



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ – ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Πρόγραμμα Μεταπτυχιακών Σπουδών
«ΠΜΣ Πληροφορική»**

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	«Ανάπτυξη εφαρμογής σε περιβάλλον Android με εξατομικευμένες προτάσεις σειρών και δυνατότητα interaction με χρήστες» «Development of an android application with series suggestions and user interaction»
Όνοματεπώνυμο Φοιτητή	ΕΥΣΤΑΘΙΟΣ ΚΑΡΑΔΗΜΗΤΡΙΟΥ
Πατρώνυμο	ΑΝΑΣΤΑΣΙΟΣ
Αριθμός Μητρώου	ΜΠΠΛ21024
Επιβλέπων	ΑΛΕΠΗΣ ΕΥΘΥΜΙΟΣ, ΑΝΑΠΛΗΡΩΤΗΣ ΚΑΘΗΓΗΤΗΣ

Ημερομηνία Παράδοσης **Ιούλιος 2023**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Ευθύμιος Αλέπης
Αναπληρωτής Καθηγητής

Μαρία Βίρβου
Καθηγήτρια

Κωνσταντίνος Πατσάκης
Αναπληρωτής Καθηγητής

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΥΧΑΡΙΣΤΙΕΣ	5
ΛΙΣΤΑ ΕΙΚΟΝΩΝ	6
ΠΕΡΙΛΗΨΗ	7
ABSTRACT	7
ΕΙΣΑΓΩΓΗ	8
ΚΕΦΑΛΑΙΟ 1	9
1.1 Η ΕΞΑΠΛΩΣΗ ΤΟΥ INTERNET	9
1.2 ΕΞΥΠΝΕΣ ΣΥΣΚΕΥΕΣ (SMARTPHONES, TABLETS).....	9
1.3 ΤΡΟΠΟΣ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ ΓΙΑ ΚΙΝΗΤΑ	10
1.4 ANDROID	11
1.5 ΒΑΣΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ ANDROID.....	13
1.5.1 Αρχείο Manifest	13
1.5.2 Δραστηριότητες (Activities).....	14
1.5.3 Κομμάτι Οθόνης (Fragment)	15
1.5.4 Προθέσεις (Intents)	17
1.5.5 Broadcast Receivers	17
1.6 ΑΣΦΑΛΕΙΑ	17
1.7 ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ ΜΕ ΤΟ ANDROID STUDIO	18
ΚΕΦΑΛΑΙΟ 2	19
2.1 ΑΝΑΠΤΥΞΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	19
2.2 ΤΟ ΠΡΟΒΛΗΜΑ.....	19
2.3 ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ	19
2.4 ΜΗ ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ	21
2.5 ΔΕΔΟΜΕΝΑ ΣΕΙΡΩΝ ΠΟΥ ΘΑ ΧΡΗΣΙΜΟΠΟΙΕΙ Η ΕΦΑΡΜΟΓΗ	21
2.5.1 Επιλογή API εφαρμογής	22
2.5.2 Δυνατότητα αξιολόγησης σειράς σαν επισκέπτης.....	23
ΚΕΦΑΛΑΙΟ 3	23
3.1 ΤΕΧΝΟΛΟΓΙΕΣ ΠΟΥ ΘΑ ΧΡΗΣΙΜΟΠΟΙΗΘΟΥΝ ΣΤΗΝ ΑΝΑΠΤΥΞΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	23
3.1.1 Kotlin (Γλώσσα Προγραμματισμού)	23
3.1.2 MVVM Clean Architecture (Architectural pattern)	24
3.1.3 Κατάτμηση εφαρμογής σε modules (Multimodular approach)	25
3.1.4 Navigation Component	27
3.1.5 Kotlin Coroutines.....	28
3.1.6 Kotlin Flows	29
3.1.7 Data Binding	29
3.1.8 Preferences DataStore	30
3.1.9 Hilt (Dependency Injection).....	30

3.1.10	Firebase Authentication	30
3.1.11	Firebase Cloud Firestore (No SQL Database).....	31
3.1.12	Firebase Storage	31
3.1.13	Retrofit (HTTP Client)	32
3.1.14	Gson	32
3.1.15	Paging 3 (Android Jetpack)	32
3.1.16	Glide	33
3.1.17	Github (Version Control System – VCS).....	33
3.2	ΣΥΜΠΛΗΡΩΜΑΤΙΚΗ ΕΦΑΡΜΟΓΗ ΠΑΡΑΓΩΓΗΣ ΣΤΑΤΙΣΤΙΚΩΝ ΣΤΟΙΧΕΙΩΝ	33
3.2.1	Στατιστικά στοιχεία που παράγει η εφαρμογή	33
ΚΕΦΑΛΑΙΟ 4	34
4.1	ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	34
4.1.1	Όνομα εφαρμογής, Λογότυπο & Styling	34
4.2	ΑΝΑΛΥΣΗ ΟΘΟΝΩΝ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	35
4.2.1	Είσοδος στην εφαρμογή - Onboarding	35
4.2.2	Dashboard	38
4.2.3	Οθόνες πληροφοριών	42
4.2.4	Προφίλ.....	45
4.2.5	Συζητήσεις (Forum)	50
4.2.6	Ρυθμίσεις (Settings)	52
4.3	ΒΟΗΘΗΤΙΚΕΣ ΥΛΟΠΟΙΗΣΕΙΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	56
ΚΕΦΑΛΑΙΟ 5	57
5.1	ΣΥΜΠΕΡΑΣΜΑΤΑ	57
5.2	ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	58
ΚΕΦΑΛΑΙΟ 6	61
6.1	ΒΙΒΛΙΟΓΡΑΦΙΑ	61

ΕΥΧΑΡΙΣΤΙΕΣ

Στο πλαίσιο της διπλωματικής μου εργασίας θα ήθελα να ευχαριστήσω όλους τους καθηγητές του Τμήματος Πληροφορικής του Πανεπιστημίου Πειραιώς για τις πολύτιμες γνώσεις που μου μετέδωσαν στην μέχρι στιγμής ακαδημαϊκή μου ζωή.

Στη συνέχεια θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, κ. Αλέπη Ευθύμιο για την καθοδήγησή και τις πολύτιμες συμβουλές του καθ' όλη τη διάρκεια των μεταπτυχιακών σπουδών μου.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου για την υποστήριξη που μου προσέφερε σε όλα τα χρόνια των σπουδών μου.

ΛΙΣΤΑ ΕΙΚΟΝΩΝ

Εικόνα 1. Το market share των smartphones (statcounter)	10
Εικόνα 2: Το λογότυπο του λειτουργικού συστήματος android.....	11
Εικόνα 3: Συσσωρευτική κατανομή των android releases.	13
Εικόνα 4. Ο κύκλος ζωής (lifecycle) ενός activity.....	15
Εικόνα 5. Ο κύκλος ζωής (lifecycle) ενός fragment	16
Εικόνα 6. Το λογότυπο του Android Studio	18
Εικόνα 7. Εικόνα από το api call testing στο TMDB API.....	22
Εικόνα 8. Περιγραφή του MVVM Clean Architecture.	25
Εικόνα 9. Περιγραφή των εξαρτήσεων του app module.....	26
Εικόνα 10. Περιγραφή των εξαρτήσεων του core module.....	27
Εικόνα 11. Περιγραφή των εξαρτήσεων του data module.....	27
Εικόνα 12. Λογότυπο της εφαρμογής (Series Mania).....	34
Εικόνα 13. Η οθόνη εγγραφής χρήστη στην εφαρμογή	37
Εικόνα 14. Η οθόνη σύνδεσης στην εφαρμογή.....	38
Εικόνα 15. Η αρχική οθόνη της εφαρμογής	39
Εικόνα 16. Η οθόνη αναζήτησης της εφαρμογής.....	40
Εικόνα 17. Η οθόνη συζητήσεων στην εφαρμογή.....	41
Εικόνα 18. Η οθόνη πληροφοριών σειράς της εφαρμογής.....	43
Εικόνα 19. Η οθόνη πληροφοριών ενός ηθοποιού στην εφαρμογή.....	44
Εικόνα 20. Η οθόνη πληροφοριών για κατηγορίες σειρών.....	45
Εικόνα 21. Το προφίλ μου στην εφαρμογή.....	47
Εικόνα 22. Η οθόνη στην οποία ανεβάζει ο χρήστης εικόνα προφίλ.....	48
Εικόνα 23. Η οθόνη επιλογής προτιμήσεων.....	49
Εικόνα 24. Η οθόνη αλλαγής ονόματος χρήστη και bio μηνύματος.....	50
Εικόνα 25. Η οθόνη προσθήκης νέας συζήτησης.....	51
Εικόνα 27. Η οθόνη πληροφοριών μιας συζήτησης.....	52
Εικόνα 28. Η οθόνη ρυθμίσεων.....	53
Εικόνα 29. Η οθόνη σχετικά με την εφαρμογή.....	54
Εικόνα 30. Η οθόνη προβολής στατιστικών στην εφαρμογή.....	55
Εικόνα 31. Η οθόνη συχνών ερωτήσεων.....	56

ΠΕΡΙΛΗΨΗ

Με την ολοένα και αυξανόμενη χρήση του διαδικτύου, όλο και περισσότεροι χρήστες έχουν έξυπνες συσκευές για να αξιοποιούν τις δυνατότητες τους προκειμένου να αναβαθμίσουν την καθημερινότητά τους. Η παρούσα διπλωματική αρχικά κάνει μια αναφορά στον κόσμο των εφαρμογών σε κινητά τηλέφωνα και πιο συγκεκριμένα εκείνα του android, ενώ από το επόμενο κεφάλαιο ξεκινά η υλοποίηση μιας εφαρμογής με εξατομικευμένες προτάσεις σειρών. Από την σύλληψη της ιδέας, την ανάλυση απαιτήσεων μέχρι και την επιλογή της πηγής δεδομένων αναλύεται βήμα βήμα η διαδικασία ανάπτυξης. Τέλος, παρουσιάζεται η εφαρμογή που αναπτύχθηκε στο πλαίσιο της διπλωματικής εργασίας και υπάρχουν προτάσεις για περαιτέρω ανάπτυξη της εφαρμογής. Οι γνώσεις που αποκτήθηκαν από την παρούσα διπλωματική ήταν πολύ σημαντικές και οδήγησαν στην βαθύτερη κατανόηση του λειτουργικού συστήματος του android τόσο σε θεωρητικό, όσο και σε πρακτικό υπόβαθρο, χρησιμοποιώντας τις τελευταίες τεχνολογίες στην αγορά.

ABSTRACT

With the ever-increasing use of the internet, more and more people have obtained smart devices in order to take advantage of its capabilities so they will upgrade their daily lives. This dissertation initially makes a reference to the world of applications on mobile phones and more specifically to android, while the next chapter gathers all the necessary information about the implementation of an application with personalized series suggestions and the capability of user interaction. From the conception of the idea and the analysis of the requirements to the selection of the data source, the development process is analyzed step by step. Finally, the application developed in the context of this dissertation is presented and there are suggestions for further development of the application later on. This dissertation led to gain useful knowledge regarding the android operating system both in theoretical and practical level, by using the latest trends and technologies in the market.

ΕΙΣΑΓΩΓΗ

Η παρούσα διπλωματική εργασία, διαπραγματεύεται την ανάπτυξη μιας Android εφαρμογής με εξατομικευμένες προτάσεις σειρών και δυνατότητα αλληλεπίδρασης μεταξύ των χρηστών της.

Στην πρώτη ενότητα της εργασίας, αρχικά αναλύεται η εξάπλωση που γνώρισε το Internet τα τελευταία χρόνια, λόγω της ανάγκης για επικοινωνία και ανταλλαγή δεδομένων μεταξύ των ανθρώπων από οποιοδήποτε σημείο του κόσμου. Στη συνέχεια, περιγράφεται ο τρόπος με τον οποίο μπορεί να περιηγηθεί κάποιος μέσα από την κινητή του συσκευή (smartphone, tablet). Επίσης, οπτικοποιείται το μερίδιο αγοράς των λειτουργικών συστημάτων, δηλαδή, ποιά μερίδα του πληθυσμού χρησιμοποιεί ποιο λειτουργικό σύστημα. Ακολουθεί μια ανάλυση του λειτουργικού συστήματος Android, τα βασικά στοιχεία που το αποτελούν όπως το αρχείο Manifest.xml, τα Activities, τα Fragments και ο κύκλος ζωής τους (activity/fragment lifecycle), η ασφάλεια σε αυτό το λειτουργικό σύστημα. Τέλος, αναφέρεται το ολοκληρωμένο περιβάλλον ανάπτυξης (IDE), το οποίο μπορεί να χρησιμοποιήσει κάποιος με στόχο την ανάπτυξη της δικιάς του εφαρμογής.

Η δεύτερη ενότητα έχει ως σκοπό την συγκέντρωση των βασικών λειτουργικών στοιχείων της εφαρμογής που πρόκειται να αναπτυχθεί. Πιο συγκεκριμένα, περιγράφεται το κενό στην αγορά το οποίο θα πρέπει να προσεγγιστεί και τελικά να επιλυθεί με την χρήση της εφαρμογής. Ακολουθούν οι λειτουργικές απαιτήσεις που θα πρέπει να καλύπτει η εφαρμογή σε μορφή μικρών σεναρίων λειτουργιών (user stories). Στη συνέχεια, περιγράφονται οι μη λειτουργικές απαιτήσεις, τις οποίες θα πρέπει να καλύπτει η εφαρμογή που πρόκειται να αναπτυχθεί. Εν κατακλείδι, αξιολογούνται οι διαφορετικές πηγές δεδομένων (APIs) που προσφέρονται δωρεάν αυτή τη στιγμή για να καλύψουν τις ανάγκες της εφαρμογής σε περιεχόμενο.

Στην τρίτη ενότητα, βρίσκονται συγκεντρωμένες οι τεχνολογίες τις οποίες πρόκειται να αξιοποιήσει η εφαρμογή, προκειμένου να καλύψει τις ανάγκες της, τόσο από πλευράς αρχιτεκτονικής και αποθήκευσης δεδομένων, όσο και γενικότερα της ροής των δεδομένων από την πηγή των δεδομένων, μέχρι και την προβολή τους μπροστά στην αντίστοιχη οθόνη. Γενικότερα, η εφαρμογή χρησιμοποιεί τις τελευταίες προτάσεις της Google στην ανάπτυξη android εφαρμογών. Συνοπτικά, οι τεχνολογίες αυτές είναι η ανάπτυξη εφαρμογής σε Kotlin, MVVM, Clean Architecture με Multimodular λογική, Coroutines, LiveData, Kotlin Flows, Hilt, Retrofit, Gson, Glide, Android Jetpack (Navigation Component, Paging 3, Data Binding, Preferences DataStore) Η εφαρμογή χρησιμοποιεί επίσης την Firebase (Auth, Storage, Firestore) σαν ολοκληρωμένο σύστημα Software as a Service (SaaS).

Η τέταρτη ενότητα πραγματεύεται την παρουσίαση της εφαρμογής που αναπτύχθηκε στα πλαίσια της διπλωματικής εργασίας. Πιο συγκεκριμένα, αναλύονται όλα τα μονοπάτια (application flows) που μπορεί να ακολουθήσει ένας χρήστης σε αυτή και αναλύονται τα χαρακτηριστικά της κάθε οθόνης.

Στην τελευταία ενότητα, συνοψίζεται η διπλωματική εργασία και αναφέρονται κάποια συμπερασματικά στοιχεία για την εκπόνηση της. Τέλος, παρουσιάζονται μερικές ιδέες και τρόποι με τους οποίους θα μπορούσε να συνεχίσει η ανάπτυξη της παρούσας εφαρμογής.

ΚΕΦΑΛΑΙΟ 1

1.1 Η ΕΞΑΠΛΩΣΗ ΤΟΥ INTERNET

Στη σύγχρονη κοινωνία η ανάγκη για επικοινωνία μεταξύ των ανθρώπων και η γρήγορη ανταλλαγή δεδομένων και πολυμέσων είναι πιο επίκαιρη από ποτέ. Το διαδίκτυο έχει λύσει τα χέρια των πολιτών με διάφορους τρόπους. Πλέον, μέσω του διαδικτύου ένας πολίτης, μπορεί να:

- Διεκπεραιώνει τις οικονομικές -και όχι μόνο- συναλλαγές προς τρίτους από το ηλεκτρονικό σύστημα (e-banking) της τράπεζας του
 - Εργάζεται για έναν οργανισμό εξ αποστάσεως με την χρήση ψηφιακών εργαλείων που του επιτρέπουν να είναι σε επαφή με την εργασία του χωρίς να είναι αναγκαία η φυσική του παρουσία.
 - Ενημερώνεται για το τί συμβαίνει στον κόσμο αυτή τη στιγμή μέσω από μια πληθώρα ιστοσελίδων και εφαρμογών.
 - Εκπαιδεύεται και να αναβαθμίζει ηλεκτρονικά τις ψηφιακές του δεξιότητες
 - Να ψυχαγωγείται σε portal πολυμεσικού περιεχομένου όπως οι συνδρομητικές υπηρεσίες ταινιών και όχι μόνο.
-

Οι οργανισμοί αξιοποιούν το διαδίκτυο προκειμένου να κάνουν πιο γνωστά τα προϊόντα τους στους καταναλωτές τους ενώ ταυτόχρονα χρησιμοποιούν την τεχνολογία για να έχουν συστήματα υποστήριξης της επιχειρηματικής διαδικασίας σε κάθε στάδιο. Με αυτόν τον τρόπο, έρχονται πιο κοντά στην στην επίτευξη της αποστολής και γενικότερα του στόχου που έχουν στην κοινωνία. Με το διαδίκτυο ένας οργανισμός μπορεί να δημιουργήσει ένα ηλεκτρονικό προφίλ και να έρθει πιο κοντά στους δυνητικούς της πελάτες, να δημιουργήσει περιεχόμενο κ.α.

Δεν θα μπορούσε να λείπει και το παράδειγμα της ηλεκτρονικής κρατικής διακυβέρνησης. Το κράτος χρόνο με τον χρόνο εξελίσσεται αξιοποιώντας τις λύσεις που προσφέρει το διαδίκτυο, απλοποιώντας χρονοβόρες επαναλαμβανόμενες υπηρεσίες προς όφελος των πολιτών της.

Οι δυνατότητες που προσφέρονται από τη χρήση του διαδικτύου είναι αμέτρητες εφόσον χρησιμοποιούνται με σύνεση. Με τη πάροδο του χρόνου οι συσκευές εξελίσσονται προσφέροντας ολοκληρωμένες λύσεις στις καθημερινές ανάγκες που συνεχώς αυξάνονται, προκειμένου να διευκολύνουν την καθημερινότητα των ανθρώπων.

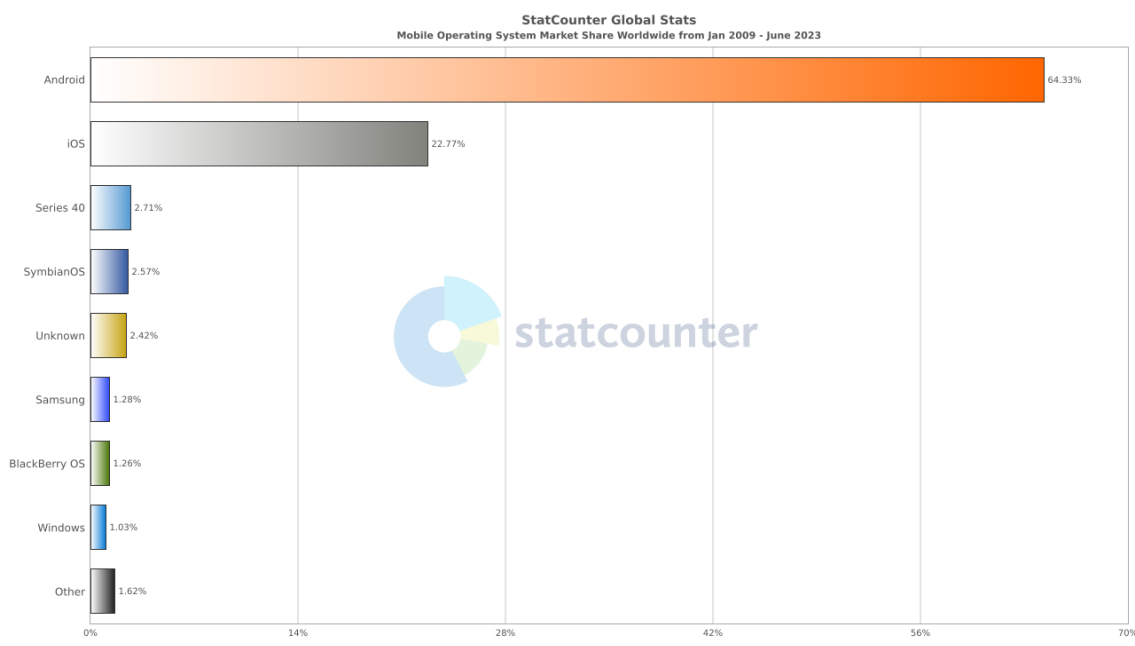
1.2 ΕΞΥΠΝΕΣ ΣΥΣΚΕΥΕΣ (SMARTPHONES, TABLETS)

Η κινητή τηλεφωνία έφερε επανάσταση στην επικοινωνία τα τελευταία χρόνια. Σήμερα, δεν μπορεί κανείς να διανοηθεί πως κάποιος δεν έχει στην κατοχή του τουλάχιστον ένα κινητό. Με τον καιρό τα κινητά τηλέφωνα έφεραν πάνω τους όλο και περισσότερα χαρακτηριστικά ενσωματώνοντας λειτουργίες από άλλες καθημερινές συσκευές, όπως η κάμερα. Φυσικά ένα τηλέφωνο δεν μπορεί σε καμία περίπτωση να ανταγωνιστεί μια κάμερα τελευταίας τεχνολογίας, ωστόσο είναι επαρκής για καθημερινή χρήση.

