



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Εφαρμογή Android για Μαγειρικές Συνταγές Cooking Recipes Android Application
Όνοματεπώνυμο Φοιτητή	Ιωάννης Ματσαρόκης
Πατρώνυμο	Νικόλαος
Αριθμός Μητρώου	ΜΠΠΛ18047
Επιβλέπων	Ευθύμιος Αλέπης, Αναπληρωτής Καθηγητής

Ημερομηνία Παράδοσης: Ιούλιος 2022

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Ευθύμιος Αλέπης
Αναπληρωτής Καθηγητής

Μαρία Βίρβου
Καθηγήτρια

Κωνσταντίνος Πατσάκης
Αναπληρωτής Καθηγητής

Ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε στα πλαίσια της ολοκλήρωσης των σπουδών μου στο Πρόγραμμα Μεταπτυχιακών Σπουδών «ΠΛΗΡΟΦΟΡΙΚΗ» του Τμήματος Πληροφορικής του Πανεπιστημίου Πειραιώς.

Ευχαριστώ τον καθηγητή μου Ευθύμιο Αλέπη για την βοήθεια που μου παρείχε καθ' όλη την διάρκεια της εκπόνησης της παρούσας εργασίας, καθώς και τους καθηγητές της τριμελούς επιτροπής που δέχτηκαν να παρευρεθούν στην εξέταση της.

Οι ευχαριστίες μου απευθύνονται επίσης τόσο στον φίλο μου και συνάδερφο Φιλώτα όσο και στην κοπέλα μου Φανή για την συνεχή παρακίνησή τους καθ' όλο το χρονικό διάστημα εκπόνησης της εργασίας. Τέλος, θα ήθελα να ευχαριστήσω μέσα από την καρδιά μου τον πατέρα μου, Νίκο, την μητέρα μου, Μαρία και τον αδερφό μου Καρυοφύλλη, οι οποίοι ήταν, είναι και το ξέρω ότι θα είναι δίπλα μου σε ό,τι τους χρειαστώ.

Ιωάννης Ματσαρόκης
Πειραιάς - 2022

1. Περίληψη

Η παρούσα μεταπτυχιακή διατριβή απασκοπεί στην ανάπτυξη μιας εφαρμογής, για κινητές συσκευές με λειτουργικό σύστημα Android, η οποία θα δίνει στον χρήστη μια εύκολη και ταυτόχρονα γρήγορη πρόσβαση σε μια μεγάλη γκάμα μαγειρικών συνταγών. Οι συνταγές αυτές παρέχονται από το API Spoonacular, το οποίο αξίζει να σημειωθεί ότι προσφέρει πρόσβαση σε πάνω από 5000 συνταγές, 2600 υλικά συνταγών κ.α. Ο χρήστης ενημερώνεται σχετικά με κάποια βασικά χαρακτηριστικά της κάθε συνταγής όπως ο απαιτούμενος χρόνος προετοιμασίας της συνταγής, αν η συνταγή είναι κατάλληλη για vegan διατροφή και το πόσο likes έχει αυτή. Εκτός από το Remote Datasource(Spoonacular), η εφαρμογή χρησιμοποιεί και ως Local Datasource μια Room Database. Με αυτόν τον τρόπο ο χρήστης έχει την δυνατότητα να αποθηκεύει τοπικά τις αγαπημένες του συνταγές κι επίσης επιτυγχάνεται η συνέχιση της καλής λειτουργίας της εφαρμογής, ακόμα κι όταν χαθεί η σύνδεση της συσκευής με το διαδίκτυο. Τέλος, αναφέρεται ότι έγινε προσπάθεια ώστε η εφαρμογή να ακολουθεί την αρχιτεκτονική ModelView-View-Model (MVVM) και όλες τις τελευταίες τεχνικές ανάπτυξης Android εφαρμογών.

This master's thesis aims to develop an application for mobile devices with Android operating system, which will give the user easy and at the same time quick access to a wide range of recipes. These recipes are provided by the Spoonacular API, which is worth noting that it offers access to over 5000 recipes, 2600 recipe ingredients etc. The user is informed about some basic features of each recipe such as the required preparation time of the recipe, if the recipe is suitable for vegan diet and how many likes it has. In addition to Remote Datasource (Spoonacular), the application also uses a Room Database as Local Datasource. In this way the user has the ability to save his favorite recipes locally and also the continuation of the good operation of the application is achieved, even when the connection of the device with the internet is lost. Finally, it is reported that an attempt was made for the application to follow the ModelView-View-Model (MVVM) architecture and all the latest Android application development techniques.

Περιεχόμενα

1. Περίληψη	4
2. Εισαγωγή	7
3. Ανασκόπηση Πεδίου	8
3.1. Allrecipes Dinner Spinner	8
3.2. Cookpad	9
3.3. SideChef	10
4. Τεχνολογίες και Εργαλεία που χρησιμοποιήθηκαν	10
4.1. Android Studio	10
4.2. Γλώσσα Προγραμματισμού Kotlin	11
4.3. Spoonacular API	12
4.4. Απαιτήσεις συστήματος	13
4.5. Αρχιτεκτονική Model-View-ViewModel (MVVM)	13
4.6. Dependency Injection – Hilt Library	14
4.7. Data Binding – Data Binding Library	15
4.8. Datastore Preferences	16
4.9. Navigation Component	17
4.10. Connectivity Manager	18
4.11. RecyclerView - DiffUtil	19
4.12. Εργαλεία UI	20
4.12.1. Splash Screen	20
4.12.2. Shimmer Effect	20
4.12.3. ViewPager	21
4.12.4. MotionLayout	22
4.12.5. Dark Theme	23
5. Ανάλυση και Παρουσίαση της Εφαρμογής	23
5.1. Όνομα και SplashScreen της εφαρμογής	24
5.2. Navigation Graph	24
5.3. Recipes Fragment	25
5.4. Details Activity	29
5.4.1. OverviewFragment	29
5.4.2. IngredientsFragment	30
5.4.3. InstructionsFragment	31
5.5. FavoritesFragment	32
5.6. TriviaFragment	34
5.7. Dark Theme	36
6. Συμπεράσματα-Μελλοντικές Επεκτάσεις	38
7. Βιβλιογραφία	39

Πίνακας Εικόνων

Εικόνα 1. Mobile Apps και Μαγειρική	8
Εικόνα 2. Εφαρμογή Allrecipes Dinner Spinner	8
Εικόνα 3. Εφαρμογή Cookpad	9
Εικόνα 4. Εφαρμογή SideChef	10
Εικόνα 5. Android Studio Chipmunk 2021.2.1	11
Εικόνα 6. Το λογότυπο της γλώσσας προγραμματισμού Kotlin	11
Εικόνα 7. Το λογότυπο του Spoonacular API	12
Εικόνα 8. Η αρχιτεκτονική Model-View-ViewModel.....	14
Εικόνα 9. Dependency Injection - Hilt Library	15
Εικόνα 10. Data Binding	16
Εικόνα 11. Datastore Preferences	17
Εικόνα 12. Android Navigation Component	18
Εικόνα 13. Απώλεια σύνδεσης στο διαδίκτυο.	18
Εικόνα 14. RecyclerView	19
Εικόνα 15. Splash Screens γνωστών εφαρμογών	20
Εικόνα 16. Shimmer Effect	21
Εικόνα 17. ViewPager	22
Εικόνα 18. MotionLayout	22
Εικόνα 19. Dark Theme	23
Εικόνα 20. Το SplashScreen της εφαρμογής "Cook It Yourself".....	24
Εικόνα 21. Navigation Graph.....	25
Εικόνα 22. RecipesFragment	26
Εικόνα 23. Αποτελέσματα αναζήτησης του όρου "banana"	27
Εικόνα 24. Bottom Sheet	28
Εικόνα 25. Συνταγές Finger Food για Paleo διατροφή.....	29
Εικόνα 26. OverviewFragment	30
Εικόνα 27. IngredientsFragment	31
Εικόνα 28. InstructionsFragment.....	32
Εικόνα 29. FavoritesFragment με και χωρίς αγαπημένες συνταγές	33
Εικόνα 30. Contextual menu του FavoritesFragment	34
Εικόνα 31. TriviaFragment	35
Εικόνα 32. Κοινοποίηση του trivia	36
Εικόνα 33. RecipesFragment και BottomSheet (Dark Theme)	37
Εικόνα 34. FavoritesFragment και TriviaFragment (Dark Theme).....	37
Εικόνα 35. OverviewFragment και IngredientsFragment (Dark Theme).....	38

2. Εισαγωγή

Στον σύγχρονο τρόπο ζωής με όλες τις καθημερινές υποχρεώσεις να τρέχουν, τους ρυθμούς συνεχώς να αυξάνονται και τον ελεύθερο χρόνο να έχει περιοριστεί όσο ποτέ άλλοτε, σπάνια βρίσκει κανείς χρόνο για να μαγειρέψει ο ίδιος και συνήθως καταλήγει να τρώει junk fast food. Ωστόσο, τα πλεονεκτήματα του «σπιτικού» φαγητού είναι πολλά και αδιαμφισβήτητα. Αρχικά μαγειρεύοντας στο σπίτι, ο καθένας είναι σε θέση να έχει πλήρη επίγνωση των υλικών καθώς και των ποσοτήτων των διάφορων μπαχαρικών που περιέχει το φαγητό του. Πράγμα που σημαίνει πιο υγιεινά γεύματα. Επιπρόσθετα, η μαγειρική στο σπίτι είναι φιλική προς το πορτοφόλι του καταναλωτή, αφού αν κανείς τρώει κάθε μέρα από έξω σίγουρα θα χρεώνεται πολλά περισσότερα. Επίσης η μαγειρική αποτελεί σημαντική δεξιότητα κάποιου και με την εξάσκηση μπορεί κανείς να εξελιχθεί ικανοποιητικά. Τέλος, η διαδικασία της προετοιμασίας ενός γεύματος μπορεί να λειτουργήσει και ψυχοαναληπτικά, αφού βοηθάει στο να ξεφεύγουμε από τους γρήγορους ρυθμούς της καθημερινότητας. Αλλά πώς να επιβιβαστεί κανείς σε αυτό το ταξίδι που λέγεται μαγειρική, όταν δεν ξέρει από που να ξεκινήσει; Και σε αυτό ακριβώς το ερώτημα έρχονται να δώσουν λύση οι εφαρμογές μαγειρικών συνταγών για κινητές συσκευές.

Αυτές οι εφαρμογές είναι τόσο βοηθητικές, αφού μπορείς να βρεις πολλούς και διαφορετικούς τύπους μαγειρικών συνταγών συγκεντρωμένες σε ένα μέρος. Πριν την εξέλιξη της τεχνολογίας και την ένταξη των mobile apps στις ζωές όλων, τα βιβλία μαγειρικής αποτελούσαν τον κανόνα. Αλλά σήμερα, δεν είναι απαραίτητο να ξεφυλλίζεις σελίδες μέχρι να βρεις την συνταγή για την οποία ενδιαφέρεσαι. Απλά γράφεις μια μόνο λέξη στο κελί της αναζήτησης μιας εφαρμογής μαγειρικής και έχεις το αποτέλεσμα μπροστά σου.

Αντικείμενο της παρούσας μεταπτυχιακής διατριβής είναι η ανάπτυξη μιας τέτοιας εφαρμογής, η οποία προορίζεται για κινητές συσκευές με λειτουργικό σύστημα Android. Ο χρήστης της εφαρμογής θα έχει πρόσβαση σε πάνω από 5000 συνταγές, οι οποίες προέρχονται από το Spoonacular API. Πιο συγκεκριμένα, για κάθε συνταγή θα παρέχονται στοιχεία, που θα βοηθούν τον χρήστη να επιλέξει την εκάστοτε συνταγή εφόσον του ταιριάζει. Πέρα από την δυνατότητα αναζήτησης συνταγών, παρέχεται η δυνατότητα της εμφάνισης συνταγών ανάλογα με τον τύπο του γεύματος (πρωινό, κυρίως γεύμα, ορεκτικό, ποτό, συνοδευτικό, επιδόρπιο, σνακ κτλ) και το επιθυμητό είδος διατροφής (vegetarian, vegan, paleo, pescetarian, whole30, ketogenic κτλ). Ακόμα ο χρήστης μπορεί να εισάγει τις συνταγές που του αρέσουν στην των αγαπημένων του συνταγών, στην οποία έχει πρόσβαση ακόμα και στην περίπτωση που η κινητή του συσκευή δεν διαθέτει σύνδεση στο διαδίκτυο. Επιπρόσθετα, υπάρχει και το trivía κομμάτι της εφαρμογής, το οποίο κάθε φορά που το ανοίγει ο χρήστης του εμφανίζει ένα αξιοπερίεργο σχετικά πάντα με κάτι που τρώγεται.

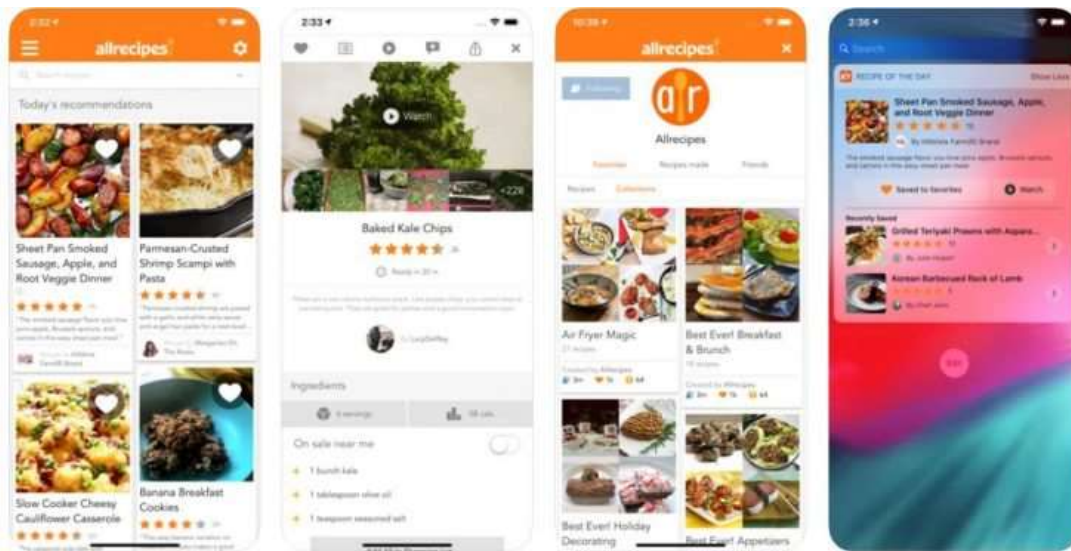
3. Ανασκόπηση Πεδίου

Σε αυτό το κεφάλαιο θα γίνει μια σύντομη παρουσίαση παρόμοιων εφαρμογών μαγειρικών συνταγών. Όπως είναι αναμενόμενο στον χώρο υπάρχουν αρκετές τέτοιες εφαρμογές, οι οποίες εύκολα μπορούν να αναζητηθούν και να εγκαταστηθούν από το Google Playstore. Το αρκετά μεγάλο πλήθος αυτών των εφαρμογών αποτελεί ακόμα ένα στοιχείο που καταδεικνύει την χρησιμότητά τους. Επίσης αυτές η εφαρμογές αξιοποιούν την ευχρηστία των σύγχρονων κινητών συσκευών.



Εικόνα 1. Mobile Apps και Μαγειρική

3.1. Allrecipes Dinner Spinner



Εικόνα 2. Εφαρμογή Allrecipes Dinner Spinner

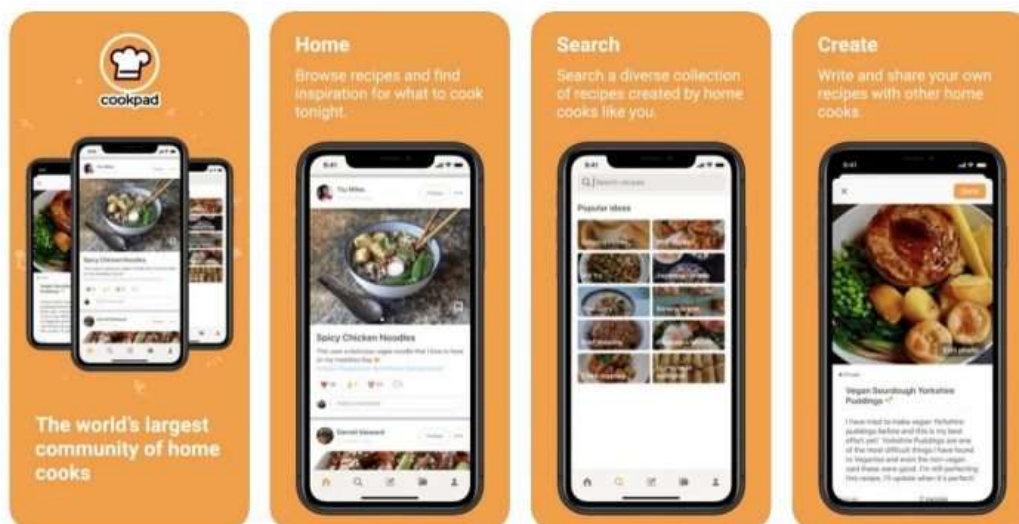
Αυτή η εφαρμογή συγκεντρώνει συνταγές από μια κοινότητα που αποτελείται από περισσότερα από 50 εκατομμύρια άτομα, πράγμα που σημαίνει ότι όλες οι συνταγές έχουν μια προσωπική αίσθηση. Αυτά τα άτομα συνεισφέρουν φωτογραφίες, αξιολογήσεις και κριτικές, μαζί με συνταγές.

