ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Πτυχιακή Εργασία

Ανάπτυξη Android Εφαρμογής για παροχή υπηρεσιών σχετικές με τη παραγωγή και διανομή φαγητού σε γλώσσα Dart μέσω Flutter

Του φοιτητή
Ξενιτίδη Φώτη
Αρ. Μητρώου: 164717

Επιβλέπων
Αντωνίου Ευστάθιος
Βαθμίδα : Καθηγητής

Θεσσαλονίκη 2024

Τίτλος Π.Ε. Ανάπτυξη Android Εφαρμογής για παροχή υπηρεσιών σχετικές με τη παραγωγή και διανομή φαγητού σε γλώσσα Dart μέσω Flutter
Κωδικός Π.Ε. 22253
Ονοματεπώνυμο φοιτητή Ξενιτίδης Φώτιος
Ονοματεπώνυμο εισηγητή Ευσταθίου Αντωνίου
Ημερομηνία ανάληψης Π.Ε. 12/10/2022
Ημερομηνία περάτωσης Π.Ε.

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραιτήτως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Θα ήθελα να αφιερώσω της συγκεκριμένη εργασία στους γονείς μου για την στήριξη που μου παρείχαν όλα αυτά τα χρόνια των σπουδών μου καθώς και στον φίλο μου τον Γιώργο που άθελα του μου έδωσε την ιδέα για αυτή την εφαρμογή μέσω μιας κατάστασης στην οποία βρέθηκε και μου διηγήθηκε.

Πρόλογος

Η συγκεκριμένη πτυχιακή εργασία δημιουργήθηκε με σκοπό την προσφορά κυρίως σε άτομα που τελειώνοντας μία μαγειρική σχολή ,έχοντας προσκομίσει σημαντικές γνώσεις, δεν τους δίνονται οι κατάλληλες ευκαιρίες να τις εφαρμόσουν. Η συγκεκριμένη εφαρμογή εμπνεύστηκε μετά τη συνειδητοποίηση της κατάστασης που βιώνουν απόφοιτοι μάγειρες. Εφόσον οι χρήστες της εφαρμογής που προσφέρουν υπηρεσίες, μπορούν να επωφεληθούν οικονομικά από αυτή, τότε ο νόμος της προσφοράς επιβάλλει να μπορούν και άλλοι χρήστες, να προσφέρουν τις υπηρεσίες τους, για αυτό δεν υπάρχει περιορισμός σχετικά με το ποιοι μπορούν να προσφέρουν υπηρεσίες.

Περίληψη

Η ραγδαία τεχνολογική ανάπτυξη σε συνδυασμό με την πανδημία με την οποία ήρθαμε αντιμέτωποι συνέβαλαν στην συνειδητοποίηση της δυνατότητας ότι ο καθένας μπορεί να εργαστεί από την οικία του. Η δυνατότητα εύκολης πρόσβασης στο διαδίκτυο ακόμα και με φορητές συσκευές έχει διευρύνει εκθετικά τη χρήση αυτού, κάτι που βοήθησε σημαντικά στην τηλεργασία τους περασμένους μήνες. Το φαινόμενο της τηλεργασίας δεν εμφανίστηκε πρώτη φορά με την πανδημία απλώς εξαπλώθηκε ραγδαία λόγω των αναγκών της παγκόσμιας αγοράς. Είναι γεγονός καθώς τέτοιου είδους μέθοδοι εργασίας συνεχίσουν να αναπτύσσονται ολοένα και περισσότεροι τρόποι δημιουργίας εισοδήματος από την άνεση του σπιτιού μας θα δημιουργούνται.

Ο κλάδος της εστίασης και ιδιαίτερα του φαγητού προσαρμόστηκε εξαιρετικά στις απαιτήσεις των αντίξοων συνθηκών που παρουσιάστηκαν.Η ανάπτυξη των εφαρμογών delivery φαγητού στην Ελλάδα έχει σημειώσει εκρηκτική αύξηση τα τελευταία χρόνια ,ανταποκρινόμενες στην αυξανόμενη ζήτηση των καταναλωτών για πιο εύκολες και γρήγορες επιλογές στο φαγητό τους. Η εμφάνιση εφαρμογών όπως η efood, η Wolt, η Box έχει αναδείξει μία νέα πραγματικότητα στον τρόπο που οι Έλληνες καταναλωτές παραγγέλνουν φαγητό.

Ένα από τα κύρια χαρακτηριστικά που ξεχωρίζει κάθε εφαρμογή είναι η ποικιλία των εστιατορίων και των καταστημάτων που προσφέρουν τις υπηρεσίες τους μέσω αυτών των πλατφορμών. Από την παραδοσιακή ελληνική κουζίνα μέχρι τα διεθνή γεύματα και τις νέες τάσεις στη γαστρονομία, οι καταναλωτές έχουν πλέον πρόσβαση σε ένα ευρύ φάσμα επιλογών.

Ένα άλλο σημαντικό στοιχείο που πρέπει να ληφθεί υπόψη είναι η τεχνολογική πρόοδος που εισήγαγαν αυτές οι εφαρμογές. Από την εύκολη πλοήγηση και τη γρήγορη παραγγελία μέχρι την αξιολόγηση και την ανάδραση των χρηστών, η εμπειρία του καταναλωτή έχει βελτιωθεί σημαντικά μέσω αυτών των εφαρμογών.Τα προγράμματα επιβράβευσης και οι ειδικές προσφορές που προσφέρουν οι εφαρμογές delivery φαγητού έχουν ενθαρρύνει τους καταναλωτές να προτιμούν αυτόν τον τρόπο αγοράς. Η παρουσία των εφαρμογών delivery φαγητού έχει ενισχύσει τον ανταγωνισμό μεταξύ των εστιατορίων, καθώς αυτά αναζητούν συνεργασίες για να διατηρήσουν την ανταγωνιστικότητά τους. Η ανάπτυξη των εφαρμογών delivery φαγητού έχει επηρεάσει τον τρόπο λειτουργίας των εστιατορίων, καθώς αυτά πρέπει να προσαρμοστούν στις αλλαγές στις προτιμήσεις των καταναλωτών. Με την εισαγωγή τεχνολογικών καινοτομιών, όπως η εξέλιξη των εφαρμογών κινητής τηλεφωνίας και η χρήση αλγορίθμων, οι εταιρείες delivery μπορούν να βελτιώσουν την αποτελεσματικότητα της διαδικασίας παράδοσης.Οι εφαρμογές delivery φαγητού έχουν επίσης δημιουργήσει νέες ευκαιρίες εργασίας, καθώς οι εταιρείες αναζητούν συνεργάτες για την παράδοση των παραγγελιών.

Η συνεχής εξέλιξη και καινοτομία στον τομέα των εφαρμογών delivery φαγητού έχει δημιουργήσει ένα περιβάλλον όπου οι επιχειρήσεις πρέπει να προσαρμοστούν συνεχώς για να παραμείνουν ανταγωνιστικές στην αγορά.

Τέλος, μια σημαντική πτυχή που πρέπει να εξεταστεί είναι η κοινωνική και οικονομική επίδραση αυτών των εφαρμογών στην τοπική αγορά. Πιο συγκεκριμένα, μέσω του διαδικτύου κατάφερε να συνεχίσει την ομαλή λειτουργία του καθώς τα καταστήματα μπορούσαν να συνεχίσουν να προσφέρουν τα προϊόντα τους μέσω online παραγγελιών τα οποία έφταναν μέχρι την πόρτα του καταναλωτή. Από τη δημιουργία νέων επιχειρηματικών ευκαιριών για τα εστιατόρια έως την αύξηση της απασχόλησης στον τομέα της παράδοσης, αυτές οι εφαρμογές έχουν επιφέρει αλλαγές στον τρόπο λειτουργίας της αγοράς φαγητού στην Ελλάδα.

Η πτυχιακή εργασία παρουσιάζει μία εφαρμογή για κινητές συσκευές που χρησιμοποιούν android λογισμικό που επικεντρώνεται στην Online αγορά φαγητού από ιδιώτες και όχι καταστήματα όπως εστιατόρια.

Android Application for ordering homecooked drinks and meals

Xenitidis Fotios

ABSTRACT

The rapid technological development combined with the pandemic we have faced has contributed to the realisation that everyone can work from home. The possibility of easy access to the Internet, even with mobile devices, has expanded the use of the Internet exponentially, which has helped teleworking considerably in recent months. The phenomenon of telecommuting did not first appear with the pandemic just spread rapidly due to the needs of the global market. It is a fact that as such methods of work continue to develop more and more ways of generating income from the comfort of our homes will be created.

