW, gmmIntentUri);
        mapIntent.setPackage("com.google.android.apps.maps");
        mContext.startActivity(mapIntent);
    }
}

```

```

});

// Find the TextView in the list_item.xml layout with the ID
TextView                hoursTextView                =
listItemView.findViewById(R.id.hours_text_view);
hoursTextView.setText(currentSubject.getHours());

// Find the ImageView in the list_item.xml layout with the ID
ImageView                tourImageView                =
listItemView.findViewById(R.id.tour_image_view);
tourImageView.setImageResource(currentSubject.getImageResourceId());

// Return the list layout (4 TextViews and 1 ImageView)
// so that it can be shown in the ListView
return listItemView;
}
}

```

Η συγκεκριμένη κλάση λειτουργεί ως συνδετικός κρίκος με σκοπό την προβολή των πληροφοριών των αξιοθέατων στα εκάστοτε fragments. Επιπλέον χρησιμοποιείται ο OnClickListener, προκειμένου να ανοίξει η τοποθεσία σε περιβάλλον Google Maps όταν επιλέγεται η αντίστοιχη διεύθυνση.

3.2.10. Strings.xml

Στο συγκεκριμένο αρχείο έχουν προστεθεί οι πληροφορίες, οι οποίες αφορούν τα αξιοθέατα, έτσι ώστε στην συνέχεια να μπορούν να χρησιμοποιηθούν και να προβληθούν στην εφαρμογή του χρήστη. Ενδεικτικά παρατίθεται screenshot, το οποίο δείχνει μέρος του αρχείου που αφορά τις πληροφορίες των μουσείων:

```
<!--Museums Fragment-->
<string name="natural_history_museum">American Museum of Natural History</string>
<string name="natural_history_museum_des">Popular museum featuring an extensive collection of preserved specimens and priceless artifacts from around the world</string>
<string name="natural_history_museum_loc">Central Park West and 79th St</string>
<string name="natural_history_museum_hours">Monday-Sunday 10am-5:45pm</string>

<string name="guggenheim_museum">Solomon R. Guggenheim Museum</string>
<string name="guggenheim_museum_des">World renowned art works exhibited in an equally iconic building designed by Frank Lloyd Wright.</string>
<string name="guggenheim_museum_loc">1071 5th Ave</string>
<string name="guggenheim_museum_hours">Monday-Sunday 10am-5:45pm</string>

<string name="world_trade_center">Ground Zero Museum Workshop</string>
<string name="world_trade_center_des">Museum dedicated to the remembrance of victims of September 11, located on the site of what once was the World Trade Cent</string>
<string name="world_trade_center_loc">420 W 14th St, Floor 2</string>
<string name="world_trade_center_hours">Wednesday-Sunday 11am-3pm</string>
```

Εικόνα 26: Πληροφορίες μουσείων στο strings.xml

Αντίστοιχο απόσπασμα για τα πάρκα:

```
<!--Parks Fragment-->
<string name="central_park">Central Park</string>
<string name="central_park_des">Iconic green area located in the heart of Manhattan offering an extensive biking and walking trail system.</string>
<string name="central_park_loc">59th to 110th Street</string>
<string name="central_park_hours">Monday-Sunday 6am-1am</string>

<string name="bryant_park">Bryant Park</string>
<string name="bryant_park_des">Located in Midtown, this park is known for its calmness and cleanness amongst the locals.</string>
<string name="bryant_park_loc">98-74 W 40th St</string>
<string name="bryant_park_hours">Monday-Sunday 7am-10pm</string>

<string name="brooklyn_bridge_park">Brooklyn Bridge Park</string>
<string name="brooklyn_bridge_des">Scenic park located under the iconic bridge with scenic views of Lower Manhattan from across the East River.</string>
<string name="brooklyn_bridge_loc">334 Furman St</string>
<string name="brooklyn_bridge_hours">Monday-Sunday 6am-1am</string>
```

Εικόνα 27: Πληροφορίες πάρκων στο strings.xml

Αντίστοιχο απόσπασμα για τα σημεία παρατήρησης:

```
<!--Observation Decks Fragment-->
<string name="one_world_obs">One World Observatory</string>
<string name="one_world_obs_des">Highest observation deck featuring 360-degree views of the city, located on top of the One World Trade Center.</string>
<string name="one_world_obs_loc">285 Fulton St</string>
<string name="one_world_obs_hours">Monday-Sunday 8am-9pm</string>

<string name="thirty_rock">Top of the Rock</string>
<string name="thirty_rock_des">Located on top of the NBC Studios at Rockefeller Plaza, it features sights overlooking the Empire State Building and Central Park.</string>
<string name="thirty_rock_loc">30 Rockefeller Plaza</string>
<string name="thirty_rock_hours">Monday-Sunday 8am-12:30am</string>

<string name="empire_state">Empire State Building Observatory</string>
<string name="empire_state_des">Popular observation deck, where you will not be able to see one of the most iconic skyscrapers.</string>
<string name="empire_state_loc">350 5th Ave</string>
<string name="empire_state_hours">Monday-Sunday 8am-2am</string>
```

Εικόνα 28: Πληροφορίες σημείων παρατήρησης στο strings.xml

Τέλος, αντίστοιχο απόσπασμα για τα λοιπά αξιοθέατα:

```
<!--Miscellaneous Fragment-->
<string name="the_highline">The Highline</string>
<string name="the_highline_des">Old train rail repurposed into an elevated walking park with walking trails and scenic overlooks of the city.</string>
<string name="the_highline_loc">820 Washington St</string>
<string name="the_highline_hours">Monday-Sunday 7am-11pm</string>

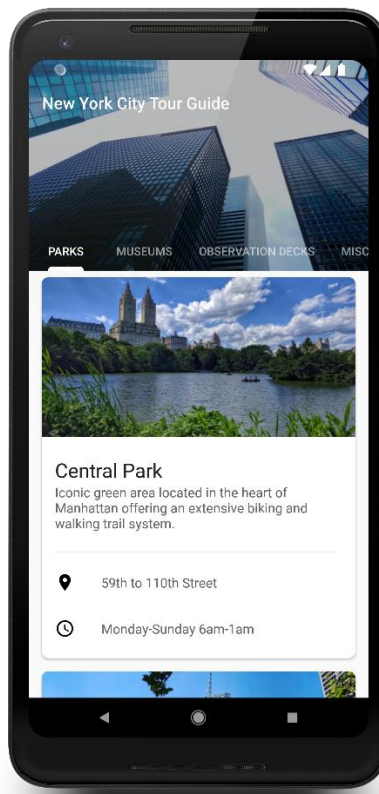
<string name="st_john">The Cathedral of St. John the Divine</string>
<string name="st_john_des">Historic cathedral in the heart of Manhattan's Morningside Heights neighborhood. Boasts title of world's largest cathedral.</string>
<string name="st_john_loc">1047 Amsterdam Ave</string>
<string name="st_john_hours">Monday-Saturday 9am-5pm, Sunday 1pm-3pm</string>