Με το Allrecipes, μπορεί κανείς να γράψει μια λίστα αγορών, να αποθηκεύσει όποιες συνταγές του αρέσουν ή να αναζητήσει συγκεκριμένες συνταγές που έχει στο μυαλό του. Μπορεί επίσης να φιλτράρει τα αποτελέσματα της αναζήτησής του με βάση το χρόνο μαγειρέματος και τη διατροφή. Το Allrecipes επιτρέπει επίσης να φιλτράρονται τυχόν τρόφιμα που δεν αρέσουν στον χρήστη.

Εφαρμογή Android για Μαγειρικές Συνταγές

Ένα χαρακτηριστικό γνώρισμα αυτής της εφαρμογής συνταγών, ωστόσο, είναι ότι προσφέρει την δυνατότητα ελέγχου των προϊόντων, που πωλούνται σε κοντινή απόσταση. Επίσης, δίνει πληροφορίες για τις φθηνότερες συνταγές που μπορεί κανείς να φτιάξει μια συγκεκριμένη μέρα. Επιπλέον, έχει συνεργασίες με ορισμένα καταστήματα, έτσι ώστε, όταν ο χρήστης της εφαρμογής μπει σε ένα από αυτά, η εφαρμογή θα του δώσει προτάσεις συνταγών.

3.2. Cookpad



Εικόνα 3. Εφαρμογή Cookpad

Όταν κανείς εγγράφεται στο Cookpad, του δίνεται ένα προφίλ που μπορεί να χρησιμοποιήσει για να περιηγηθεί σε συνταγές που έχουν ανεβάσει άλλοι χρήστες. Η περιήγηση γίνεται εύκολη αφού οι συνταγές είναι οργανωμένες σε διαφορετικές κατηγορίες. Το Cookpad επιτρέπει επίσης την παρακολούθηση του Cookrplan του χρήστη χωρίς να χρειάζεται σύνδεση στο Διαδίκτυο.

Παλαιότερα γνωστό ως All the Cooks, το Cookpad είναι μια από τις εφαρμογές συνταγών που ενσωματώνει απρόσκοπτα μια κοινωνική πτυχή στη μαγειρική. Ο χρήστης της εφαρμογής έχει την δυνατότητα να ανεβάσει μόνος του μια συνταγή, συμπληρωμένη με φωτογραφίες που θέλει να συμπεριλάβει. Άλλοι χρήστες έχουν επίσης τη δυνατότητα να κάνουν like και να σχολιάσουν τις συνταγές άλλων.

Το Cookpad είναι σε μεγάλο βαθμό μια εφαρμογή συνταγών που βασίζεται στην κοινότητα, αλλά αυτό δεν είναι απαραίτητα κακό. Εξάλλου, η σύνδεση και η επικοινωνία με άλλους ενθουσιώδεις ανθρώπους που μοιράζονται το ίδιο πάθος είναι ένας πολύ καλός τρόπος για να βελτιωθούν οι δεξιότητες όλων.

3.3. SideChef



Εικόνα 4. Εφαρμογή SideChef

Το SideChef είναι μια εφαρμογή που απευθύνεται ιδιαίτερα σε όσους είναι νέοι στον κόσμο της μαγειρικής. Με αυτήν την εφαρμογή συνταγών, μπορεί κανείς να περιηγηθεί σε περισσότερες από 11.000 συνταγές, τις οποίες μπορεί να αναζητήσει ανά συστατικό ή κατηγορία. Κάθε συνταγή συνοδεύεται από εύκολα βήματα, βίντεο και φωτογραφίες. Ο χρήστης της εφαρμογής μπορεί ακόμη να χρησιμοποιήσει χρονοδιακόπτες και φωνητικές εντολές για να κάνει τις προετοιμασίες ευκολότερες.

Εάν ο επίδοξος σεφ βρει μια συνταγή που θέλει να φτιάξει, αλλά είναι υπερβολική για ένα άτομο, μπορεί να προσαρμόσει το μέγεθος της μερίδας και η εφαρμογή θα αλλάξει αυτόματα τα συστατικά στις κατάλληλες αναλογίες και ποσότητες. Μπορεί επίσης να αυξήσει το μέγεθος της μερίδας. Επιπλέον, μπορεί κανείς να αποθηκεύσει συνταγές και ακόμη και να τις συγκεντρώσει σε μια λίστα που μπορεί στη συνέχεια να στείλει μέσω email στον εαυτό του. Ίσως το πιο ελκυστικό χαρακτηριστικό, ωστόσο, είναι ότι μπορεί να αγοράσει συστατικά απευθείας από την εφαρμογή. Μετά την αγορά, τα συστατικά θα παραδοθούν απευθείας στον χώρο του.

Όπως και με μερικές άλλες εφαρμογές συνταγών, το SideChef ενσωματώνει επίσης τη μαγειρική με την κοινωνικοποίηση. Ο χρήστης έχει την δυνατότητα να ακολουθήσει διάσημους σεφ, food bloggers ή άλλα άτομα που γνωρίζει. Μπορεί επίσης να μοιραστεί φωτογραφίες από τα πιάτα που έχει ετοιμάσει.

4. Τεχνολογίες και Εργαλεία που χρησιμοποιήθηκαν

Σε αυτήν την ενότητα θα γίνει μια παρουσίαση των βασικότερων εργαλείων και τεχνολογιών, που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής. Ακόμα θα γίνει αναφορά στις ελάχιστες απαιτήσεις του συστήματος, προκειμένου να τρέξει με επιτυχία η εφαρμογή.

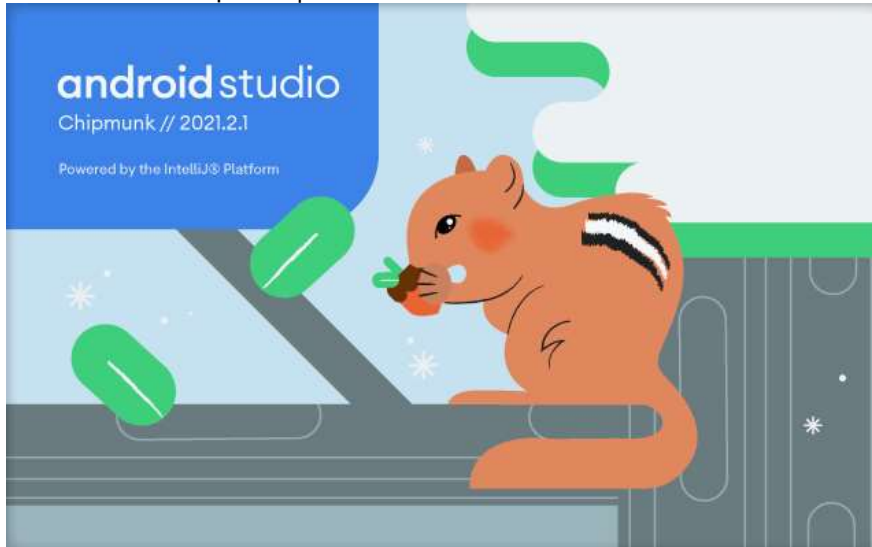
4.1. Android Studio

Το Android Studio είναι ένα ολοκληρωμένο προγραμματιστικό περιβάλλον (IDE) για ανάπτυξη εφαρμογών στην πλατφόρμα Android. Ανακοινώθηκε στις 16 Μαΐου 2013 στο συνέδριο Google I/O από την Google Product Manager, Katherine Chou. Το Android Studio είναι διαθέσιμο ελεύθερα με την άδεια Apache License 2.0.

Το Android Studio ήταν διαθέσιμο σε πρώιμο στάδιο για προεπισκόπηση ξεκινώντας από την έκδοση 0.1 τον Μάιο του 2013, έπειτα ξεκίνησε το δοκιμαστικό στάδιο από την έκδοση 0.8 που βγήκε τον Ιούνιο του 2014. Η πρώτη σταθερή έκδοση βγήκε το Δεκέμβριο του 2014, με την έκδοση 1.0.

Βασισμένο στο λογισμικό της JetBrains' IntelliJ IDEA, το Android Studio σχεδιάστηκε αποκλειστικά για προγραμματισμό Android. Είναι διαθέσιμο για Windows, Mac OS X και Linux, και αντικατέστησε τα Eclipse Android Development Tools (ADT) ως το κύριο IDE της Google για ανάπτυξη εφαρμογών Android.

Για την ανάπτυξη της εφαρμογής της παρούσας μεταπτυχιακής διατριβής χρησιμοποιήθηκε η έκδοση Android Studio Chipmunk|2021.2.1.



Εικόνα 5. Android Studio Chipmunk|2021.2.1

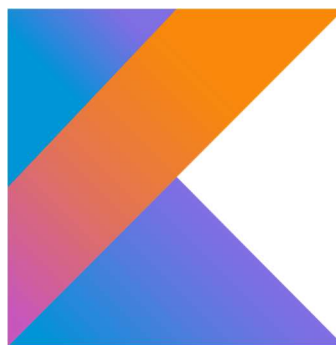
4.2. Γλώσσα Προγραμματισμού Kotlin

Η Kotlin είναι μια αντικειμενοστρεφής γλώσσα προγραμματισμού που σχεδιάστηκε από την ομάδα JetBrains.

Η γλώσσα Kotlin τρέχει πάνω στην Εικονική Μηχανή της Java (JVM). Έχει επηρεαστεί από τις γλώσσες Java, Scala, C#, Groovy, είναι συμβατή με την Java και το Android, έχει σχεδιαστεί για να τρέχει σαν εγγενής κώδικας στα iOS και macOS. Απο τον Φεβρουάριο του 2012 η Kotlin αναπτύσσεται ως γλώσσα ανοικτού κώδικα. Η πρώτη επίσημη έκδοση 1.0 δημοσιεύτηκε τον Φεβρουάριο του 2016. Τον Μάιο του 2017 η Google ανακοίνωσε ότι η Kotlin θα είναι επίσημη γλώσσα του Android.

Η Kotlin αξιοποιεί τις υπάρχουσες βιβλιοθήκες για την Εικονική Μηχανή της Java (JVM), για Android και web browser. Ένα πρόγραμμα Kotlin μπορεί να αναπτυχθεί σε Ολοκληρωμένο περιβάλλον ανάπτυξης IntelliJ IDEA, Android Studio, Eclipse, NetBeans, Visual Studio Code. Ο κώδικας Kotlin μπορεί να περιέχει κώδικα Java και ο κώδικας Java να περιέχει Kotlin. Ακόμα ο κώδικας Java μπορεί να μετατραπεί σε κώδικα Kotlin.

Η έκδοση της γλώσσας που χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής είναι η Kotlin 1.6.0.



Εικόνα 6. Το λογότυπο της γλώσσας προγραμματισμού Kotlin

4.3. Spoonacular API

Ένα άλλο από τα βασικά εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής είναι το Spoonacular API.

Μια διεπαφή προγραμματισμού εφαρμογών (ή αλλιώς API) είναι μια σύνδεση μεταξύ υπολογιστών ή μεταξύ προγραμμάτων υπολογιστή. Είναι ένας τύπος διεπαφής λογισμικού, που προσφέρει υπηρεσία σε άλλα κομμάτια λογισμικού.

Σε αντίθεση με μια διεπαφή χρήστη, η οποία συνδέει έναν υπολογιστή με ένα άτομο, μια διεπαφή προγραμματισμού εφαρμογών συνδέει υπολογιστές ή κομμάτια λογισμικού μεταξύ τους. Δεν προορίζεται να χρησιμοποιηθεί απευθείας από άτομο (τον τελικό χρήστη) εκτός από προγραμματιστή υπολογιστή που το ενσωματώνει στο λογισμικό. Ένα API αποτελείται συχνά από διαφορετικά μέρη που λειτουργούν ως εργαλεία ή υπηρεσίες που είναι διαθέσιμα στον προγραμματιστή. Ένα πρόγραμμα ή ένας προγραμματιστής που χρησιμοποιεί ένα από αυτά τα μέρη λέγεται ότι καλεί αυτό το τμήμα του API. Οι κλήσεις που αποτελούν το API είναι επίσης γνωστές ως υπορουτίνες, μέθοδοι, αιτήματα ή τελικά σημεία.



Εικόνα 7. Το λογότυπο του Spoonacular API

Ένα τέτοιο API είναι και αυτό του Spoonacular. Το συγκεκριμένο API αφορά μαγειρικές συνταγές, φαγητά, προϊόντα, υλικά συνταγών, όπως επίσης εμπλέκει και διατροφολογικά στοιχεία (είδη διατροφής, αλλεργιογόνα συστατικά κτλ.) Πιο συγκεκριμένα μέσω του Spoonacular API κανείς έχει πρόσβαση σε:

- Συστατικά συνταγών
 - Διατροφικά δεδομένα
 - Τιμές
 - Μαγειρικές συμβουλές
 - Πληροφορίες για την υγεία
 - Υποκατάστατα
 - Παραλλαγές
- Μαγειρικές συνταγές
 - Διατροφική ανάλυση
 - Ανάλυση κόστους
 - Σχετικές συνταγές
 - Σημασιολογική αναζήτηση
 - Περιεχόμενο διαθέσιμο για αγορά
- Προϊόντα
 - Ανάλυση συστατικών
 - Διατροφικά δεδομένα
 - Διατροφική Οπτικοποίηση
 - Περιγραφές
 - Συσχετισμός προϊόντων
 - Αναζήτηση προϊόντων

Με όλα αυτά κι άλλα πολλά το Spoonacular API αποτελεί κυριολεκτικά την πηγή όλης της πληροφορίας και των δεδομένων σχετικά με τις μαγειρικές συνταγές της εφαρμογής μας.

4.4. Απαιτήσεις συστήματος

Οι απαιτήσεις της εφαρμογής είναι ένα smartphone με λειτουργικό σύστημα Android του οποίου η έκδοση του Software Development Kit (SDK) να είναι 31. Η έκδοση του SDK δύναται να είναι και μικρότερη από 31, αλλά θα πρέπει να είναι απαραίτητα μεγαλύτερη από 24. Επίσης ενδείκνυται η συσκευή να διαθέτει τουλάχιστον 2GB μνήμη RAM και να έχει πρόσβαση στο διαδίκτυο (τουλάχιστον για την πρώτη φορά που θα τρέξει η εφαρμογή στην εκάστοτε συσκευή).

4.5. Αρχιτεκτονική Model-View-ViewModel (MVVM)

Model–View–Viewmodel (MVVM) είναι ένα αρχιτεκτονικό μοτίβο λογισμικού που διευκολύνει τον διαχωρισμό της ανάπτυξης της γραφικής διεπαφής χρήστη (View) – είτε μέσω μιας γλώσσας σήμανσης είτε μέσω ενός κώδικα GUI – από την ανάπτυξη της business logic ή το back-end logic (Model), έτσι ώστε το επίπεδο View να μην εξαρτάται από κάποια συγκεκριμένη πλατφόρμα μοντέλου. Το επίπεδο του ViewModel του MVVM είναι ένας μετατροπέας τιμών, που σημαίνει ότι το ViewModel είναι υπεύθυνο για την έκθεση (μετατροπή) των αντικειμένων δεδομένων από το επίπεδο του Model με τέτοιο τρόπο ώστε τα αντικείμενα να διαχειρίζονται και να παρουσιάζονται εύκολα. Από αυτή την άποψη, το ViewModel είναι περισσότερο Model παρά View και χειρίζεται το μεγαλύτερο μέρος, αν όχι όλη τη λογική εμφάνισης του View. Το ViewModel μπορεί να εφαρμόσει ένα μοτίβο διαμεσολαβητή, οργανώνοντας την πρόσβαση στην back-end λογική γύρω από το σύνολο των use cases που υποστηρίζονται από το επίπεδο του View.

Τα κυριότερα πλεονεκτήματα από την υιοθέτηση της αρχιτεκτονικής MVVM είναι τα εξής:

- Συντηρησιμότητα

Ο καθαρός διαχωρισμός των διαφορετικών ειδών του κώδικα διευκολύνει την πρόσβαση σε ένα ή περισσότερα από τα εστιασμένα κομμάτια και την πραγματοποίηση αλλαγών σε αυτά, χωρίς ιδιαίτερα προβλήματα. Αυτό προσφέρει προγραμματιστική ευελιξία και την γρήγορη μετάβαση σε νέες εκδόσεις της εφαρμογής.

- Δυνατότητα testing

Μέσω της αρχιτεκτονικής MVVM, το κάθε κομμάτι κώδικα είναι πιο στοιχειώδες, κι εφόσον αυτό έχει γραφεί σωστά, τα εξωτερικά και εσωτερικά dependencies βρίσκονται σε διακριτά μέρη κώδικα από τα προς τεστάρισμα κομμάτια λογικής. Αυτό κάνει πολύ πιο εύκολη την συγγραφή μονάδων για τεστάρισμα.

- Επεκτασιμότητα

Πολλές φορές επικαλύπτεται με την συντηρησιμότητα, λόγω των καθαρών ορίων διαχωρισμού και των πιο λεπτομερών κομματιών κώδικα. Δίνεται η δυνατότητα της επαναχρησιμοποίησης οποιουδήποτε από τα εξαρτήματα, όπως επίσης και η δυνατότητα της αντικατάστασης ή προσθήκης κομματιών κώδικα, τα οποία πραγματοποιούν παρόμοιες λειτουργίες στα κατάλληλα σημεία της αρχιτεκτονικής.