Οι πρώτες προσπάθειες ενσωμάτωσης του διαδικτύου στα κινητά τηλέφωνα ξεκίνησε στη δεκαετία του 1990. Το 1996 η Nokia παρουσίασε το πρώτο κινητό τηλέφωνο με πρόσβαση στο WAP (Wireless Application Protocol), ένα πρότυπο για την πρόσβαση σε ιστοσελίδες.

Σήμερα, οι κινητές συσκευές έχουν εξελιχθεί δραματικά, προσφέροντας δυνατότητες όπως γρήγορη πλοήγηση σε ιστοσελίδες πλούσιο σε περιεχόμενο με αλληλεπίδραση του χρήστη. Ο χρήστης είτε χρησιμοποιώντας το Internet είτε εφαρμογές στη συσκευή του, μπορεί να επικοινωνεί, να συμμετέχει σε κοινωνικά δίκτυα και να αλληλεπιδρά με άλλους χρήστες. Μπορεί επίσης να διαβάζει τα νέα της επικαιρότητας, να πλοηγείται σε άγνωστα σε αυτόν σημεία της πόλης του χρησιμοποιώντας τις δυνατότητες του GPS της συσκευής του, να τραβάει φωτογραφίες ή βίντεο στιγμές της καθημερινότητας του, να ψυχαγωγείται μέσω online παιχνιδιών.

Σύμφωνα με την statcounter (<https://gs.statcounter.com/>), τα μερίδια αγοράς παρουσιάζονται παρακάτω. Παρατηρείται πως το Android έχει ποσοστό αγοράς 64.33% ενώ ακολουθεί το iOS με ποσοστό 22.77%.



Εικόνα 1. Το market share των smartphones (statcounter)

1.3 ΤΡΟΠΟΣ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ ΓΙΑ ΚΙΝΗΤΑ

Μια εφαρμογή ανάλογα με τον τρόπο ανάπτυξης της μπορεί να είναι Native, Hybrid ή Web app. Αυτός ο διαχωρισμός πρακτικά αναφέρεται στις τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής.

«Ανάπτυξη εφαρμογής σε περιβάλλον Android με εξατομικευμένες προτάσεις σειρών και δυνατότητα interaction με 10 χρήστες»

Native: Όταν αναφέρεται κάποιος σε μια native εφαρμογή, συνήθως αναφέρεται σε μια εφαρμογή που έχει αναπτυχθεί χρησιμοποιώντας την native γλώσσα προγραμματισμού του εκάστοτε λειτουργικού συστήματος. Για παράδειγμα, μια android εφαρμογή μπορεί να είναι γραμμένη με Kotlin ή Java ενώ μια εφαρμογή για iOS έχει αναπτυχθεί χρησιμοποιώντας την Swift ή την Objective-C. Με την ανάπτυξη μιας native εφαρμογής, οι προγραμματιστές έχουν πλήρη πρόσβαση στις δυνατότητες και λειτουργίες της εκάστοτε συσκευής, όπως οι σένσορες, η λίστα τηλεφωνικού καταλόγου κ.α.

Hybrid: Μια υβριδική (hybrid) εφαρμογή χρησιμοποιεί συστατικά της native πλατφόρμας ενώ είναι γραμμένη σε διαφορετικές τεχνολογίες. Μια hybrid εφαρμογή θα μπορούσε να είναι ανεπτυγμένη με την χρήση της React Native (React και JavaScript), θα μπορούσε να είναι μια Flutter εφαρμογή γραμμένη σε Dart, μια Ionic εφαρμογή (cross-platform mobile application που βασίζεται σε τεχνολογίες HTML, CSS, JavaScript) ή Xamarin (C# και .NET framework).

Web apps: Ένα Web application είναι μια εφαρμογή που έχει αναπτυχθεί χρησιμοποιώντας τις γλώσσες HTML, CSS και JavaScript. Μια εφαρμογή web αυτόματα περιορίζει τον χρήστη να αξιοποιεί τις δυνατότητες της εφαρμογής μέσα από τον browser της εφαρμογής.

1.4 ANDROID



Εικόνα 2: Το λογότυπο του λειτουργικού συστήματος android

Το Android είναι ένα λειτουργικό σύστημα (Λ.Σ) ανοιχτού κώδικα (open source). Το Android ξεκίνησε το 2003 όταν ο Andry Rubin, Rich Miner, Nick Sears και ο Chris White ίδρυσαν μια εταιρεία που ονομαζόταν Android Inc. Αυτή η εταιρεία είχε ως βασικό στόχο να αναπτύξει ένα λειτουργικό σύστημα για ψηφιακές φωτογραφικές μηχανές. Όταν κατάλαβαν την αξία του έργου έστρεψαν την προσοχή τους στην ανάπτυξη ενός Λ.Σ για κινητά τηλέφωνα.

Το Android βασίστηκε στο λογισμικό Linux (έκδοση 2.6 Kernel) και σύμφωνα με την Google χρησιμοποιείται σε 2.5 δισεκατομμύρια συσκευές. Αυτές οι συσκευές δεν είναι αποκλειστικά smartphones ή tablets, αλλά και IoT (Internet of Things) συσκευές ή έξυπνες τηλεοράσεις (Smart TVs) ή Android Auto.

Η κύρια γλώσσα για ανάπτυξη εφαρμογών Android είναι η Kotlin, όπως ανακοινώθηκε από τη Google στο Google I/O (συνέδριο προγραμματιστών) το 2019. Εκτός από την Kotlin, προκειμένου να αναπτυχθεί μια εφαρμογή κινητών τηλεφώνων μπορεί να χρησιμοποιηθεί και η Java.

Το Android είναι ανοιχτό για ανοιχτό για όλους, προγραμματιστές, σχεδιαστές και κατασκευαστές συσκευών. Κάθε άνθρωπος είναι μοναδικός και για αυτό τον σκοπό το android προσπαθεί να λαμβάνει υπόψη τις ιδιαιτερότητες των χρηστών του και στη συνέχεια να αναπτύσει προϊόντα και υπηρεσίες που να τις καλύπτουν με τον καλύτερο δυνατό τρόπο. Ένας χρήστης μπορεί να χρησιμοποιήσει αναγνώστες θόνης, οδηγούς πεζοπορίας AR και πολλά ακόμη. Το Android είναι ένα ασφαλές Λ.Σ και κάθε εφαρμογή τρέχει σε ένα δικό της sandbox περιβάλλον. Επίσης, στο Play Store που διατίθενται οι εφαρμογές, υπάρχει το Google Play Protect το οποίο σκανάρει όλες τις εφαρμογές, το λειτουργικό λαμβάνει συχνές βελτιώσεις ασφαλείας και η γενικότερη πλατφόρμα αναπτύσσεται συνεχώς.

Το Android συνεχώς εξελίσσεται προκειμένου να συμβαδίσει με τις συνεχώς αυξανόμενες ανάγκες των χρηστών του. Οι εκδόσεις του android καθώς και η ημερομηνία λανσαρίσματος τους παρουσιάζονται παρακάτω:

Android 1.0 (Android One) - 23 Σεπτεμβρίου 2008
Android 1.5 Cupcake - 30 Απριλίου 2009
Android 1.6 Donut - 15 Σεπτεμβρίου 2009
Android 2.0/2.1 Eclair - 26 Οκτωβρίου 2009
Android 2.2 Froyo - 20 Μαΐου 2010
Android 2.3 Gingerbread - 6 Δεκεμβρίου 2010
Android 3.0/3.1/3.2 Honeycomb - 22 Φεβρουαρίου 2011
Android 4.0 Ice Cream Sandwich - 18 Οκτωβρίου 2011
Android 4.1/4.2/4.3 Jelly Bean - 9 Ιουλίου 2012
Android 4.4 KitKat - 31 Οκτωβρίου 2013
Android 5.0/5.1 Lollipop - 12 Νοεμβρίου 2014
Android 6.0 Marshmallow - 5 Οκτωβρίου 2015
Android 7.0/7.1 Nougat - 22 Αυγούστου 2016
Android 8.0/8.1 Oreo - 21 Αυγούστου 2017
Android 9 Pie - 6 Αυγούστου 2018
Android 10 (Q) - 3 Σεπτεμβρίου 2019
Android 11 (11) - 8 Σεπτεμβρίου 2020
Android 12 (S) - 4 Οκτωβρίου 2021
Android 13 (T) - 15 Αυγούστου 2022
Android 14 - Φλεβάρης 2023 (Developer Preview)

Ενδιαφέρον έχει επίσης το ποσοστό των συσκευών που χρησιμοποιούν την κάθε έκδοση android. Παρακάτω απεικονίζεται το διάγραμμα με την αθροιστική κατανομή πιθανότητας (σωρευτική κατανομή - cumulative distribution) των εκδόσεων του android στις συσκευές.



Εικόνα 3: Συσσωρευτική κατανομή των android releases.

1.5 ΒΑΣΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ ANDROID

1.5.1 Αρχείο Manifest

Το αρχείο AndroidManifest.xml υπάρχει πάντα σε μια android εφαρμογή και παίζει πολύ σημαντικό ρόλο. Αυτό το αρχείο μπορεί να εντοπιστεί στο project structure στον root folder. Αυτό το αρχείο συνοψίζει ουσιαστικά ολόκληρη την εφαρμογή που έχει αναπτυχθεί. Πιο συγκεκριμένα, σε αυτό το αρχείο βρίσκονται συγκεκριμένα όλα τα Activities που έχει η εφαρμογή, τα services, τους broadcast receivers και τους content providers. Στο πάνω μέρος του αρχείου οι προγραμματιστές της εφαρμογής δηλώνουν τα permissions της εφαρμογής. Ακόμα, ορίζονται χρήσιμα στοιχεία για την συνολική εικόνα της εφαρμογής, όπως το λογότυπο

που θα εμφανίζεται όταν είναι εγκατεστημένη η εφαρμογή. Τέλος, κρατάει την πληροφορία για το τι hardware ή software χαρακτηριστικά χρειάζεται η εφαρμογή ώστε να λειτουργεί σωστά.

1.5.2 Δραστηριότητες (Activities)

Στο Android υπάρχει η έννοια της Δραστηριότητας (Activity). Ένα Activity είναι ένα στοιχείο (unit) με το οποίο αλληλεπιδρά ο χρήστης. Σε ένα Activity, μπορούν να ζούν μία ή πολλές οθόνες. Το Activity έχει δικό του κύκλο ζωής (lifecycle). Το σύστημα επικαλείται (invokes) διάφορες μεθόδους του κύκλου ζωής μόλις αλλάζει κατάσταση (state).

Γενικά το android προσθέτει τα Activities της εφαρμογής σε μια στοίβα (backstack) ακριβώς με την σειρά στην οποία κάθε activity ξεκίνησε. Ο κύκλος ζωής ενός Activity αναλυτικότερα παρουσιάζεται παρακάτω.

onCreate: Το Activity μόλις δημιουργήθηκε. Αυτή είναι η μέθοδος στην οποία θα πρέπει να καλείται η setContentView() προκειμένου να οριστεί το layout της οθόνης. Επίσης, στο σημείο αυτό μπορούν να αρχικοποιηθούν μεταβλητές χρήσιμες για αυτό. Η οθόνη δεν είναι ορατή στον χρήστη.

onStart: Η onStart καλείται μετά από την onCreate() ή μετά από την onRestart(). Όταν δηλαδή το activity είχε σταματήσει (onStop) να είναι ορατό στον χρήστη προηγουμένως. Η οθόνη δεν είναι ορατή στον χρήστη.

onResume: Το Activity βρίσκεται πλέον στην κορυφή της στοίβας (backstack). Η οθόνη γίνεται ορατή στον χρήστη. Ο χρήστης μπορεί να αλληλεπιδράσει με τα στοιχεία της οθόνης.

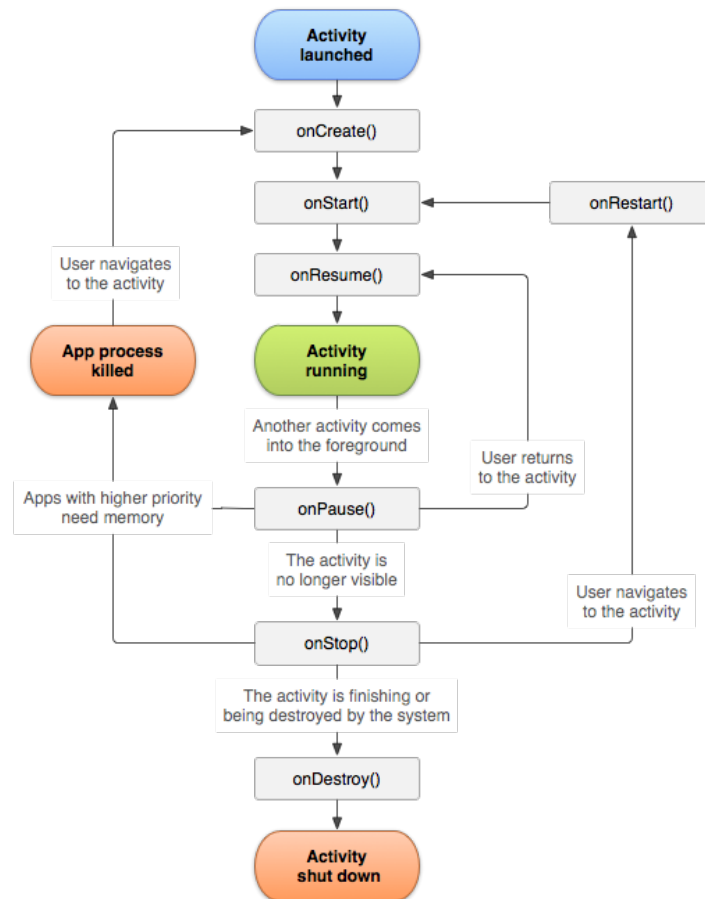
onPause: Η onPause() καλείται όταν ο χρήστης δεν έχει άμεση επαφή με την οθόνη του Activity, αλλά η οθόνη συνεχίζει να είναι ορατή στον χρήστη. Για παράδειγμα, άνοιξε κάποιο dialog που ζητάει την είσοδο του χρήστη. Ο χρήστης μπορεί να επιστρέψει στο Activity

onStop: Καλείται όταν το Activity δεν είναι πλέον ορατό στον χρήστη. Ανάλογα με την διάθεση του χρήστη, μπορεί είτε να καλεστεί η onRestart() και ο χρήστης να επανέλθει στην εφαρμογή, είτε να καλεστεί η onDestroy().

onRestart: Καλείται μετά την onStop() όταν ο χρήστης επιστρέψει στην εφαρμογή και το Activity θα πρέπει να ξαναεμφανιστεί στον χρήστη.

onDestroy: Καλείται όταν το Activity καταστρέφεται για να απελευθερωθούν πόροι συστήματος. Αυτό μπορεί να συμβαίνει καθώς το Activity ολοκληρώθηκε (καλέστηκε η finish()) σε αυτό είτε λόγω το σύστημα κατέστρεψε το Activity για να σώσει χώρο στη συσκευή.

Παρακάτω ακολουθεί μια γραφική απεικόνιση του κύκλου ζωής ενός Activity.



Εικόνα 4. Ο κύκλος ζωής (lifecycle) ενός activity

1.5.3 Κομμάτι Οθόνης (Fragment)

Ένα Fragment στο Android αντιπροσωπεύει ένα κομμάτι του User Interface της εφαρμογής. Ένα Fragment φιλοξενείται σε κάποιο Activity. Η χρήση των Fragments στο android είναι μια προσέγγιση με την οποία πετυχαίνουμε κατάτμηση της λογικής (modularity). Κάθε Fragment έχει το δικό του κύκλο ζωής και μπορεί να ακούει τα δικά του events.

Ο κύκλος ζωής ενός Fragment αναλυτικότερα παρουσιάζεται παρακάτω:

onAttach: Η onAttach() καλείται μόλις το Fragment ενσωματωθεί σε κάποιο Activity

onCreate: Η onCreate() χρησιμοποιείται για να αρχικοποιηθεί το εκάστοτε Fragment.

onCreateView: Η onCreateView() καλείται για να δημιουργήσει και να επιστρέψει το layout του Fragment.

onStart: Η onStart() καλείται όταν το Fragment πάει να γίνει ορατό στον χρήστη.

onResume: Η οθόνη του Fragment γίνεται ορατή στον χρήστη. Ο χρήστης μπορεί να αλληλεπιδράσει με τα στοιχεία της οθόνης.

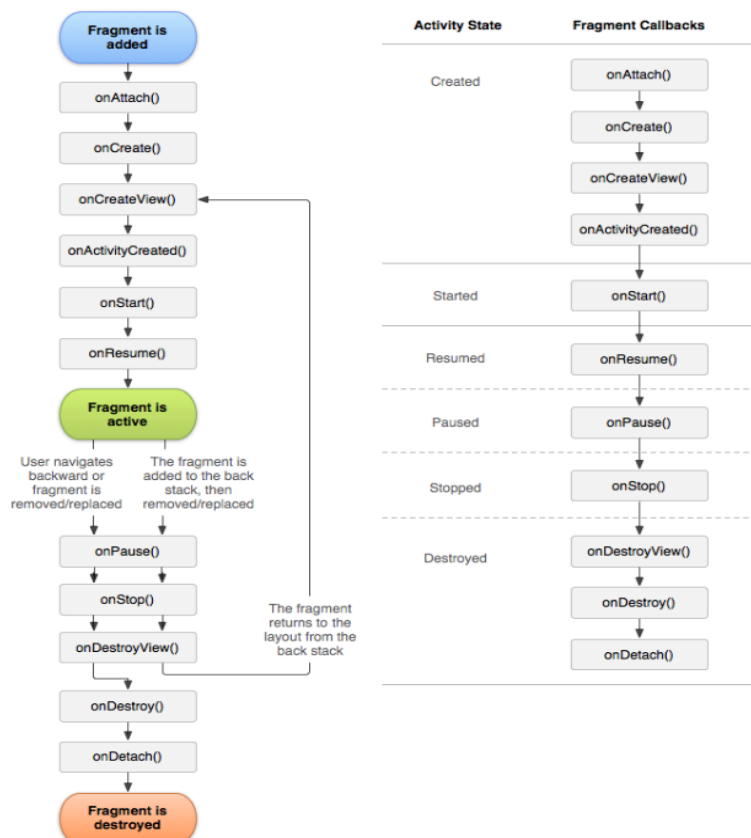
onPause: Η onPause() καλείται όταν ο χρήστης δεν έχει άμεση επαφή με την οθόνη του Fragment, αλλά η οθόνη συνεχίζει να είναι ορατή στον χρήστη. Για παράδειγμα, άνοιξε κάποιο dialog που ζητάει την είσοδο του χρήστη.

onStop: Καλείται όταν το Fragment δεν είναι πλέον ορατό στον χρήστη. Ανάλογα με την διάθεση του χρήστη, μπορεί είτε να καλεστεί η onRestart() και ο χρήστης να επανέλθει στην εφαρμογή, είτε να καλεστεί η onDestroyView().

onDestroyView: Η onDestroyView() χρησιμοποιείται προκειμένου να καθαριστούν τα resources που χρησιμοποιούνταν από το Fragment.

onDestroy: Καλείται όταν το Fragment σταματάει να χρησιμοποιείται πια από κάποιο Activity.

onDetach: Η onDetach() καλείται μόλις το Fragment σταματά να είναι ενσωματωμένο σε κάποιο Activity.



Εικόνα 5. Ο κύκλος ζωής (lifecycle) ενός fragment

1.5.4 Προθέσεις (Intents)

Στην ανάπτυξη εφαρμογών Android, ένα Intent αναπαριστά μια πρόθεση ή μια ενέργεια που εκτελείται εντός του συστήματος. Ένα Intent μπορεί να ξεκινήσει ένα νέο Activity, να επικοινωνήσει με τρίτες εφαρμογές ή να τροποποιήσει τη συμπεριφορά του συστήματος.

Σε γενικές γραμμές, τα Intents χρησιμοποιούνται για την εκτέλεση ενεργειών μεταξύ διαφορετικών στοιχείων. Σε κάποιο Intent μπορεί ο προγραμματιστής να περάσει χρήσιμη πληροφορία. Συνήθως, η βασική πληροφορία που μπορεί κάποιος να περάσει σε ένα Intent είναι το action το οποίο περιγράφει τι ακριβώς θα γίνει (ACTION_VIEW, ACTION_EDIT) και τα data σε τύπο Uri.

1.5.5 Broadcast Receivers

Στο android ένας broadcast receiver είναι ένα από τα βασικά στοιχεία του android και επιτρέπει στην εφαρμογή να λαμβάνει και να ανταποκρίνεται σε συγκεκριμένα μηνύματα (broadcast messages) σε όλο το σύστημα ή σε συγκεκριμένο κομμάτι της εφαρμογής. Αυτά τα μηνύματα, ονομάζονται Intents και μπορούν να σταλούν από το σύστημα, άλλες εφαρμογές ή από κομμάτια της εφαρμογής.

Οι broadcast receivers είναι χρήσιμοι σε σενάρια όπου χρειάζεται η εφαρμογή να αντιδρά σε ορισμένα συμβάντα, όπως αλλαγή στη συνδεσιμότητα δικτύου, εισερχόμενα μηνύματα SMS, επανεκκινήσεις συστήματος, προειδοποιήσεις χαμηλής μπαταρίας ή ακόμα και προσαρμοσμένα συμβάντα εντός της εφαρμογής. Με την εγγραφή (register) ενός broadcast receiver, η εφαρμογή μπορεί να ειδοποιηθεί για αυτά τα συμβάντα και να λάβει τις κατάλληλες ενέργειες.