The catering industry and especially the food service industry has adapted extremely well to the demands of the adverse conditions that have been presented. The growth of food delivery applications in Greece has seen explosive growth in recent years ,responding to the growing consumer demand for easier and quicker options in their food. The emergence of apps such as efood, Wolt, Box has highlighted a new reality in the way Greek consumers order food.

One of the main features that sets each app apart is the variety of restaurants and shops that offer their services through these platforms. From traditional Greek cuisine to international dishes and new trends in gastronomy, consumers now have access to a wide range of choices.

Another important element to consider is the technological advances introduced by these applications. From easy navigation and quick ordering to user rating and feedback, the consumer experience has been greatly improved through these apps.

Finally, an important aspect to consider is the social and economic impact of these apps on the local market. More specifically, through the internet it has been able to continue its smooth operation as shops could continue to offer their products through online orders which reached the consumer's doorstep. From creating new business opportunities for restaurants to increasing employment in the delivery sector, these applications have brought about changes in the way the food market in Greece operates.

This thesis presents a mobile application for mobile devices using android software that focuses on online food purchasing by individuals rather than stores such as restaurants.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω του γονείς μου για την στήριξη που μου παρείχαν όλα αυτά τα χρόνια ,τον υπεύθυνο καθηγητή κύριο Αντωνίου Ευστάθιο για την άψογη συνεργασία μας καθώς και τον φίλο μου τον Γιώργο για την ιδέα που μου έδωσε για αυτή την εφαρμογή άθελα του.

Περιεχόμενα

Κατάλογος Εικόνων

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία

Κεφάλαιο 1ο: Σκοπός της πτυχιακής εργασίας – Το πρόβλημα που επιλύει

Η πτυχιακή εργασία ανήκει στην κατηγορία των εφαρμογών για κινητές συσκευές και ηλεκτρονικού εμπορίου (e-commerce) για τον κλάδο της εστίασης και αναπτύχθηκε με την εξέλιξη του μεγάλου ανταγωνισμού και των δύσκολων συνθηκών που βιώνουμε. Ως ηλεκτρονικό εμπόριο ορίζεται το εμπόριο παροχής αγαθών και υπηρεσιών που πραγματοποιείται εξ αποστάσεων με ηλεκτρονικά μέσα χωρίς να καθίσταται απαραίτητη η φυσική παρουσία των εμπλεκόμενων σε αυτό ατόμων. Συνεχώς και περισσότερες επιχειρήσεις του κλάδου της εστίασης υποστηρίζουν τις υπηρεσίες τους ηλεκτρονικά μέσω δικών τους εφαρμογών ή εφαρμογών που υποστηρίζουν πολλά καταστήματα του ίδιου κλάδου. Οι εν δυνάμει πελάτες μπορούν μεταξύ άλλων να αναζητήσουν και να συγκρίνουν τα αγαθά που τους ενδιαφέρουν από οποιαδήποτε κατάστημα επιθυμούν με βάση την τιμή ,τη βαθμολογία, την απόσταση και άλλες παραμέτρους έτσι ώστε να επιλέξουν την λύση που τους ταιριάζει περισσότερο.

Οι κύριες λειτουργίες της εν λόγω διαδικτυακής εφαρμογής είναι να παρέχει από τη μία την δυνατότητα κέρδους στους ιδιώτες που προσφέρουν υπηρεσίες και τα αγαθά και από την άλλη να λύνει μία βασική ανάγκη της σύγχρονης κοινωνίας που είναι η γρήγορα και εύκολη πρόσβαση σε υπηρεσίες εστίασης. Πιο συγκεκριμένα ο χρήστης που προσφέρει τις υπηρεσίες του θα μπορεί να δημιουργήσει λογαριασμό στην εφαρμογή και να δώσει την λίστα με τα προϊόντα που προσφέρει σε ένα συγκεκριμένο χρονικό διάστημα ,επίσης θα πρέπει να αναφέρει το αντίτιμο για τις υπηρεσίες που προσφέρει καθώς και το που βρίσκεται, ποιες ώρες είναι διαθέσιμες οι υπηρεσίες του και έναν εκτιμώμενο χρόνο της παροχής της εκάστοτε υπηρεσίας από την ώρα της παραγγελίας. Ο χρήστης θα μπορεί πολύ εύκολα και γρήγορα να ενημερώσει και να τροποποιήσει ανά πάσα στιγμή οποιαδήποτε από τις προαναφερθείσες πληροφορίες που τον αφορούν. Επιπρόσθετα η πλατφόρμα παρέχει στον χρήστη που θέλει να αγοράσει φαγητό ή ποτό την δυνατότητα να εγγραφεί και επώνυμα πλέον να αναζητήσει, να συγκρίνει και να αγοράσει το αγαθό της αρεσκείας του. Η αναζήτηση μπορεί να γίνει με βάση μία πληθώρα φίλτρων ανάλογα με την επιλογή του χρήστη και η αγορά είναι εύκολη και γρήγορη. Επιπλέον να προστεθεί εδώ ότι ένας χρήστης μπορεί να εγγραφεί στην εφαρμογή και για τις δύο κατηγορίες χρηστών.

Κλείνοντας , η εφαρμογή προσπαθεί να πετύχει στοιχειώδεις αρχές που κάθε εφαρμογή αυτού του τύπου πρέπει να έχει, όπως το να είναι φιλικό προς όλα τα είδη χρηστών αλλά και προς όλα τα είδη συσκευών ανάλογα με το μέγεθος της οθόνης τους και να ικανοποιεί όσο δυνατόν περισσότερες ανάγκες των χρηστών.

Κεφάλαιο 2ο: Μελέτη και αξιολόγηση υφιστάμενων λύσεων (υπηρεσιών) που έχουν αναπτυχθεί

Σε αυτό το κεφάλαιο παρουσιάζονται μερικές από τις πιο γνωστές εφαρμογές που αξιοποιήθηκαν και οι οποίες δραστηριοποιούνται στον ίδιο χώρο με αυτό της GourMeet (της εφαρμογής που αναπτύχθηκε στα πλαίσια της Πτυχιακής εργασίας), δηλαδή στον κλάδο της εστίασης και ιδιαίτερα στην online παραγγελία φαγητού. Εφαρμογές όπως αυτές που θα αναφερθούν έπαιξαν και παίζουν κυρίαρχο ρόλο στην ανάπτυξη του κλάδου της εστίασης καθώς και στον τρόπο με τον οποίο ο απλός πολίτης έχει πρόσβαση εύκολα και γρήγορα στο φαγητό της αρεσκείας του. Άξιο αναφοράς σε αυτό το σημείο είναι ότι καμία από τις αναφερθείσες εφαρμογές δεν έχει ακριβώς τον ίδιο σκοπό και λειτουργικότητα με την GourMeet αλλά παρόμοια χαρακτηριστικά και διαδικασίες.

2.1 E-food

Η υπηρεσία e-food ιδρύθηκε το 2011 από δύο αδέρφια, τον Κωνσταντίνο και Παμίνο Κυρκίνη οι οποίοι θελήσαν να αλλάξουν τις κατ'οίκον διανομές στην Ελλάδα καθώς εκείνο τον καιρό γινόταν ακόμα με το τηλέφωνο. Φυσικά υπήρξαν και άλλα μέλη από την αρχή του συγκεκριμένου εγχειρήματος. Η έννοια της online παραγγελίας δεν ήταν άγνωστη εκτός των ελληνικών συνόρων απλώς δεν είχε εφαρμοστεί στην Ελληνική αγορά μέχρι εκείνη τη στιγμή. Το e-food καθιέρωσε σταδιακά τις online παραγγελίες φαγητού άμεσα μέσω της εφαρμογής που ανέπτυξε, εξελίσσοντας το προϊόν με υπηρεσίες ταξινομημένες ανάλογα με την απόσταση, τον χρόνο παράδοσης, την ελάχιστη τιμή ή την αξιολόγηση, δίνοντας ακόμη επιλογές για φιλοδωρήματα και κριτικές για την κάθε επιχείρηση με την οποία συνεργάζεται, η οποία με τη σειρά της αποκτά πρόσβαση σε ένα διευρυμένο κοινό χάρτη στην διασύνδεση που παρέχει η εταιρεία και στην συμφωνία προμήθειας που συνάπτουν οι πλευρές. Επίσης η εφαρμογή επιβραβεύει τα μέλη της με προσφορές όπως η τυχερή πινιάτα δηλαδή με την παραγγελία από ένα μόνο κατάστημα για ένα χρηματικό ποσό και πάνω ο χρήστης έχει έκπτωση ανάλογα με το ποσό καθώς υπάρχουν και διαβαθμίσεις στο ποσό αυτό. Επίσης υπάρχουν κι άλλες προσφορές που παρέχει η εφαρμογή όπως με την αγορά ενός προϊόντος δώρο άλλο ένα αλλά και με την αγορά μερικών συγκεκραμένων προϊόντων ανά κατάστημα δώρο κάποιο ακόμα προϊόν. Επίσης επιβραβεύει την παραγγελία του χρήστη από το ίδιο κατάστημα για περισσότερες από μία φορές με την συλλογή των rubies Η εταιρεία αργότερα εξαγοράστηκε από τον γερμανικό κολοσσό του delivery, Delivery Hero, έναντι άγνωστου ποσού χωρίς όμως αυτό να επηρεάσει την δυναμικότητα της στην ελληνική αγορά.