Κατά την ανάπτυξη της εφαρμογής της παρούσας διπλωματικής διατριβής, υιοθετήθηκε η αρχιτεκτονική MVVM. Ακολουθεί μια σύντομη παρουσίασή της.

Αρχικά, όπως έχει ήδη αναφερθεί, η πηγή όλων των πληροφοριών σχετικά με τις μαγειρικές συνταγές που περιέχει η εφαρμογή, προέρχεται από το Spoonacular API. Έχουμε πρόσβαση σε όλα τα απαραίτητα δεδομένα που υπάρχουν εκεί χρησιμοποιώντας την βιβλιοθήκη του Retrofit. Η παραπάνω βιβλιοθήκη μας προσφέρει όλες εκείνες τις αναγκαίες συναρτήσεις μέσω των οποίων πραγματοποιούμε τα GET requests μας στο Spoonacular και παίρνουμε ως απάντηση δεδομένα. Όλο αυτό το κομμάτι αποτελεί και το Remote Data Source της εφαρμογής.

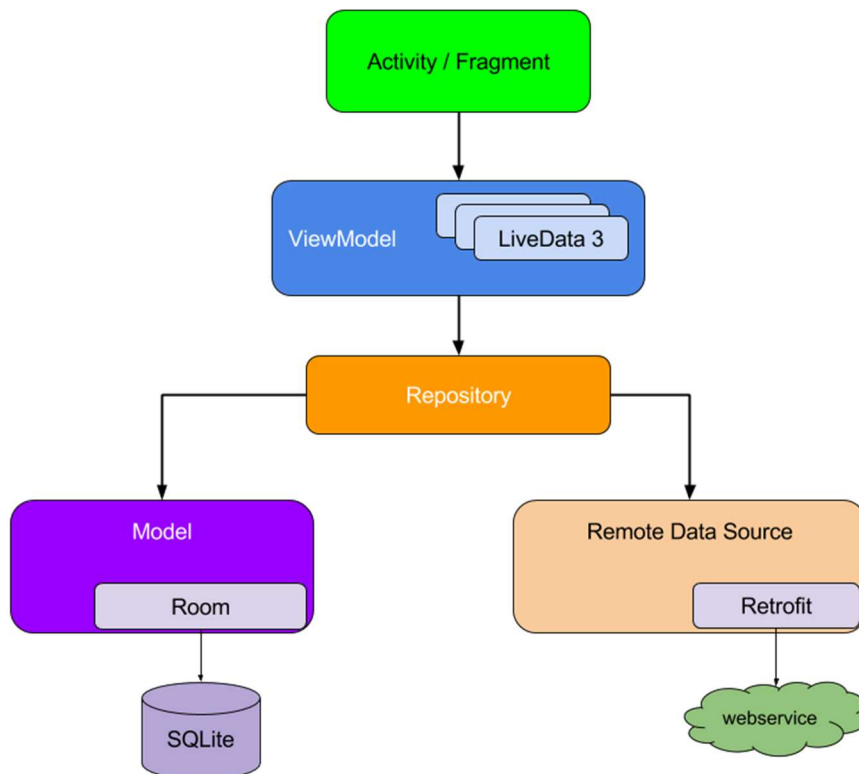
Αφού έχουμε πάρει τα δεδομένα στα χέρια μας, τα αποθηκεύουμε τοπικά στην ROOM Database που έχουμε υλοποιήσει για αυτόν τον λόγο. Συγκεκριμένα στην ROOM Database αποθηκεύονται όλες οι αγαπημένες συνταγές μας και παραμένουν εκεί μέχρι να τις διαγράψει ο χρήστης της εφαρμογής. Ακόμα εκεί αποθηκεύονται προσωρινά και οι εκάστοτε συνταγές που εμφανίζονται στην αρχική οθόνη της εφαρμογής, όσο και τα «αξιοπεριέργα» της οθόνης των trivía.

Με αυτόν τον τρόπο επιτυγχάνεται η απαραίτητη λειτουργία της εφαρμογής ακόμα και στην ατυχή περίπτωση που η κινητή συσκευή μας χάσει την σύνδεσή της στο διαδίκτυο. Όπως είναι λογικό η ROOM Database αποτελεί ουσιαστικά την Local Data Source της εφαρμογής.

Σε αυτό το σημείο αξίζει να τονιστεί ότι, σύμφωνα πάντα με την αρχιτεκτονική MVVM, η Remote Data Source και η Local Data Source εγκολπώνονται στο εσωτερικό του Repository. Αυτή η μέθοδος έχει ακολουθηθεί και κατά την ανάπτυξη της παρούσας εφαρμογής. Το Repository μαζί με την Remote Data Source και την Local Data Source αποτελούν το επίπεδο του Model της αρχιτεκτονικής MVVM.

Όσο αφορά το επίπεδο του ViewModel, αυτό στην εφαρμογή έχει υλοποιηθεί μέσω δύο κλάσεων (MainViewModel και RecipesViewModel). Αυτές οι δύο κλάσεις αποτελούν ουσιαστικά ένα είδος γέφυρας μεταξύ του επιπέδου του Model και του επιπέδου του View. Πιο συγκεκριμένα, αλληλεπιδρούν με το Model, προσπελαίνουν τα δεδομένα του, τα επεξεργάζονται καταλλήλως ώστε να τα φέρουν στην επιθυμητή μορφή και τέλος εκθέτουν αυτά τα δεδομένα προς το επίπεδο του View. Εδώ αξίζει να σημειωθεί ότι μέσω της χρήσης Kotlin Flows και LiveData επιτεύχθηκε η λεγόμενη lifecycle awareness των αντικειμένων και με αυτόν τον τρόπο έγινε ακόμα πιο αποτελεσματική χρήση των πόρων.

Τέλος, το ανώτερο επίπεδο της αρχιτεκτονικής MVVM, που δεν είναι άλλο από το επίπεδο View, αποτελείται ουσιαστικά από όλα τα Activities, τα Fragments και τα αρχεία XML της εφαρμογής. Είναι το επίπεδο με το οποίο αλληλεπιδρά άμεσα ο χρήστης. Ακόμα στα καθήκοντα αυτού του επιπέδου συγκαταλέγεται και η μεταφορά των ενεργειών του χρήστη προς το επίπεδο του ViewModel. Όλα τα παραπάνω φαίνονται πολύ πιο αναπαραστατικά παρακάτω στην Εικόνα 8.



Εικόνα 8. Η αρχιτεκτονική Model-View-ViewModel

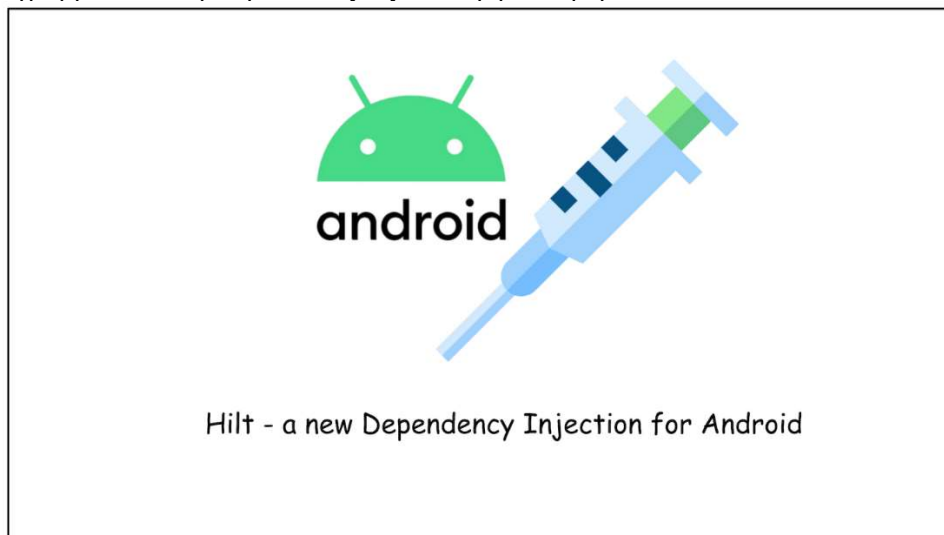
4.6. Dependency Injection – Hilt Library

Μία έννοια, η οποία συνοδεύει σχεδόν πάντα την υιοθέτηση της αρχιτεκτονικής MVVM και χρήση της οποίας έγινε κατά την ανάπτυξη της εφαρμογής της παρούσας διπλωματικής διατριβής, είναι αυτή του Dependency Injection. Το Dependency Injection αποτελεί μια προγραμματιστική τεχνική,

σύμφωνα με την οποία παρέχονται σε κάθε αντικείμενο, όλα τα instances των αντικειμένων, τα οποία του είναι απαραίτητα. Στόχος του Dependency Injection είναι ο διαχωρισμός της έννοιας της κατασκευής ενός αντικειμένου και της έννοιας της χρήσης του. Μέσω αυτής της τεχνικής το κάθε αντικείμενο δεν χρειάζεται να ξέρει το πως θα κατασκευάσει ένα instance ενός άλλου αντικειμένου που χρειάζεται. Αντ' αυτού, το instance του δεύτερου αντικειμένου (μαζί με όλα τα δικά του dependencies) παρέχεται στο πρώτο αντικείμενο μέσω εξωτερικού κώδικα, τον οποίο το αντικείμενο δεν γνωρίζει. Μέσω του Dependency Injection επιτυγχάνονται τα εξής:

- Μια κλάση μπορεί να είναι ανεξάρτητη από την δημιουργία των αντικειμένων τα οποία της είναι απαραίτητα.
- Μια εφαρμογή και τα αντικείμενα, τα οποία χρησιμοποιεί, υποστηρίζουν διαφορετικές διαμορφώσεις.
- Η συμπεριφορά ενός κομματιού κώδικα μπορεί να αλλάξει, χωρίς να τον επεξεργαστούμε άμεσα.

Το μειονέκτημα του manual Dependency Injection έγκειται στο γεγονός ότι ο προγραμματιστής υποχρεούται να κατασκευάσει κάθε κλάση και τα dependencies της γράφοντάς τα με το χέρι και να χρησιμοποιήσει έναν container προκειμένου να μπορέσει να επαναχρησιμοποιήσει και να διαχειριστεί τα dependencies. Κι εδώ έρχεται να διευκολύνει το έργο των προγραμματιστών η Dependency Injection βιβλιοθήκη του Hilt.



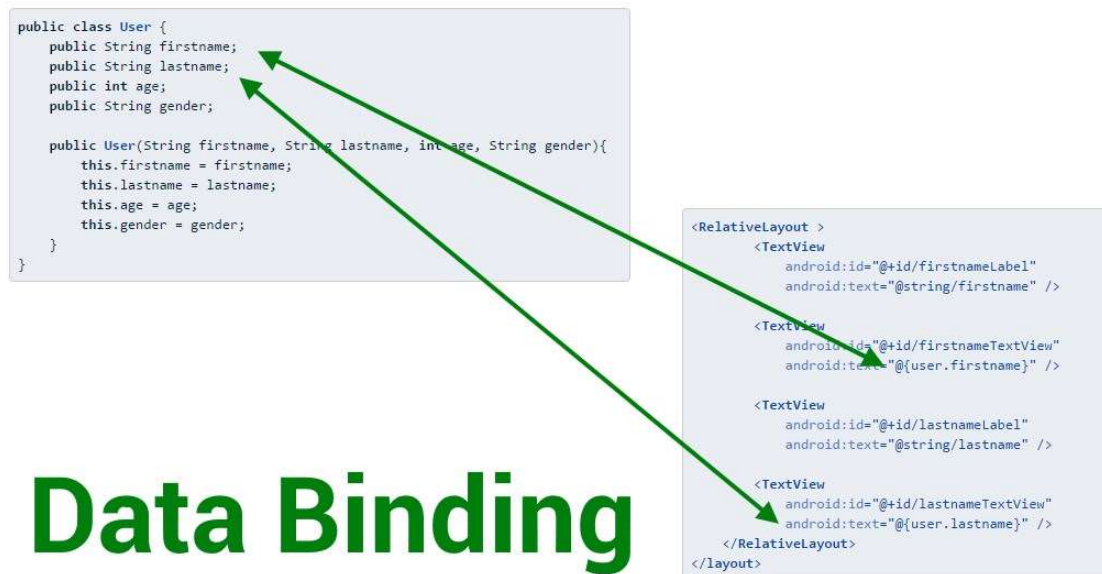
Εικόνα 9. Dependency Injection - Hilt Library

Η βιβλιοθήκη Hilt αναφέρεται στο Android και μειώνει κατά μεγάλο βαθμό τετριμμένο κώδικα, ο οποίος έχει γραφεί manual και πολλές φορές μπορεί και να επαναλαμβάνεται. Μέσω της Hilt παρέχεται ένας τυπικός τρόπος εφαρμογής του Dependency Injection στην εφαρμογή, αφού διατίθενται οι containers για κάθε κλάση του project και διαχειρίζονται αυτόματα τα lifecycles τους. Η Hilt είναι χτισμένη με θεμέλια, την γνωστή βιβλιοθήκη για Dependency Injection, την Dagger. Με αυτόν τον τρόπο, επωφελείται από την ορθότητα σε χρόνο μεταγλώττισης, την απόδοση χρόνου εκτέλεσης, την επεκτασιμότητα και την υποστήριξη στο Android Studio, που παρέχει η Dagger.

4.7. Data Binding – Data Binding Library

Ακόμη μία συχνά χρησιμοποιούμενη στον προγραμματισμό τεχνική, της οποίας χρήση έγινε και στην εφαρμογή των μαγειρικών συνταγών, είναι αυτή του Data Binding. Το Data Binding αποτελεί μια τεχνική, η οποία όπως λέει και το όνομά της, συνδέει και συγχρονίζει δεδομένα μεταξύ ενός consumer (καταναλωτής δεδομένων) κι ενός provider (πάροχος δεδομένων). Χάρη στο Data Binding, κάθε αλλαγή που συμβαίνει στα δεδομένα αντανakλάται αυτόματα σε εκείνα τα στοιχεία, τα οποία είναι συνδεδεμένα με τα δεδομένα που άλλαξαν. Για την διευκόλυνση του έργου των

προγραμματιστών όσο αφορά το Data Binding έχει αναπτυχθεί η βιβλιοθήκη Data Binding, χρήση της οποίας έγινε και στην εφαρμογή.



Εικόνα 10. Data Binding

Η βιβλιοθήκη Data Binding αποτελεί μια υποβοηθητική βιβλιοθήκη, η οποία επιτρέπει την σύνδεση των User Interface στοιχείων των layouts αρχείων με τα data sources της εφαρμογής, χρησιμοποιώντας μια δηλωτική μορφή κι όχι προγραμματιστικά. Κατ' αυτόν τον τρόπο επιτυγχάνονται τα εξής:

- Αποφυγή της κλήσης της μεθόδου findViewById(), εξασφαλίζοντας μεγαλύτερη ταχύτητα εκτέλεσης.
- Διευκόλυνση της σύνδεσης των δεδομένων με τα views των XML αρχείων.

4.8. Datastore Preferences

Συχνά σε μια εφαρμογή για κινητές συσκευές, υπάρχει η απαίτηση της αποθήκευσης μικρών ή απλών data sets. Για παράδειγμα, στην εφαρμογή της παρούσας διπλωματικής διατριβής, υπάρχει η ανάγκη της αποθήκευσης των επιλογών του χρήστη όσο αφορά τον τύπο γεύματος και το είδος της διατροφής στα οποία ανήκουν οι μαγειρικές συνταγές που εμφανίζονται στην οθόνη. Με αυτόν τον τρόπο εξασφαλίζεται ότι την επόμενη φορά που ο χρήστης θα ανοίξει την παρούσα εφαρμογή, οι επιλογές που είχε κάνει την τελευταία φορά θα βρίσκονται εκεί. Αυτό γίνεται συνήθως με χρήση των shared preferences. Ωστόσο η Google εισήγαγε έναν καινούριο τρόπο για να επιτύχει κανείς αυτό, τα Datastore preferences, τα οποία μελλοντικά πρόκειται να αντικαταστήσουν τα shared preferences. Τα Datastore preferences χρησιμοποιήθηκαν, λοιπόν, και στην συγκεκριμένη εφαρμογή.

Το Datastore δεν είναι κάτι άλλο από μια αποθήκη δεδομένων, τα οποία αποθηκεύονται με την μορφή ζευγών κλειδιού-τιμής, όπως ακριβώς και στα shared preferences. Ωστόσο η διαφορά στα Datastore preferences έγκειται στο γεγονός ότι γίνεται χρήση της Kotlin, των coroutines και των flows, για να διαχειριστεί τις λειτουργίες της εγγραφής και της ανάγνωσης σε διαφορετικά threads. Σε αντίθεση με τα shared preferences, ο προγραμματιστής μπορεί να λάβει αντικείμενα της Kotlin αντί για JSON strings, όταν διαβάσει δεδομένα, πράγμα που κάνει πολύ πιο εύκολο το να δουλεύει με το Datastore παρά με τα shared preferences.