1.6 ΑΣΦΑΛΕΙΑ

Οι εφαρμογές στο λειτουργικό σύστημα του Android τρέχουν σε ένα δικό τους προστατευμένο περιβάλλον (sandbox) στο οποίο αν κάτι δεν πάει σωστά, δεν ρισκάρει την συνολική ασφάλεια της συσκευής. Όταν γίνεται διάθεση μια εφαρμογής που έχει αναπτυχθεί στο android, λόγω της διαδικασίας κατά την οποία προστίθενται τα χαρακτηριστικά του ατόμου ή του οργανισμού που ανέπτυξε την εκάστοτε εφαρμογή, αποτρέπεται τους προγραμματιστές από το να χρησιμοποιούν άσκοπα διεπαφές και άδειες χωρίς να τους έχουν εκχωρηθεί από τον χρήστη.

Από το Android 9 και μετά δίνεται η δυνατότητα μέσω του Biometric Prompt API οι εφαρμογές να μπορούν να ζητήσουν από τον χρήστη να κάνει αυθεντικοποίηση μέσω βιομετρικών χαρακτηριστικών. Τέλος, το Android προσφέρει το Android Keystore για σκοπούς ασφαλείας, με υποστήριξη υλικού που παρέχει δημιουργία κλειδιών, εισαγωγή και εξαγωγή ασύμμετρων κλειδιών, εισαγωγή μη συμμετρικών κλειδιών, ασύμμετρη κρυπτογράφηση και αποκρυπτογράφηση με κατάλληλες λειτουργίες padding modes και πολλά άλλα.

1.7 ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ ΜΕ ΤΟ ANDROID STUDIO



Εικόνα 6. Το λογότυπο του Android Studio

Το Android Studio είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE - Integrated Development Environment) που προτείνει η Google για την ανάπτυξη android εφαρμογών. Ανακοινώθηκε στις 16 Μαΐου 2013 στο συνέδριο Google I/O. Το android studio είναι ένα λογισμικό ανάπτυξης που διατίθεται ελεύθερα με την άδεια Apache License 2.0 και έχει αναπτυχθεί με βάση το IntelliJ IDEA. Αποτελείται από ένα πλήρες σετ εργαλείων που στοχεύουν στην δημιουργία android εφαρμογών και διευκολύνουν στην διαδικασία της αποσφαλμάτωσης (debugging) της εφαρμογής, την δοκιμή (testing) και την εκτέλεση android εφαρμογών. Ο προγραμματιστής μπορεί να δημιουργήσει εικονικές συσκευές (emulators) επιλέγοντας από μια λίστα τόσο την συσκευή όσο και την έκδοση του λειτουργικού συστήματος android που επιθυμεί να τρέχει η εικονική συσκευή.

Εκτός από το σύστημα οπτικής απεικόνισης του κώδικα το οποίο είναι σχεδιασμένο να υποστηρίζει τις γλώσσες προγραμματισμού, ενσωματώνει πολλές χρήσιμες λειτουργίες όπως εκείνη της δημιουργίας APKs προκειμένου να γίνουν προωθηθούν τα εκτελέσιμα αρχεία των εφαρμογών στην αγορά. Ακόμα, διαθέτει προηγμένα εργαλεία για την ανάλυση της απόδοσης της εφαρμογής και τη βελτιστοποίηση της.

ΚΕΦΑΛΑΙΟ 2

2.1 ΑΝΑΠΤΥΞΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Στον κύκλο της ανάπτυξης λογισμικού, πριν από το στάδιο της ανάπτυξης της εφαρμογής είναι πολύ σημαντικό να ξεκαθαρίζεται το πρόβλημα το οποίο υπάρχει και το τι ακριβώς καλούμαστε να επιλύσουμε. Αφού καταγραφεί το πρόβλημα, θα πρέπει να συγκεντρωθούν οι λειτουργικές και οι μη λειτουργικές απαιτήσεις του συστήματος το οποίο πρόκειται να υλοποιηθεί. Ακολουθεί μια ανάλυση του προβλήματος, καθώς και οι λειτουργικές και μη λειτουργικές απαιτήσεις της εφαρμογής που θα αναπτυχθεί στα επόμενα κεφάλαια.

2.2 ΤΟ ΠΡΟΒΛΗΜΑ

Το πρόβλημα που καλούμαστε να προσεγγίσουμε είναι η ανάπτυξη μιας εφαρμογής η οποία θα δίνει την δυνατότητα στον χρήστη να πλοηγηθεί και να βρει σειρές, να διαβάσει πληροφορίες για σειρές και ηθοποιούς. Ακόμα θα πρέπει να δίνεται η δυνατότητα να δημιουργήσει το προσωπικό του προφίλ στην εφαρμογή και να αλληλεπιδράσει με άλλους χρήστες της εφαρμογής ακολουθώντας τους χτίζοντας ένα δίκτυο χρηστών (community). Τέλος, ο χρήστης της εφαρμογής θα πρέπει να μπορεί να δημιουργήσει συζητήσεις και να απαντήσει σε ήδη υπάρχουσες της εφαρμογής με άλλους χρήστες.

2.3 ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ

Μια λειτουργική απαίτηση μοιάζει με έναν κανόνα ή μια οδηγία που λέει σε ένα σύστημα πώς πρέπει να συμπεριφέρεται. Είναι σαν μια λίστα με χαρακτηριστικά που θα πρέπει να καλύπτει το σύστημα για να καλύπτει τις ανάγκες του χρήστη που το χρησιμοποιεί. Στην εφαρμογή που πρόκειται να αναπτυχθεί οι λειτουργικές απαιτήσεις που έχουν προκύψει παρουσιάζονται παρακάτω και παρουσιάζονται με την μορφή μικρών σεναρίων λειτουργιών (user stories).

Είσοδος στην εφαρμογή

- Σαν χρήστης, θέλω να εγγραφώ ώστε να δημιουργήσω τον λογαριασμό μου στην εφαρμογή.
 - Σαν χρήστης, θέλω να μπορώ να συνδεθώ στον λογαριασμό που έχω δημιουργήσει ώστε να μπω στην εφαρμογή.
-

Σειρές

-
- Σαν χρήστης, θέλω να βλέπω μέσα στην εφαρμογή λίστες με σειρές ανάλογα την κατηγορία που ανήκουν (σειρές που έρχονται σύντομα, κορυφαίες, τάσεις, ανά κατηγορία, προτάσεις).
 - Σαν χρήστης θέλω να μπορώ να πατήσω σε μια σειρά και να διαβάσω τις πληροφορίες της συγκεκριμένης σειράς.
 - Σαν χρήστης θέλω να μπορώ να επισκεφθώ το προφίλ του ηθοποιού που παίζει σε κάποια σειρά
 - Σαν χρήστης θέλω να δω τις ταινίες που είναι γνωστος κάποιος ηθοποιός απο κάποια σειρά στην εφαρμογή
 - Σαν χρήστης θέλω να μπορώ να αναζητήσω σειρές με βάση την κατηγορία στην οποία ανήκουν.
 - Σαν χρήστης θέλω να μπορώ να αναζητήσω σειρές με βάση τον τίτλο της σειράς.
 - Σαν χρήστης θέλω να μπορώ να υποβάλλω μια αξιολόγηση για κάποια σειρά
 - Σαν χρήστης θέλω να μπορώ να γράψω μια κριτική αλλά και να διαβάσω όσες ήδη υπάρχουν για κάποια σειρά.
 - Σαν χρήστης θέλω να μπορώ να δω προτάσεις σειρών ανάλογα με αυτή που διαβάζω.
 - Σαν χρήστης θέλω να μπορώ να βάλω μια σειρά στη λίστα προβολής μου.
-

Προφίλ

-
- Σαν χρήστης θέλω να μπορώ να επισκεφθώ το προφίλ μου.
 - Σαν χρήστης θέλω να μπορώ να διαβάσω τις πληροφορίες του προφίλ μου.
 - Σαν χρήστης θέλω να μπορώ να δω πόσους χρήστες ακολουθώ, ποιοι με ακολουθούν και πόσες ταινίες έχω στη λίστα προβολής μου.
 - Σαν χρήστης θέλω να μπορώ να ορίζω προτιμήσεις για τις σειρές που μου αρέσουν στην εφαρμογή
 - Σαν χρήστης θέλω να μπορώ να επεξεργαστώ τα βασικά στοιχεία του λογαριασμού μου (εικόνα, όνομα χρήστη και ενημερωτικό κείμενο χρήστη - bio)
 - Σαν χρήστης θέλω να μπορώ να δω το προφίλ ενός άλλου χρήστη μέσα στην εφαρμογή
 - Σαν χρήστης θέλω να μπορώ να ακολουθήσω κάποιον άλλον χρήστη.
 - Σαν χρήστης θέλω να μπορώ να αποσυνδεθώ από τον λογαριασμό μου.
-

Συζητήσεις

-
- Σαν χρήστης επιθυμώ να μπορώ να διαβάσω τις συζητήσεις που υπάρχουν ανοιχτές στην εφαρμογή.
 - Σαν χρήστης θέλω να μπορώ να γράψω μια νέα συζήτηση (thread) στην εφαρμογή.
 - Σαν χρήστης θέλω να μπορώ να διαβάσω τις απαντήσεις μιας συζήτησης.
 - Σαν χρήστης θέλω να μπορώ να απαντήσω σε μια συζήτηση.
-

2.4 ΜΗ ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ

Επιδόσεις: Η εφαρμογή θα πρέπει να φορτώνει τα δεδομένα που χρειάζεται ανάλογα την περίπτωση χωρίς καθυστερήσεις, γρήγορα και απλά. Μεγάλες καθυστερήσεις μπορούν να αποτελέσουν αποτρεπτικό παράγοντα χρήσης της εφαρμογής από τους χρήστες.

Κατανάλωση δεδομένων: Η εφαρμογή προκειμένου να στείλει και να λάβει πληροφορία καταναλώνει δεδομένα. Θα πρέπει η εφαρμογή να λαμβάνει υπόψη πως το να καταναλώνεται μεγάλος όγκος δεδομένων δεν είναι σωστό. Θα πρέπει η εφαρμογή να πραγματοποιεί μόνο τις απαραίτητες κλήσεις για να φορτώσει το διαθέσιμο περιεχόμενο. Όπου είναι εφικτό, θα πρέπει να χρησιμοποιείται ο σελιδοποίηση (paging) για να φορτώνεται σταδιακά το περιεχόμενο ή να χρησιμοποιείται τεχνικές προσωρινής αποθήκευσης (caching).

Ασφάλεια και ακεραιότητα δεδομένων: Όλα τα στοιχεία που υπάρχουν στην εφαρμογή θα πρέπει να διασφαλίζεται ότι αποθηκεύονται στη βάση δεδομένων με ασφαλή τρόπο έτσι ώστε να μην χάνονται ούτε να μπορούν να απαντηθούν από τρίτα πρόσωπα μη εξουσιοδοτημένα στην εφαρμογή.

Εμπειρία χρήστη: Μια εφαρμογή είναι ένας ζωντανός οργανισμός που συνεχώς εξελίσσεται. Η εμπειρία χρήστη (User Experience) σε μια εφαρμογή, δηλαδή η εμπειρία την οποία βιώνει ένας χρήτης όταν χρησιμοποιεί μια εφαρμογή τόσο από πλευράς χρησικότητας, όσο και η οπτική απεικόνιση των στοιχείων που την απαρτίζουν παίζουν καθοριστικό ρόλο επιλογής και διατήρησης των χρηστών εντός της εφαρμογής. Βασικός στόχος ανάπτυξης είναι μια εφαρμογή η οποία είναι χρηστική και καλύπτει τις πραγματικές ανάγκες των χρηστών, είναι εύκολη στη χρήση, η μετάβαση από μια οθόνη σε κάποια άλλη είναι απλή και γενικότερα δεν δημιουργείται σύγχυση στον χρήστη στην χρήση της.

Συμβατότητα: Το android είναι ένα λειτουργικό σύστημα που έχει πολλές εκδόσεις. Η εφαρμογή θα πρέπει να είναι συμβατή με όσο το δυνατόν πιο πολλές εκδόσεις android υποστηρίζοντας τους χρήστες που επιλέγουν να μην αναβαθμίσουν, θα πρέπει να απεικονίζεται επαρκώς σε διαφορετικές διαστάσεις των οθονών αλλά και σε οθόνες με διαφορετική ανάλυση.

Συντήρηση: Η εφαρμογή θα πρέπει να είναι κατατημένη σε ξεχωριστές κλάσεις έτσι ώστε να είναι εύκολη η συντήρηση της και θα είναι ευκολότερο να αναπτυχθεί (scale) περαιτέρω στο μέλλον.

2.5 ΔΕΔΟΜΕΝΑ ΣΕΙΡΩΝ ΠΟΥ ΘΑ ΧΡΗΣΙΜΟΠΟΙΕΙ Η ΕΦΑΡΜΟΓΗ

Η εφαρμογή που θα αναπτυχθεί θα πρέπει να τραβάει δεδομένα από κάποιο σημείο στο διαδίκτυο για να εμφανίζει λίστες με σειρές, το περιεχόμενό τους, πληροφορίες για ηθοποιούς και σειρές που είναι γνωστοί, είδη σειρών και δυνατότητα αναζήτησης με βάση τον τίτλο της σειράς. Για τον σκοπό αυτό, η εφαρμογή θα πρέπει να αλληλεπιδράσει με ένα ή περισσότερα APIs (Application Programming Interface - Διασύνδεση Προγραμματισμού Εφαρμογών). Τα πιο δημοφιλή APIs παρουσιάζονται παρακάτω προκειμένου να επιλεγεί τελικά αυτό που θα καλύπτει καλύτερα τις προγραμματιστικές ανάγκες της εφαρμογής. Σύντομη παρουσίαση εναλλακτικών APIs για την εφαρμογή:

2.5.2 Δυνατότητα αξιολόγησης σειράς σαν επισκέπτης

Στην εφαρμογή θα πρέπει να δίνεται η δυνατότητα στον χρήστη να υποβάλλει μια αξιολόγηση σε κάποια σειρά. Αυτό με το TMDb χρειάζεται μια μικρή προεργασία. Για την ακρίβεια, προκειμένου να σταλεί κάποια αξιολόγηση (POST request), θα πρέπει να γίνει παραγωγή (generate) από το API ενός guest session ID. Αυτό το guest session ID θα πρέπει να είναι μυστικό και να το κρατάει αποθηκευμένο η εφαρμογή και να το χρησιμοποιεί για όσο είναι ενεργό. Αν λήξει, θα πρέπει να παράγει ένα νέο.

Συνοψίζοντας την διαδικασία παραγωγής ενός guest session ID, η εφαρμογή θα πρέπει να στέλνει ένα GET HTTP request στο παρακάτω endpoint:

https://api.themoviedb.org/3/authentication/guest_session/new

Το παραπάνω endpoint θα μας απαντήσει ιδανικά ένα guest session token και θα περιέχει την ημερομηνία λήξης του. Στη συνέχεια, όταν έρθει η στιγμή προσθήκης μιας αξιολόγησης στην εφαρμογή ο χρήστης στην κλήση της κριτικής θα πρέπει να ορίζει το guest session id ως query parameter του post request.

ΚΕΦΑΛΑΙΟ 3

3.1 ΤΕΧΝΟΛΟΓΙΕΣ ΠΟΥ ΘΑ ΧΡΗΣΙΜΟΠΟΙΗΘΟΥΝ ΣΤΗΝ ΑΝΑΠΤΥΞΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

3.1.1 Kotlin (Γλώσσα Προγραμματισμού)

Η Kotlin είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού η οποία δημιουργήθηκε το 2011 από την JetBrains, μια εταιρία ανάπτυξης λογισμικού. Η Kotlin τρέχει πάνω στον JVM (Java Virtual Machine) και είναι πλήρως διαλειτουργική με την Java, δηλαδή μπορούν κλάσεις και αρχεία γραμμένα σε Kotlin να συνυπάρχουν με εκείνα που είναι γραμμένα σε Java. Έχει επηρεαστεί από τις γλώσσες Java, Scala, C# και Groovy.

-
- Η Kotlin μπορεί να χρησιμοποιηθεί σε μεγάλο εύρος εφαρμογών όπως:
 - Ανάπτυξη εφαρμογών για mobile συσκευές (τόσο για native android development όσο και με το kotlin multiplatform - την δυνατότητα δηλαδή να γραφτεί μια κοινή βάση κώδικα σε μια εφαρμογή η οποία θα χρησιμοποιηθεί σε διάφορες πλατφόρμες αργότερα όπως Android, iOS, Web και Backend).
 - Ανάπτυξη εφαρμογών σε servers
 - Ανάπτυξη εφαρμογών web
 - Ανάπτυξη desktop εφαρμογών
-

Η Google στο Google I/O (συνέδριο προγραμματιστών) το 2019 έκανε μια σημαντική ανακοίνωση. Πιο συγκεκριμένα, ανακοίνωσε την Kotlin σαν την γλώσσα που προτείνει πλέον για ανάπτυξη εφαρμογών σε περιβάλλον android (Kotlin first applications). Αυτό το γεγονός έκανε την Kotlin να γνωρίσει μεγάλη δημοφιλία καθώς όλο και περισσότεροι προγραμματιστές πείστηκαν να αναβαθμίσουν το Java codebase της εφαρμογής τους σε Kotlin, λόγω της ευκολίας στην χρήση και τα πλεονεκτήματά της.

Ενδιαφέρον έχει το γεγονός πως σύμφωνα με κάποια στατιστικά που έχει δημοσιοποιήσει η JetBrains, περίπου το 70% των προγραμματιστών που έχουν υιοθετήσει την Kotlin ως βασική γλώσσα προγραμματισμού για την ανάπτυξη εφαρμογών τους αναφέρουν πως η γλώσσα αυτή τους βοηθάει να είναι πιο παραγωγικοί στην εργασία τους συγκριτικά με κάποια άλλη γλώσσα. Τέλος, σύμφωνα με τα εσωτερικά δεδομένα της Google ως προς το ποσοστό των crash rates στις εφαρμογές, αναφέρει πως οι εφαρμογές οι οποίες έχουν αναπτυχθεί με την Kotlin ως βασική γλώσσα, είναι 20% λιγότερο πιθανό να έχουν crash κατά την εκτέλεση του κώδικα.

3.1.2 MVVM Clean Architecture (Architectural pattern)

Το MVVM Clean Architecture ένα μοντέλο αρχιτεκτονικής (architecture pattern) συνδυάζει το πρότυπο του MVVM με το Clean Architecture για την ανάπτυξη της εφαρμογής. Ο βασικός λόγος επιλογής του συγκεκριμένου architectural pattern για την ανάπτυξη της εφαρμογής είναι πως με την χρήση του διαχωρίζονται βασικές έννοιες κάνοντας χρήση της κατάτμησης (modularization) των κλάσεων προκειμένου ο κώδικας να είναι οργανωμένος και δομημένος. Επίσης, με τον διαχωρισμό του κώδικα σε επίπεδα, δίνεται η δυνατότητα να είναι ο κώδικας πιο εύκολα συντηρήσιμος και να μπορεί να ελεγχθεί κατά το testing πιο εύκολα μειώνοντας την πολυπλοκότητα.

Ο κώδικας χωρίζεται σε τρία βασικά επίπεδα (layers) τα οποία αλληλεπιδρούν στενά μεταξύ τους προκειμένου να επιτευχθεί η ροή της πληροφορίας από το ένα επίπεδο στο άλλο. Αυτά τα επίπεδα είναι το Presentation Layer, το Domain Layer και το Data Layer.

Presentation Layer: Στο Presentation Layer εντάσσονται τα Views (π.χ Activity, Fragments) και τα State Holder classes (ViewModels). Σκοπός του View είναι να ζητά πληροφορία από το ViewModel και να περιμένει να λάβει αυτά τα δεδομένα. Το View Layer δεν περιέχει application logic, παρα μόνο αλληλεπίδραση με τον χρήστη σε επίπεδο οθόνης. Ένα ViewModel κάνει expose τα τα κανάλια πληροφορίας (data streams) τα οποία ακούει (Observe, Collect - Ανάλογα την τεχνολογία υλοποίησης) το View.

Domain Layer: Το Domain Layer υπάρχει στην εφαρμογή ανάμεσα από το Representation και το Data Layer. Σκοπός του είναι να περιέχει την επαναλαμβανόμενη λογική που χρησιμοποιείται από πολλά State holder classes (ViewModels). Περιέχει τα interfaces των repositories που γίνονται implement σε επίπεδο data layer, αλλά και τις περιπτώσεις χρήσης (use cases) της εφαρμογής τα οποία χρησιμοποιούν τα ViewModels για να πάρουν δεδομένα.

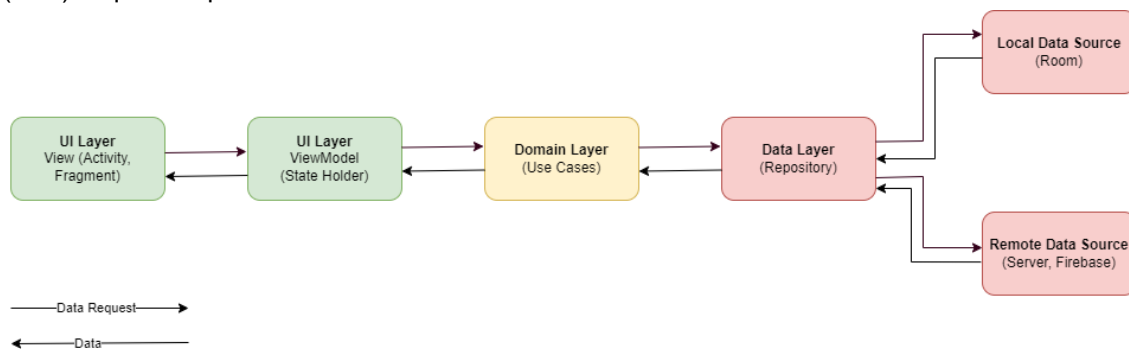
Data Layer: Στο Data Layer υπάρχουν όλες οι κλάσεις οι οποίες διαχειρίζονται το business logic της εφαρμογής, δηλαδή με ποιόν τρόπο η εφαρμογή παίρνει, διαχειρίζεται και αποθηκεύει τα δεδομένα της. Πιο συγκεκριμένα, στο data layer υπάρχει το implementation των repositories καθώς και τα data sources της εφαρμογής, δηλαδή αν θα χρησιμοποιεί κάποιο api προκειμένου «Ανάπτυξη εφαρμογής σε περιβάλλον Android με εξατομικευμένες προτάσεις σειρών και δυνατότητα interaction με χρήστες»

να λάβει δεδομένα, αν χρησιμοποιεί την Firebase κ.λ.π. Ουσιαστικά, το Data Layer έχει ως βασικό σκοπό να μεταφέρει την πληροφορία που λαμβάνει η εφαρμογή από το εξωτερικό της περιβάλλον προς το εσωτερικό διασφαλίζοντας την ακεραιότητα της.