Η εταιρεία είχε μάλιστα καταφέρει να είναι η πρώτη επιλογή στο Google Playstore, ειδικότερα κατά την διάρκεια της πανδημίας. Με την είσοδο της στην Delivery Hero, η ίδια ανέπτυξε περαιτέρω τις τεχνολογικές της παροχές, φτάνοντας 5 χρόνια μετά να επιτρέπει την δυνατότητα ανέπαφων συναλλαγών. Ο λόγος για τις παραγγελίες που έφταναν παντού χωρίς να χρειάζεται ο πελάτης να έρθει σε επαφή με τον διανομέα, αποφεύγοντας έτσι την επιπλέον διασπορά του κορονοϊού με αφορμή την υγειονομική κρίση που ξέσπασε διεθνώς.

Την ίδια περίοδο έδωσαν την ευκαιρία στους πελάτες να κάνουν εικονικά τα ψώνια τους στο σούπερ μάρκετ, αποφεύγοντας τον συνωστισμό και εξυπηρετώντας τους μεγαλύτερης ηλικίας καταναλωτές με συχνές παραδόσεις ακόμη και την ίδια μέρα, όπως διαθέσιμο έγινε στην πορεία και το e-φαρμακείο με mini markets, ανθοπωλεία και περίπτερα. Το e-food θεωρείται η νούμερο ένα επιλογή για online delivery στην Ελλάδα από την αρχή λειτουργίας του μέχρι και σήμερα και δεν έχει σταματήσει ποτέ να προσφέρει καινούργιες υπηρεσίες και να βελτιώνεται συνεχώς.

2.2 Wolt

Η Wolt είναι μία Φινλαδική τεχνολογική εταιρεία που είναι ευρέως γνωστή για την delivery πλατφόρμα της για φαγητό. Ιδρύθηκε το 2014 από έξι συνιδρυτές και πλέον αριθμεί 25 χώρες και πάνω από 300 πόλεις στις οποίες δραστηριοποιείται ενεργά. Στην εφαρμογή της Wolt οι πελάτες μπορούν να παραγγείλουν το φαγητό και είδη οικιακής χρήσης από τους συνεργάτες της εφαρμογής και έχουν επίσης την επιλογή για delivery στην τοποθεσία που θα επιλέξουν ή να παραλάβουν οι ίδιοι τη παραγγελία τους. Η Wolt το 2021 εξαγοράστηκε από τον Αμερικάνικο κολοσσό DoorDash. Αρχικά η εφαρμογή εξυπηρετούσε μόνο παραλαβές από τον ίδιο τον πελάτη και αργότερα πρόσθεσε και την επιλογή του delivery στην πλατφόρμα της, δεν έμεινε μόνο εκεί αφού πειραματίστηκε με αυτοκινούμενα delivery robots στο Ταλίν. Η εφαρμογή της Wolt προσφέρει επίσης υπηρεσίες ταξινομημένες ανάλογα με την απόσταση, τον χρόνο παράδοσης, το ελάχιστο ποσό παραγγελίας ή την αξιολόγηση, δίνοντας ακόμη επιλογές για φιλοδωρήματα και κριτικές για την κάθε επιχείρηση με την οποία συνεργάζεται. Η κάθε επιχείρηση-συνεργάτης με τη σειρά της αποκτά πρόσβαση σε ένα διευρυμένο κοινό χάρτη στην διασύνδεση που παρέχει η εταιρεία και στην συμφωνία προμήθειας που συνάπτουν οι πλευρές. Η Wolt έχει μία διαφορετική προσέγγιση όσο αναφορά το περιβάλλον χρηστή με συνεχής προτάσεις που μπορούν να διαφέρουν ανάλογα με την γιορτή της ημέρας όπως π.χ. η γιορτή του πατέρα αλλά και κατηγορίες όπως τι να φάω σήμερα, κρυμμένοι θησαυροί και άλλα. Η Wolt έχει και αυτή επίσης τα δικά της προγράμματα ανταμοιβής των χρηστών όπως οι προσφορές 1 + 1 , προσφορές και εκπλήξεις, δοκίμασε με 30% και άλλα. Η προσέγγιση του User Interface (περιβάλλον χρήστη) είναι αρκετά προσωποποιημένη και προσφέρει μια εξαιρετική εξατομικευμένη εμπειρία στην χρήστη σύμφωνα με τις ανάγκες και τα θέλω του όπως καμία άλλη εφαρμογή Online Delivery αυτή τη στιγμή στην Ελλάδα. Η Wolt επίσης παρέχει και άλλες υπηρεσίες ανά την υφήλιο όπως logistics και οικονομικές λύσεις για επιχειρήσεις. Η Wolt έχει ως στόχο να ενώσει όσο δυνατόν περισσότερες επιχειρήσεις με σκοπό το κοινό όφελος για όλες καθώς και φυσικά για τον χρήστη που δεν προσφέρει υπηρεσίες απλά τις αγοράζει, βελτιστοποιώντας έτσι τις διαδικασίες που απαιτούνται για να παραχθεί ένα προϊόν ή να προσφερθεί μία υπηρεσία στο ευρύ κοινό.

2.3 Box

Η Box είναι υπηρεσία online παραγγελίας φαγητού που δημιούργησε ο όμιλος ΟΤΕ με σκοπό την δραστηριοποίηση του ομίλου στον τομέα του online delivery. Μέσω του Box app ο καταναλωτής θα μπορούσε για πρώτη φορά να δημιουργήσει το δικό του γευστικό προφίλ σύμφωνα με τις προτιμήσεις και τις ανάγκες του αλλά και θα μπορούσε να προγραμματίσει την ώρα και την τοποθεσία παράδοσης της παραγγελίας τους. Η εφαρμογή της Box δίνει τις ίδιες δυνατότητες με τις προαναφερθείσες για αξιολόγηση καταστημάτων, δυνατότητας φιλοδωρημάτων καθώς και αναζήτηση και ταξινόμηση καταστημάτων ανάλογα με τις επιλογές του χρήστη. Καλό θα ήταν στο σημείο αυτό αν σημειωθεί ότι η box app ήταν η πρώτη ευρέως γνωστή εφαρμογή στην Ελλάδα που έφερε πρόγραμμα ανταμοιβής χρηστών σύμφωνα με πόντους που συλλέγει ο χρήστης από προηγούμενες παραγγελίες. Πιο συγκεκριμένα η εφαρμογή προσφέρει τη δυνατότητα στον χρήστη να μαζεύει πόντους μα κάθε παραγγελία του οι οποίοι αντιστοιχίζονται με ευρώ τα οποία μπορεί αν εξαργυρώσει στην εφαρμογή σε οποιαδήποτε παραγγελία του. Μία επιπλέον λειτουργικότητα της Box είναι η δυνατότητα το χρήστη να μπορεί να επιλέξει ώρα παράδοσης η παραλαβής της παραγγελίας του, δυνατότητα που δεν προσφέρει καμία άλλη εφαρμογή στην Ελλάδα μέχρι και αυτή τη στιγμή. Η εφαρμογή επίσης παρέχει προσφορές 1+1 προϊόντα και άλλες προσφορές που είναι μόνο στην συγκεκριμένη εφαρμογή.

2.4 Επίλογος

Οι παραπάνω εφαρμογές είναι κάποιες από τις εφαρμογές που έχουν αναπτυχθεί και προσφέρουν υπηρεσίες online delivery παραγγελιών φαγητού, φυσικά υπάρχουν πολλές ακόμα, και από τις οποίες η GourMeet αντλεί έμπνευση κυρίως αναφορικά με τα εργαλεία που παρέχονται σε όλα τα είδη των χρηστών. Ο παρακάτω πίνακας απεικονίζει ομοιότητες και διαφορές που παρατηρήθηκαν ανάμεσα στις προαναφερθείσες εφαρμογές .