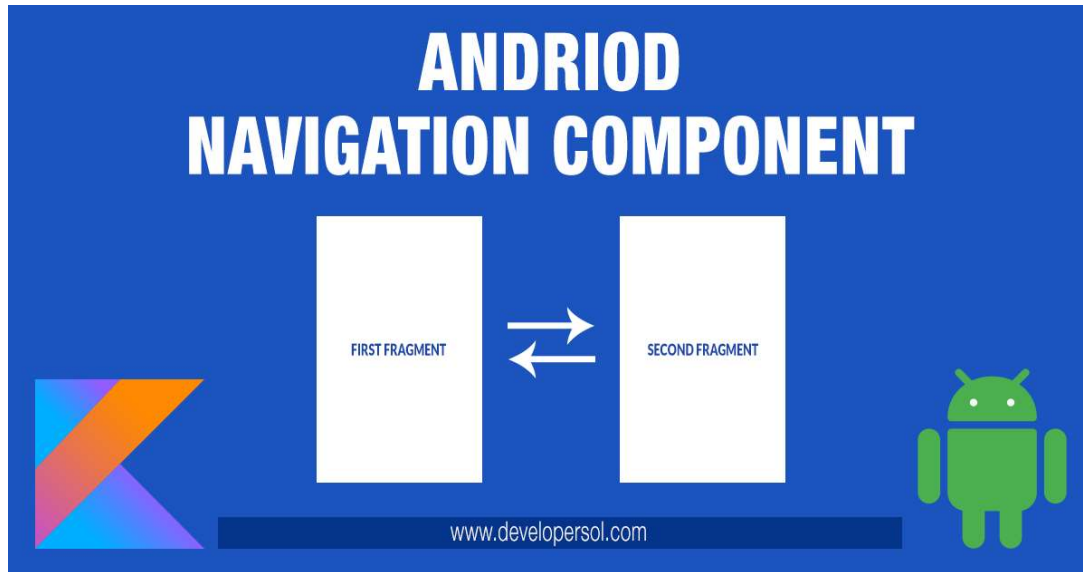
Εικόνα 11. Datastore Preferences

Η χρήση των Datastore preferences έχει αρκετά πλεονεκτήματα σε σχέση με τα shared preferences. Μερικά από αυτά είναι τα εξής:

- Τα shared preferences τρέχουν στο main thread, ενώ τα Datastore preferences τρέχουν σε ξεχωριστό thread, πράγμα που τα καθιστά thread-safe.
- Η χρήση των Datastore preferences είναι πιο εύκολη από την αντίστοιχη των shared preferences.
- Τα shared preferences δεν έχουν μηχανισμό σηματοδότησης σφαλμάτων, ενώ το Datastore αποθηκεύει δεδομένα ασύγχρονα και με συνέπεια.

4.9. Navigation Component

Η πλοήγηση (navigation) αναφέρεται στις αλληλεπιδράσεις, οι οποίες επιτρέπουν τους χρήστες των εφαρμογών να πλοηγούνται στα διάφορα κομμάτια (Activities, Fragments κτλ) της εκάστοτε εφαρμογής. Το Navigation Component είναι αυτό το οποίο διευκολύνει τον προγραμματιστή να υλοποιήσει την πλοήγηση από τα απλά clicks μέχρι και τα πιο σύνθετα σχήματα. Ακόμα το Navigation Component διασφαλίζει συνέπεια και προβλεψιμότητα στην εμπειρία του χρήστη τηρώντας ένα καθιερωμένο σύνολο αρχών. Επιπρόσθετα με τον παραπάνω τρόπο ο προγραμματιστής μπορεί πολύ εύκολα να οπτικοποιήσει την ροή πλοήγησης της εφαρμογής του σε ένα σημείο.



Εικόνα 12. Android Navigation Component

Η βιβλιοθήκη του Navigation Component παρέχει επιπλέον τα εξής πλεονεκτήματα:

- Αυτόματος χειρισμός της εναλλαγής μεταξύ των Fragments της εφαρμογής.
- Εξ' ορισμού οι ενέργειες up και back χειρίζονται σωστά.
- Προκαθορισμένα animation κατά την μετάβαση από μια οθόνη σε μια άλλη.
- Υλοποίηση UI μοτίβων πλοήγησης (π.χ. bottom navigation) με ελάχιστο επιπλέον κώδικα.
- Εξασφάλιση type safety για τα δεδομένα που ανταλλάσσονται κατά την πλοήγηση.

Λόγω όλων των παραπάνω πλεονεκτημάτων που παρουσιάζει το Navigation Component, έγινε χρήση του κατά την ανάπτυξη της εφαρμογής της παρούσας μεταπτυχιακής διατριβής.

4.10. Connectivity Manager

Μια ακόμα κλάση της οποίας έγινε χρήση κατά την ανάπτυξη της εφαρμογής είναι η Connectivity Manager. Κύριος ρόλος της είναι να αποκρίνεται σε ερωτήματα που έχουν να κάνουν με την παρούσα κατάσταση της σύνδεσης της κινητής συσκευής με το διαδίκτυο. Επίσης, ενημερώνει την εφαρμογή κάθε φορά που σημειώνεται κάποια αλλαγή στην σύνδεση με το διαδίκτυο (απώλεια σύνδεσης, ανάκτηση σύνδεσης κτλ).

[CONNECTION LOST]



Εικόνα 13. Απώλεια σύνδεσης στο διαδίκτυο.

Οι κυριότερες υποχρεώσεις του Connectivity Manager είναι οι εξής:

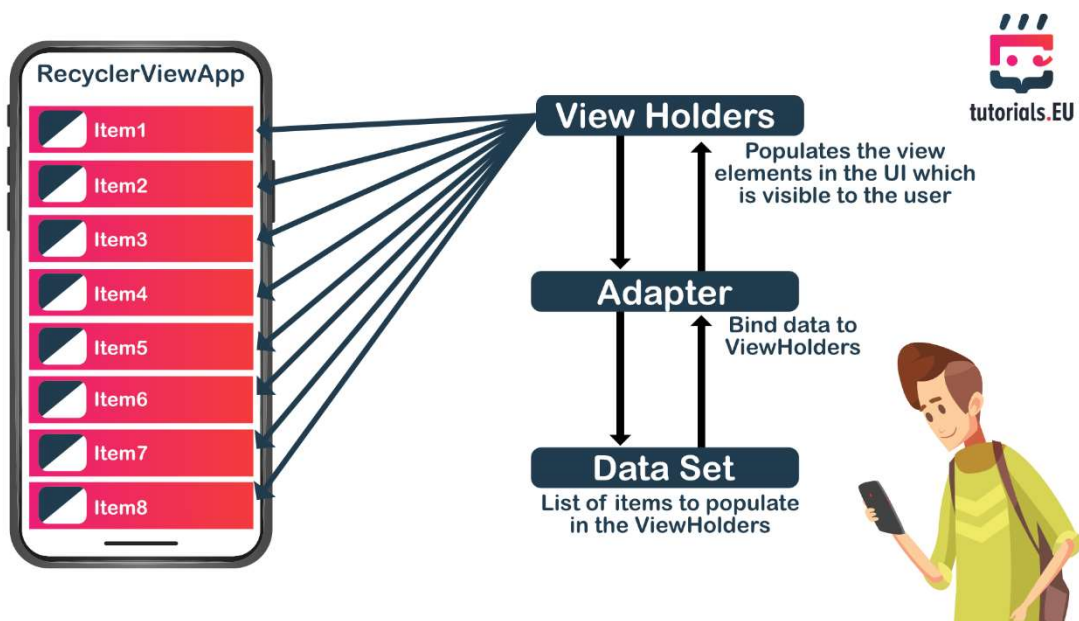
- Παρακολούθηση όλων των συνδέσεων (Wi-Fi, GPRS, UMTS κτλ).
- Αποστολή intents εκπομπής όταν αλλάζει η συνδεσιμότητα του δικτύου.

- Προσπάθεια για μετάβαση σε άλλο δίκτυο, όταν χάνεται η σύνδεση σε ένα δίκτυο.
- Παροχή ενός API στις εφαρμογές, μέσω του οποίου αυτές μπορούν να υποβάλλουν ερωτήματα σχετικά με την γενική ή την λεπτομερή κατάσταση των διαθέσιμων δικτύων.
- Παροχή ενός API στις εφαρμογές, μέσω του οποίου τις επιτρέπει να ζητούν και να επιλέγουν τα επιθυμητά δίκτυα για την διακίνηση των δεδομένων.

Χάρη στη χρήση του Connectivity Manager στην εφαρμογή μας, επιτεύχθηκε η απρόσκοπτη λειτουργία της, ακόμα κι όταν χάνεται η σύνδεση στο διαδίκτυο, και η άμεση ενημέρωση του χρήστη σχετικά με την απώλεια ή την ανάκτηση της σύνδεσης.

4.11. RecyclerView - DiffUtil

Η αρχική οθόνη της εφαρμογής μας παρουσιάζει ουσιαστικά στον χρήστη μια λίστα από μαγειρικές συνταγές βάσει των επιλογών του. Το default πλήθος αυτών των συνταγών είναι 50, πράγμα που σημαίνει ότι θα πρέπει να προβληθεί στην μικρή οθόνη του κινητού μας, ένας αρκετά μεγάλος όγκος από δεδομένα. Αυτό επιτυγχάνεται μέσω της χρήσης του RecyclerView, το οποίο αποτελεί έναν από τους πιο αποτελεσματικούς και αποδοτικούς τρόπους για αντιμετώπιση τέτοιων καταστάσεων. Ο προγραμματιστής από την πλευρά του απαιτείται να παρέχει τα δεδομένα και να καθορίσει την μορφή του κάθε στοιχείου του RecyclerView και στη συνέχεια η βιβλιοθήκη αυτού με την σειρά της δημιουργεί δυναμικά τα στοιχεία του RecyclerView, όταν αυτά χρειαστούν. Όπως υποδηλώνει και το όνομά του, το RecyclerView ανακυκλώνει αυτά τα μεμονωμένα στοιχεία. Όταν ένα από αυτά τα στοιχεία γίνει scrolled εκτός της οθόνης της κινητής συσκευής, το RecyclerView δεν καταστρέφει το View του στοιχείου αυτού. Αντιθέτως το επαναχρησιμοποιεί το View για τα νέα στοιχεία που έχουν γίνει scroll εντός της οθόνης του κινητού.



Εικόνα 14. RecyclerView

Αυτή η επαναχρησιμοποίηση αυξάνει σημαντικά την απόδοση, βελτιώνοντας την ανταπόκριση της εφαρμογής και μειώνοντας την κατανάλωση ενέργειας. Με αυτόν τον τρόπο επιτυγχάνεται και αποτελεσματικότητα στην διαχείριση των διαθέσιμων πόρων. Επιπρόσθετα, αξίζει να σημειωθεί ότι η χρήση του RecyclerView συνδυάζεται με την χρήση της κλάσης DiffUtil. Η τελευταία διευκολύνει πάρα πολύ την διαδικασία κατά την οποία η λίστα των στοιχείων του RecyclerView πρέπει να ανανεωθεί. Σε αυτήν την περίπτωση η κλάση DiffUtil δίνει την δυνατότητα της άμεσης σύγκρισης της παλιάς με την νέα λίστα των στοιχείων, επιτρέποντας την αντικατάσταση μόνο εκείνων των στοιχείων της λίστας τα οποία διαφέρουν. Με αυτόν τον τρόπο επιτυγχάνεται ακόμα μεγαλύτερη ταχύτητα στην εκτέλεση της εφαρμογής. Τέλος, σημειώνεται ότι

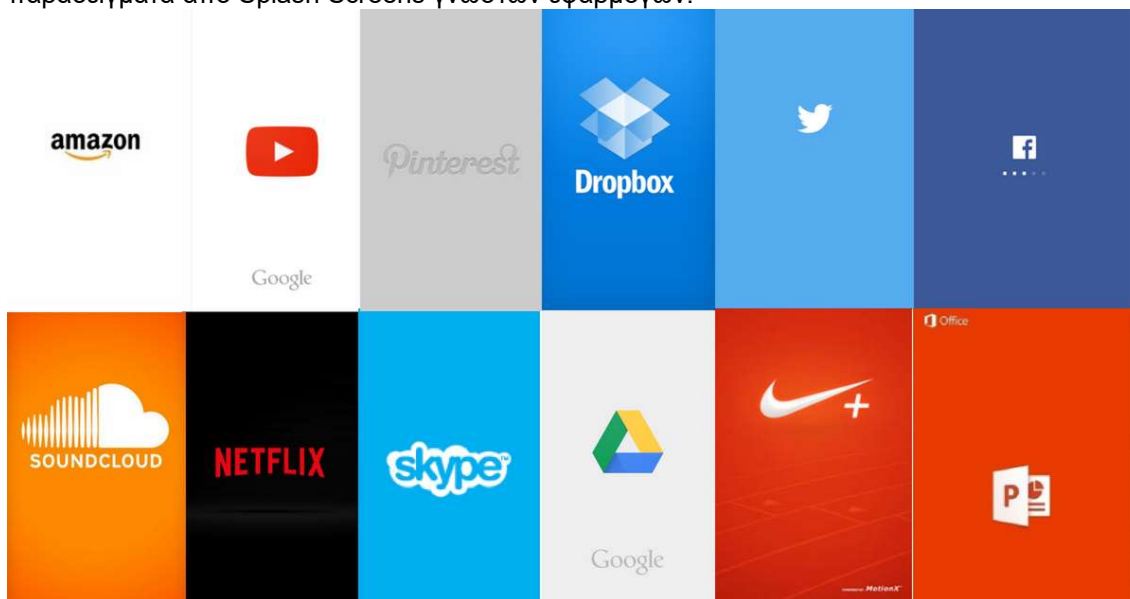
η χρήση των ανωτέρω εργαλείων έγινε και στην οθόνη με τις αγαπημένες μαγειρικές συνταγές του χρήστη, αφού κι εκεί έχουμε μια λίστα συνταγών που πρέπει να προβληθούν στην οθόνη.

4.12. Εργαλεία UI

Σε αυτήν την ενότητα γίνεται μια σύντομη παρουσίαση μερικών από τα εργαλεία που χρησιμοποιήθηκαν προκειμένου να βελτιωθεί η εμπειρία του χρήστη και να εξωραϊστεί περαιτέρω η εφαρμογή μας.

4.12.1. Splash Screen

Όλες οι σύγχρονες εφαρμογές για κινητές συσκευές πλέον διαθέτουν το Splash Screen τους. Το Splash Screen δεν είναι κάτι άλλο από την πρώτη οθόνη, την οποία βλέπει ο χρήστης αμέσως μόλις ξεκινήσει η εκτέλεση της εφαρμογής. Παρακάτω στην Εικόνα 15 φαίνονται μερικά παραδείγματα από Splash Screens γνωστών εφαρμογών.



Εικόνα 15. Splash Screens γνωστών εφαρμογών

Αποτελεί μια από τις πιο βασικές οθόνες της εφαρμογής, διότι είναι η πρωταρχική εμπειρία που έχει ο χρήστης. Η εμφάνιση των Splash Screens διαρκεί συνήθως ελάχιστα δευτερόλεπτα και μόλις ολοκληρωθεί η φόρτωση ο χρήστης οδηγείται αυτομάτως στην πρώτη λειτουργική οθόνη της εφαρμογής. Συνήθως ένα Splash Screen περιέχει το λογότυπο της εκάστοτε εφαρμογής και το όνομά της.

4.12.2. Shimmer Effect

Ένα ακόμα στοιχείο το οποίο μπορεί να βελτιώσει σημαντικά την εμπειρία του χρήστη είναι το Shimmer Effect. Αυτό αποτελεί ουσιαστικά έναν αποτελεσματικό και συνάμα φιλικό προς τον χρήστη τρόπο της προβολής της κατάστασης φόρτωσης μιας διαδικασίας. Σημειώνεται ότι προτιμάται αντί για μια progressbar και χρησιμοποιείται ιδίως όταν έχουμε να κάνουμε με στοιχεία μιας λίστας που φορτώνουν. Κανείς μπορεί να καταλάβει την σημασία που έχει το Shimmer Effect για τον πιο όμορφο σχεδιασμό της εφαρμογής, αν αναλογιστεί πόσες φορές έχει τύχει να περιμένει προκειμένου να φορτώσουν τα δεδομένα σε κάποια εφαρμογή.