Προκειμένου να κατανοηθεί σε βάθος το μοντέλο αρχιτεκτονικής, θα χρησιμοποιηθεί το παράδειγμα της οθόνης πληροφοριών μιας σειράς (series detail screen) και πως τα δεδομένα ταξιδεύουν από το εξωτερικό περιβάλλον προς το εσωτερικό και τελικά εμφανίζονται στην οθόνη του χρήστη.

Για την περίπτωση μας, θα είχαμε κάποιο View αρχικά. Εστω ότι υπάρχει ως View το SeriesDetailsFragment το οποίο χρησιμοποιεί το SeriesDetailsViewModel για να ζητήσει τις πληροφορίες της εκάστοτε σειράς με βάση το ID της. Η οθόνη με κάποια μέθοδο θα ζητούσε από το ViewModel αυτή την πληροφορία. Το ViewModel με τη σειρά του θα καλούσε ένα use case (π.χ FetchSeriesDetailsUseCase) εκχωρώντας του το ID της σειράς. Το use case με τη σειρά του θα καλούσε τον SeriesRepository προκειμένου να λάβει την πληροφορία. Τέλος, το repository θα καλούσε το api των σειρών της εφαρμογής και θα περίμενε να λάβει την πληροφορία.

Στο σημείο αυτό, η πληροφορία έρχεται από το api service και ξεκινά η αντίστροφη διαδικασία προκειμένου η πληροφορία να φτάσει μπροστά στην εφαρμογή και να απεικονιστεί στην οθόνη. Το repository χρησιμοποιεί έναν mapper προκειμένου να μετασχηματίσει την nullable πληροφορία σε non-null και να την προωθήσει προς το use case που την ζήτησε. Το use case προωθεί την πληροφορία στο ViewModel και από εκεί το viewModel την στέλνει στο View. Τέλος, το View αναλαμβάνει τον τρόπο να μεταφέρει την πληροφορία και να την δείξει (bind) στην οθόνη.



Εικόνα 8. Περιγραφή του MVVM Clean Architecture.

3.1.3 Κατάτμηση εφαρμογής σε modules (Multimodular approach)

Μια εφαρμογή λογισμικού είναι ένας ζωντανός οργανισμός ο οποίος συνεχώς εξελίσσεται. Εφόσον συνεχώς εξελίσσεται με διάφορους τρόπους, μεγαλώνει σε μέγεθος και επομένως είναι πολύ δύσκολο να οργανωθεί σωστά αν υπάρχει σε ένα module.

Η προσέγγιση της κατάτμησης των αρχείων και η οργάνωσή τους σε διαφορετικά modules δίνει την δυνατότητα στην ομάδα ανάπτυξης να έχουν ένα οργανωμένο project σπασμένο σε ξεκάθαρα packages τα οποία είναι αυτόνομα προκειμένου να εκτελέσουν μια

συγκεκριμένη ενέργεια τη φορά. Τα πλεονεκτήματα ενός multi-module application είναι τα παρακάτω:

Επεκτασιμότητα (Scalability) της εφαρμογής: Με το να υπάρχουν ξεκάθαρα modules τα οποία επιτελούν συγκεκριμένες ενέργειες και περιέχουν συγκεκριμένες κλάσεις, μειώνεται η εξάρτηση από άλλα σημεία οδηγεί στην προγραμματιστική αρχή της κατάτμησης των στοιχείων (separation of concerns).

Readability & Reusability: Τα αρχεία κώδικα είναι ξεκάθαρα οργανωμένα σε modules και packages τα οποία μειώνουν την συνολική πολυπλοκότητα του κώδικα. Είναι πιο εύκολα προσβάσιμα καθώς υπάρχει ξεκάθαρη αρχή οργάνωσης των αρχείων και όταν χρειαστεί να χρησιμοποιηθεί κάποιο συγκεκριμένο στοιχείο του module, αρκεί να προστεθεί η κατάλληλη εξάρτηση σε αυτό.

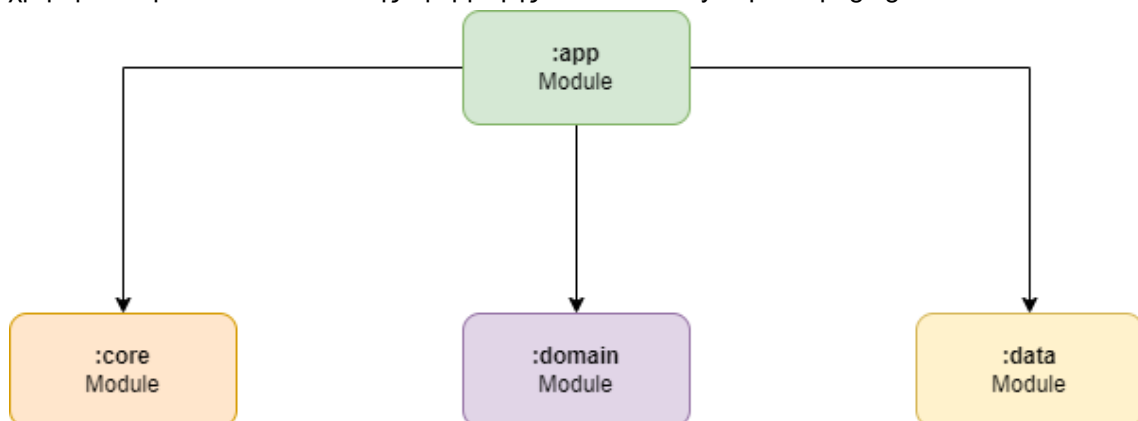
Testability: Ένα από τα πιο κρίσιμα στοιχεία μιας εφαρμογής είναι ο τρόπος με τον οποίο θα αξιολογηθεί ότι η εφαρμογή λειτουργεί σωστά και με τον τρόπο που προβλέπεται. Για αυτό τον σκοπό υπάρχουν διάφοροι τρόποι στο πλαίσιο της φάσης ελέγχου (testing phase) του κύκλου ανάπτυξης λογισμικού. Με το να υπάρχει ο κώδικας της εφαρμογής κατατμημένος σε modules οι προγραμματιστές είναι σε θέση να μπορούν να ελέγξουν συγκεκριμένα πράγματα του κώδικα τους απομονώνοντας τα από το υπόλοιπο codebase της εφαρμογής.

Visibility control: Οι προγραμματιστές της εφαρμογής επιλέγουν ποιο module θα αλληλεπιδρά με ποιο άλλο επιλέγοντας τους κανόνες του visibility μεταξύ τους. Χωρίς αυτούς τους κανόνες, ένα module δεν μπορεί να έχει πρόσβαση στον κώδικα ενός άλλου.

Μειωμένος χρόνος του Build time της εφαρμογής: Κατά το Gradle build της εφαρμογής, το να υπάρχουν διαφορετικά modules καθιστά την εφαρμογή να πραγματοποιήσει παράλληλα build σε αυτά και συνεπώς να αυξήσει το build performance.

Η εφαρμογή χρησιμοποιεί τρία (3) βοηθητικά modules. Το core, το data και το domain και χρησιμοποιούνται ως εξής:

App Module: Περιέχει τις οθόνες της εφαρμογής. Οι οθόνες είναι οργανωμένες σε packages ανάλογα με το activity στο οποίο ανήκουν. Το app βλέπει το core module προκειμένου να χρησιμοποιήσει τα απαραίτητα resources για την εφαρμογή, το domain module προκειμένου να χρησιμοποιήσει τα use cases της εφαρμογής και το data layer για το paging.



Εικόνα 9. Περιγραφή των εξαρτήσεων του app module.

Core Module: Περιέχει όλες τις βοηθητικές μεθόδους (extension functions), binding adapters, generic components (π.χ bottomsheets), τα λεκτικά της εφαρμογής (strings.xml), theming (themes.xml) και βοηθητικά utilities που μπορούν να χρησιμοποιηθούν. Το Core module βλέπει το domain module όπως περιγράφεται στο παρακάτω διάγραμμα.



Εικόνα 10. Περιγραφή των εξαρτήσεων του core module.

Domain Module: Στο domain module υπάρχουν τα models και τα use cases της εφαρμογής. Πρακτικά δηλαδή ότι ενέργεια μπορεί να γίνει μέσα στην εφαρμογή. Ακόμα, υπάρχουν τα interfaces των repositories της εφαρμογής.

Data Module: Στο data module υπάρχουν τα endpoints που χρησιμοποιεί η εφαρμογή (remote datasource), οι mappers που μετασχηματίζουν τα dto models σε domain models και οι implementation των repositories του domain module. Το data module έχει εξάρτηση από το domain module.



Εικόνα 11. Περιγραφή των εξαρτήσεων του data module.

3.1.4 Navigation Component

Το Navigation Component είναι μια τεχνολογία που υπάρχει στο Android Jetpack Library της Google. Πιο συγκεκριμένα, το Navigation Component έχει ως στόχο να απλοποιήσει τα navigation actions μέσα στην εφαρμογή. Είναι ένα πρόσθετο (plugin) το οποίο διαχειρίζεται την ουρά οθονών (backstack) της εφαρμογής και προσφέρει γρήγορο και εύκολο navigation στην εφαρμογή.

Στο παρελθόν, αν χρειαζόταν να πραγματοποιηθεί μια μετάβαση πατώντας ένα κουμπί από μια οθόνη σε μια άλλη υπήρχαν δύο τρόποι. Είτε να χρησιμοποιηθεί ένα intent προς κάποιο activity, είτε να γίνει χρήση κάποιου fragment transaction προκειμένου να ανοίξει ένα νέο fragment της εφαρμογής. Τα κύρια στοιχεία του Navigation Component είναι τα παρακάτω:

Navigation Graph: Ένα Navigation Graph είναι ένα αρχείο xml που περιγράφει ποιιά Fragments είναι μέρος ενός navigation flow στο android και τις αλληλεπιδράσεις όσον αφορά το navigation μεταξύ τους. Τοποθετείται στον φάκελο των resources κάτω από τον φάκελο navigation.

Destination Action: Ένα action είναι ένας τρόπος να ενωθούν δύο οθόνες. Περιγράφει αυστηρά πως προσεγγίζονται οι οθόνες τόσο σε front όσο και σε back navigation της εφαρμογής.

NavHostFragment: Το NavHostFragment είναι ένα container που ζεί στο Activity xml αρχείο του Activity που θα φιλοξενήσει το navigation. Λειτουργεί ως container για να εμφανίσει το ενεργό fragment στον χρήστη.

Navigation Editor: Ο Navigation Editor είναι ένας εύκολος τρόπος να δει ο προγραμματιστής το Navigation Graph του. Ο editor αυτός αποτελεί ένα διαδραστικό πίνακα στον οποίο εμφανίζονται οι πληροφορίες ενός Navigation Graph, οπτικοποιούνται τα Fragments και οι μεταβάσεις των οθονών. Χρησιμοποιώντας τον editor, ο χρήστης μπορεί να διαχειριστεί ένα Navigation Graph (π.χ προσθέτοντας ένα νέο Fragment, ή δημιουργώντας μεταβάσεις (Destination Actions) από μια οθόνη σε μια άλλη).

Safe Args: Τα Safe Args είναι ένας τρόπος να περαστεί πληροφορία σε ένα destination action ενός navigation flow. Πιο συγκεκριμένα, στην οθόνη που πρόκειται να γίνει το destination ορίζεται ο τύπος της πληροφορίας και το όνομα της και το σύστημα αναλαμβάνει να δημιουργήσει ένα αρχείο με αυτό ώστε να γίνει ανάκτηση της (retrieve) στην επόμενη οθόνη. Ο προγραμματιστής απλώς θα πρέπει να περάσει αυτή την πληροφορία μέσα στο destination action καθώς πλέον η επόμενη οθόνη εξαρτάται από αυτή την πληροφορία.

Navigation Controller: Ο Navigation Controller δίνει ζωή στο navigation component. Πιο συγκεκριμένα, διαβάζει το navigation graph, διαχειρίζεται τις μεταβάσεις των οθονών και το backstack των οθονών, ενώ ταυτόχρονα κρατά και χρήσιμη πληροφορία για την εφαρμογή.

Deeplinks: Δίνεται η δυνατότητα να μπορεί να γίνει μετάβαση δυναμικά σε οποιοδήποτε destination της εφαρμογής χωρίς να ακολουθηθεί το προδιαγεγραμμένο μονοπάτι για αυτή. Αρκεί να δηλωθεί το id του αντίστοιχου fragment κατά το navigation.

Το Navigation Component αλλάζει την διαδικασία μετάβασης συγκριτικά με τον παραδοσιακό τρόπο από ένα fragment σε ένα άλλο με τον εξής τρόπο. Αρχικά θα πρέπει να υπάρχει ένα Activity στο οποίο θα ζούν τα fragments της εφαρμογής και θα αποτελούν ένα flow μέσα στην εφαρμογή. Αυτό το Activity στο xml αρχείο του layout της, θα έχει ένα NavHostFragment. Θα πρέπει να δημιουργηθεί ένα νέο navigation graph xml που θα περιέχει τα fragments που θα υπάρχουν σε αυτό. Μέσω του Navigation Editor θα οριστούν οι μεταβάσεις των οθονών (destinations) με βελάκια. Προαιρετικά, μπορεί να γίνει χρήση των SafeArgs για τις μεταβάσεις προκειμένου να σταλούν δεδομένα από την μια οθόνη σε μια άλλη.

3.1.5 Kotlin Coroutines

Τα coroutines στην Kotlin είναι ασύγχρονα operations που χρησιμοποιούνται για λόγους asynchronous programming στην ανάπτυξη android εφαρμογών και όχι μόνο. Ο τυπικός ορισμός ενός coroutine είναι μια λειτουργία η οποία μπορεί να ξεκινήσει, να γίνει παύση και να συνεχιστεί. Στο android, είναι πολύ σημαντικό να μην μπλοκάρεται το main thread σε πολλές περιπτώσεις.

Προκειμένου να κρατήσουμε τις λειτουργίες που είναι χρονοβόρες (π.χ να πάρουμε δεδομένα απο κάποιο endpoint, να διαβάσουμε ή να γράψουμε σε κάποια βάση δεδομένων), υπάρχει η δυνατότητα να χρησιμοποιηθούν οι παρακάτω dispatchers στα coroutines:

Dispatchers.Main: Το thread αυτό χρησιμοποιείται για να αλληλεπιδράσουμε με το UI και να πραγματοποιήσουμε γρήγορες εντολές

Dispatchers.IO: Προορίζεται για λειτουργίες δίσκου ή δικτύου (π.χ fetch data ή read files) οι οποίες δεν πρέπει να μπλοκάρουν το main thread της εφαρμογής μέχρι την ολοκλήρωσή τους.

Dispatchers.Default: Ο dispatcher αυτός προορίζεται για λειτουργίες που απαιτούν μεγάλη χρήση της CPU (για παράδειγμα sort μιας λίστας, parse JSON file)

3.1.6 Kotlin Flows

Τα Kotlin flows αποτελούν μια εξέλιξη των coroutines. Τα flows είναι ένα type με το οποίο μπορεί να παράξει και να στείλει (emit) πολλές τιμές αντί για μια μόνο τιμή. Ουσιαστικά πρόκειται για ένα κανάλι(stream) πληροφορίας το οποίο ξεκινά από ένα σημείο (data layer) και καταλήγει κάπου αλλού (presentation layer) μέσα στην εφαρμογή.

Τα flows βασίζονται πάνω στην λογική του παραγωγού (Producer) - μεσολαβητή (Intermediary) - καταναλωτή (Consumer). Στο παράδειγμα της εφαρμογής, αν πάρουμε ως παράδειγμα μια οθόνη, ένας producer θα ήταν ένα repository που ως σκοπό έχει να στείλει πληροφορία (data), ο μεσολαβητής θα ήταν κάποια ενδιάμεση ίσως κλάση που θα μετασχηματίζει (map) αυτή την πληροφορία σε κάτι άλλο χρήσιμο για την εφαρμογή και τέλος ο καταναλωτής θα ήταν το αντίστοιχο View στο οποίο θα αποτυπωνόταν η πληροφορία.

3.1.7 Data Binding

Το Data Binding αποτελεί μέρος του Android Jetpack Library και σύμφωνα με το documentation του Android developer, είναι μια διαδικασία κατά την οποία το UI (ή View) συνδέεται με τα δεδομένα (data) τα οποία χρειάζεται να προβάλει κατευθείαν μέσα στο UI αντί να οριστούν με προγραμματιστικό τρόπο στην αντίστοιχη κλάση του view. Η χρήση του data binding σε ένα android project οδηγεί σε γρηγορότερο development time, πιο ευπαρουσίαστο και ευκολότερα διαχειρισίμο κώδικα.

Αν αναλογιστούμε ένα τυπικό Activity (xml & class file), αν θέλαμε να έχουμε μια οθόνη που περιέχει ένα TextView με έναν μετρητή (counter) και ένα Button και κάθε φορά που πατάω το button να αυξάνει τον counter κατα μία μονάδα, με Data Binding και MVVM pattern θα γινόταν ως εξής:

Στο XML του Activity, θα προσθέταμε στο πάνω μέρος το παρακάτω code snippet το οποίο ορίζει ότι στο View μας περιμένουμε μια μεταβλητή τύπου integer.

```
<data>
    <variable
        name="counter"
```

```
type="Integer" />
</data>
```

Έπειτα, θα μπορούσαμε να κάνουμε set κατευθείαν το value της τιμής του counter στο TextView του xml ως εξής:

```
android:text="@{String.valueOf(counter)}"
```

Στον κώδικα (.kt file) θα έπρεπε να γραφτεί λογική έτσι ώστε κάθε φορά που πατάει ο χρήστης το κουμπί, να ζητείται από το viewModel να αυξήσει την τιμή του counter κατά μία μονάδα και να την επιστρέψει με LiveData στο Activity το οποίο στη συνέχεια θα γίνει observe απο το view. Τέλος, η τιμή θα οριζόταν στον counter ως εξής: binding.counter = myCounterValue

3.1.8 Preferences DataStore

Το DataStore είναι μια νέα εναλλακτική λύση στην αποθήκευση πληροφορίας τοπικά στη συσκευή και ως σκοπό έχει να αντικαταστήσει τα Android Shared Preferences. Η πληροφορία που σώζεται στο DataStore έχει την μορφή key, value pairs όπως ακριβώς γινόταν στα Shared Preferences. Το DataStore χρησιμοποιεί Kotlin Flows για να μεταβιβάσει την πληροφορία από το τοπικό αποθηκευτικό χώρο προς την εφαρμογή.

3.1.9 Hilt (Dependency Injection)

Η Hilt είναι ένα library με το οποίο μπορούμε να υλοποιήσουμε την προγραμματιστική τεχνική του dependency injection στον κώδικα μας. Το dependency injection είναι μια τεχνική που βοηθά στην οργάνωση του κώδικα και την αποφυγή προβλημάτων όπως η υπερβολική σύνδεση των κλάσεων μεταξύ τους και η δυσκολία στην επαναχρησιμοποίηση κώδικα. Το Hilt διευκολύνει την υλοποίηση του dependency injection στο Android, καθιστώντας την πιο εύκολη και ασφαλή για τους προγραμματιστές.

Το Hilt χρησιμοποιεί το annotation processing για τη δημιουργία ενός γραφήματος εξαρτήσεων (dependency graph) και τη διαχείριση των αντικειμένων που το σχηματίζουν. Οι κλάσεις που χρειάζονται αντικείμενα εξαρτημένα από άλλες κλάσεις δηλώνονται ως @Inject, ενώ η δημιουργία των αντικειμένων που σχηματίζουν το γράφημα γίνεται από το Hilt.

Υπάρχουν πολλά διαφορετικά annotations προκειμένου να ορίσουμε στο dependency graph τι ακριβώς χρειάζεται να γίνει. Έτσι λοιπόν έχουμε το @Provides για να δηλώσουμε ότι κάτι γίνεται provide, το @Singleton αν θέλουμε να ορίσουμε κάποια κλάση ως singleton κ.α.

3.1.10 Firebase Authentication

Το Firebase Authentication είναι μια ολοκληρωμένη λύση ενός συστήματος αυθεντικοποίησης χρηστών (authentication system) η οποία προσφέρεται από το Google Firebase Platform. Χρησιμοποιώντας αυτή τη λύση σαν cloud auth system, δεν υπάρχει ανάγκη ανάπτυξης ενός authentication system από τους προγραμματιστές της εκάστοτε εφαρμογής, καθώς το Firebase Auth διαχειρίζεται αυτή την διαδικασία αυτόματα με τους δικούς της μηχανισμούς. Οι τρόποι που προσφέρονται από την πλατφόρμα είναι αρκετές και καλύπτουν σχεδόν κάθε ανάγκη που μπορεί να προκύψει. Πιο συγκεκριμένα, το Firebase Auth μπορεί να χρησιμοποιηθεί με τους παρακάτω τρόπους για σύνδεση στην εφαρμογή:

-
- Email/password login
 - Phone login
 - Google login
 - Facebook login
 - Anonymous login
-

Ακόμα, δίνεται η δυνατότητα ενεργοποίησης ενός TFA (Two Factor Auth) μηχανισμού κατευθείαν μέσω της Firebase Auth.