	Ομοιότητες	Διαφορές
Προσφορές/ Προγράμματα Επιβράβευσης Χρηστών	Όλες οι εφαρμογές έχουν την 1+1 προσφορά αλλά και προσφορές όπως π.χ. με την παραγγελία τριών προϊόντων δώρο άλλο ένα	Η E-food έχει την τυχερή πινιάτα καθώς και τα rubies Η Wolt έχει επίσης τη κατηγορία δοκίμασε με 30% όπου ο χρήστης έχει έκπτωση για την παραγγελία του Η Box έχει τις δικές της εξατομικευμένες προσφορές
User Interface	Ανοίγοντας και τις τρεις εφαρμογές ο χρήστης βλέπει εστιατόρια καθώς και την επιλογή να τα ταξινομήσει ανάλογα με τις ανάγκες του καθώς και τις προσφορές από τις οποίες μπορεί να επιλέξει. Επίσης εμφανίζεται σε ένα πλαίσιο επάνω η διεύθυνση του χρήστη την οποία μπορεί να αλλάξει. Το User Exprerinece(UX-εμπειρία χρήστη) είναι αρκετά παρόμοια για όλες τις εφαρμογές καθώς ο χρήστης κάνοντας είσοδο στην εφαρμογή μπορεί να επιλέξει το κατάστημα που επιθυμεί και ύστερα να παραγγείλει με συνοπτικές διαδικασίες με αρκετά παρόμοιο τρόπο μεταξύ των εφαρμογών	Οι διαφορές έγκεινται κυρίως στο είδος των επιλογών που έχει ο χρήστης και πόσο προσωποποιημένες είναι αυτές οι επιλογές.
Αναζήτηση/ Ταξινόμηση	Και οι τρεις εφαρμογές δίνουν τη δυνατότητα για αναζήτηση ενός συγκεκριμένου καταστήματος ή προϊόντος καθώς και την ταξινόμηση των καταστημάτων με βάση το είδος της κουζίνας τους, τις τιμές που προσφέρουν, την απόσταση που απέχουν από τον χρήστη, τις βαθμολογίες χρηστών αλλά και το delivery fee	Καμία σε αυτή τη κατηγορία
Επιλογές παράδοσης	Και οι τρεις δίνουν τη δυνατότητα για επιλογή καταστημάτων ανάλογα με τον χρόνο παράδοσης τον οποίο τον ορίζει το κατάστημα	Μόνο η Box προσφέρει τη δυνατότητα στον χρήστη να επιλέξει την ώρα που θα παραλάβει την παραγγελία του είτε αυτό είναι delivery είτε είναι takeaway

Κεφάλαιο 3ο: Περιγραφή του τεχνολογικού περιβάλλοντος που χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής

Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκαν μερικές τεχνολογίες όπως το Flutter Framework με γλώσσα Dart και η Firebase Platform οι οποίες περιγράφονται παρακάτω.

3.1 Flutter Framework και Dart

Το Flutter Framework είναι ένα open source framework, ης γνωστής σε όλους μας Google, γεγονός που του δίνει ιδιαίτερη αξία και ακεραιότητα, για την κατασκευή όμορφων,εγγενώς μεταγλωτισμένων(natively compiled), που μπορούν να τρέξουν σε πολλές διαφορετικές πλατφόρμες(multi-platform) εφαρμογών από μία μοναδική πηγή κώδικα. Αναφέρεται ως εγγενώς μεταγλωτισμένη(natively compiled) γιατί ο compiler (μεταγλωτιστής) δουλεύει τη μεταγλώττισή (compilation) της εφαρμογής στην ίδια τεχνολογία στην οποία τρέχει η εφαρμογή. Χρησιμοποιεί το ίδιο λειτουργικό σύστημα ή πλατφόρμα όπως το λογισμικό για το οποίο παράγει την γλώσσα μηχανής (machine language). Με πιο απλά λόγια μπορεί ο προγραμματιστής να δει compile errors στον κώδικα του την ώρα που γράφει τον κώδικα γεγονός που του εξοικονομεί σημαντικό και πολύτιμο χρόνο. Επίσης ένας από τους πιο σημαντικούς λόγους που ξεχωρίζει η flutter σαν framework και επιλέγεται από τους προγραμματιστές και τις development teams είναι η cross-platform ιδιότητα του. Πιο συγκεκριμένα, το παραγόμενο αρχείο για την εφαρμογή είναι συμβατό με πολλά mobile συστήματα λογισμικού που σημαίνει ότι μπορεί να χρησιμοποιηθεί ανεξαρτήτως συστήματος λογισμικού. Μέσω τις flutter εκτός από διάφορα λογισμικά για κινητές συσκευές ο ίδιος κώδικας μπορεί να χρησιμοποιηθεί για desktop και web εφαρμογές. Ένα πολύ σημαντικό θετικό στοιχείο της flutter είναι ότι ο προγραμματιστής μπορεί να παραμετροποιήσει τον κώδικα της εφαρμογής έτσι ώστε σε κάθε περίπτωση λογισμικού η πλατφόρμας να υπάρχει διαφορετική λειτουργικότητα και εμφάνιση.

Εικόνα 3:1: Το λογότυπο της Flutter

Πέρα από τα παραπάνω, σημαντικό ρόλο στην ανάπτυξη της εφαρμογής έπαιξε και η γλώσσα Dart. Η flutter υποστηρίζει άμεσα δύο γλώσσες προγραμματισμού την Python και την Dart με

την δεύτερη να είναι πιο συνυφασμένη με το συγκεκριμένο framework. Για τη συγκεκριμένη εργασία και εφαρμογή η γλώσσα που χρησιμοποιήθηκε ήταν η Dart. Η Dart είναι αντικειμενοστραφής, class-based, garbage-collected γλώσσα με σύνταξη παρόμοια με αυτή της C. Μπορεί επίσης να μεταγλωττίσει σε JavaScript η WebAssembly καθώς ταυτόχρονα υποστηρίζει interfaces, mixins, abstract classes, reified generics και type inference. Με άλλα λόγια η Dart είναι μία καλύτερη έκδοση της JavaScript καθώς είναι δύο φορές ταχύτερη, type-safe και μπορεί να μεταγλωττιστεί από AOT και JIT μεταγλωττιστές (compilers).

Εικόνα 3.2: Το λογότυπο της Dart

3.2 Βάσεις Δεδομένων και Διαχείριση

Για τις ανάγκες της αποθήκευσης των δεδομένων της εφαρμογής χρησιμοποιήθηκε η Firebase Realtime Database, μία βάση δεδομένων σηκωμένη στο cloud. Τα αρχεία αποθηκεύονται στη βάση με τον μορφή JSON και είναι συγχρονισμένα σε πραγματικό χρόνο με τον κάθε client που έχει πρόσβαση σε αυτά. Ένα μεγάλο θετικό και σημαντικός λόγος που επιλέχτηκε η συγκεκριμένη πλατφόρμα είναι η αμεσότητα και η ταχύτητα που παρέχει τα δεδομένα και με κάθε αλλαγή τους. Η Firebase προσφέρει μία πληθώρα άλλων υπηρεσιών-βοηθειών στον προγραμματιστή όπως OffLine χρήση, μη αναγκαία ύπαρξη server για την ανταλλαγή κλήσεων καθώς μπορεί να αποκτήσει πρόσβαση στα δεδομένα κανείς ακόμα και μέσα από ένα αίτημα που θα στείλει από κάποιον Browser. Επίσης μία πολύ σημαντική λειτουργία είναι το authentication που προσφέρει η πλατφόρμα για τον έλεγχο των ενεργειών που μπορεί να κάνει ένας χρήστης πάνω στα δεδομένα γεγονός που προσφέρει μεγάλη βοήθεια στους προγραμματιστές καθώς η προστασία των δεδομένων σε μία εφαρμογή είναι μεγάλης σημασίας με πολλές πιθανές και μεγάλες επιπτώσεις σε περίπτωση κάποιος προσπαθήσει να δει ή επεξεργαστεί ευαίσθητα κυρίως προσωπικά δεδομένα. Η βάση δεδομένων που χρησιμοποιείται για την κινητή συσκευή είναι η Hive. Η Hive είναι μια ελαφριά και ταχύτατη βάση δεδομένων κλειδιών-τιμών γραμμένη σε καθαρή Dart.

Εικόνα 3:3 Hive Mobile Firebase Logo

Εικόνα 3:4 Firebase Realtime Database

3.3 Περιβάλλον Ανάπτυξης

Η ανάπτυξη της εφαρμογής στο Visual Studio Code που είναι ένας editor που δημιούργησε η ομάδα της Microsoft ίσως ο πιο διαδεδομένος και πολυχρησιμοποιημένος editor ανάμεσα σε developers. Οι λειτουργίες και υπηρεσίες που προσφέρει είναι πραγματικά παραπάνω από όσες μπορεί ένας και μόνο προγραμματιστής να γνωρίζει μιας και δίνει τη δυνατότητα για εισαγωγή plugins , δηλαδή έξτρα υπηρεσιών, που πραγματικά μπορούν να διευκολύνουν σε τεράστιο βαθμό τον προγραμματιστή. Μερικά features που προσφέρονται είναι debugging, syntax highlighting, intelligent code completion, code refacatoring και φυσικά την εισαγωγή extension για την βελτιστοποίηση της βοήθειας που προσφέρει στον χρήστη. Το Visual studio Code διαθέτει βασική στήριξη για τις περισσότερες πολυχρησιμοποιούμενες γλώσσες προγραμματισμού γεγονός που το καθιστά ιδανικό για μια πολύ μεγάλη γκάμα εφαρμογών. Το Visual Studio Code είναι μία εξαιρετική επιλογή για την ανάπτυξη android αφαρμογών με Flutter καθώς το debugging είναι πολύ εύκολο και γρήγορο και στις εφαρμογές; για κινητά γίνεται συνέχεια αλλά και η εύκολη διασύνδεση με το GitHub για το ανέβασμα του κώδικα είναι επίσης κάτι εξαιρετικά χρήσιμο.