Using Shimmer Effect Placeholder

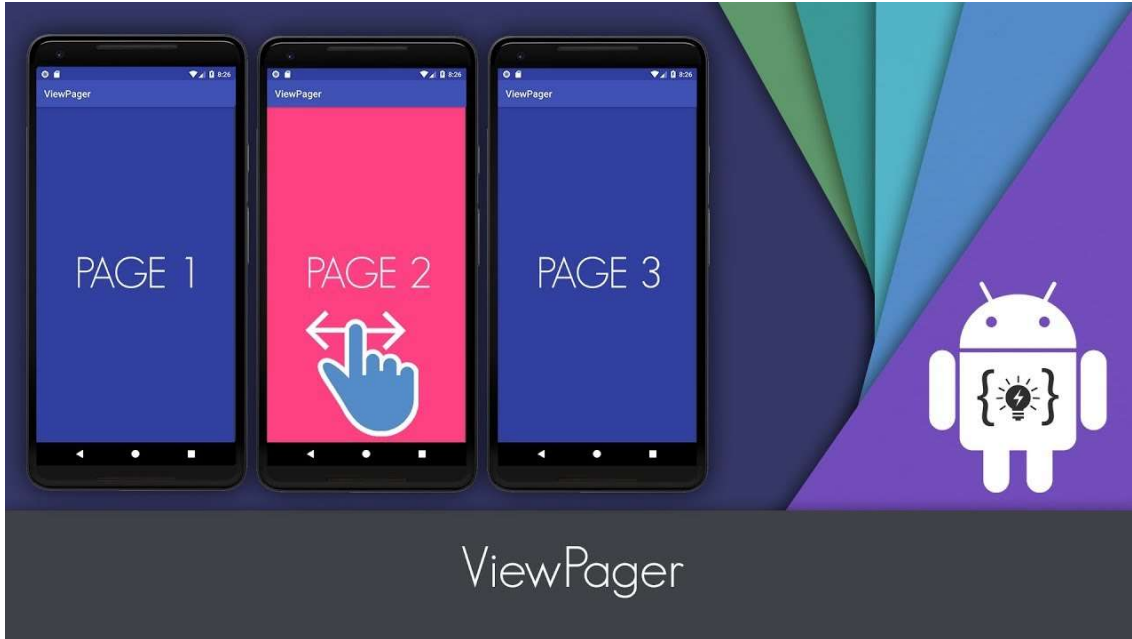


Εικόνα 16. Shimmer Effect

Στην εφαρμογή των μαγειρικών συνταγών το Shimmer Effect χρησιμοποιείται στο RecyclerView της πρώτης οθόνης με τις συνταγές, διότι στην περίπτωση που φορτωθούν συνταγές απομακρυσμένα και καθ' όλη την διάρκεια της φόρτωσής τους είναι καλό ο χρήστης να καταλάβει ότι οι συνταγές βρίσκονται σε κατάσταση φόρτωσης.

4.12.3. ViewPager

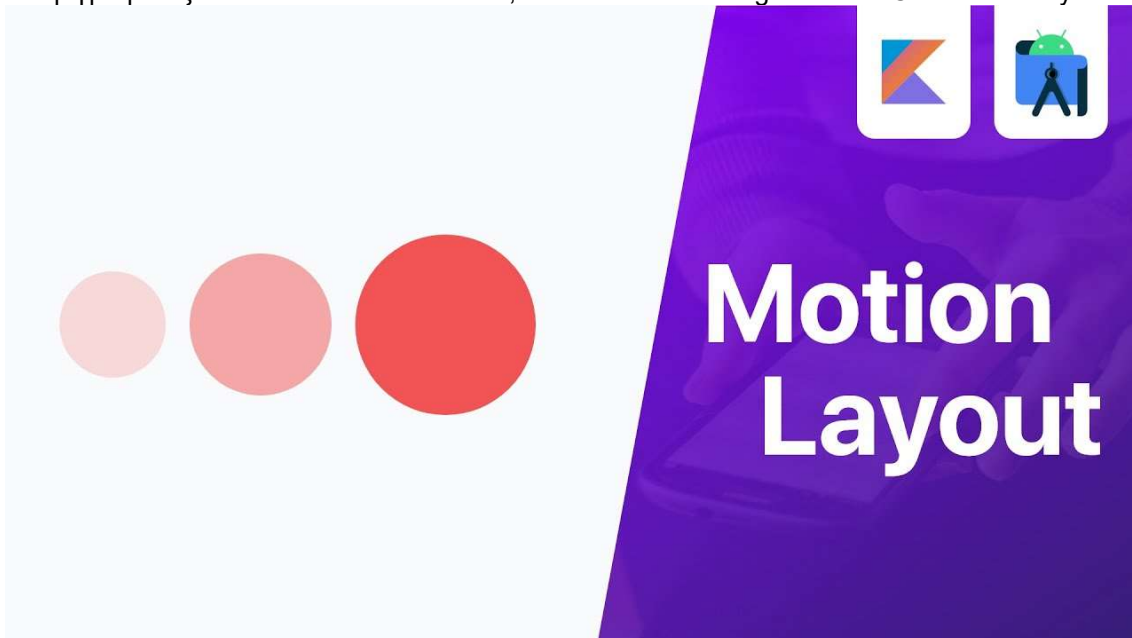
Το ViewPager είναι εκείνο το γραφικό component, το οποίο δίνει την δυνατότητα στον χρήστη της εκάστοτε εφαρμογής να κάνει swiipe δεξιά και αριστερά την οθόνη της εφαρμογής προκειμένου να πλοηγηθεί σε μια καινούρια οθόνη. Ουσιαστικά αποτελεί ένα αποτελεσματικό τρόπο για να προβληθούν στον χρήστη πολλά διαφορετικά tabs. Επιπρόσθετα υποστηρίζει την δυνατότητα της δυναμικής προσθήκης και αφαίρεσης σελιδών ή tabs οποτεδήποτε. Με άλλα λόγια, αποτελεί ουσιαστικά έναν τρόπο διαχείρισης για τα layouts, επιτρέποντας τον χρήστη να κάνει flip μεταξύ διαφορετικών σελίδων δεδομένων. Εν συντομία, μπορούμε να κάνουμε swiipe δεξιά και αριστερά μια λίστα από κάρτες. Η παραπάνω βιβλιοθήκη χρησιμοποιήθηκε στο detailsActivity της εφαρμογής μας, στο οποίο παρουσιάζονται λεπτομέρειες για κάποια μαγειρική συνταγή που έχει επιλέξει ο χρήστης.



Εικόνα 17. ViewPager

4.12.4. MotionLayout

Το MotionLayout, όπως αποκαλύπτει και το όνομά του δεν πρόκειται για κάτι άλλο από ενός είδους layout, το οποίο επιτρέπει την χωροθέτηση των στοιχείων. Πρόκειται για μια υποκλάση του ConstraintLayout, χτισμένο πάνω στις πλούσιες layout δυνατότητες του τελευταίου. Το MotionLayout δημιουργήθηκε προκειμένου να γεφυρωθεί το κενό μεταξύ των layout μεταβάσεων και του σύνθετου χειρισμού κίνησης. Μπορεί κανείς να το σκεφτεί από άποψη δυνατοτήτων ως ένα μίγμα μεταξύ του animation framework, του TransitionManager και του CoordinatorLayout.



Εικόνα 18. MotionLayout

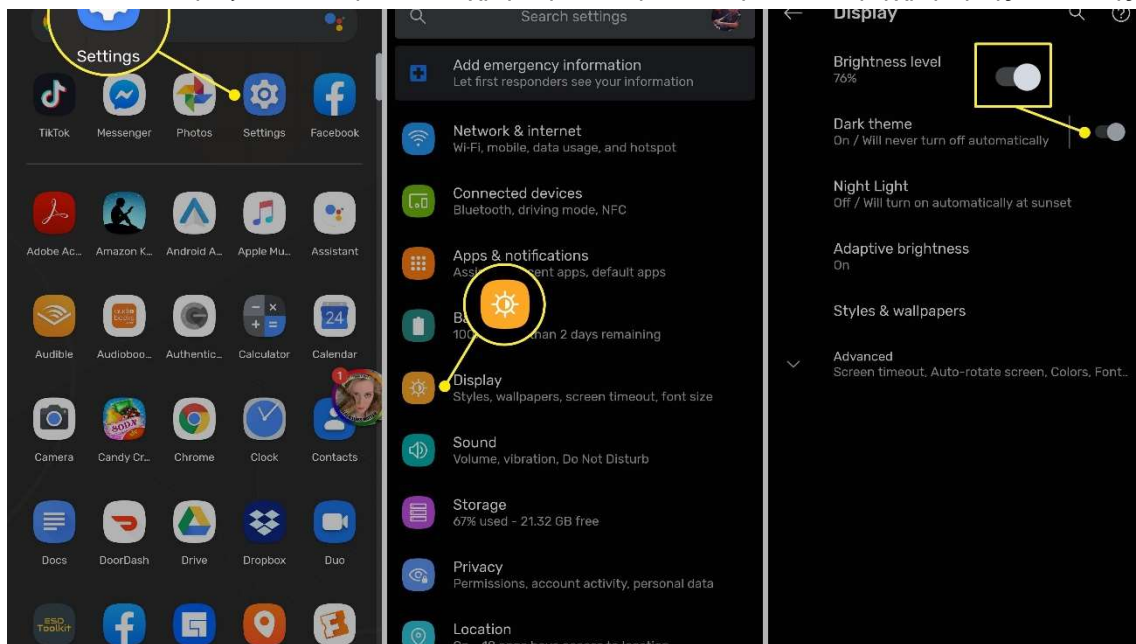
Στην εφαρμογή μας η χρήση του MotionLayout έγινε στο OverviewFragment, στο οποίο παρουσιάζεται μια σύνοψη της μαγειρικής συνταγής που έχει επιλέξει ο χρήστης. Πιο συγκεκριμένα, όταν ο χρήστης κάνει swiipe up αυτήν την οθόνη, η φωτογραφία της συνταγής που

φαίνεται στο ανώτερο μέρος της οθόνης, βγαίνει βαθμιαία εκτός της οθόνης ενώ ταυτόχρονα ξεθωριάζει κιόλας, μέχρι το σημείο που εξαφανίζεται τελείως.

4.12.5. Dark Theme

Το Dark Theme είναι διαθέσιμο στο Android 10 (API level 29) και στα επόμενα. Πρόκειται ουσιαστικά για μια επιλογή προβολής σύμφωνα με την οποία αντί για την εμφάνιση σκοτεινόχρωμου κειμένου σε ανοιχτόχρωμο φόντο, το κείμενο εμφανίζεται ανοιχτόχρωμο ενώ το φόντο σκουρόχρωμο. Ωστόσο η default επιλογή στις περισσότερες κινητές συσκευές είναι το Light Theme κι όχι το Dark. Η ιδέα πίσω από το Dark Theme είναι ότι μειώνει το φως που εκπέμπεται από τις οθόνες των συσκευών, διατηρώντας παράλληλα τις ελάχιστες αναλογίες αντίθεσης χρώματος που απαιτούνται για την αναγνωσιμότητα. Ταυτόχρονα παρουσιάζει τα εξής πλεονεκτήματα:

- Μπορεί να μειώσει σημαντικά την κατανάλωση ενέργειας και ως εκ τούτου να αυξήσει τον χρόνο ζωής της μπαταρίας μια κινητής συσκευής (αναλόγως και την τεχνολογία της οθόνης της συσκευής).
- Βελτιώνει την ορατότητα για χρήστες με χαμηλά επίπεδα όρασης και για όσους έχουν φωτοευαισθησία.
- Διευκολύνει τον καθένα να χρησιμοποιήσει την συσκευή του σε περιβάλλον χαμηλού φωτισμού.
- Δεν κουράζει τόσο τα μάτια του χρήστη όταν πρόκειται για πολύωρη χρήση της συσκευής.



Εικόνα 19. Dark Theme

Η ενεργοποίηση του Dark Theme πραγματοποιείται από τα System Settings. Όταν αυτή ενεργοποιηθεί όλες οι εφαρμογές που υποστηρίζουν Dark Theme μεταβαίνουν σε αυτό. Για όλα τα παραπάνω πλεονεκτήματα του Dark Theme, αποφασίστηκε και υλοποιήθηκε ώστε και η δική μας εφαρμογή να το υποστηρίζει.

5. Ανάλυση και Παρουσίαση της Εφαρμογής

Σε αυτό το κεφάλαιο θα γίνει μια αναλυτική παρουσίαση της εφαρμογής των μαγειρικών συνταγών, όλων των διαθέσιμων οθονών της και όλων των διαθέσιμων ενεργειών που έχει ο χρήστης σε κάθε μία από αυτές τις οθόνες.

5.1. Όνομα και SplashScreen της εφαρμογής

Το όνομα της εφαρμογής των μαγειρικών συνταγών της παρούσας διπλωματικής διατριβής είναι "Cook It Yourself" και το SplashScreen της φαίνεται παρακάτω στην Εικόνα 20. Αυτό το SplashScreen είναι η πρώτη οθόνη που αντικρίζει ο χρήστης με το που ανοίγει την εφαρμογή. Παραμένει στην οθόνη του κινητού για 2 δευτερόλεπτα και στη συνέχεια μεταβαίνουμε αυτόματα στο RecipesFragment της εφαρμογής το οποίο περιγράφεται αναλυτικά παρακάτω.



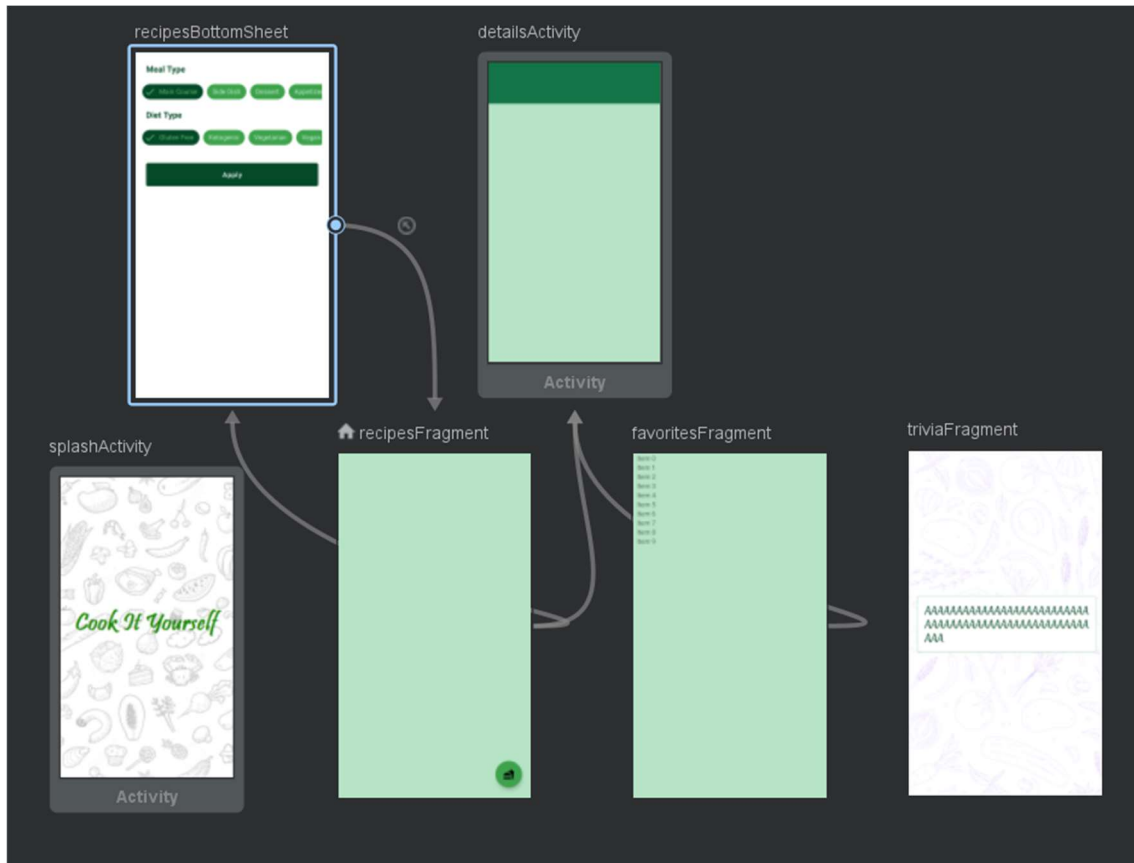
Εικόνα 20. Το SplashScreen της εφαρμογής "Cook It Yourself"

5.2. Navigation Graph

Στην Εικόνα 21 παρακάτω φαίνεται το Navigation Graph της εφαρμογής μας. Αυτό έχει προκύψει με την βοήθεια του Android Navigation Component, όπως αναφέρθηκε και παραπάνω στην Ενότητα 4.9. Σε αυτό απεικονίζονται όλες οι κύριες οθόνες στις οποίες μπορεί να πλοηγηθεί ο χρήστης μέσα στην εφαρμογή. Πιο συγκεκριμένα, το Navigation Graph περιλαμβάνει τα εξής:

- Το SplashScreen.
- Το RecipesFragment.
- Το BottomSheet του RecipesFragment.
- Το FavoritesFragment.
- Το TriviaFragment.
- Και το DetailsActivity.