3.1.11 Firebase Cloud Firestore (No SQL Database)

Το Firebase Firestore είναι μια υπηρεσία βάσης δεδομένων που προσφέρεται από την Google Firebase. Η Firestore είναι μια NOSQL βάση δεδομένων που έχει σχεδιαστεί για να προσφέρει γρήγορα αποτελέσματα. Μπορεί να χρησιμοποιηθεί και ως βάση και ως backend server καθώς τα δεδομένα που βρίσκονται μέσα της αποθηκεύονται με την ακόλουθη λογική:

Collections > documents > document

Ένα collection θα μπορούσε να είναι οι χρήστες που έχουν εγγραφεί στην εφαρμογή (users). Για κάθε ξεχωριστό χρήστη θα υπήρχε ένα document με βάση το document id του, συνεπώς τα documents είναι μια λίστα από ids, ενώ ένα document αποτελεί τις πληροφορίες του εκάστοτε χρήστη.

Το Firebase Firestore παρέχει συνέπεια στην ανάγνωση και εγγραφή δεδομένων σε πραγματικό χρόνο, επιτρέποντας την ενημέρωση των πελατών με νέα δεδομένα αυτόματα. Επίσης, προσφέρει προχωρημένες δυνατότητες όπως ταυτόχρονη ανάγνωση-εγγραφή, σύνθετες ερωτήσεις, φιλτράρισμα και ταξινόμηση δεδομένων, καθώς και τη δυνατότητα να συνδέεται με άλλες υπηρεσίες Firebase όπως το Firebase Authentication και το Firebase Cloud Functions.

3.1.12 Firebase Storage

Το Firebase Storage είναι μια υπηρεσία που προσφέρεται από την Google Firebase Platform. Όπως προδιαθέτει και το όνομα της online υπηρεσίας, πρόκειται για μια υπηρεσία αποθήκευσης αρχείων. Στο Storage δίνεται η δυνατότητα αποθήκευσης και ανάκτησης

περιεχομένου των χρηστών μέσα από την εφαρμογή και όχι μόνο, όπως φωτογραφίες, βίντεο κ.α.

Προκειμένου να χρησιμοποιηθεί το Firebase Storage σε μια android εφαρμογή, αρκεί να χρησιμοποιηθεί το Storage API της Firebase. Επίσης μπορεί να γίνει restrict η χρήση του σε συγκεκριμένους χρήστες (εκείνους π.χ που έχουν πραγματοποιήσει login) έτσι ώστε το content της εφαρμογής μας να παραμείνει ασφαλές από μη εξουσιοδοτημένους χρήστες.

Η παρούσα εφαρμογή χρησιμοποιεί το firebase storage, προκειμένου να μπορεί ο χρήστης να αποθηκεύσει φωτογραφίες του προφίλ του.

3.1.13 Retrofit (HTTP Client)

Το Retrofit είναι ένας HTTP client και προορίζεται για java και android development. Απλοποιεί τον τρόπο με τον οποίο μπορεί μια εφαρμογή να επικοινωνήσει με το εξωτερικό της περιβάλλον. Το Retrofit ουσιαστικά σε μια εφαρμογή διαχειρίζεται την διαδικασία της αποστολής ενός HTTP request και της λήψης του response.

Ένα τυπικό παράδειγμα του retrofit, στην εφαρμογή χρησιμοποιείται το interface SeriesApi στο οποίο βρίσκονται όλα τα endpoints της εφαρμογής με τα GET, PUT η POST annotations τους ως μέθοδοι καθώς και τον τύπο που επιστρέφουν. Το μόνο που αρκεί να γίνει είναι να δημιουργηθεί ο http client και στη συνέχεια να χρησιμοποιηθεί απο τα repositories προκειμένου να επικοινωνήσουν με το API των σειρών.

3.1.14 Gson

Το Gson είναι μια βιβλιοθήκη η οποία μπορεί να μετασχηματίσει μια java κλάση σε ένα JSON αρχείο (key-value pairs). Αντιστοίχως, μπορεί να κάνει και την αντίστροφη διαδικασία, δηλαδή απο ένα JSON file, να παράξει ένα instance μιας java κλάσης.

3.1.15 Paging 3 (Android Jetpack)

Μια εφαρμογή κινητών ζητά δεδομένα από data sources. Αυτά τα data sources, αντί να στέλνουν ολόκληρη την πληροφορία ως απάντηση, πραγματοποιούν κατάτμηση της σε σελίδες (pages). Αν μιλάμε για μεγάλο όγκο πληροφορίας, αυτό θα είχε επίπτωση στο network bandwidth και στα system resources της κάθε συσκευής. Με το paging πετυχαίνουμε να λαμβάνουμε την πληροφορία σε πακέτα και μόνο όταν είναι αναγκαίο να ζητά η εφαρμογή την επόμενη προκειμένου να την απεικονίσει.

Έτσι, αν για παράδειγμα μέσα στην εφαρμογή πατήσει ο χρήστης να δει τις ταινίες με κατηγορία "Action", η εφαρμογή θα ζητήσει απο το API την πρώτη σελίδα και θα την εμφανίσει μπροστα. Αν ο χρήστης κάνει scroll προς τα κάτω, λίγο πριν τελειώσει η λίστα, η εφαρμογή θα αναλάβει να ζητήσει την επόμενη σελίδα κ.ο.κ. Το paging 3 είναι η πιο πρόσφατη έκδοση της paging βιβλιοθήκης και είναι μέρος του Jetpack library.

3.1.16 Glide

Το Glide είναι ένας γρήγορος τρόπος να φορτώσουμε φωτογραφίες από κάποιο url. Το Glide υποστηρίζει το fetch και το decode, ενώ μπορεί να φορτώσει video, photos και GIFs.

Για να φορτώσουμε μια φωτογραφία ένα παράδειγμα είναι το παρακάτω:
`Glide.with(context).load(url).into(imageView)`

Φυσικά, προσφέρει ακόμα περισσότερες λειτουργίες προκειμένου να τροποποιηθούν οι εικόνες της εφαρμογής.

Όπου context το context από κάποια οθόνη (Activity/Fragment), url το url της εικόνας και imageView το reference από κάποιο ImageView UI component. Φυσικά, υπάρχουν πολλές chain method calls που επιτρέπουν να παραμετροποιηθεί περαιτέρω η εικόνα την οποία επιθυμούμε να φορτώσουμε στο View.

Εναλλακτικές του Glide είναι το Coil και το Picasso που επιτελούν την ίδια δουλειά.

3.1.17 Github (Version Control System – VCS)

Προκειμένου να διασφαλιστεί ότι ο κώδικας θα βρίσκεται σε ένα ενιαίο και ασφαλές σημείο χρησιμοποιήθηκε το Github ως VCS για την ανάπτυξη του project. Το Github είναι μια δημοφιλής πλατφόρμα αποθετηρίων κώδικα (code repositories) το οποίο χρησιμοποιείται ευρέως από προγραμματιστές προκειμένου να διαχειρίζονται τον πηγαίο κώδικα των εφαρμογών τους. Οι προγραμματιστές μπορούν να συνεργάζονται πάνω σε ένα κοινό κώδικα δημιουργώντας branches για να υλοποιήσουν το feature που έχει ανατεθεί και στη συνέχεια ανοίγουν pull request για να ελεγχθεί ο κώδικας από τα υπόλοιπα μέλη της ομάδας και τελικά να προστεθεί (merge) στο υπόλοιπο κομμάτι της εφαρμογής.

Για να αποφευχθούν τα περιττά commits του κώδικα στο βασικό branch της ανάπτυξης, χρησιμοποιήθηκαν sub branches τα οποία μετά το πέρας της ανάπτυξης του εκάστοτε feature γινόταν merge στο master branch.

Ο πηγαίος κώδικας (source code) της εφαρμογής υπάρχει στο ακόλουθο url:
<https://github.com/skaradimitriou/unipi-seriesmania-app>

3.2 ΣΥΜΠΛΗΡΩΜΑΤΙΚΗ ΕΦΑΡΜΟΓΗ ΠΑΡΑΓΩΓΗΣ ΣΤΑΤΙΣΤΙΚΩΝ ΣΤΟΙΧΕΙΩΝ

3.2.1 Στατιστικά στοιχεία που παράγει η εφαρμογή

Πιο συγκεκριμένα, η συμπληρωματική εφαρμογή παράγει τα παρακάτω στατιστικά στοιχεία:

Γενικά στοιχεία: Πόσους χρήστες έχει η εφαρμογή, πόσες συζητήσεις υπάρχουν στο forum της εφαρμογής, πόσες αξιολογήσεις έχουν γίνει μέσα από την εφαρμογή, πόσες κριτικές έχουν γίνει μέσα από την εφαρμογή

Κορυφαίες σειρές που έχουν αξιολογηθεί μέσα στην εφαρμογή: XXX

Στατιστικά Χρηστών: Ποιός είναι ο χρήστης που έχει δώσει τις περισσότερες αξιολογήσεις μέσα στην εφαρμογή, ο χρήστης με την μεγαλύτερη λίστα προβολής (Watchlist), ο χρήστης με τους περισσότερους ακόλουθους (followers), ο χρήστης που απαντάει πιο συχνά στο forum

Στατιστικά Συζητήσεων: Ποιό είναι το πιο δημοφιλές forum thread μέσα στην εφαρμογή. Υπολογίζεται βάσει των απαντήσεων που διαθέτει η συζήτηση.

Στατιστικά Προτιμήσεων Χρηστών: Οι χρήστες μέσα στην εφαρμογή μπορούν απο το προφίλ τους να ορίσουν μέχρι και τρεις προτιμήσεις σχετικά με το είδος των σειρών που τους αρέσουν. Με βάσει αυτό το γνώρισμα στους χρήστες, προκύπτουν τα κορυφαία τρία είδη προτιμήσεων συνολικά για την εφαρμογή.

ΚΕΦΑΛΑΙΟ 4

4.1 ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

4.1.1 Όνομα εφαρμογής, Λογότυπο & Styling

Το όνομα που δόθηκε στην εφαρμογή είναι "Series Mania" ενώ το λογότυπο που χρησιμοποιήθηκε στην εφαρμογή παρουσιάζεται παρακάτω.



Εικόνα 12. Λογότυπο της εφαρμογής (Series Mania)

Βασικός στόχος της ανάπτυξης της εφαρμογής ήταν η σχεδίαση της εφαρμογής προκειμένου να φαίνεται ελκυστική στον χρήστη. Σε όλη την εφαρμογή, χρησιμοποιήθηκε χρωματική παλέτα η οποία δεν κουράζει τον χρήστη ενώ η τυπογραφία στα διαφορετικά μεγέθη βοηθά τον χρήστη να ακολουθήσει τις διάφορες κατηγορίες αναβαθμίζοντας την συνολική εμπειρία πλοήγησης του.

Η χρωματική παλέτα της εφαρμογής αποτελείται από χρώματα στις διαβαθμίσεις του μπλε, γκρί, πορτοκαλί και κίτρινου. Απο την εφαρμογή δεν θα μπορούσε να λείπει το άσπρο και το μαύρο χρώμα.

4.2 ΑΝΑΛΥΣΗ ΟΘΟΝΩΝ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Η εφαρμογή αποτελείται συνολικά από έξι (6) Activities και είκοσιένα (21) Fragments. Πιο συγκεκριμένα, η εφαρμογή αποτελείται από έξι (6) μονοπάτια (flows). Το κάθε flow ζει σε ένα δικό του Activity. Τα flows της εφαρμογής παρουσιάζονται παρακάτω:

- **Onboarding:** Το onboarding flow αφορά την είσοδο του χρήστη στην εφαρμογή. Περιλαμβάνει τις οθόνες εισόδου (splash), εγγραφής (register) και σύνδεσης (login).
 - **Dashboard:** Το dashboard flow αφορά τις βασικές τρεις οθόνες τις εφαρμογές από τις οποίες μπορεί να εξερευνηθεί όλη η υπόλοιπη εφαρμογή. Περιλαμβάνει την αρχική οθόνη (home), την οθόνη εξερεύνησης (discover) και την οθόνη των συζητήσεων (forum).
 - **Settings:** Το settings flow περιλαμβάνει συμπληρωματικά πράγματα για την εφαρμογή. Περιλαμβάνει την οθόνη των ερωτοαπαντήσεων (FAQs), τα στατιστικά στοιχεία της εφαρμογής (analytics) και την οθόνη σχετικά με την εφαρμογή (about).
 - **Forum:** Το forum flow περιλαμβάνει τις οθόνες εισαγωγής νέας συζήτησης (add new thread) και οθόνη προβολής στοιχείων συζήτησης (thread details).
 - **Profile:** Το flow του προφίλ είναι από τα πιο σημαντικά flows στην εφαρμογή. Περιλαμβάνει τις οθόνες του προφίλ του χρήστη (profile), εισαγωγής φωτογραφίας (upload image screen), επεξεργασία προφίλ (update profile), προτιμήσεων (preferences), λίστας σειρών (watchlist).
 - **Results:** Πρόκειται για το κομμάτι που δίνει πληροφορίες στον χρήστη. Περιλαμβάνει τις οθόνες προβολής πληροφοριών για σειρές (detail info) και ηθοποιών (cast details). Ακόμα, περιλαμβάνει την οθόνη προβολής αποτελεσμάτων κατα τις αναζητήσεις (results).
-

Οι δυνατότητες του χρήστη εντός της εφαρμογής παρουσιάζονται παρακάτω ανάλογα με το flow στο οποίο ανήκουν.

4.2.1 Είσοδος στην εφαρμογή - Onboarding

Splash screen

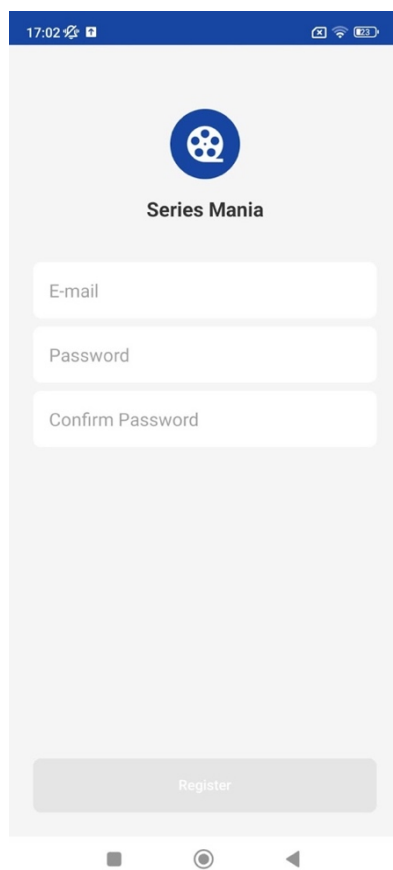
Μια splash screen σε μια εφαρμογή κινητών αποτελεί μια εισαγωγική οθόνη κατά την έναρξη της εφαρμογής (launch) από τον χρήστη. Σε αυτή την οθόνη, ο χρήστης μπορεί να

διακρίνει το λογότυπο της εφαρμογής καθώς και το όνομα της. Αυτή η οθόνη, συχνά περιέχει επίσης και κάποιο δείκτη προόδου (progress bar) που σηματοδοτεί ότι η εφαρμογή ξεκινά.

Στο Series Mania, η splash screen αποτελείται από το λογότυπο και το όνομα της εφαρμογής. Η διάρκεια κατά την οποία ο χρήστης θα μείνει σε αυτή την οθόνη, εξαρτάται από το αν ο χρήστης είναι ήδη συνδεδεμένος ή όχι. Πιο συγκεκριμένα, η οθόνη αυτή πραγματοποιεί αυτόν τον έλεγχο και στην συνέχεια αν ο χρήστης είναι ήδη συνδεδεμένος ξεκινά την κεντρική οθόνη της εφαρμογής. Σε διαφορετική περίπτωση, τον ανακατευθύνει στην οθόνη σύνδεσης λογαριασμού προκειμένου να συνδεθεί.

Register

Η οθόνη σύνδεσης λογαριασμού δίνει την δυνατότητα στον χρήστη να πραγματοποιήσει μια νέα εγγραφή ενός λογαριασμού στο Series Mania. Προκειμένου ένας χρήστης να δημιουργήσει έναν λογαριασμό, του ζητείται να συμπληρώσει τα πεδία e-mail, κωδικός πρόσβασης και επιβεβαίωση κωδικού πρόσβασης. Η επιλογή για εγγραφή ("Register") ενεργοποιείται μόλις τα πεδία αυτά είναι συμπληρωμένα και δίνουν την δυνατότητα να ξεκινήσει η διαδικασία της εγγραφής του χρήστη στην εφαρμογή.

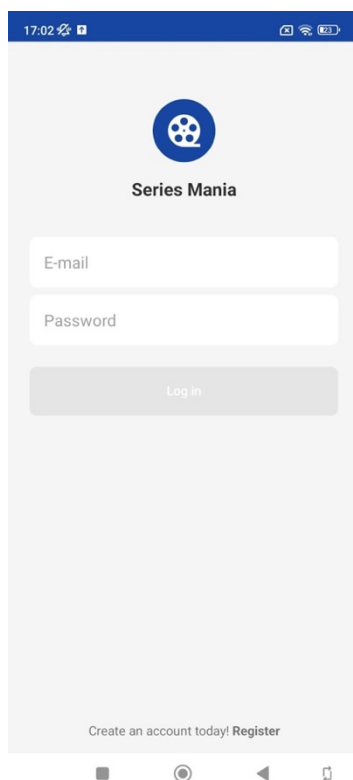


Εικόνα 13. Η οθόνη εγγραφής χρήστη στην εφαρμογή

Login

Η οθόνη σύνδεσης λογαριασμού δίνει την δυνατότητα στον χρήστη να συνδεθεί στον λογαριασμό του. Για μια επιτυχή σύνδεση στην εφαρμογή ζητείται το e-mail και ο κωδικός πρόσβασης του λογαριασμού του χρήστη. Η επιλογή για σύνδεση (“Log in”) ενεργοποιείται μόλις τα δύο πεδία αυτά δεν είναι άδεια και δίνουν την δυνατότητα να ξεκινήσει η διαδικασία της ταυτοποίησης του χρήστη, η οποία ανάλογα με το αποτέλεσμα της, είτε θα κατευθύνει τον χρήστη στην κεντρική οθόνη της εφαρμογής (home) είτε θα εμφανίσει μήνυμα σφάλματος σε μορφή Snackbar στο κάτω μέρος της οθόνης.

Σε περίπτωση που ο χρήστης δεν διαθέτει κάποιο λογαριασμό στην εφαρμογή, στο κάτω μέρος της οθόνης στο κουμπί εγγραφής (“Register”) δίνεται η δυνατότητα να ανοίξει ο χρήστης την οθόνη εγγραφής και να δημιουργήσει τον λογαριασμό του.



Εικόνα 14. Η οθόνη σύνδεσης στην εφαρμογή.

4.2.2 Dashboard

Κεντρική οθόνη εφαρμογής (Home)

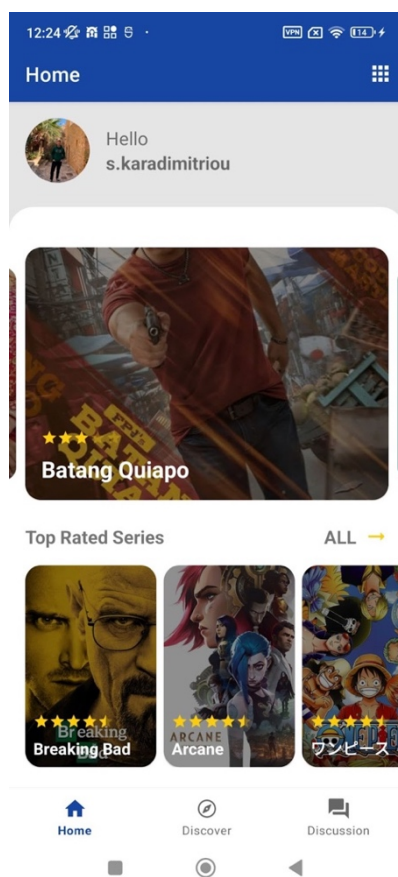
Η κεντρική οθόνη της εφαρμογής αποτελεί την βασική οθόνη της εφαρμογής καθώς είναι η πρώτη οθόνη με την οποία έρχεται σε επαφή ο χρήστης μετά από μια επιτυχημένη σύνδεση λογαριασμού.

Στην οθόνη αυτή, στο πάνω μέρος της οθόνης, ο χρήστης μπορεί να δει το username του καθώς και την φωτογραφία προφίλ που έχει. Μπορεί επίσης να πλοηθεί κάθετα και να λάβει πληροφορίες σχετικά με διάφορες κατηγορίες σειρών που είναι διαθέσιμες. Πιο συγκεκριμένα, εμφανίζονται με την σειρά είκοσι (20) αποτελέσματα από τις εξής κατηγορίες σειρών:

- Δημοφιλείς σειρές (Popular series)
- Σειρές με τις περισσότερες θετικές κριτικές (Top rated series)
- Τάσεις σειρών (Trending series)

- Προβάλλονται Σήμερα (Airing today series)
- Σειρές που ο χρήστης έχει στη λίστα προβολής του (Watchlist) εφόσον υπάρχουν
- Σειρές των τριών κατηγοριών προτιμήσεων που μπορεί να ορίσει ο χρήστης από το προφίλ του (User Series Preferences). Αυτό θα μπορούσε να είναι οποιαδήποτε από τις 16 κατηγορίες που υποστηρίζει η εφαρμογή. Για παράδειγμα, ένας χρήστης με προτιμήσεις στις κατηγορίες “Action”, “Crime”, “Documentary” θα έβλεπε στην αρχική οθόνη τρεις κατηγορίες ακόμα με 20 αποτελέσματα σειρών στην κάθε μία.