Εικόνα 3:5 Λογότυπο του IDE Visual Studio Code

3.4 Επεκτάσεις και βιβλιοθήκες

Κατά την ανάπτυξη της εφαρμογής χρησιμοποιήθηκαν μία πληθώρα από επεκτάσεις και βιβλιοθήκες με σκοπό γρηγορότερη και ευκολότερη ανάπτυξη της εφαρμογής.

3.4.1 **Dart:**

Το Dart extension είναι μία επέκταση για την υποστήριξη της γλώσσας Dart που βοηθάει στο editing, refactoring, running και reloading Flutter mobile εφαρμογές.

3.4.2 **Flutter:**

Υποστηρίζει το Flutter Framework και εξαρτάται από το προηγούμενο extension το οποίο θα εγκαταστήσει σε περίπτωση που δεν υπάρχει.

3.4.3 **Bloc:**

Το bloc extension είναι μία βιβλιοθήκη που προσφέρει εργαλεία για την αποτελεσματική διαείριση των Blocs μέσω των οποίων γίνεται η διαχείριση των states όλης της εφαρμογής.

3.4.4 Rainbow Tags:

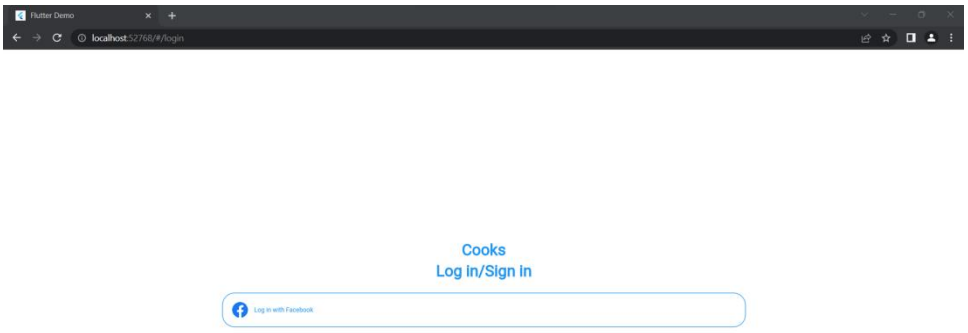
Είναι ένα extension το οποίο είναι πραγματικά αναγκαίο στην ανάπτυξη Flutter εφαρμογών με Dart καθώς με την μεγάλη πολυπλοκότητα που αποκτά ο κώδικας συχνά γίνεται πολύ δύσκολο να κατατοπιστείς μέσα στο Visual Studio Code και τα χρώματα που δίνει η συγκεκριμένη βιβλιοθήκη στα Brackets και όχι μόνο προσδίδουν μεγάλη αξία.

Κεφάλαιο 4ο: Οδηγός χρήσης λογισμικού για τον συνεργάτη(μάγειρα) της εφαρμογής

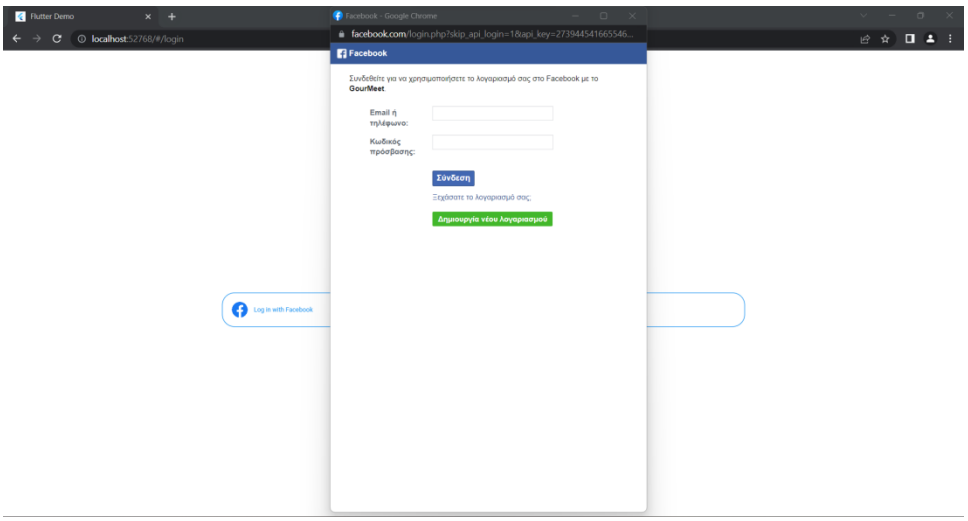
Η εφαρμογή που αναπτύχθηκε προσφέρει εργαλεία για τη διαχείριση του προφίλ του συνεργάτη της εφαρμογής. Μέσω της διαχείρισης του προφίλ ο διαχειριστής (admin) μπορεί να προσθέσει λεπτομέρειες για τις υπηρεσίες που προσφέρει όπως τις ώρες και τις μέρες που μπορεί να εξυπηρετήσει, το menu που προσφέρει μαζί με τις τιμές των αγαθών και την ώρα που θα διαρκέσει η προετοιμασία της παραγγελιάς.

4.1 Είσοδος στην εφαρμογή από πλευρά συνεργάτη

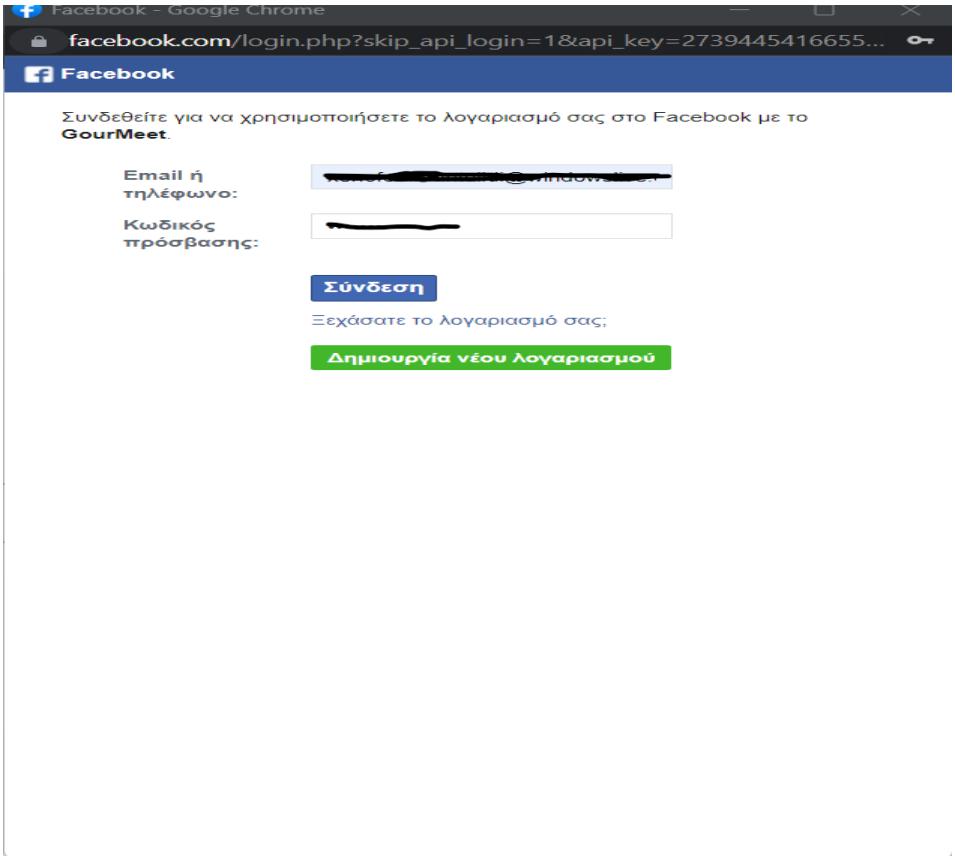
Ο διαχειριστής-συνεργάτης μπορεί να πραγματοποιήσει είσοδο στην εφαρμογή μόνο μέσω του λογαριασμού του στο Facebook από όπου θα πάρει τα στοιχεία του η εφαρμογή για την ταυτοποίηση του χρήστη. Ο χρήστης αρχικά θα πλοηγηθεί στην πρώτη οθόνη της εφαρμογής όπου εμφανίζεται ένα κουμπί και ο τίτλος της σελίδας, πατώντας ο χρήστης κουμπί θα ανοίξει ένα νέο παράθυρο μέσω του οποίου ο χρήστης συμπληρώνοντας το email και το Password του Facebook λογαριασμού του και πατώντας το κουμπί σύνδεση που εμφανίζεται μπορεί να έχει πρόσβαση στην εφαρμογή.