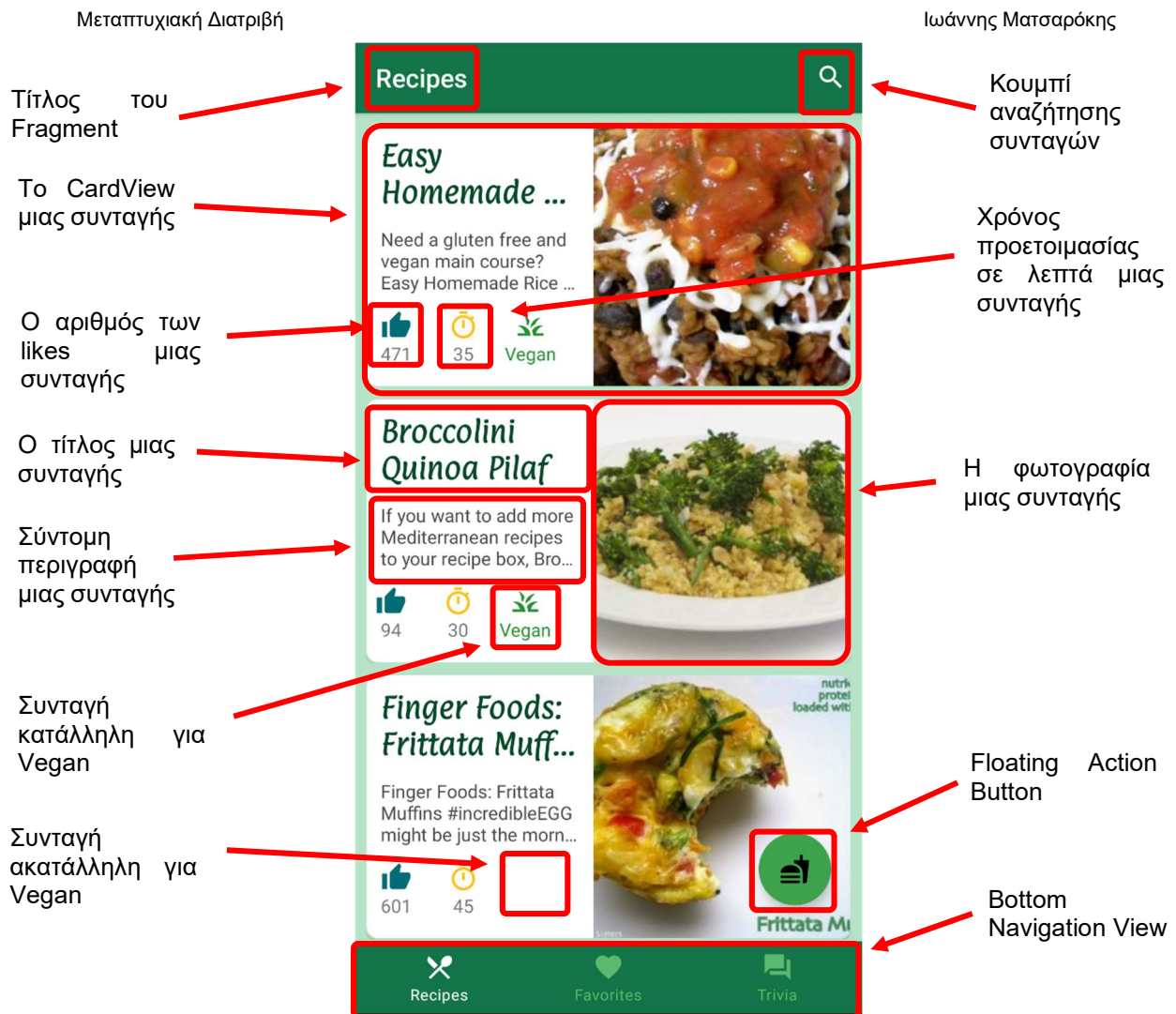
Εκτός από τα παραπάνω, στο Navigation Graph απεικονίζονται και όλα τα δυνατά actions, μέσω των οποίων ο χρήστης μεταβαίνει από την μια οθόνη του Graph σε κάποια άλλη.



Εικόνα 21. Navigation Graph

5.3. Recipes Fragment

Αμέσως μετά το SplashScreen της εφαρμογής εμφανίζεται η πρώτη λειτουργική οθόνη της, η οποία είναι το RecipesFragment. Παρακάτω στην Εικόνα 22 φαίνεται ένα παράδειγμα ενός πιθανού RecipesFragment.



Εικόνα 22. RecipesFragment

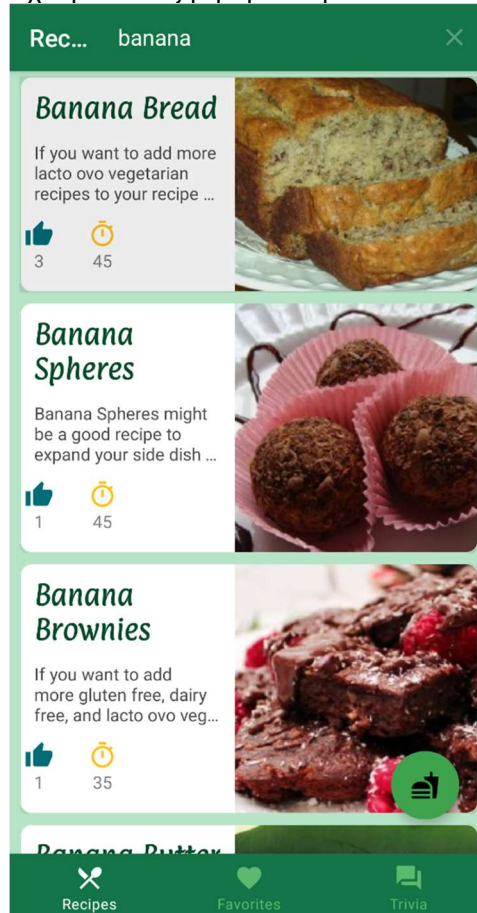
Ξεκινώντας από την κορυφή της οθόνης και κατεβαίνοντας προς τα κάτω το πρώτο πράγμα που συναντούμε είναι το μενυ του RecipesFragment. Αυτό αποτελείται από τον τίτλο του Fragment κι από ένα κουμπί για αναζήτηση μαγειρικών συνταγών. Στην συνέχεια συναντάμε τον RecyclerView, ο οποίος εγκολπώνει CardViews. Κάθε ένα από τα CardViews αφορά σε μια συγκεκριμένη μαγειρική συνταγή και περιέχει τα εξής:

- Τον τίτλο της μαγειρικής συνταγής.
- Μια φωτογραφία από το τελικό προϊόν της μαγειρικής συνταγής.
- Μια σύντομη περιγραφή της μαγειρικής συνταγής.
- Τον αριθμό των likes της μαγειρικής συνταγής.
- Τον χρόνο προετοιμασίας της μαγειρικής συνταγής μετρημένο σε λεπτά.
- Ένα εικονίδιο το οποίο υποδεικνύει αν η μαγειρική συνταγή απευθύνεται σε vegans ή όχι. Ανάλογα με το αν εμφανίζεται ή όχι το συγκεκριμένο εικονίδιο.

Στην κάτω δεξιά γωνία του RecipesFragment υπάρχει το Floating Action Button, το οποίο πατώντας το ο χρήστης έχει πρόσβαση στο Bottom Sheet του Fragment. Τέλος στο κάτω μέρος της οθόνης του RecipesFragment βρίσκεται το Bottom Navigation Menu, μέσω του οποίου ο χρήστης δύναται να πλοηγηθεί στο RecipesFragment, στο FavoritesFragment και στο TriviaFragment. Στο σημείο αυτό αξίζει να σημειωθεί ότι το RecyclerView του RecipesFragment περιέχει 50 μαγειρικές συνταγές by default.

Πέρα από την δυνατότητα πλοήγησης στις συνταγές οι οποίες υπάρχουν ήδη στον RecyclerView του RecipesFragment, ο χρήστης μπορεί να αναζητήσει συνταγές βάσει κάποιου δικού του όρου αναζήτησης χρησιμοποιώντας το κουμπί της αναζήτησης στην πάνω δεξιά γωνία Εφαρμογή Android για Μαγειρικές Συνταγές

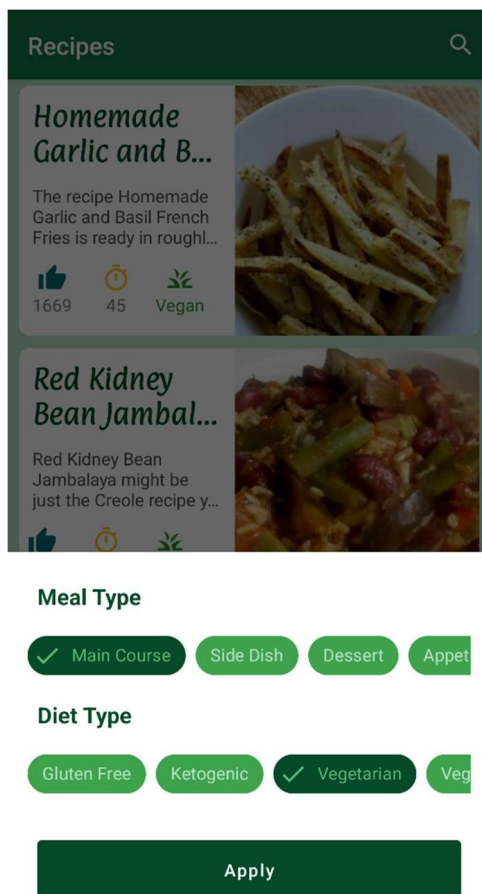
της οθόνης. Στην Εικόνα 23 παρακάτω εμφανίζονται όλες οι μαγειρικές συνταγές που μας επιστρέφει η εφαρμογή, αφού έχει γίνει αναζήτηση του όρου «banana» από τον χρήστη.



Εικόνα 23. Αποτελέσματα αναζήτησης του όρου "banana"

Όπως φαίνεται όλες οι επιστραφείς μαγειρικές συνταγές εμπεριέχουν ως βασικό συστατικό την μπανάνα, δηλαδή τον όρο της αναζήτησής μας.

Επιπρόσθετα, η εφαρμογή δίνει την δυνατότητα πρόσβασης στον χρήστη σε μαγειρικές συνταγές βάσει του είδους του επιθυμητού γεύματος και του τύπου της επιθυμητής διατροφής. Αυτό γίνεται μέσω του Bottom Sheet που εμφανίζεται όταν ο χρήστης πατήσει στο Floating Action Button του RecipesFragment. Στην Εικόνα 24 παρακάτω παρουσιάζεται το προαναφερθέν Bottom Sheet.



Εικόνα 24. Bottom Sheet

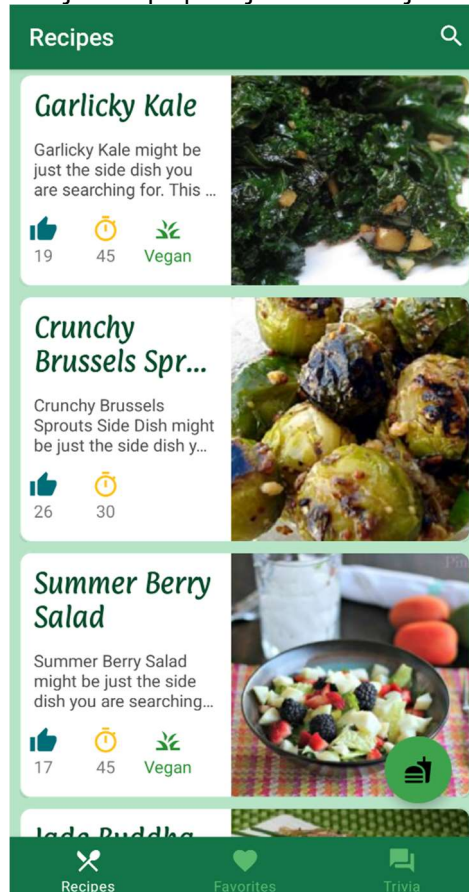
Μέσω του Bottom Sheet του RecipesFragment ο χρήστης έχει πρόσβαση στους εξής τύπους γεύματος:

- Κυρίως πιάτο.
- Συνοδευτικό.
- Επιδόρπιο.
- Ορεκτικό.
- Σαλάτα.
- Ψωμί.
- Σούπα.
- Ποτό.
- Σως.
- Μαρινάδα.
- Finger food.
- Σνακ.
- Ρόφημα.

Ενώ τα είδη της διατροφής στα οποία έχει πρόσβαση ο χρήστης είναι τα εξής:

- Χωρίς γλουτένη.
- Κετογονική.
- Χορτοφαγική.
- Vegan.
- Ιχθυοχορτοφαγική.
- Paleo.
- Primal.
- Whole30.

Αφού ο χρήστης κάνει τις επιλογές του, πατώντας το κουμπί “Apply” στο κάτω μέρος του Bottom Sheet, εμφανίζονται όλες οι συνταγές που αντιστοιχούν σε αυτές τις επιλογές. Ακόμα οι επιλογές αυτές αποθηκεύονται στα Datastore Preferences και με αυτόν τον τρόπο κάθε φορά που ο χρήστης μεταβαίνει στο RecipesFragment, θα του εμφανίζονται οι συνταγές που αντιστοιχούν σε αυτές τις επιλογές, έως ότου γίνουν καινούργιες. Παρακάτω φαίνεται το RecipesFragment έπειτα από επιλογές για Finger Food ως τύπο γεύματος και Paleo ως είδος διατροφής.



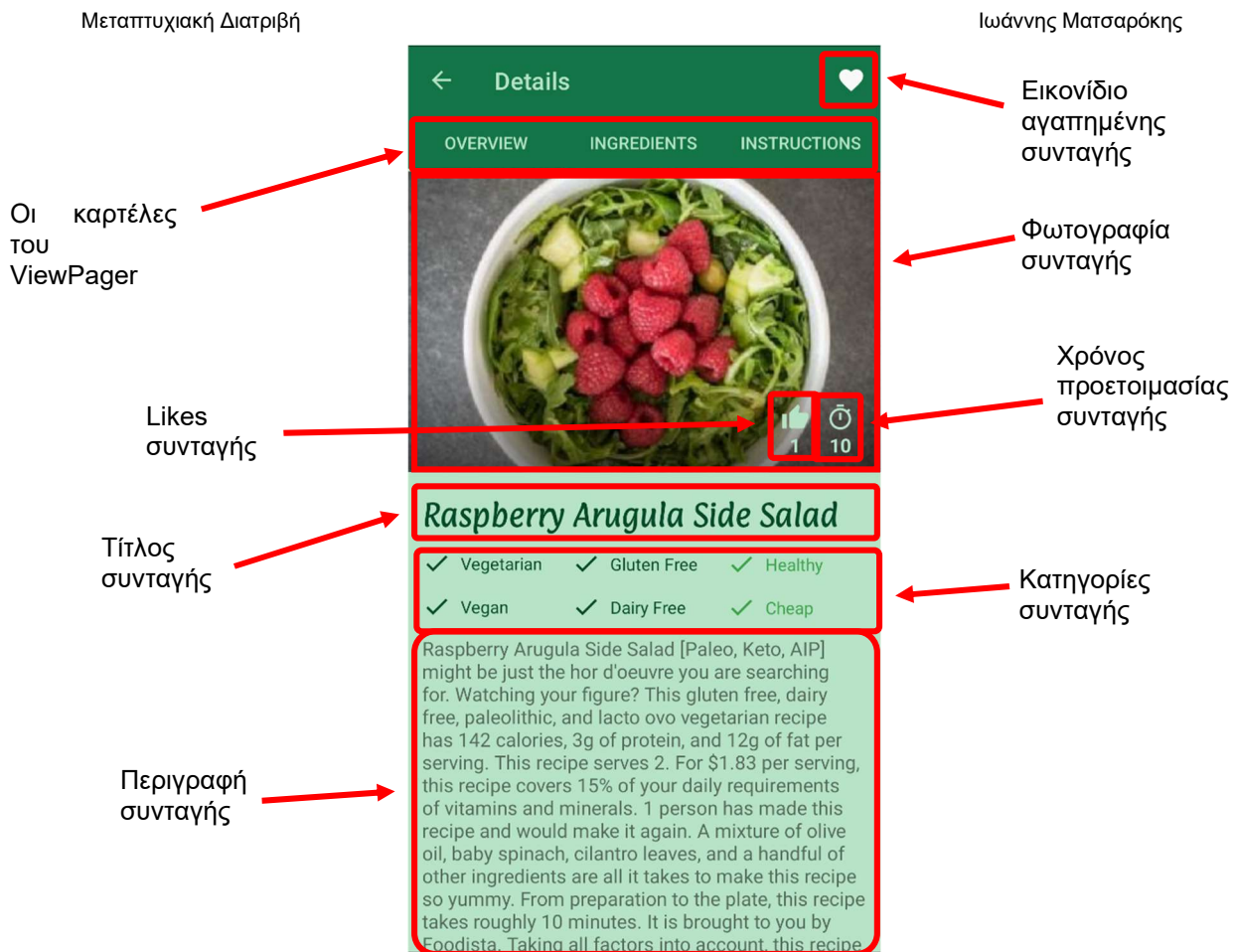
Εικόνα 25. Συνταγές Finger Food για Paleo διατροφή

5.4. Details Activity

Όταν ο χρήστης πατήσει πάνω σε μια από τις μαγειρικές συνταγές του RecyclerView του RecipesFragment ή του FavoritesFragment, τότε μεταβαίνει στο DetailsActivity. Όπως υποδηλώνει και το όνομά του αυτό είναι ένα Activity, το οποίο περιέχει πολλές περισσότερες λεπτομέρειες για την επιλεγθείσα μαγειρική συνταγή. Σε αυτό το Activity γίνεται χρήση ενός ViewPager για την εύκολη πρόσβαση του χρήστη σε 3 επιμέρους Fragments: το OverviewFragment, το IngredientsFragment και το InstructionsFragment, τα οποία περιγράφονται στις παρακάτω υποενότητες.

5.4.1. OverviewFragment

Αποτελεί το πρώτο Fragment, το οποίο εμφανίζεται στον χρήστη με το που ανοίξει το DetailsActivity, πατώντας πάνω σε κάποια μαγειρική συνταγή. Στην παρακάτω Εικόνα 26 φαίνεται το OverviewFragment μιας μαγειρικής συνταγής.