<string name="chelsea_market">Chelsea Market</string>
<string name="chelsea_market_des">Local indoor marketplace offering a selection of unique handmade goods, hip restaurants, and gourmet imports.</string>
<string name="chelsea_market_loc">75 9th Ave</string>
<string name="chelsea_market_hours">Monday-Saturday 7am-10pm, Sunday 8am-8pm</string>
```

Εικόνα 29: Πληροφορίες λοιπών αξιοθέατων στο strings.xml

3.3. Screenshots Εφαρμογής

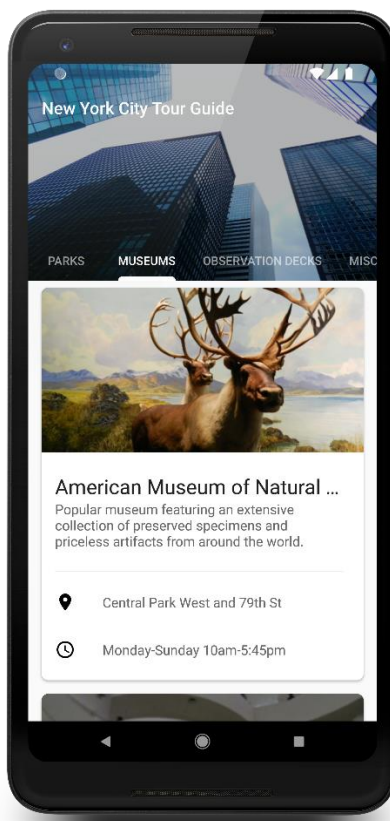
Στα πλαίσια της συγκεκριμένης ενότητας, παρουσιάζονται screenshots, τα οποία επιδεικνύουν την λειτουργικότητα της εφαρμογής. Αρχικά παρατίθεται screenshot που αφορά την ενότητα των πάρκων:



Εικόνα 30: Screenshot 1 - Παρουσίαση πάρκων

Στην παραπάνω εικόνα, παρουσιάζεται το πάρκο Central Park, όπου και εμφανίζεται η περιγραφή του, το ωράριο λειτουργίας τους, καθώς και η διεύθυνσή του.

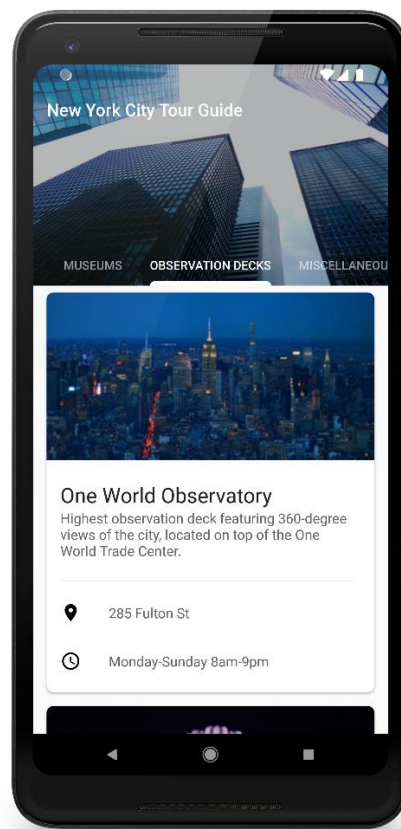
Ακολουθεί επόμενο screenshot, το οποίο επιδεικνύει την παρουσίαση των μουσείων:



Εικόνα 31: Screenshot 2 - Παρουσίαση μουσείων

Στην παραπάνω εικόνα παρουσιάζονται οι πληροφορίες του μουσείου American Museum of Natural History.

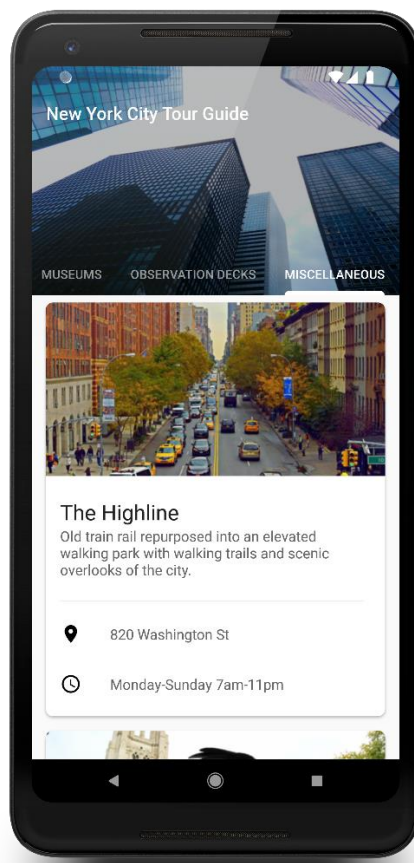
Ακολουθεί screenshot που αφορά την ενότητα των σημείων παρατήρησης:



Εικόνα 32: Screenshot 3 - Παρουσίαση σημείων παρατήρησης

Πιο συγκεκριμένα, στην παραπάνω εικόνα παρουσιάζονται οι πληροφορίες του σημείου: One World Observatory.

Τέλος, το τελευταίο screenshot που αφορά τα λοιπά αξιοθέατα:



Εικόνα 33: Screenshot 4 - Παρουσίαση λοιπών αξιοθέατων

Στην συγκεκριμένη εικόνα παρουσιάζεται η περίπτωση του αξιοθέατου: The Highline.

Συμπεράσματα

Στα πλαίσια της συγκεκριμένης εργασίας αρχικά πραγματοποιήθηκε βιβλιογραφική ανασκόπηση όσο αφορά τις περιπτώσεις εφαρμογών που έχουν σαν βασικό σκοπό την πρόταση αξιοθέατων. Στην συνέχεια παρουσιάστηκαν τα βασικά συστατικά μιας Android εφαρμογής, καθώς και η βασική λειτουργικότητα του εργαλείου Android Studio.