Πατώντας στην φωτογραφία του προφίλ του, ανοίγει η οθόνη του προφίλ του συνδεδεμένου χρήστη ενώ πατώντας πάνω σε μία σειρά ανοίγει η οθόνη των πληροφοριών της επιλεγμένης σειράς.

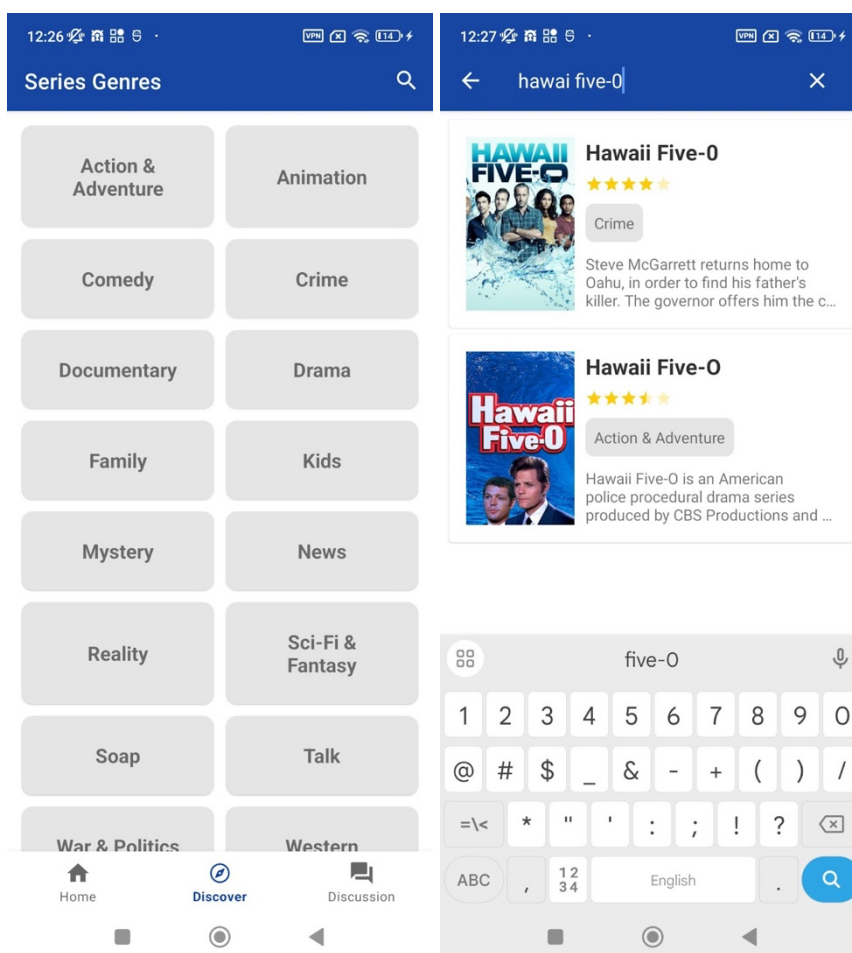


Εικόνα 15. Η αρχική οθόνη της εφαρμογής

Αναζήτηση (Discover)

Στην οθόνη της αναζήτησης (“Discover”) ο χρήστης μπορεί να αναζητήσει σειρές στην εφαρμογή. Αυτό μπορεί να γίνει με δύο τρόπους. Ο πρώτος τρόπος είναι να αναζητήσει σειρές βάσει της κατηγορίας στην οποία ανήκουν. Για παράδειγμα, πατώντας πάνω στην κατηγορία “Crime” για να έρθουν τα αποτελέσματα για την κατηγορία και από εκεί να αναζητήσει σειρές που τον ενδιαφέρουν και να αλληλεπιδράσει με αυτές.

Ο δεύτερος τρόπος είναι να χρησιμοποιήσει την μπάρα αναζήτησης πατώντας το εικονίδιο του μεγεθυντικού φακού στο πάνω μέρος της οθόνης. Με το πάτημα στο εικονίδιο, ανοίγει η μπάρα αναζήτησης και ο χρήστης όσο πληκτρολογεί, εμφανίζονται αποτελέσματα (εφόσον υπάρχουν). Αν ο χρήστης σβήσει το κείμενο από την μπάρα, επανέρχονται οι κατηγορίες των σειρών μπροστά.



Εικόνα 16. Η οθόνη αναζήτησης της εφαρμογής

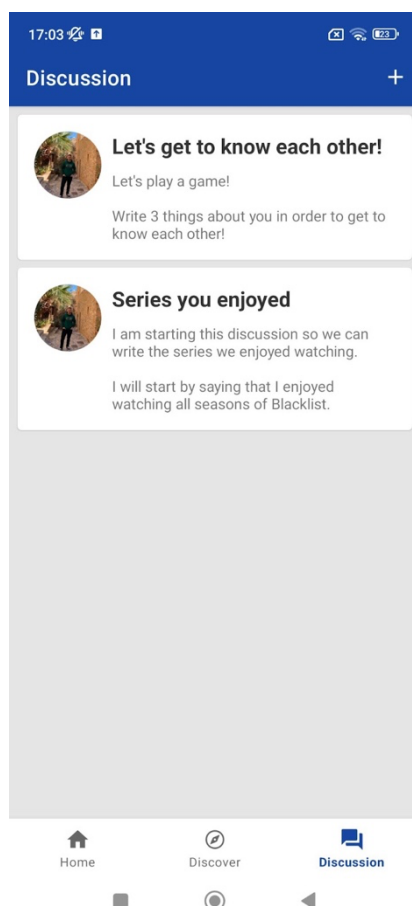
Συζητήσεις (Discussions/Forum)

«Ανάπτυξη εφαρμογής σε περιβάλλον Android με εξατομικευμένες προτάσεις σειρών και δυνατότητα interaction με 40 χρήστες»

Η δυνατότητα αλληλεπίδρασης μεταξύ των χρηστών είναι ένα σημαντικό κομμάτι που έχει ληφθεί υπόψη στην ανάπτυξη της εφαρμογής. Οι χρήστες μπορούν να αλληλεπιδρούν μεταξύ τους και να δημιουργούν μικρά communities εντός της εφαρμογής στην ενότητα συζητήσεις ("Discussions") της εφαρμογής.

Όταν ο χρήστης πλοηγηθεί στην οθόνη των συζητήσεων πατώντας στο τρίτο εικονίδιο από την μπάρα στο κάτω μέρος της εφαρμογής, ανοίγει η οθόνη στην οποία φαίνονται οι ενεργές συζητήσεις της εφαρμογής. Εκεί, με μια γρήγορη ματιά, ο χρήστης μπορεί να δει ποιος χρήστης έχει ανοίξει ποιά συζήτηση, να δημιουργήσει μια νέα πατώντας στο εικονίδιο στο πάνω μέρος της οθόνης, ή να ανοίξει μια συζήτηση προκειμένου να διαβάσει ή να συμμετάσχει σε αυτή και να εκφέρει τη γνώμη του.

Πατώντας στο εικονίδιο του κάθε χρήστη, ανοίγει το προφίλ του εκάστοτε χρήστη προκειμένου να δει ο χρήστης τις πληροφορίες του προφίλ του ατόμου του οποίου άνοιξε.



Εικόνα 17. Η οθόνη συζητήσεων στην εφαρμογή

4.2.3 Οθόνες πληροφοριών

Οθόνη Πληροφοριών Σειράς (Series Details)

Η κεντρική Στην οθόνη των πληροφοριών μιας σειράς, ο χρήστης μπορεί να πλοηγηθεί αφού έχει πατήσει πάνω σε μια σειρά από οποιοδήποτε σημείο της εφαρμογής. Στην οθόνη αυτή, ο χρήστης μπορεί να πλοηγηθεί κάθετα και να διαβάσει χρήσιμες πληροφορίες για την σειρά την οποία επέλεξε όπως τα παρακάτω:

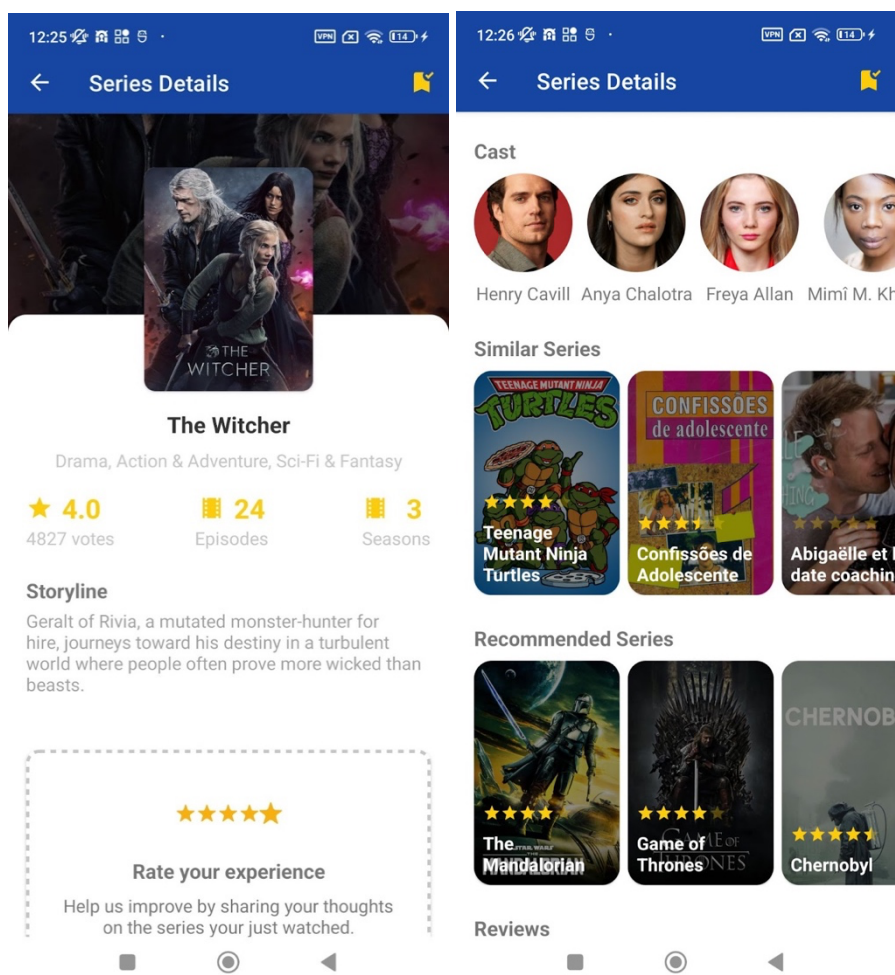
- Βασικές πληροφορίες της εκάστοτε σειράς όπως: Εικόνα, τίτλος, είδος (ή είδη), βαθμολογία (πλήθος και τιμή), πλήθος επεισοδίων, πλήθος σεζόν και περιγραφή.
 - Ηθοποιοί που πήραν μέρος στην εκάστοτε σειρά (φωτογραφία και ονοματεπώνυμο ηθοποιού).
 - Παρόμοιες σειρές (Similar series section) στο οποίο εμφανίζονται είκοσι αποτελέσματα με παρόμοιες σειρές
 - Προτάσεις σειρών βάσει της επιλεγμένης σειράς (Recommended series)
 - Κριτικές σειρές (εφόσον υπάρχουν).
-

Απο την οθόνη αυτή, δίνεται η δυνατότητα στον χρήστη να πραγματοποιήσει τις ακόλουθες ενέργειες:

Μπορεί να προσθέσει την σειρά στην λίστα προβολής του (Watchlist) πατώντας το εικονίδιο του σελιδοδείκτη στο πάνω μέρος της οθόνης. Στην περίπτωση που η σειρά προστεθεί επιτυχώς στην λίστα προβολής του χρήστη, το εικονίδιο αλλάζει χρώμα (γίνεται κίτρινο).

Μπορεί να αξιολογήσει την σειρά πατώντας την επιλογή "Rate now" στην κάρτα αξιολόγησης. Στο κλικ της επιλογής για μια νέα αξιολόγηση της σειράς, εμφανίζεται ένα bottomsheet στο οποίο ο χρήστης καλείται να ορίσει το πόσα αστέρια θεωρεί ότι πρέπει να βάλει στην σειρά και εφόσον το επιθυμεί, μπορεί να προσθέσει και μια μικρή κριτική. Πατώντας το κουμπί "Submit Rating" η κριτική του σώζεται στην βάση της εφαρμογής και πλέον στην οθόνη της σειράς, φαίνεται πως ο χρήστης την έχει ήδη αξιολογήσει. Στην περίπτωση που έχει προσθέσει και κείμενο στην αξιολόγηση του, η κριτική του θα εμφανίζεται πλέον στο κάτω μέρος της οθόνης στην ενότητα κριτικές (Reviews).

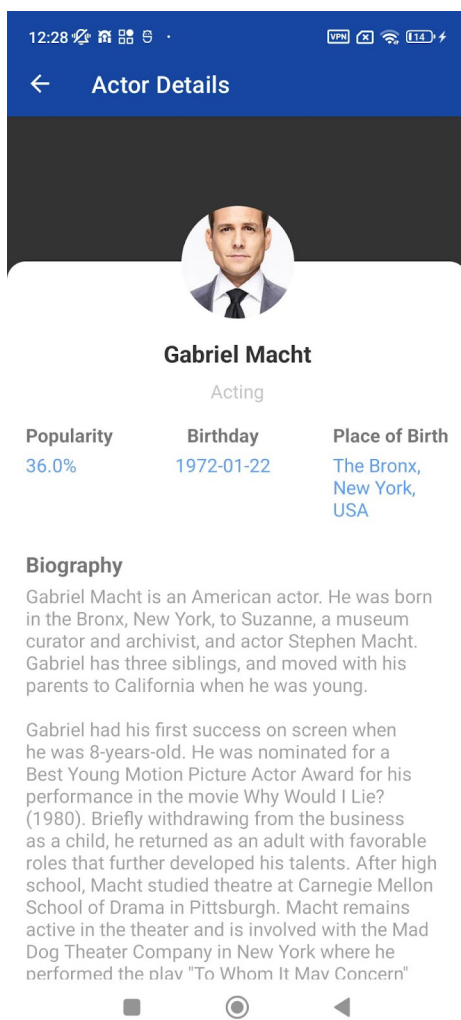
Μπορεί να ανοίξει τις πληροφορίες του εκάστοτε ηθοποιού πατώντας πάνω στην φωτογραφία του, ή να ανοίξει εκ νέου μια οθόνη πληροφοριών για κάποια άλλη από τις σειρές που εμφανίζονται στις ενότητες Similar Series ή Recommended Series πατώντας πάνω σε αυτή που τον ενδιαφέρει.



Εικόνα 18. Η οθόνη πληροφοριών σειράς της εφαρμογής

Οθόνη Πληροφοριών Ηθοποιού (Cast Details Screen)

Η οθόνη των πληροφοριών ενός ηθοποιού είναι προσβάσιμη μόνο από την οθόνη πληροφοριών μιας σειράς. Στην οθόνη αυτή, ο χρήστης μπορεί να λάβει πληροφορίες για τον ηθοποιό που επέλεξε. Πιο συγκεκριμένα, εμφανίζονται κάθετα τα προσωπικά του στοιχεία (Όνοματεπώνυμο, ο λόγος που είναι γνωστός ο/η συγκεκριμένος/η ηθοποιός, το ποσοστό διασημότητας του, η ημερομηνία γενεθλίων του και πότε γεννήθηκε, η βιογραφία του και τέλος οι σειρές για τις οποίες είναι γνωστός.

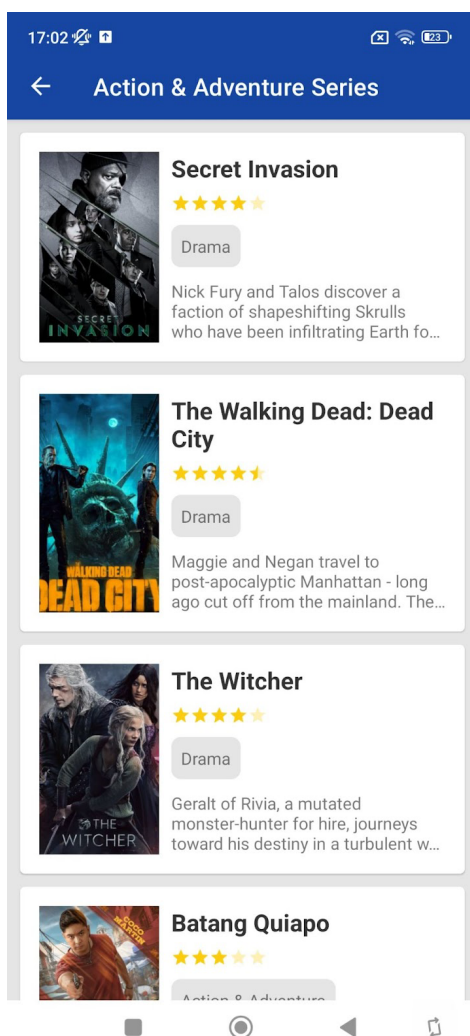


Εικόνα 19. Η οθόνη πληροφοριών ενός ηθοποιού στην εφαρμογή.

Οθόνη αποτελεσμάτων (General Results Screen)

Η οθόνη προβολής αποτελεσμάτων αποτελεί μια βασική οθόνη της εφαρμογής. Αναλαμβάνει να δείξει σειρές βάσει της κατηγορίας που επιλέχθηκε από τον χρήστη. Η οθόνη αυτή είναι προσβάσιμη από διάφορα σημεία της εφαρμογής που διαθέτουν το κουμπί "ALL" στα δεξιά της κατηγορίας. Η οθόνη αυτή εμφανίζει αποτελέσματα σειρών και όσο ο χρήστης αποφασίσει να scrollάρει πιο κάτω, γίνονται εκ νέου κλήσεις στην επόμενη σελίδα (page) των αποτελεσμάτων. Με αυτόν τον τρόπο, ο χρήστης δεν είναι περιορισμένος στο να διαβάσει μόνο την πρώτη σελίδα των αποτελεσμάτων, αλλά είναι σε θέση να έχει ένα μεγάλο εύρος επιλογών στη διάθεση του.

«Ανάπτυξη εφαρμογής σε περιβάλλον Android με εξατομικευμένες προτάσεις σειρών και δυνατότητα interaction με 44 χρήστες»



Εικόνα 20. Η οθόνη πληροφοριών για κατηγορίες σειρών

4.2.4 Προφίλ

My Profile & User Profile

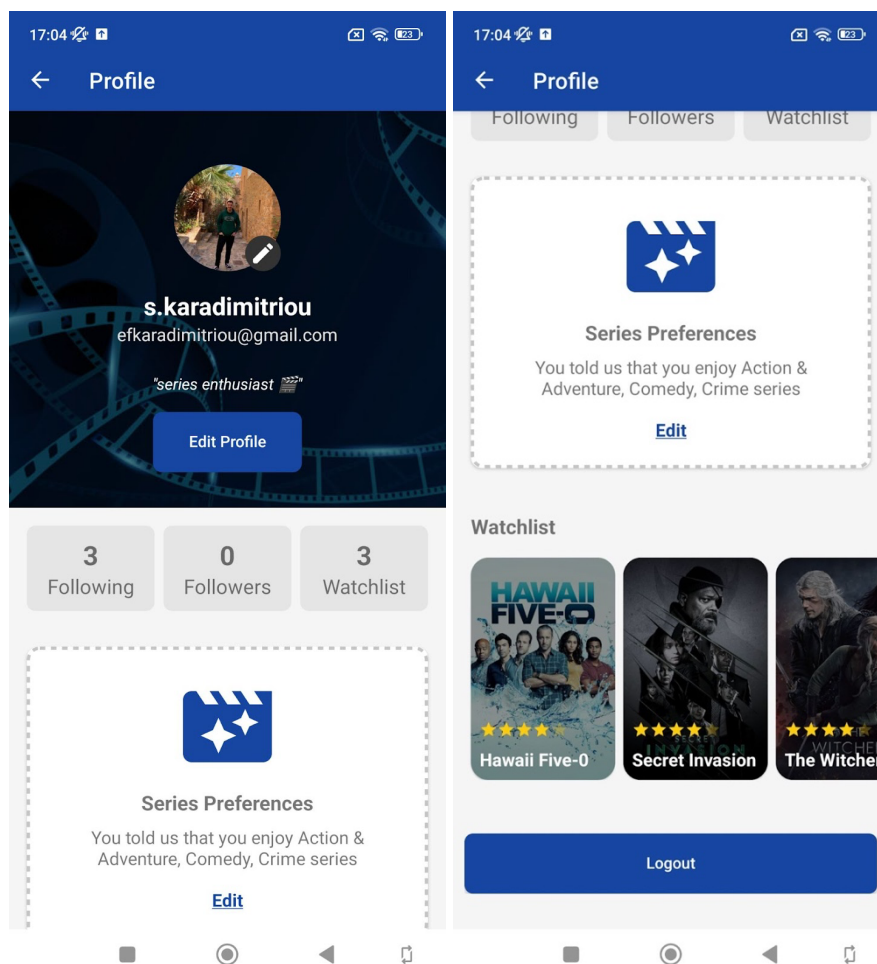
Το προφίλ του χρήστη είναι προσβάσιμο είτε απο την αρχική οθόνη, είτε από οποιαδήποτε άλλη οθόνη εμφανίζει κάποια φωτογραφία κάποιου χρήστη. Στην περίπτωση που ο χρήστης για τον οποίο θέλω να δω το προφίλ είναι ο συνδεδεμένος χρήστης της εφαρμογής, μεταφέρομαι στο

προφίλ μου και μου δίνονται οι δυνατότητες διαχειριστή στο προφίλ, δηλαδή μπορώ να το επεξεργαστώ. Σε διαφορετική περίπτωση, έχω δικαιώματα προβολής του προφίλ.