Εικόνα 4:1 Πρώτη οθόνη συνεργάτη για Login/Sign in μέσω Facebook

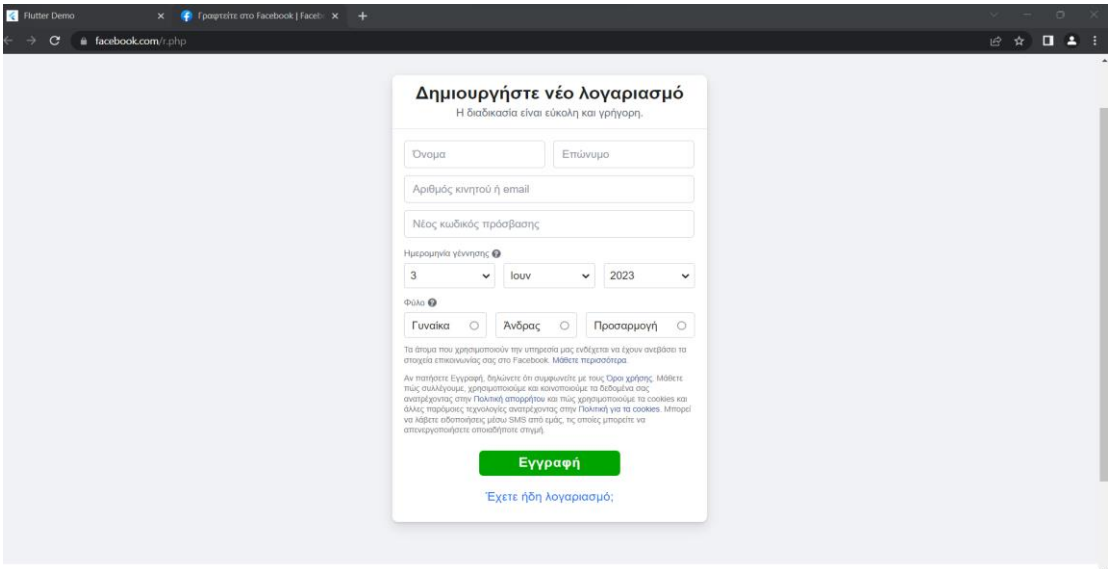


Εικόνα 4:2 Ξεκινώντας το LogIn/SignIn μέσω Facebook



Εικόνα 4:3 Login μέσω Facebook

Σε περίπτωση που ο χρήστης δεν έχει λογαριασμό Facebook ξεκινώντας πάλι από την αρχική σελίδα της εφαρμογής μπορεί να πατήσει το κουμπί και χωρίς να συμπληρώσει στοιχεία, να επιλέξει το κουμπί Δημιουργία νέου λογαριασμού. Μόλις επιλέξει το κουμπί θα ανοίξει καινούργια καρτέλα στο υπάρχον παράθυρο στο οποίο ο χρήστη μπορεί να συμπληρώσει τα στοιχεία του και πατώντας το κουμπί Εγγραφή να δημιουργήσει έναν νέο λογαριασμό στην κοινωνική πλατφόρμα του Facebook.

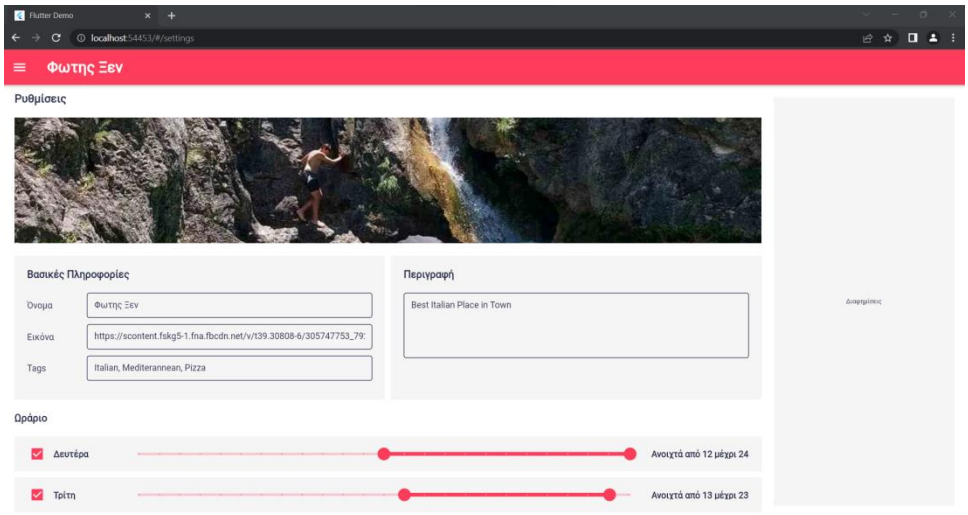


Εικόνα 4:4 Sign In μέσω Facebook

Συμπληρώνοντας ο χρήστης Email ή τηλέφωνο και τον κωδικό πρόσβασης και πατώντας το κουμπί σύνδεση εφόσον είναι σωστά τα στοιχεία και αντιστοιχούν σε λογαριασμό ο χρήστης θα μεταφερθεί στην πρώτη σελίδα της διαχείρισης του λογαριασμού του.

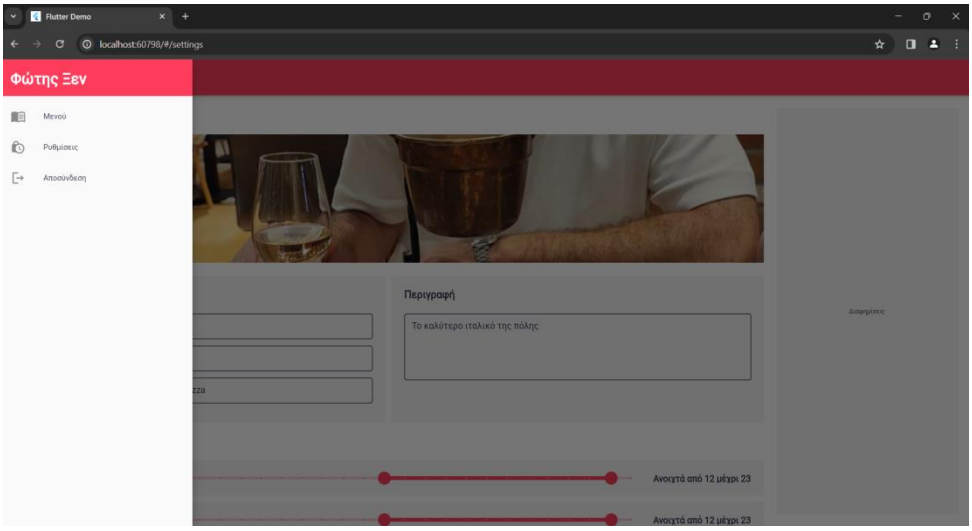
4.2 Περιβάλλον εφαρμογής

Εφόσον ο χρήστης ταυτοποιηθεί θα μπει στην αρχική σελίδα του λογαριασμού του στην οποία φαίνονται τα προϊόντα που έχει, οι κατηγορίες που υπάρχουν διαθέσιμες, που είναι για όλους του χρήστες κοινές, κάποιες διαφημίσεις που πιθανόν να προκύψουν όταν η εφαρμογή βγεί παραγωγικά καθώς και η δυνατότητα για προσθήκη ενός νέου προϊόντος. Εδώ να προστεθεί ότι το όνομα που εμφανίζεται είναι αυτό που έχει ο χρήστης στη Facebook και το παίρνει η εφαρμογή με το που γίνεται η είσοδος.