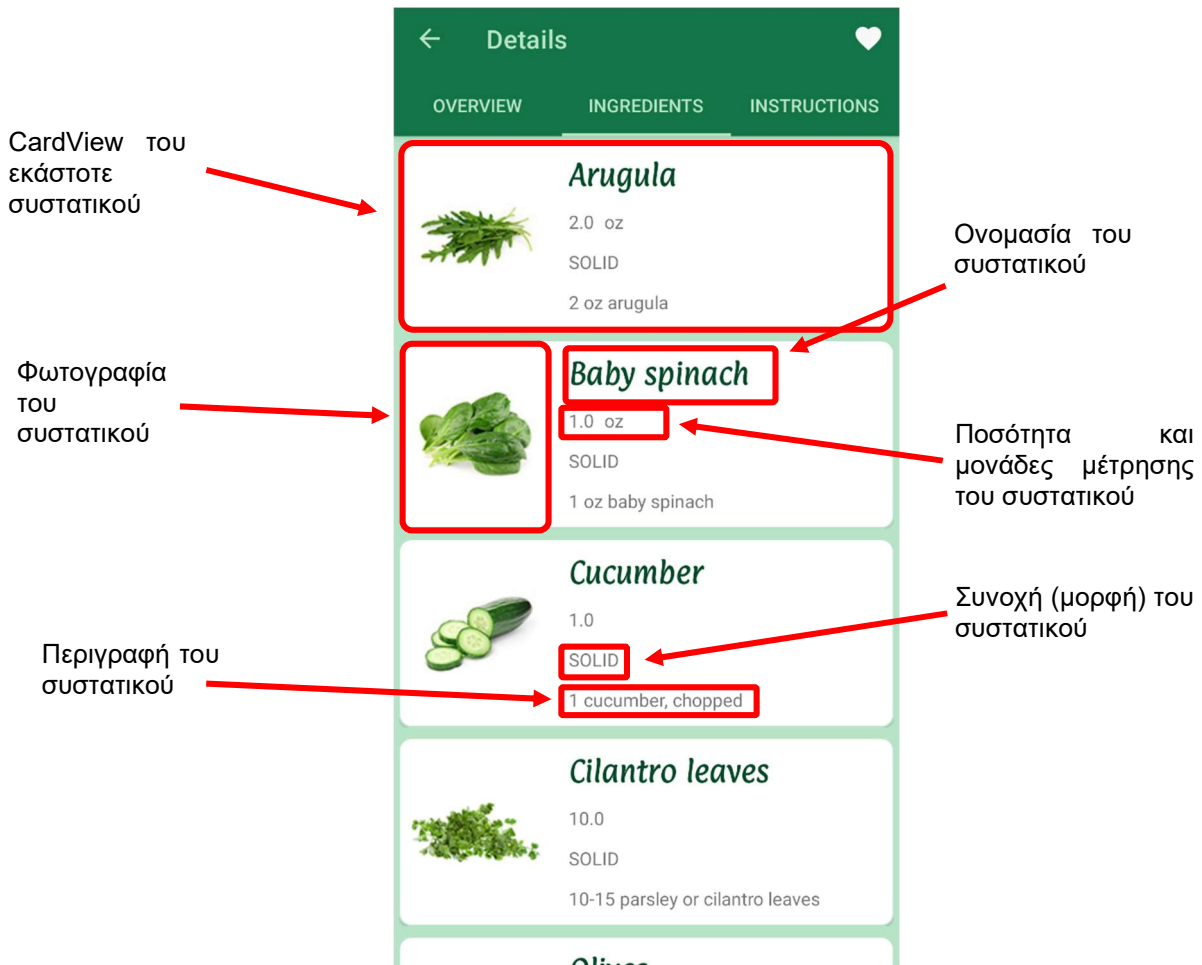


Εικόνα 26. OverviewFragment

Ξεκινώντας από το πάνω μέρος του Fragment και κατεβαίνοντας προς τα κάτω, το πρώτο πράγμα που συναντά ο χρήστης είναι ο τίτλος του Activity και το menu του. Στο menu υπάρχει ένα κουμπί σε σχήμα καρδιάς το οποίο πατώντας το ο χρήστης προσθέτει την τρέχουσα συνταγή στην λίστα των αγαπημένων του συνταγών και το κουμπί από λευκό γίνεται κόκκινο. Στη συνέχεια συναντάμε τις 3 καρτέλες που υποδηλώνουν τα επιμέρους Fragments του ViewPager (Overview, Ingredients, Instructions). Ακόμα από εδώ ο χρήστης πληροφορείται σχετικά με το σε ποιο Fragment βρίσκεται. Πιο κάτω ακολουθεί η φωτογραφία με το τελικό προϊόν της μαγειρικής συνταγής, όπως επίσης και τα likes της συνταγής και ο χρόνος προετοιμασίας της (σε λεπτά). Ακολουθούν ο τίτλος της συνταγής κι ένα γκρουπ κατηγοριών, στις οποίες μπορεί να ανήκει ή όχι η συνταγή ανάλογα με το αν η αντίστοιχη κατηγορία εμφανίζεται σκουρόχρωμη (ανήκει) ή ανοιχτόχρωμη (δεν ανήκει). Τέλος στο κάτω μέρος του OverviewFragment υπάρχει μια αναλυτική περιγραφή της συνταγής.

5.4.2. IngredientsFragment

Το αμέσως επόμενο Fragment του ViewPager του DetailsActivity είναι αυτό του Ingredients. Κι εδώ όπως υποδηλώνει το όνομά του αυτό το Fragment παρέχει μία λίστα με όλα τα απαραίτητα συστατικά που χρειάζεται ο χρήστης για την εκτέλεση της τρέχουσας μαγειρικής συνταγής. Παρακάτω στην Εικόνα 27 φαίνεται το Ingredients Fragment για την συνταγή της Εικόνας 26.

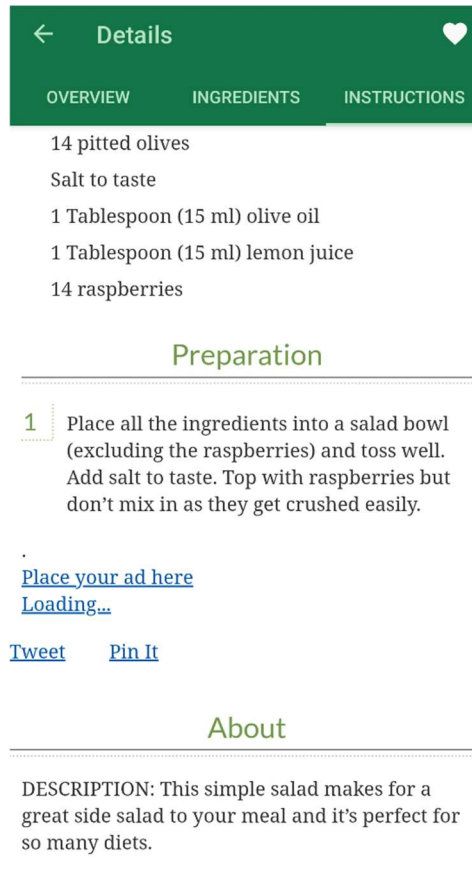


Εικόνα 27. IngredientsFragment

Ξεκινώντας για ακόμη μια φορά από την κορυφή του IngredientsFragment, το πρώτο πράγμα που συναντάει ο χρήστης είναι ο τίτλος του Activity μαζί με το κουμπί της καρδιάς (κουμπί αγαπημένης συνταγής). Ακολουθούν οι 3 καρτέλες των Fragments του DetailsActivity. Τέλος υπάρχει ένας RecyclerView που εγκολπώνει CardViews, κάθε ένα από τα οποία αναφέρεται σε κάποιο συστατικό της τρέχουσας συνταγής. Κάθε CardView περιλαμβάνει την ονομασία του συστατικού, μία φωτογραφία του, την απαιτούμενη (για την συνταγή) ποσότητα και τις μονάδες μέτρησης της, την μορφή του συστατικού (στερεό, υγρό κτλ.) και τέλος μια σύντομη περιγραφή του συστατικού. Αυτό η περιγραφή μπορεί να προσδιορίζει λεπτομέρειες σχετικά με τον χειρισμό του εκάστοτε συστατικού. Για παράδειγμα, μπορεί να υποδεικνύει στον χρήστη με ποιον τρόπο ενδείκνυται να κόψει το κρεμμύδι πριν το προσθέσει στη συνταγή.

5.4.3. InstructionsFragment

Τελευταίο Fragment του ViewPager του DetailsActivity είναι το InstructionsFragment. Αυτό το Fragment περιλαμβάνει οδηγίες σχετικά με την εκτέλεση της συνταγής. Παρακάτω στην Εικόνα 28 φαίνεται το InstructionsFragment της συνταγής των Εικόνων 26 και 27.

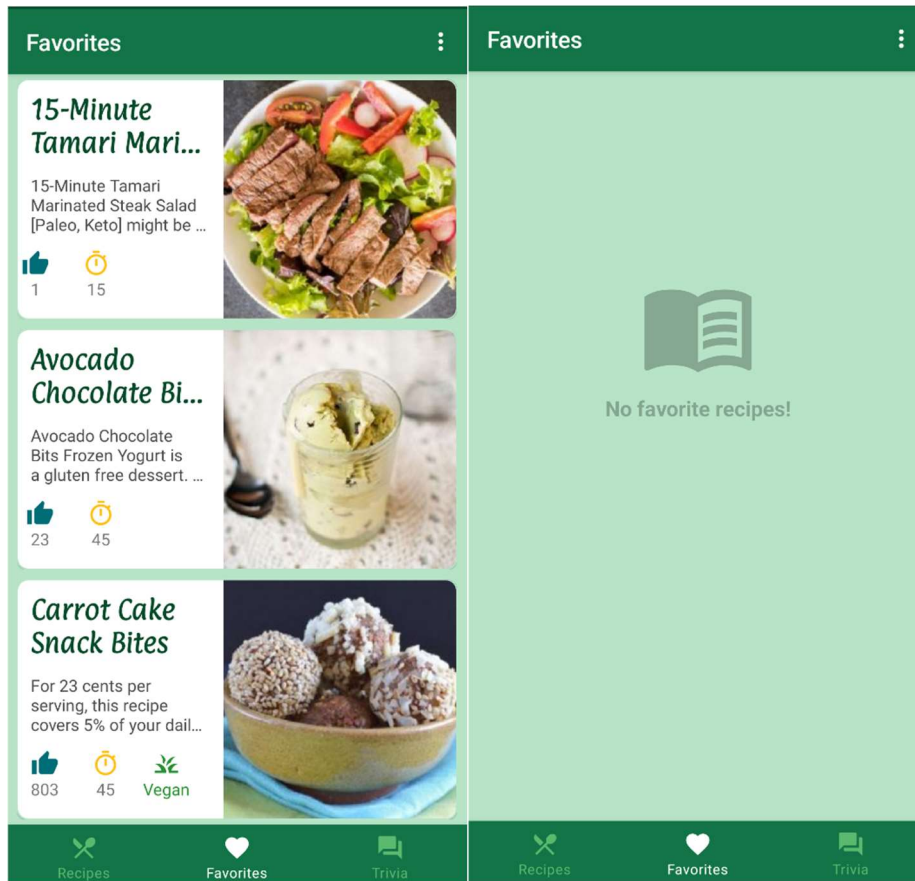


Εικόνα 28. InstructionsFragment

Το InstructionsFragment, όπως και τα άλλα δύο Fragments του DetailsActivity, εμπεριέχει τον τίτλο του Activity και το menu με το κουμπί της αγαπημένης συνταγής, όπως επίσης και τις 3 καρτέλες με τις ονομασίες των Fragments. Το κυρίως σώμα του αποτελείται από ένα WebView, το οποίο εμπεριέχει την ιστοσελίδα-πηγή της τρέχουσας μαγειρικής συνταγής. Επομένως το περιεχόμενο του εκάστοτε InstructionsFragment μπορεί να διαφέρει από συνταγή σε συνταγή. Ωστόσο, συνήθως θα περιλαμβάνει τον τίτλο, την φωτογραφία, την λίστα των απαιτούμενων συστατικών, την περιγραφή της διαδικασίας εκτέλεσης και το ονοματεπώνυμο του συντάκτη της μαγειρικής συνταγής.

5.5. FavoritesFragment

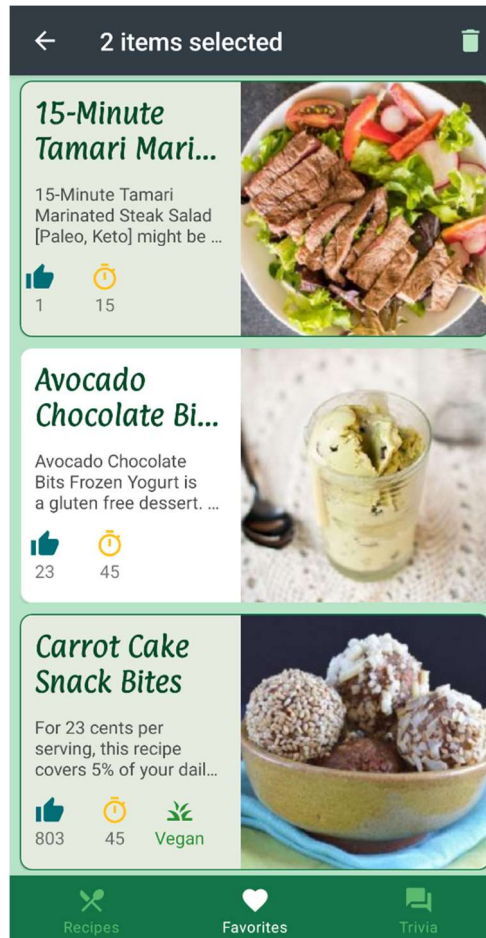
Ο χρήστης πατώντας στο κουμπί Favorites του Bottom Navigation View, πλοηγείται στο FavoritesFragment. Στην Εικόνα 29 παρακάτω φαίνονται 2 screenshots από αυτό το Fragment. Το πρώτο screenshot είναι από την περίπτωση που ο χρήστης έχει προσθέσει αγαπημένες συνταγές στην λίστα του, ενώ στο δεύτερο όχι.



Εικόνα 29. FavoritesFragment με και χωρίς αγαπημένες συνταγές

Ξεκινώντας και πάλι από το πάνω μέρος του FavoritesFragment και κατεβαίνοντας προς τα κάτω, το πρώτο πράγμα που αντικρίζει ο χρήστης είναι ο τίτλος του Fragment μαζί με ένα κουμπί (συμβολίζεται με 3 τελείες) στο menu. Στην συνέχεια ακολουθεί ένας RecyclerView, ο οποίος περιέχει ένα CardView για κάθε μία από τις μαγειρικές συνταγές, που ο χρήστης έχει δηλώσει ως αγαπημένες (πατώντας το κουμπί της καρδιάς στην πάνω δεξιά γωνία του DetailsActivity της εκάστοτε συνταγής). Πατώντας πάνω σε οποιαδήποτε από τις συνταγές της λίστας των αγαπημένων του, ο χρήστης μεταβαίνει στο DetailsActivity της συγκεκριμένης συνταγής.

Επιπρόσθετα ο χρήστης έχει την δυνατότητα διαγραφής μαγειρικών συνταγών από την λίστα των αγαπημένων του με 2 τρόπους. Ο πρώτος τρόπος είναι να διαγράψει όλες τις αγαπημένες του συνταγές πατώντας στο κουμπί που βρίσκεται στην πάνω δεξιά γωνία του FavoritesFragment. Ο δεύτερος τρόπος είναι μέσω το contextual menu του Fragment. Αυτό το menu ενεργοποιείται έπειτα από ένα παρατεταμένο click του χρήστη πάνω σε κάποιο από τα CardViews των αγαπημένων του συνταγών. Έπειτα ο χρήστης μπορεί να επιλέξει (πέρα από την αρχική συνταγή πάνω στην οποία πάτησε παρατεταμένα) περισσότερες συνταγές κάνοντας απλό click πάνω στα CardViews τους. Τα CardViews των επιλεγθεισών μαγειρικών συνταγών εμφανίζονται πιο σκουρόχρωμα, ώστε να ξεχωρίζουν από τις μη επιλεγθεισες συνταγές. Στην Εικόνα 30 παρακάτω φαίνεται ενεργοποιημένο το contextual menu του FavoritesFragment και 2 συνταγές έχουν επιλεγθεί από τον χρήστη.

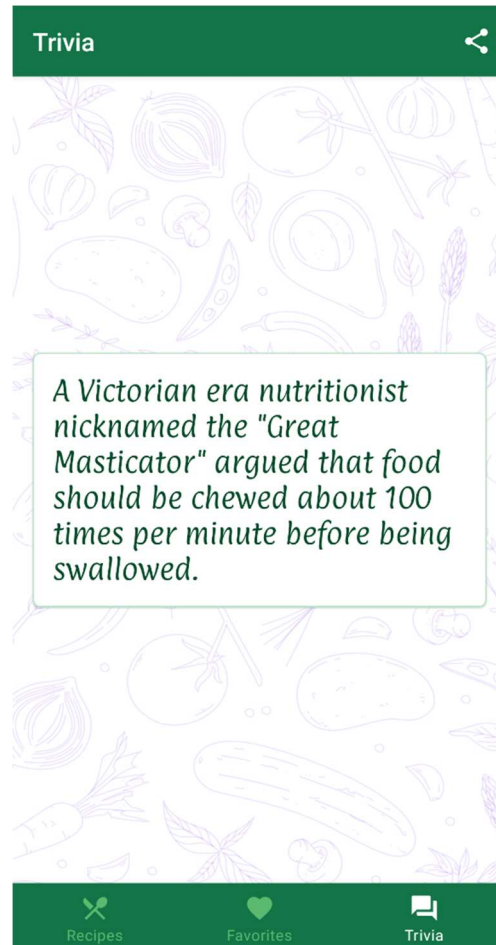


Εικόνα 30. Contextual menu του FavoritesFragment

Αφού ο χρήστης επιλέξει όλες τις επιθυμητές προς διαγραφή μαγειρικές συνταγές μπορεί, πατώντας το κουμπί σε σχήμα κάδου στην πάνω δεξιά γωνία του contextual menu, να αφαιρέσει αυτές τις συνταγές από την λίστα των αγαπημένων του. Μετά την διαγραφή το contextual menu απενεργοποιείται. Άλλος τρόπος απενεργοποίησης του contextual menu είναι η αποεπιλογή όλων των επιλεγμένων συνταγών.