Πιο συγκεκριμένα, στην περίπτωση του προφίλ μου ισχύουν τα παρακάτω:

Ο χρήστης αν εισέλθει στο προφίλ του, βλέπει αρχικά τα βασικά στοιχεία του προφίλ του (φωτογραφία, username, email, bio περιγραφή). Ο χρήστης μπορεί να μεταβεί στη σελίδα αλλαγής φωτογραφίας πατώντας το κουμπί με το μολύβι δίπλα από την φωτογραφία. Στη συνέχεια, υπάρχει το κουμπί επεξεργασίας προφίλ ("Edit profile") με το οποίο ο χρήστης ανοίγει την σελίδα αλλαγής στοιχείων. Ακολουθούν τα στατιστικά στοιχεία που αφορούν το πόσους χρήστες ακολουθεί, ποιοί τον ακολουθούν και πόσες σειρές έχει στην λίστα προβολής του (watchlist). Πατώντας σε μια από τις τρεις αυτές επιλογές, ανοίγει η σελίδα των αποτελεσμάτων. Κάτω από τα στατιστικά στοιχεία, υπάρχει η κάρτα των προτιμήσεων του χρήστη. Εκεί εμφανίζονται οι προτιμήσεις του χρήστη (αν ο χρήστης τις έχει ορίσει προηγουμένως), σε διαφορετική περίπτωση υπάρχει κατάλληλο μήνυμα το οποίο παρακινεί τον χρήστη να προσθέσει τις προτιμήσεις του. Εν συνεχεία, εμφανίζεται οριζόντια η λίστα προβολής του χρήστη εφόσον υπάρχει. Αν δεν υπάρχει, εμφανίζεται μια ακόμα κάρτα που τον παρακινεί να προσθέσει κάποια σειρά στην λίστα του. Τέλος, δίνεται η επιλογή της αποσύνδεσης πατώντας το κουμπί της αποσύνδεσης ("Logout").

Στην περίπτωση της προβολής ενός προφίλ άλλου χρήστη, ισχύουν οι ίδιοι κανόνες ως προς την προβολή των στοιχείων (εκτός της επιλογής για αλλαγή φωτογραφίας και της επεξεργασίας προφίλ). Στη θέση του, υπάρχει ένα κουμπί με το οποίο μπορεί ο χρήστης να ακολουθήσει κάποιον και λαμβάνει δύο καταστάσεις. Η πρώτη κατάσταση αν δεν ακολουθώ τον χρήστη αυτόν είναι να γράφει "Follow" ενώ αν τον ακολουθήσω "Unfollow" υποδηλώνοντας έτσι την ενέργεια που πρόκειται να εκτελεστεί. Από την κατηγορία των στατιστικών μπορώ να ανοίξω και να δω ποιους χρήστες ακολουθεί ο εκάστοτε χρήστης, ποιοί τον ακολουθούν καθώς και την λίστα προβολής του.

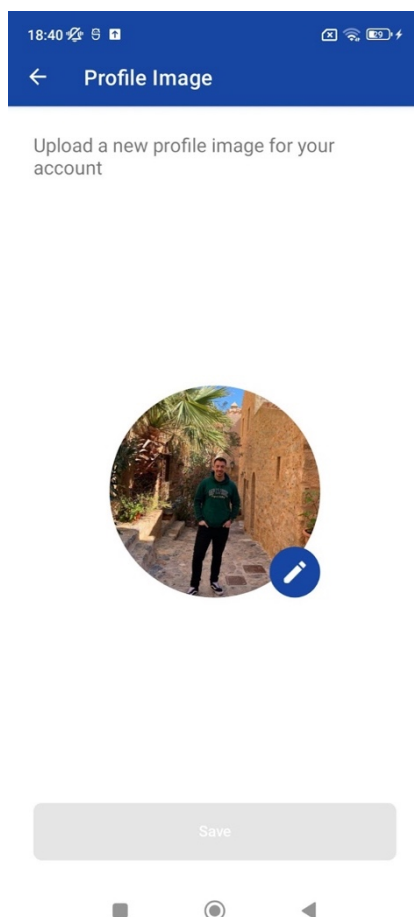


Εικόνα 21. Το προφίλ μου στην εφαρμογή.

Upload image

Η οθόνη αλλαγής φωτογραφίας (upload image) είναι προσβάσιμη από το προφίλ του χρήστη πατώντας το κυκλικό εικονίδιο δίπλα από την φωτογραφία του χρήστη στην οθόνη του προφίλ του συνδεδεμένου χρήστη. Πατώντας το κυκλικό μπλε εικονίδιο ανοίγει στον χρήστη ένα μενού επιλογών (bottomsheet) στο κάτω μέρος προκειμένου να επιλέξει τον τρόπο με τον οποίο θα ανεβάσει την φωτογραφία του. Οι επιλογές που έχει ο χρήστης είναι είτε ανέβασμα από την συλλογή, είτε με την χρήση της κάμερας της συσκευής.

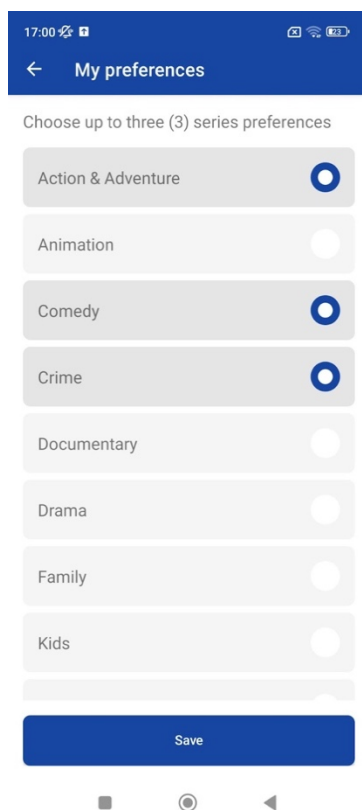
Αφού ο χρήστης επιλέξει και ανεβάσει τελικά μια νέα φωτογραφία, ενεργοποιείται το κουμπί στο κάτω μέρος της οθόνης "Save" με το οποίο ο χρήστης μπορεί να αποθηκεύσει τις αλλαγές στο προφίλ του. Με την επιτυχή αποθήκευση της φωτογραφίας, ο χρήστης επιστρέφει στο προφίλ του.



Εικόνα 22. Η οθόνη στην οποία ανεβάζει ο χρήστης εικόνα προφίλ.

Preferences

Στην οθόνη των προτιμήσεων, εμφανίζονται οι κατηγορίες των σειρών που υποστηρίζει η εφαρμογή. Ο χρήστης καλείται να επιλέξει μέχρι τρεις επιλογές από την λίστα που προβάλλεται. Όταν επιλέξει μια από αυτές, εμφανίζεται το επιλεγμένο μπλε κυκλάκι στα δεξιά της οθόνης προκειμένου να είναι εμφανές τι έχει επιλέξει ο χρήστης. Στο πάτημα του κουμπιού αποθήκευσης “Save” σώζονται οι επιλογές του χρήστη στο προφίλ του.



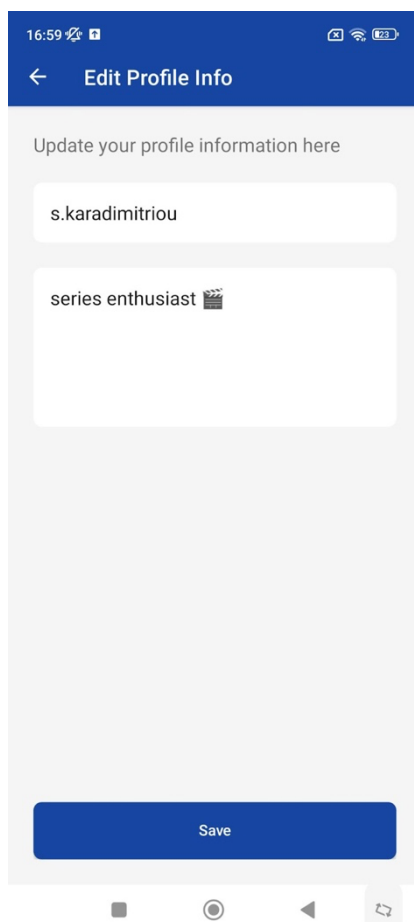
Εικόνα 23. Η οθόνη επιλογής προτιμήσεων.

Results

Η οθόνη των αποτελεσμάτων είναι προσβάσιμη από το προφίλ πατώντας πάνω στους following, followers ή watchlist. Πιο συγκεκριμένα, η οθόνη αναλαμβάνει να τραβήξει από την βάση τα στοιχεία εκείνα που ικανοποιούν τις ανάγκες προβολής του χρήστη ανάλογα την περίπτωση. Έτσι, είτε η λίστα θα επιστρέψει χρήστες είτε σειρές προκειμένου ο χρήστης να ξεκινήσει εκ νέου κάποια αναζήτηση. Σε περίπτωση κανενός αποτελέσματος, εμφανίζεται μια κάρτα ενημερώνοντας τον χρήστη για την άδεια λίστα του.

Update Profile Information

Η οθόνη της αλλαγής των στοιχείων του προφίλ είναι προσβάσιμη από το προφίλ του συνδεδεμένου χρήστη. Στην οθόνη αυτή, ο χρήστης μπορεί να τροποποιήσει το username και να ορίσει την bio περιγραφή του προφίλ του. Όποιο και αν αποφασίσει ο χρήστης να αλλάξει, πατώντας το κουμπί της αποθήκευσης, σώζει τις αλλαγές που πραγματοποίησε και επιστρέφει στο προφίλ του στο οποίο πλέον φαίνονται τα ανανεωμένα στοιχεία.

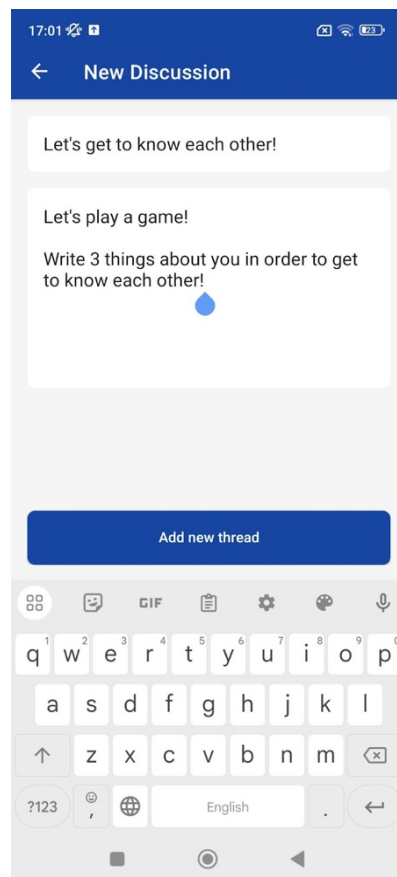


Εικόνα 24. Η οθόνη αλλαγής ονόματος χρήστη και bio μηνύματος.

4.2.5 Συζητήσεις (Forum)

Add new thread

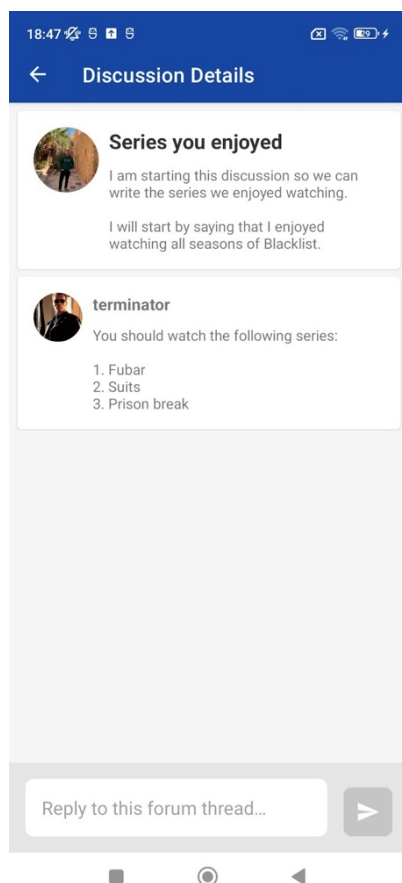
Ο χρήστης μεταφέρεται στην οθόνη συγγραφής μιας νέας συζήτησης (thread). Εκεί, ζητείται από τον χρήστη να δώσει τον τίτλο της ο οποίος δεν μπορεί να υπερβαίνει την μια γραμμή, καθώς και το κείμενο που θα την συνοδεύει. Το κουμπί της προσθήκης είναι ανενεργό ενώ μόλις συμπληρωθούν και τα δύο πεδία ενεργοποιείται. Κατά την επιτυχή δημιουργία μιας νέας συνομιλίας, ο χρήστης επιστρέφει στην οθόνη προβολής των συζητήσεων.



Εικόνα 25. Η οθόνη προσθήκης νέας συζήτησης

Thread Details

Στην οθόνη των πληροφοριών μιας συζήτησης (forum thread) ο χρήστης μεταβαίνει πατώντας μια συζήτηση (thread) στην οθόνη των συζητήσεων. Ο χρήστης έχει την δυνατότητα να αλληλεπιδράσει με άλλους χρήστες της εφαρμογής απαντώντας στην συζήτηση που έχει επιλέξει. Το κουμπί της αποστολής μηνύματος είναι ανενεργό σε κενό μήνυμα, ενώ αν ο χρήστης πληκτρολογήσει κάτι ενεργοποιείται. Αν ο χρήστης αποφασίσει να πατήσει πάνω στην εικόνα κάποιου χρήστη, μεταβαίνει στο προφίλ του. Επίσης, σέρνοντας προς τα κάτω την οθόνη, ενεργοποιείται ο swipe to refresh μηχανισμός με τον οποίο γίνεται refresh η λίστα των μηνυμάτων. Σε περίπτωση που υπάρχουν πολλές απαντήσεις σε μια συζήτηση, κατα την είσοδο η την αποστολή μιας νέας απάντησης ο χρήστης μεταφέρεται στο τέλος της λίστας (στο πιο πρόσφατο μήνυμα).

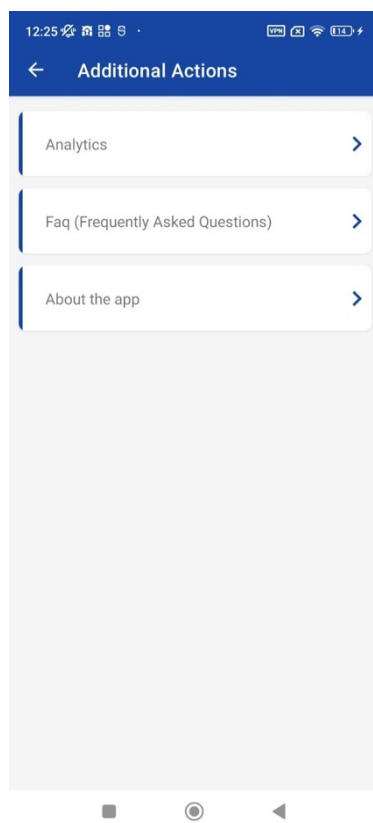


Εικόνα 27. Η οθόνη πληροφοριών μιας συζήτησης.

4.2.6 Ρυθμίσεις (Settings)

Οθόνη Επιλογών (Intro)

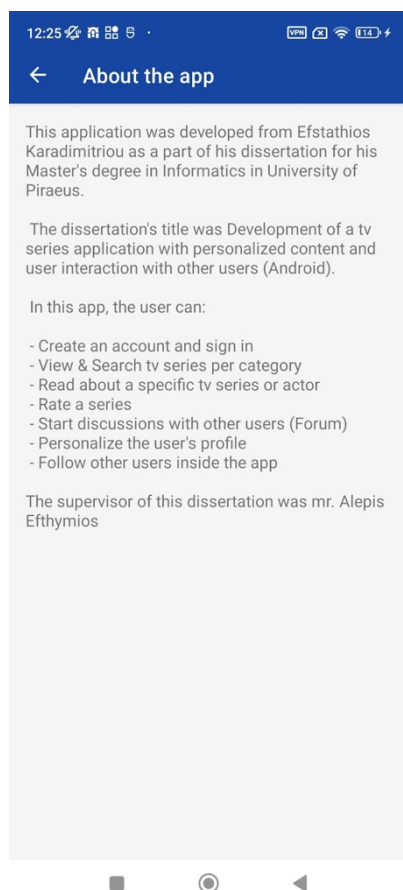
Η οθόνη αυτή είναι προσβάσιμη πατώντας το εικονίδιο των επιλογών στο πάνω δεξιά μέρος της μπάρας στην κεντρική οθόνη. Ο χρήστης ενώ βρίσκεται στην οθόνη, του εμφανίζεται μια λίστα επιλογών με τρεις κατηγορίες (Analytics, FAQ και About the app) στις οποίες μπορεί να πατήσει πάνω για να λάβει πρόσθετη πληροφορία σχετικά με την εφαρμογή και τις λειτουργίες της.



Εικόνα 28. Η οθόνη ρυθμίσεων.

About

Η οθόνη αυτή είναι προσβάσιμη πατώντας την επιλογή “About the app”. Όπως μπορεί κάποιος να φανταστεί, η οθόνη αυτή περιέχει ένα κείμενο σχετικά με τον λόγο της ανάπτυξης της εφαρμογής επιγραμματικά σε περίπτωση που επιθυμεί ο χρήστης να γνωρίζει. Επίσης, στο κείμενο αυτό, περιέχονται οι βασικές λειτουργίες που επιτελεί η εφαρμογή.



Εικόνα 29. Η οθόνη σχετικά με την εφαρμογή

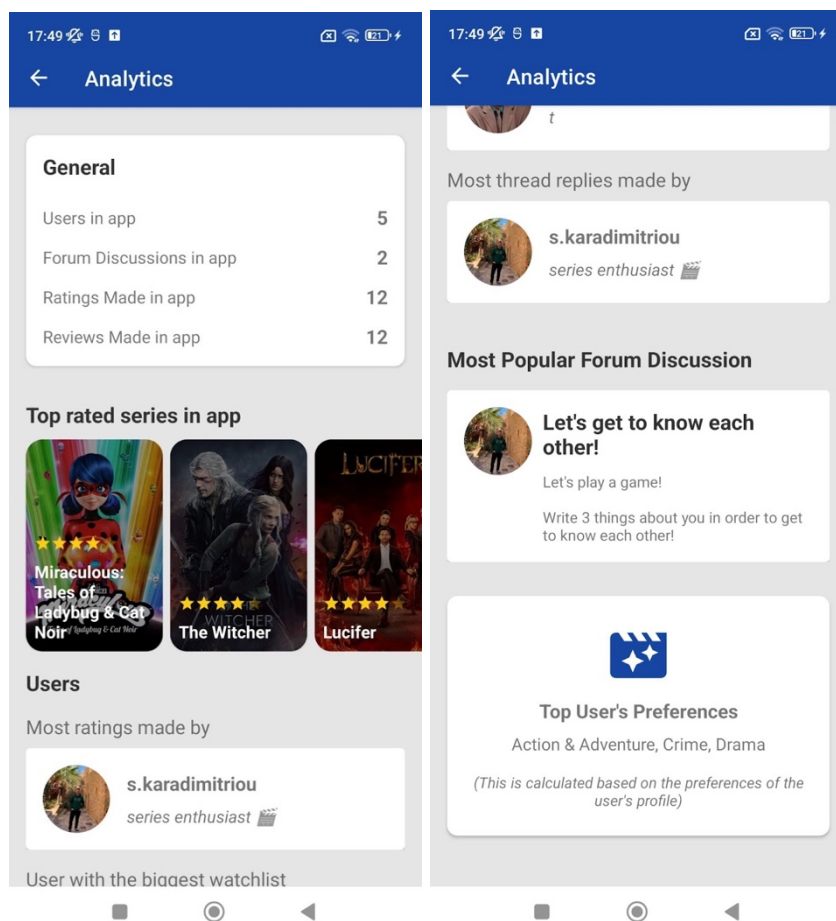
Στατιστικά στοιχεία εφαρμογής (Analytics)

Η οθόνη αυτή είναι προσβάσιμη πατώντας την επιλογή “Analytics” από την λίστα της οθόνης επιλογών. Πιο συγκεκριμένα, ο χρήστης σε αυτή την οθόνη μπορεί να αντλήσει πληροφορίες σχετικά με τα στατιστικά στοιχεία που προκύπτουν από την χρήση της εφαρμογής. Για την ακρίβεια, όπως έχει αναφερθεί και παραπάνω, υλοποιήθηκε ένα backend application το οποίο είναι προγραμματισμένο να τρέχει σε ένα συγκεκριμένο χρονικό διάστημα και να διαβάζει την βάση δεδομένων, να την επεξεργάζεται και να παράγει αυτά τα analytics για τους χρήστες.

Συγκεκριμένα, αυτό που μπορεί να διαβάσει ο χρήστης είναι γενικά στοιχεία, όπως το πόσους χρήστες έχει η εφαρμογή αυτή τη στιγμή, πόσες συζητήσεις υπάρχουν στο forum, πόσες αξιολογήσεις και πόσες κριτικές έχουν γίνει μέσα από την εφαρμογή. Ακόμα, στο κομμάτι των σειρών, υπάρχει η κατηγορία των top rated series στην οποία εμφανίζονται οι σειρές με τα περισσότερα in app ratings από τους χρήστες. Στη συνέχεια, υπάρχει η κατηγορία των χρηστών στην οποία κάθετα εμφανίζονται οι χρήστες με τις περισσότερες αξιολογήσεις, το μεγαλύτερο

«Ανάπτυξη εφαρμογής σε περιβάλλον Android με εξατομικευμένες προτάσεις σειρών και δυνατότητα interaction με 54 χρήστες»

watchlist, τους περισσότερους followers και τους πιο ενεργούς χρήστες στο forum. Ακολουθεί το πιο popular forum thread στην εφαρμογή ενώ δεν θα μπορούσε να λείπει το κομμάτι υπολογισμού των τριών πιο δημοφιλών προτιμήσεων των χρηστών στο profile τους.