Όσο αφορά το πρακτικό μέρος, υλοποιήθηκε μια Android εφαρμογή σε Android Studio η οποία είχε σαν βασικό αντικείμενο την πρόταση αξιοθέατων όσο αφορά την πόλη της Νέας Υόρκης. Τα συμπεράσματα, τα οποία προέκυψαν, συνοψίζονται στην συνέχεια:

- Η εφαρμογή ήταν πλήρως λειτουργικά και απλή στην χρήση
- Επετεύχθη ο σκοπός της πρότασης αξιοθέατων για μια συγκεκριμένη πόλη
- Μπορεί να επεκταθεί ώστε να υποστηρίζεται η προβολή αξιοθέατων για περισσότερες πόλεις στον κόσμο.
- Μπορεί να επεκταθεί ώστε να παρέχει πολυγλωσσική υποστήριξη.
- Προτείνεται να δημιουργηθεί έκδοση για συσκευές iPhone.

ΒΙΒΛΙΟΓΡΑΦΙΑ – ΠΗΓΕΣ

- [1].Lee, H. A., Law, R., & Murphy, J. (2011). Helpful reviewers in TripAdvisor, an online travel community. *Journal of Travel & Tourism Marketing*, 28(7), 675-688.
- [2].Mehta, H., Kanani, P., & Lande, P. (2019). Google maps. *International Journal of Computer Applications*, 178(8), 41-46.
- [3].<https://www.lonelyplanet.com/guides>
- [4].<https://travel.sygic.com/en>
- [5].<https://www.viator.com/>
- [6].<https://theculturetrip.com/our-app>
- [7].<https://roadtrippers.com/>
- [8].<https://rove.me/>
- [9].Sia, P. Y. H., Saidin, S. S., & Iskandar, Y. H. P. (2023). Systematic review of mobile travel apps and their smart features and challenges. *Journal of Hospitality and Tourism Insights*, 6(5), 2115-2138.
- [10]. Ostrander, J. (2012). *Android UI Fundamentals: Develop and Design*. Peachpit press.
- [11]. Gargenta, M. (2011). *Learning android*. " O'Reilly Media, Inc."
- [12]. DiMarzio, J. F. (2008). *Android™ A Programmer's Guide*.

- [13]. Brahler, S. (2010). Analysis of the android architecture. Karlsruhe institute for technology, 7(8).
- [14]. Gerber, A., Craig, C., & Selvaraj, D. (2015). Learn Android Studio: Build Android Apps Quickly and Effectively. Apress.
- [15]. Yener, M., & Dunder, O. (2016). Expert Android Studio. John Wiley & Sons.
- [16]. Craig, C., & Gerber, A. (2015). Learn android studio: build android apps quickly and effectively. Apress.
- [17]. Van Drongelen, M. (2015). Android studio cookbook. Packt Publishing Ltd.