Εικόνα 4:5 Οθόνη Ρυθμίσεις συνεργάτη

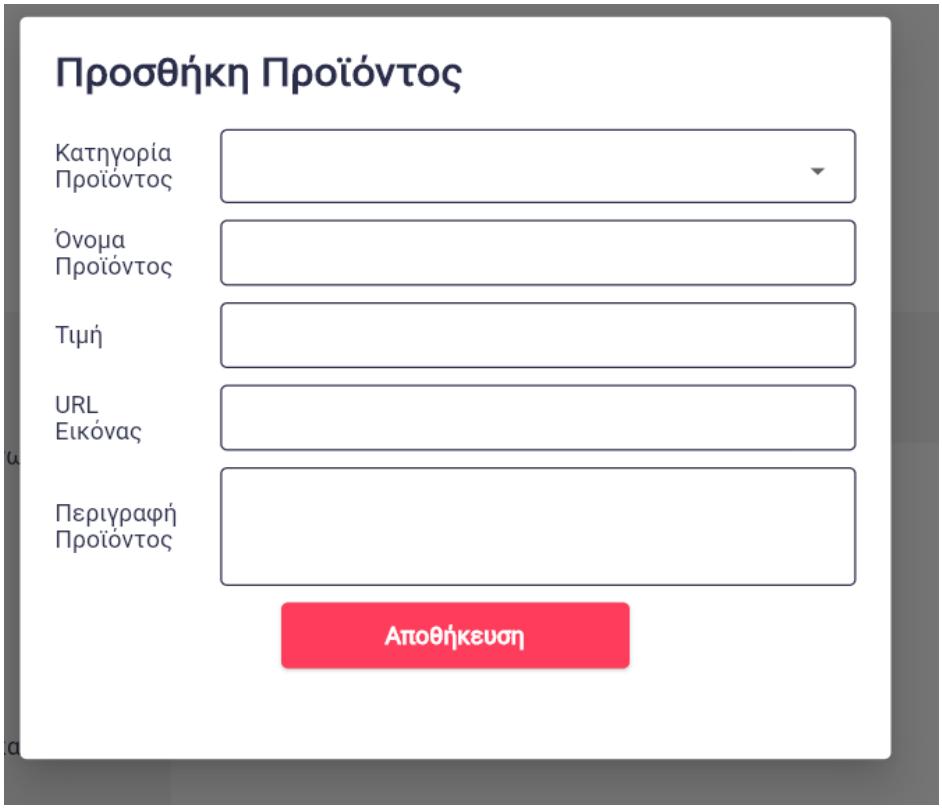
Στο επάνω μέρος της σελίδας, στα αριστερά δίπλα από το όνομα, εμφανίζεται το κουμπί του μενού όπου ο χρήστης μπορεί να πλοηγηθεί εκτός από την αρχική καρτέλα του μενού στην καρτέλα με τις ρυθμίσεις του λογαριασμού του όπου μπορεί να αλλάξει το όνομα του λογαριασμού του, την φωτογραφία του λογαριασμού του , τα αναγνωριστικά του φαγητού που προσφέρει (tags) καθώς και την περιγραφή όπου θα μπορεί να προσθέσει εξτρά πληροφορίες για τα προϊόντα του και όχι μόνο. Επίσης εκεί προσθέτει και τις μέρες και ώρες όπου είναι διαθέσιμος να προσφέρει αυτές του τις υπηρεσίες. Στο μενού υπάρχει και η επιλογή του για αποσύνδεση όπου ο χρήστης θα ανακατευθυνθεί στην σελίδα της εισόδου.



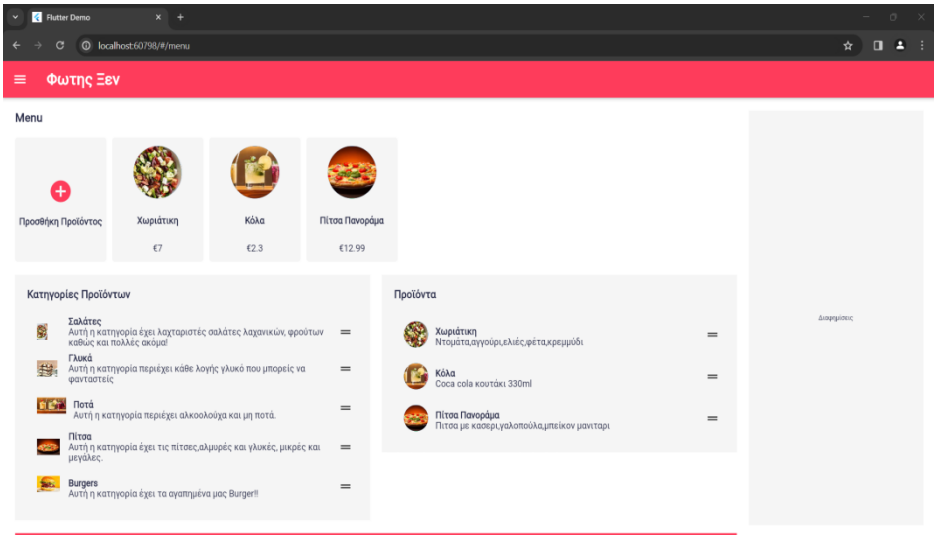
Εικόνα 4:6 Drawer Menu Συνεργάτη

4.3 Εισαγωγή προϊόντων

Ο χρήστης μπορεί να προσθέσει καινούργια προϊόντα πατώντας πάνω στο κουμπί προσθήκη προϊόντος. Μόλις το πατήσει θα του εμφανιστεί μία φόρμα την οποία μπορεί να συμπληρώσει ο χρήστης με τα στοιχεία του προϊόντος που θέλει να προσθέσει και πατώντας μετά το κουμπί Αποθήκευση θα αποθηκεύσει τη καινούργια προσθήκη και το pop-up menu θα εξαφανιστεί και το νέο προϊόν θα έχει εμφανιστεί και στη λίστα δίπλα από το κουμπί Προσθήκη Προϊόντος και στη κατηγορία Προϊόντα. Αλλάζοντας τη σειρά από το δεξί κουμπί στη κατηγορία Προϊόντα θα αλλάξει και η σειρά που εμφανίζονται τα προϊόντα δίπλα στο κουμπί προσθήκη προϊόντος.



Εικόνα 4:7 Εισαγωγή Προϊόντων

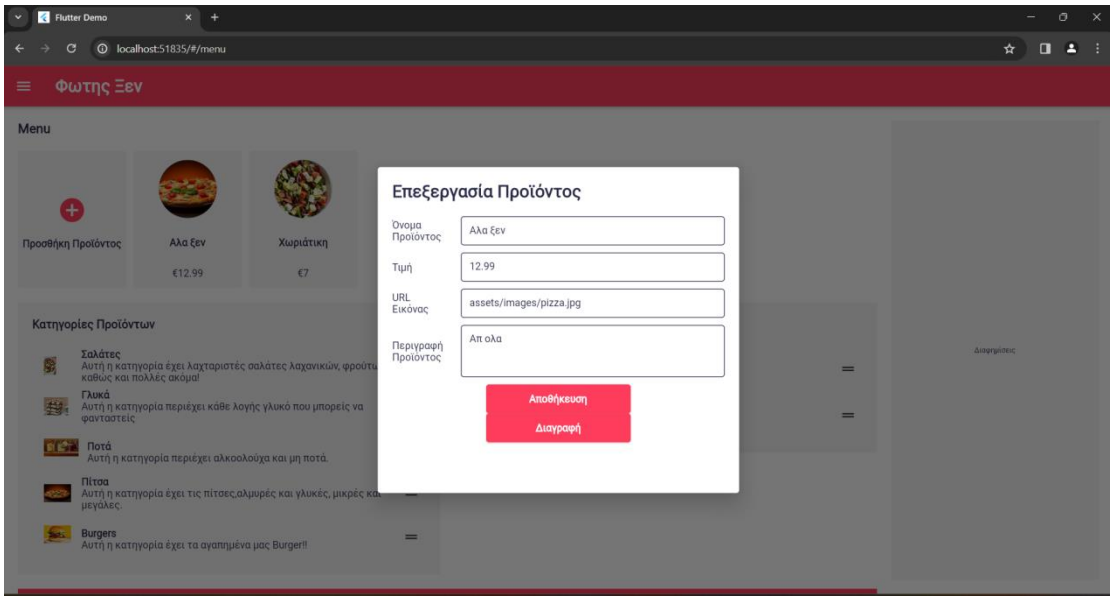


Εικόνα 4:8 Μενου Συνεργάτη

4.4 Επεξεργασία και διαγραφή προϊόντων

Ο χρήστης μπορεί κάνοντας διπλό κλικ πάνω στο προϊόν να δει τη καρτέλα του προϊόντος και να επεξεργαστεί τις διάφορες τιμές του ακόμα και να το διαγράψει πατώντας στο αντίστοιχο κουμπί.

Μετά την επεξεργασία χρειάζεται απλά να πατήσει την αποθήκευση για να αποθηκευτούν οι τιμές.