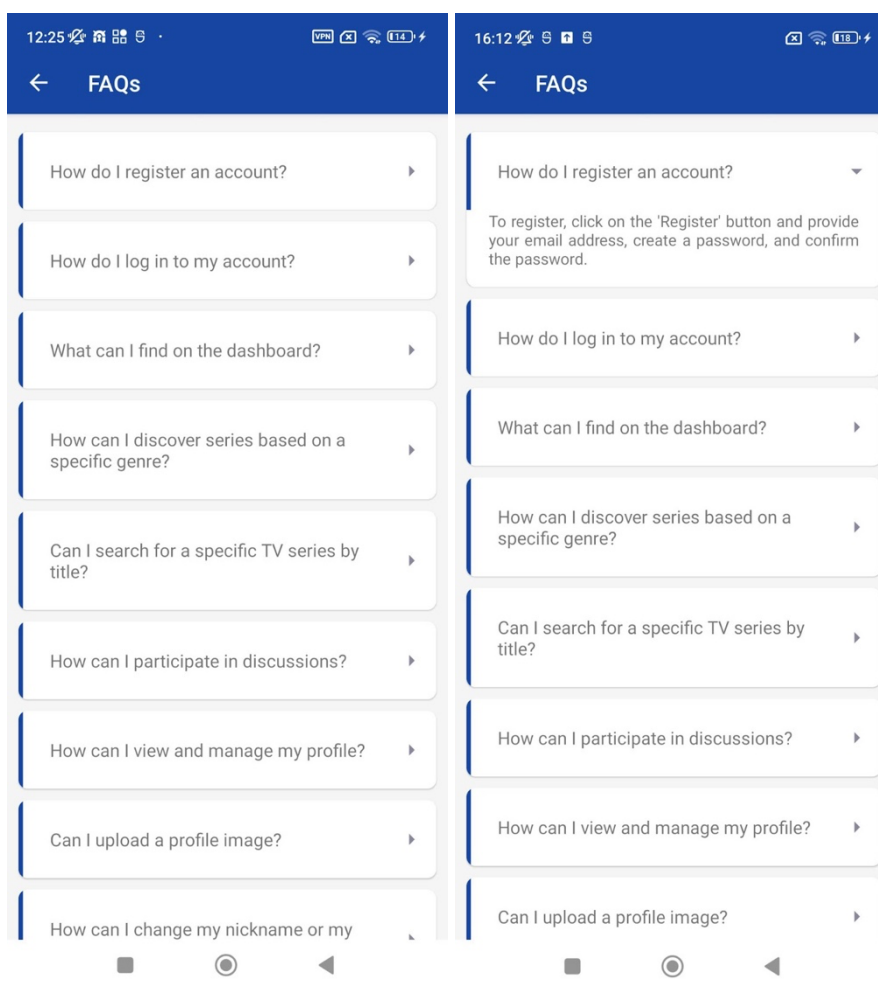
Εικόνα 30. Η οθόνη προβολής στατιστικών στην εφαρμογή.

FAQs

Ο χρήστης μπορεί να πλοηγηθεί στην οθόνη των FAQs από την οθόνη επιλογών πατώντας την επιλογή "Faq (Frequently Asked Questions)". Στην οθόνη αυτή, δίνεται η δυνατότητα στον χρήστη να διαβάσει τις ερωταπαντήσεις που υπάρχουν στην βάση δεδομένων της Cloud Firestore (Firebase). Το περιεχόμενο της λίστας που προβάλλεται μπορεί να αλλάξει ανα πάσα στιγμή από τους διαχειριστές της εφαρμογής, χωρίς να απαιτείται ενημέρωση της εφαρμογής.

Στην λίστα των FAQs, βρίσκονται συγκεντρωμένες ερωτήσεις που εξηγούν τις λειτουργίες της εφαρμογής στις οποίες μπορεί ο χρήστης να ανατρέξει οποιαδήποτε στιγμή. Ο

χρήστης όταν επιλέγει μια ερώτηση, το κουτάκι μεγαλώνει και παρουσιάζεται η απάντηση της ερώτησης.



Εικόνα 31. Η οθόνη συχνών ερωτήσεων

4.3 ΒΟΗΘΗΤΙΚΕΣ ΥΛΟΠΟΙΗΣΕΙΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Navigator

Η εφαρμογή χρησιμοποιεί μια custom κλάση για το navigation στα flows της η οποία ονομάζεται Navigator. Ο Navigator στην εφαρμογή χρησιμοποιεί το navigation component που αναφέρθηκε

«Ανάπτυξη εφαρμογής σε περιβάλλον Android με εξατομικευμένες προτάσεις σειρών και δυνατότητα interaction με χρήστες» 56

παραπάνω για την μετάβαση από μια οθόνη σε μια άλλη. Ο navigation controller του navigation component κρατά όλη την χρήσιμη πληροφορία για το navigation. Ο Navigator παίρνει σαν γνώρισμα στον constructor του το ενεργό Activity και τον navigation controller του flow. Ορίζονται ξεκάθαρα enums όσα τα Actions που μπορούν να γίνουν στο εκάστοτε flow (για παράδειγμα CTA_BUTTON_CLICK). Ο navigator αναλαμβάνει να κάνει navigation με βάση τα enum actions του κάθε flow. Με αυτόν τον τρόπο, η λογική του navigation υπάρχει συγκεντρωμένη σε μια κλάση και δεν βρίσκεται στο activity. Αυτός είναι ένας τρόπος να απομονωθεί η λογική σε ένα κεντρικό σημείο και να μπορεί να γίνει test πιο εύκολα στα σενάρια της εφαρμογής.

Authenticator

Ο Authenticator μέσα στην εφαρμογή παίζει κεντρικό ρόλο στο onboarding section της εφαρμογής. Πιο συγκεκριμένα, διεκπαιρεύει τις διαδικασίες της εγγραφής, σύνδεσης και αποσύνδεσης του χρήστη στην εφαρμογή. Ακόμα, κρατάει την πληροφορία για τον ενεργό χρήστη στην εφαρμογή (εφόσον υπάρχει).

Session Manager

Ο Session Manager στην εφαρμογή είναι ένα από τα πιο βασικά στοιχεία της εφαρμογής. Η χρήση του είναι να γνωρίζει ανα πάσα ώρα και στιγμή τα στοιχεία του χρήστη που αποτελούν το ολοκληρωμένο του προφίλ. Στην περίπτωση της εφαρμογής, ένας χρήστης αποτελείται από το email, username, bio κείμενο, image url, μια λίστα με preferences και το firebase id του. Ο Session Manager κάνει χρήση του Jetpack DataStore που αναφέρθηκε παραπάνω προκειμένου να αποθηκεύει και να τραβάει από το local storage της εφαρμογής τον ενεργό χρήστη όπου χρειάζεται. Η πληροφορία που είναι αποθηκευμένη στο DataStore, στέλνεται με την βοήθεια των Kotlin flows.

ΚΕΦΑΛΑΙΟ 5

5.1 ΣΥΜΠΕΡΑΣΜΑΤΑ

Συνοψίζοντας την παρούσα εργασία, αναμφίβολα το λειτουργικό σύστημα του android είναι κυρίαρχο στην αγορά των smartphones. Η ανάγκη για γρήγορες και αξιόπιστες εφαρμογές είναι περισσότερο παρόν από κάθε άλλη χρονική στιγμή λόγω της εξάπλωσης του διαδικτύου. Είναι πραγματικά πολύ ενδιαφέρον πως μια εφαρμογή μπορεί επιδράσει θετικά στην κοινωνία αναβαθμίζοντας την καθημερινότητα των χρηστών της.

Η ανάπτυξη μιας εφαρμογής για κινητές συσκευές είναι ένας πολύ ενδιαφέρον κλάδος απασχόλησης. Η εμπειρία του χρήστη σε μια εφαρμογή παίζει βασικό ρόλο στην ανάπτυξη εφαρμογών και στην διατήρηση (retention) ενός χρήστη σε αυτή. Μια εφαρμογή θα πρέπει να είναι όσο το δυνατόν πιο γρήγορη γίνεται και να προσφέρει την μέγιστη εμπειρία χρήσης σε αυτή, ενώ ταυτόχρονα λαμβάνει υπόψη τους περιορισμούς μιας μικρότερης σε μέγεθος συσκευής. Ακόμη, θα πρέπει να λαμβάνονται υπόψη όλες εκείνες τις δυσκολίες που θα

μπορούσε να αντιμετωπίζει ένας δυνητικός χρήστη σε αυτή και να τις καλύπτει με όσο το δυνατόν καλύτερο τρόπο.

Στην παρούσα εργασία αναπτύχθηκε μια εφαρμογή για android συσκευές η οποία δίνει στον χρήστη την δυνατότητα να πραγματοποιεί εγγραφή και σύνδεση με τον δικό του λογαριασμό, να βλέπει, να διαβάζει και να αναζητά σειρές είτε απο προδιαγεγραμμένες λίστες με σειρές, είτε να αναζητήσει σειρές βάσει ονόματος ή κατηγορίας. Ακόμα, δίνεται η δυνατότητα να παραμετροποιήσει ο χρήστης το προφίλ του λογαριασμού του προσθέτοντας το δικό του μοναδικό όνομα χρήστη και φωτογραφία, τις προτιμήσεις για τις σειρές του και έτσι να φέρνει την εφαρμογή στα δικά του μέτρα. Ο χρήστης μπορεί επίσης να αλληλεπιδράσει με άλλους χρήστες μέσω του feature των συζητήσεων εντός της εφαρμογής και να ακολουθήσει άλλους χρήστες. Τέλος, δίνεται η δυνατότητα προσθήκης σειρών στην δικιά του λίστα προβολής (watchlist).

Μέσα από την συγκεκριμένη εργασία αποκτήθηκαν σημαντικές γνώσεις στο κομμάτι της ανάπτυξης εφαρμογών για android συσκευές καθώς χρησιμοποιήθηκαν οι τελευταίες τάσεις στα εργαλεία που θα μπορούσαν να την απαρτίζουν, τα οποία περιγράφονται στην ενότητα των τεχνολογιών ανάπτυξης. Δώθηκε επίσης η δυνατότητα να αποκτηθούν γνώσεις πάνω στην ανάπτυξη εφαρμογών με την χρήση της Kotlin ως android-first γλώσσα προγραμματισμού τόσο στην android εφαρμογή όσο και στην συμπληρωματική εφαρμογή για την παραγωγή των στατιστικών στοιχείων.

5.2 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Backend System για την εφαρμογή

Η εφαρμογή χρησιμοποιεί συγκεκριμένα κανάλια (data sources) προκειμένου να λάβει πληροφορίες για τις λίστες με τις σειρές, τους ηθοποιούς, τα αποτελέσματα αναζήτησης κ.τ.λ. Όλη η επεξεργασία των δεδομένων αυτή την στιγμή γίνεται σε επίπεδο client. Πιο συγκεκριμένα, η εφαρμογή καλείται να λάβει δεδομένα από πολλά data sources σε οθόνες έτσι ώστε να απεικονίσει όλη την πληροφορία που είναι αναγκαία στην οθονη.

Θα μπορούσε σε δεύτερο χρόνο να αναπτυχθεί ένα ολόκληρο back end application με spring boot το οποίο θα λειτουργεί ως μεσάζοντας μεταξύ του TMDB API και της εφαρμογής. Αν λάβουμε ως παράδειγμα την οθόνη πληροφοριών μιας σειράς, το backend application θα είχε ως σκοπό να απομονώσει ένα μεγάλο κομμάτι που υπάρχει αυτή την στιγμή στον client (android app) και αφορά την λήψη των πληροφοριών της σειράς με βάση το id της, την λήψη της λίστας παρόμοιων σειρών, προτάσεων και κριτικών. Αυτά τα δεδομένα θα μπορούσε να τα συγκεντρώνει το backend application και να τα επιστρέφει με κάποια κλήση την οποία θα χρησιμοποιεί η εφαρμογή.

Με την χρήση του Spring Boot είναι πολύ γρήγορο να φτιαχτεί μια REST υπηρεσία που να εξυπηρετεί αυτή την ανάγκη. Επίσης, είναι προτιμότερο όλα τα δεδομένα να συγκεντρώνονται σε κάποιο κεντρικό Backend σύστημα προκειμένου να είναι προσβάσιμα από διαφορετικούς clients που ίσως υπάρξουν και στο μέλλον (π.χ web app, iOS app).

Reactions στις απαντήσεις των χρηστών

Εντός της εφαρμογής, δίνεται η δυνατότητα στους χρήστες να απαντούν σε ανοιχτές συζητήσεις (forum threads). Εκτός από την απάντηση την οποία δίνουν θα μπορούσε να υλοποιηθεί ένας μηχανισμός με γρήγορες αντιδράσεις (reactions). Πολλές εφαρμογές κοινωνικής δικτύωσης έχουν αντίστοιχες υλοποιήσεις στις αναρτήσεις των χρηστών και θα μπορούσε να χρησιμοποιηθεί για περαιτέρω ανάπτυξη των analytics της εφαρμογής, με σκοπό να εξαχθούν συμπεράσματα για την συμπεριφορά των χρηστών κατά τη χρήση της εφαρμογής.

Προσθήκη ταινιών στην εφαρμογή

Η εφαρμογή αυτή την στιγμή είναι περιορισμένη να εμφανίζει περιεχόμενο σειρών μόνο. Για τον σκοπό αυτό, χρησιμοποιεί μόνο τα μισά διαθέσιμα endpoints του TMDB API. Μελλοντικά θα μπορούσε να επεκταθεί η εφαρμογή έτσι ώστε να καλύπτει και περιεχόμενο ταινιών μέσα στην εφαρμογή έτσι ώστε ένας χρήστης να μπορεί να λάβει ολοκληρωμένες υπηρεσίες από την εφαρμογή και να μην χρειάζεται να ανοίξει τρίτες εφαρμογές σε περίπτωση που αναζητά κάτι.

Σκούρο Θέμα (Dark Theme)

Το dark theme είναι ένα από τα πιο δημοφιλή χαρακτηριστικά που υπάρχει στις εφαρμογές κινητών και όχι μόνο σήμερα. Δίνει την δυνατότητα στον χρήστη να αντιστρέψει τα χρώματα της οθόνης από τους φωτεινούς τόνους χρωμάτων σε σκούρους.

Το dark theme σε μια εφαρμογή είναι μια καλή πρακτική για τους παρακάτω τρεις βασικούς λόγους:

- Μια εφαρμογή με σκούρο θέμα, η οποία δεν χρησιμοποιεί έντονα χρώματα μειώνει την καταπόνηση των ματιών των χρηστών της. Επίσης, δίνεται η δυνατότητα ο χρήστης να επιλέξει το θέμα της εφαρμογής του έτσι ώστε να ταιριάζει στο στυλ και την προσωπικότητα του.
- Εξοικονόμηση μπαταρίας. Οι εφαρμογές που χρησιμοποιούν dark theme πετυχαίνουν καλύτερη διαχείριση της μπαταρίας κατά την χρήση της εφαρμογής.
- Σύμφωνα με έρευνες, τα κείμενα σε μια εφαρμογή που έχει ενεργοποιημένο το dark theme πετυχαίνει οι χρήστες της να μπορούν να διαβάσουν τα κείμενα της εφαρμογής πιο απλά λόγω της αντίθεσης.

Αναζήτηση Χρηστών εντός της εφαρμογής

Αυτή τη στιγμή η εφαρμογή είναι σε ένα στάδιο με τα ελάχιστα δυνατά χαρακτηριστικά που θα μπορούσε να διαθέτει (MVP - Minimum Viable Product). Θα μπορούσε να μεγαλώσει παραπάνω υλοποιώντας ένα χαρακτηριστικό (feature) στο οποίο ο χρήστης θα μπορούσε να αναζητά με βάση το όνομα χρήστη άλλους χρήστες εντός της εφαρμογής. Στην λίστα αποτελεσμάτων που θα προέκυπτε από την αναζήτηση θα μπορούσε ο χρήστης να οδηγείται στο προφίλ του εκάστοτε χρήστη και να αλληλεπιδρά μαζί του.

Δημιουργία πολλών custom λιστών σειρών

Ένας χρήστης αυτή τη στιγμή στην εφαρμογή έχει μια λίστα στην οποία μπορεί να προσθέτει σειρές και ονομάζεται "Watchlist". Μια δημοφιλής υπηρεσία μουσικής δίνει την δυνατότητα ο χρήστης να δημιουργεί διάφορες λίστες, να τις ονομάζει με ότι όνομα επιθυμεί

εκείνος και να αποθηκεύει περιεχόμενο σε αυτές. Στη συνέχεια, αυτές οι λίστες θα μπορούσαν να γίνουν ακόμα και προώθηση εντός ή εκτός της εφαρμογής σε άλλους χρήστες αυξάνοντας την αλληλεπίδραση των χρηστών της εφαρμογής. Αυτό το χαρακτηριστικό θα μπορούσε να αναπτυχθεί και στο Series Mania.

Ειδοποιήσεις (Notifications)

Στην εφαρμογή υπάρχει αυξημένη αλληλεπίδραση του χρήστη με την ίδια την εφαρμογή αλλά και με άλλους χρήστες. Ένας χρήστης μπορεί να προσθέτει σειρές στην λίστα του προκειμένου να τις παρακολουθήσει ή απλά να τις αποθηκεύσει, να δημιουργήσει συζητήσεις, να ακολουθήσει άλλους χρήστες και να αλληλεπιδράσει μαζί τους. Όλες αυτές οι ενέργειες θα μπορούσαν να αποτυπωθούν εντός της εφαρμογής σε ένα section ειδοποιήσεων. Κάθε φορά που θα συμβαίνει κάτι εντός της εφαρμογής, ο χρήστης θα μπορούσε να λαμβάνει μια real-time ειδοποίηση που να τον ειδοποιεί για το τί ακριβώς συνέβη (π.χ κάποιος χρήστης με ακολούθησε).

Αιτήματα φιλίας

Εκτός από το χαρακτηριστικό που υπάρχει αυτή τη στιγμή στην έκδοση της εφαρμογής και δίνει τη δυνατότητα στον χρήστη να ακολουθεί άλλους χρήστες, θα μπορούσε να υλοποιηθεί ένας αντίστοιχος μηχανισμός με αιτήματα φιλίας μεταξύ δύο χρηστών. Τα αιτήματα αυτά θα μπορούσαν να εγκρίνονται ή να απορρίπτονται ανάλογα με το τί ακριβώς επιθυμεί ο κάθε χρήστης.

Ιδιωτικά Προφίλ Χρηστών

Θα μπορούσαν οι χρήστες εντός της εφαρμογής να έχουν ανοιχτό ή κλειστό προφίλ προς τον υπόλοιπο κόσμο της εφαρμογής.

Κλειστές συζητήσεις

Θα μπορούσαν οι συζητήσεις να είναι ανοιχτές προς όλους τους χρήστες, ή να θέλουν κάποια πρόσκληση προκειμένου να αποκτήσει κάποιος άλλος χρήστης πρόσβαση σε αυτές.

Διαφημίσεις εντός της εφαρμογής

Τελευταία ιδέα περαιτέρω ανάπτυξης της εφαρμογής αλλά εξίσου σημαντική, είναι η προσθήκη διαφημίσεων εντός της εφαρμογής. Πιο συγκεκριμένα, θα μπορούσε η εφαρμογή να φέρνει κάποια χρήματα προς τον προγραμματιστή ή την ομάδα ανάπτυξης της εφαρμογής. Οι διαφημίσεις στο android είναι μια συχνή στρατηγική που χρησιμοποιείται στην ανάπτυξη εφαρμογών για android περιβάλλον.

ΚΕΦΑΛΑΙΟ 6

6.1 ΒΙΒΛΙΟΓΡΑΦΙΑ

Διαδικτυακοί Τόποι

1. Statcounter - <https://gs.statcounter.com/>
2. Hybrid apps - <https://ionic.io/resources/articles/what-is-hybrid-app-development>
3. Google developer (Activities) - <https://developer.android.com/guide/components/activities/intro-activities>
4. General Android Knowledge - <https://developer.android.com/guide/components/fundamentals>
5. Android Architecture - <https://developer.android.com/topic/>
6. Google developer (Fragments) - <https://developer.android.com/guide/fragments>
7. Android Fragments Lifecycle - <https://developer.android.com/guide/fragments/lifecycle>
8. Google developer (Android Jetpack) - <https://developer.android.com/jetpack>
9. Android Security - <https://source.android.com/docs/security>
10. Android Security features - <https://source.android.com/docs/security/features>
11. Kotlin - <https://kotlinlang.org/docs/home.html>
12. Android App architecture guide - <https://developer.android.com/topic/architecture>
13. Modularization in Android - <https://developer.android.com/topic/modularization>
14. Dependency Injection Android - <https://developer.android.com/training/dependency-injection>
15. Navigation Component - <https://developer.android.com/guide/navigation>
16. Firebase Authentication - <https://firebase.google.com/docs/auth>
17. Firebase Firestore - <https://firebase.google.com/docs/firestore>
18. Firebase Storage - <https://firebase.google.com/docs/storage/android/start>
19. Retrofit - <https://square.github.io/retrofit/>
20. Paging 3 - <https://developer.android.com/topic/libraries/architecture/paging/v3-overview>
21. Coroutines - <https://medium.com/androiddevelopers/coroutines-on-android-part-i-getting-the-background-3e0e54d20bb>
22. Coroutines - <https://developer.android.com/kotlin/coroutines>
23. Kotlin flows - <https://developer.android.com/kotlin/flow>
24. View Binding - <https://developer.android.com/topic/libraries/view-binding>
25. Data Binding - <https://developer.android.com/topic/libraries/data-binding>
26. IMDB API - <https://developer.imdb.com/documentation>
27. TMDb API - <https://developer.themoviedb.org/docs>
28. TVmaze API - <https://www.tvmaze.com/>