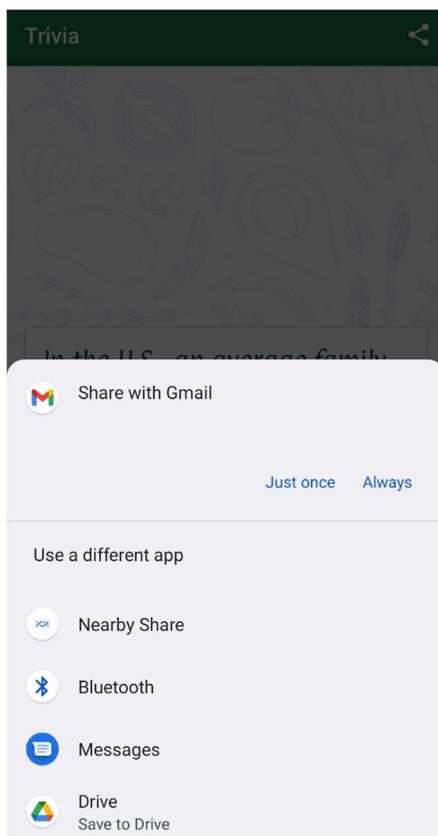
5.6. TriviaFragment

Ο χρήστης πατώντας στο κουμπί Trivia του Bottom Navigation View, πλοηγείται στο TriviaFragment. Στην Εικόνα 31 παρακάτω φαίνεται ένα screenshot από αυτό το Fragment.



Εικόνα 31. TriviaFragment

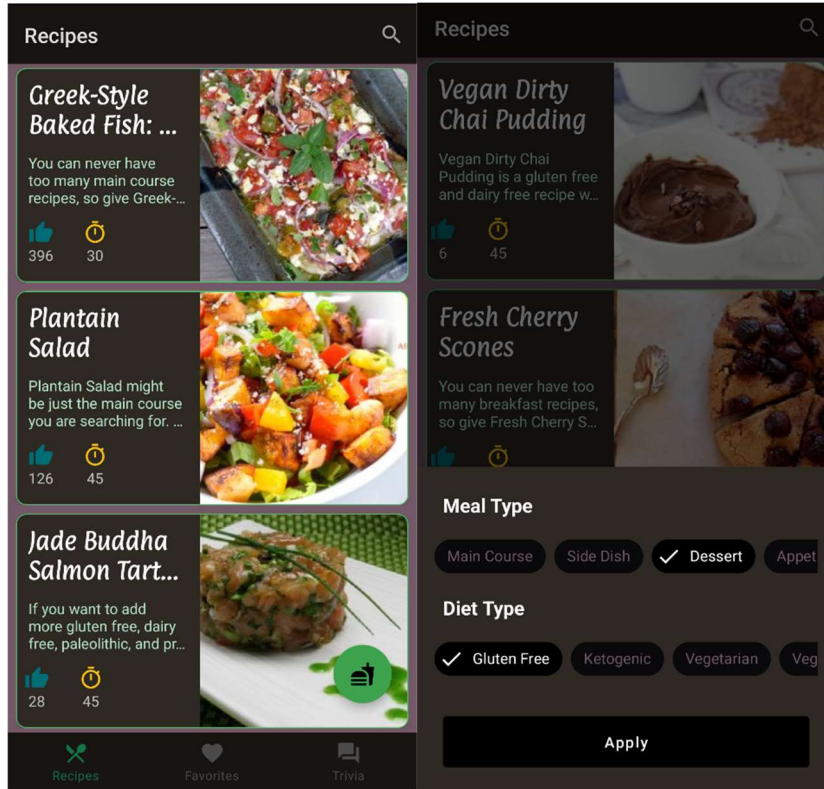
Το TriviaFragment αποτελείται από τον τίτλο του Fragment, ένα κουμπί share στο menu του Fragment και ένα κείμενο στο κυρίως σώμα του. Το κείμενο αυτό περιέχει κάποιο περίεργο γεγονός ή κάποιο άλλο αξιοσημείωτο, το οποίο έχει πάντα σχέση με το φαγητό ή την μαγειρική. Σημειώνεται ότι κάθε φορά που ο χρήστης μεταβαίνει σε αυτό το Fragment του εμφανίζεται ένα καινούριο αξιοσημείωτο, το οποίο αντλείται από το API. Τέλος αναφέρεται ότι ο χρήστης πατώντας το κουμπί της κοινοποίησης έχει την δυνατότητα να μοιραστεί το τρέχον trivia μέσω mail ή sms ή Bluetooth ή και ακόμα και να το αποθηκεύσει στο drive του. Στην Εικόνα 32 παρακάτω φαίνονται όλες αυτές οι διαθέσιμες επιλογές κοινοποίησης του trivia.



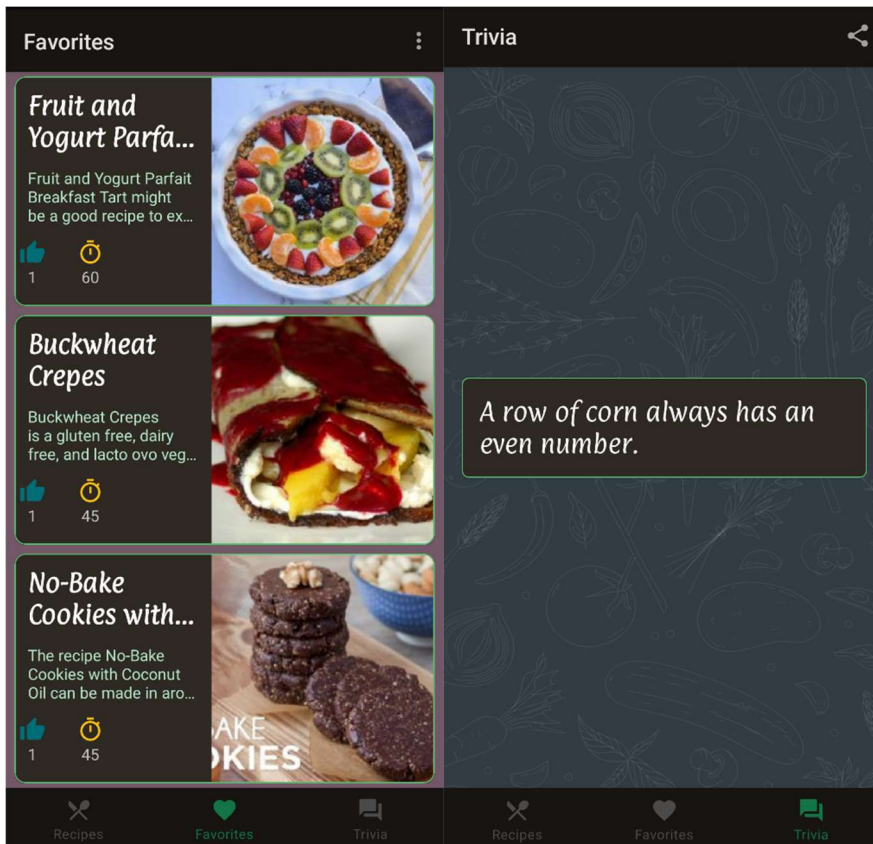
Εικόνα 32. Κοινοποίηση του trivia

5.7. Dark Theme

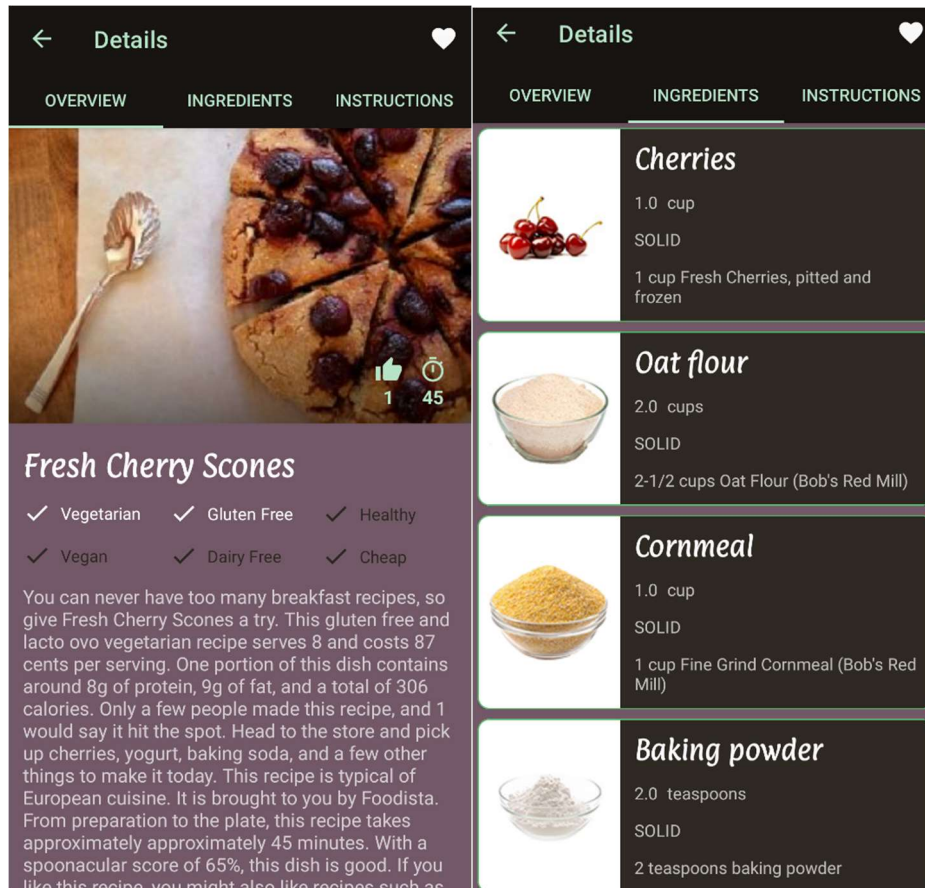
Όπως προαναφέρθηκε και στην ενότητα 4.12.5 ανωτέρω, η εφαρμογή αναπτύχθηκε με τέτοιο τρόπο προκειμένου να υποστηρίζει και Dark Theme. Αυτό σημαίνει πως όταν ο χρήστης έχει ενεργοποιήσει από τα System Settings της συσκευής του την λειτουργία του Dark Theme, αντί να εμφανίζεται το κείμενο της εφαρμογής σκουρόχρωμο πάνω σε ανοιχτόχρωμο φόντο, εμφανίζεται ανοιχτόχρωμο το κείμενο πάνω σε σκουρόχρωμο φόντο. Σημειώνεται ότι αυτή η υλοποίηση πραγματοποιήθηκε για όλα τα Activities, τα Fragments και τα υπόλοιπα components της εφαρμογής. Στις παρακάτω Εικόνες ακολουθούν μερικά screenshots από τις διάφορες οθόνες της εφαρμογής, αλλά αυτή την φορά σε Dark Theme.



Εικόνα 33. RecipesFragment και BottomSheet (Dark Theme)



Εικόνα 34. FavoritesFragment και TriviaFragment (Dark Theme)



Εικόνα 35. OverviewFragment και IngredientsFragment (Dark Theme)

6. Συμπεράσματα-Μελλοντικές Επεκτάσεις

Καθ'όλη την διάρκεια της ανάπτυξης και υλοποίησης της εφαρμογής για τις μαγειρικές συνταγές της παρούσας διπλωματικής διατριβής έγινε μια προσπάθεια για χρήση πολλών σύγχρονων τεχνικών και νέων components με τα οποία ως φοιτητής ήρθα σε επαφή για πρώτη φορά. Το πρώτο από αυτά ήταν η γλώσσα προγραμματισμού Kotlin, που πλέον αποτελεί ίσως την κυριότερη γλώσσα για ανάπτυξη εφαρμογών για κινητές συσκευές με λειτουργικό σύστημα Android, αλλά και όχι μόνο. Επιπρόσθετα, καταβλήθηκε μια προσπάθεια για διάρθρωση της εφαρμογής σύμφωνα με όλες τις αρχές της αρχιτεκτονικής MVVM, η οποία αποτελεί την πλέον ενδεδειγμένη αρχιτεκτονική για mobile apps. Και με την αρχιτεκτονική MVVM ήρθα για πρώτη φορά σε επαφή πράγμα το οποίο με έκανε να μελετήσω εντατικά κάθε επίπεδο της και θεωρώ ότι το τελικό αποτέλεσμα σε αυτόν τον τομέα είναι καλύτερο από το αναμενόμενο από πλευράς μου. Ακόμα, η έννοια του Dependency Injection και η υλοποίησή του μέσω της βιβλιοθήκης του Hilt ήταν κάτι καινούριο για μένα. Ωστόσο, το γεγονός αυτό ήταν που με έκανε να προσπαθήσω ώστε να κατανοήσω σε βάθος την τεχνική αυτή όπως και την χρησιμότητά της στις εφαρμογές για κινητά και όχι μόνο. Επιπλέον, δεν είχα εργαστεί ποτέ πριν με το Data Binding ούτε και με τα Datastore Preferences, δύο πολύ χρήσιμα εργαλεία, τα οποία σίγουρα σκοπεύω να επιστρατεύσω και στο μέλλον. Εν κατακλείδι, η τριβή και η εντατική ενασχόληση με όλες τις παραπάνω έννοιες, όπως και με πολλές άλλες σχετικές με την ανάπτυξη εφαρμογών για κινητές συσκευές, συνέβαλλε στο να ανοίξουν οι προγραμματιστικοί μου ορίζοντες, στην γνωριμία μου με όλες τις τελευταίες τάσεις του χώρου καθώς και στον εμπλουτισμό της προγραμματιστικής μου φαρέτρας με ποικίλα και χρήσιμα εργαλεία. Ως εκ τούτου, θεωρώ ότι ο σκοπός της εκπόνησης της παρούσας διπλωματικής διατριβής στέφθηκε με επιτυχία.

Η παρούσα εφαρμογή δύναται να υποβληθεί μελλοντικά σε αρκετές βελτιώσεις-επεκτάσεις. Μερικές από αυτές είναι και οι εξής:

- Χρήση του toolkit του Jetpack Compose. Μέσω αυτού μπορεί να επιτευχθεί μείωση του υπάρχοντος κώδικα και πιο εύκολη και γρήγορη υλοποίηση του UI της εφαρμογής.
- Υποστήριξη περισσότερων γλωσσών. Αυτή τη στιγμή η εφαρμογή υποστηρίζει μόνο την αγγλική γλώσσα, αλλά πολλοί χρήστες θα το έβρισκαν σημαντικό να μπορούν να διαβάζουν και να αναζητούν τις συνταγές που τους ενδιαφέρουν στις δικές τους γλώσσες.
- Δημιουργία ενός κομματιού της εφαρμογής στο οποίο οι χρήστες θα μπορούσαν να έρθουν σε επαφή μεταξύ τους ανταλλάσσοντας μηνύματα και τις απόψεις τους πάνω στις συνταγές. Ακόμα θα μπορούσαν να ανεβάζουν τις δικές τους μοναδικές συνταγές.
- Προσθήκη δυνατότητας του χρήστη για φιλτράρισμα των συνταγών βάσει συγκεκριμένων τροφίμων ή συστατικών, στα οποία μπορεί να έχει αλλεργία, ή βάσει του χρόνου μαγειρέματος. Επίσης θα μπορούσε να υπάρχει η δυνατότητα ταξινόμησης των συνταγών ανάλογα με τον αριθμό των likes τους ή και άλλα κριτήρια.
- Τέλος θα μπορούσε να προστεθεί ως δυνατότητα, η απ'ευθείας παραγγελία όλων των απαραίτητων συστατικών της εκάστοτε συνταγής μέσω της εφαρμογής για παράδοση στον χώρο του χρήστη.

7. Βιβλιογραφία

Kotlin in Action by Dmitry Jemerov & Svetlana Isakova, 2019

Head First Kotlin: A Brain-Friendly Guide by Dawn Griffiths & David Griffiths, 2019

Head First Android Development 2e: A Brain-Friendly Guide by Dawn Griffiths & David Griffiths, 2017

Android Programming: The Big Nerd Ranch Guide by Chris Stewart, 2019

<https://cellularnews.com/mobile-apps/best-recipe-apps/>

<https://medium.com/>

<https://www.geeksforgeeks.org>

<https://proandroiddev.com>

<https://developer.android.com/>

<https://en.wikipedia.org>

https://www.tutorialspoint.com/mvvm/mvvm_advantages.htm