Εικόνα 4:9 Επεξεργασία/Διαγραφή Προϊόντος

Κεφάλαιο 5ο: Περιγραφή της εφαρμογής σε επίπεδο τελικού χρήστη

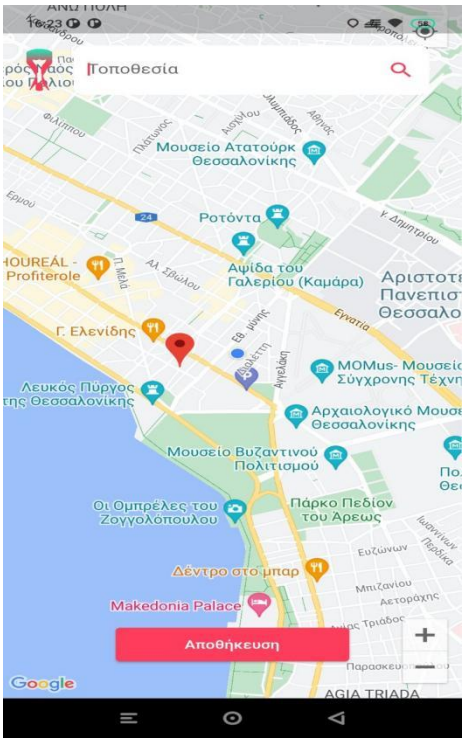
5.1 Εισαγωγή

Η εφαρμογή που αναπτύχθηκε δίνει ιδιαίτερη έμφαση στην εμπειρία πλοήγησης καθώς έχει ως στόχο την βέλτιστη εμπειρία του χρήστη. Οι χρήστες που θα χρησιμοποιήσουν την εφαρμογή ως εν δυνάμει πελάτες μπορούν να πλοηγηθούν εκτός από την κύρια σελίδα που περιλαμβάνει όλα τα κύρια χαρακτηριστικά που ένας χρήστης ψάχνει για να επιλέξει τον συνεργάτη της εφαρμογής που επιθυμεί και στις καρτέλες των ίδιων των συνεργατών για να ολοκληρώσει τη παραγγελία. Μέσα από την αρχική σελίδα μπορεί να φιλτράρει και να ταξινομήσει τις επιλογές των συνεργατών ανάλογα με τις ανάγκες του καθώς η εφαρμογή του δίνει μια πληθώρα επιλογών για να το πραγματοποιήσει. Τέλος , να αναφερθεί ότι το UI είναι προσαρμοσμένο να είναι responsive δηλαδή να προσαρμόζεται σε διάφορες διαστάσεις κινητών οθονών. Στη συνέχεια ακολουθεί μία ανάλυση των επιμέρους στοιχείων του περιβάλλοντος της εφαρμογής.

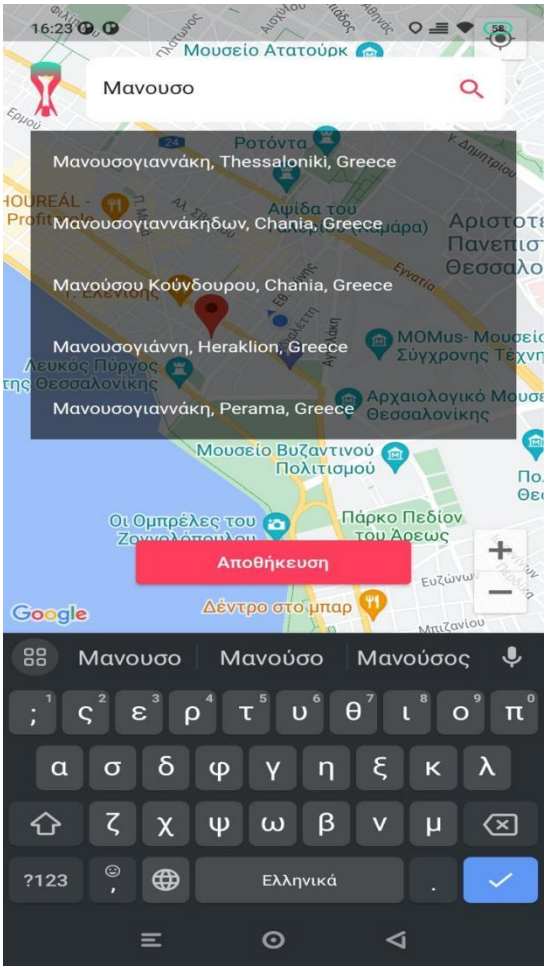
5.2 Περιήγηση στο περιβάλλον εφαρμογής

Όπως έχουμε ήδη αναφέρει, ο χρήστης μπορεί να πλοηγηθεί σε διάφορες σελίδες μέσα στην εφαρμογή ανάλογα με τη διαδικασία που διεκπεραιώνει κάθε φορά(π.χ. αναζήτηση μάγειρα, ολοκλήρωση παραγγελίας). Στην ενότητα αυτή περιγράφουμε αναλυτικά το περιεχόμενο που θα συναντήσει.

Ξεκινώντας , ο χρήστης βρίσκεται στην αρχική σελίδα της πλατφόρμας που περιέχει έναν χάρτη στον οποίο φαίνεται το στίγμα του χρήστη, αφού πρώτα έχει δώσει δικαιώματα στην εφαρμογή για πρόσβαση στις πληροφορίες τοποθεσίας, καθώς και οι διαθέσιμοι συνεργάτες της εφαρμογής σε απόσταση μέχρι και δέκα χιλιόμετρα από τον χρήστη. Ο χρήστης μπορεί να πλοηγηθεί στον χάρτη και να εισάγει την τοποθεσία του και πατώντας στο κουμπί αποθήκευση στο κάτω μέρος της σελίδας θα μεταφερθεί στην επόμενη σελίδα.Σε περίπτωση που βάλει κάποια άλλη τοποθεσία από αυτή που έχει πάρει το στίγμα στη συσκευή του θα εμφανιστούν και οι ανάλογοι συνεργάτες.



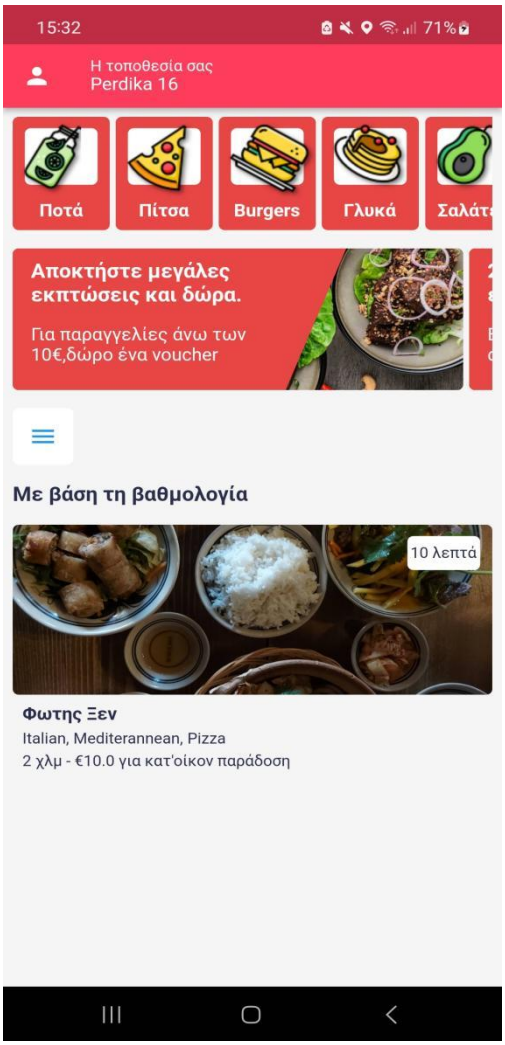
Εικόνα 5:1 Χάρτης Πελάτη



Εικόνα 5:2 Εισαγωγή Διεύθυνσης Πελάτη

Στη συνέχεια ο χρήστης έχοντας εισάγει την γεωγραφική του θέση έχει τη δυνατότητα να ξεκινήσει να ψάχνει τον/την μάγειρα/μαγείρισσα που του ταιριάζει προκειμένου να διεκπεραιώσει την παραγγελία του. Ο χρήστης θα μπορεί να δει μάγειρες σε σχέση με την τοποθεσία που έχει βάλει συγκεκριμένα σε απόσταση 10 χλμ από αυτήν, με αποτέλεσμα να προσδίδεται μεγαλύτερα λειτουργικότητα και αξία στην ιδιότητα της εφαρμογής που δίνει τη δυνατότητα στον χρήστη να παραγγείλει για το προσεχές μέλλον. Έτσι ο χρήστης για παράδειγμα μπορεί να έχει έτοιμη τη παραγγελία στο σπίτι του ,που βρίσκεται πχ 40χλμ μακριά από την εργασία του, ακόμα και αν την κάνει την ώρα που είναι στην δουλειά του κάποιες ώρες πριν.

Οι επιλογές του χρήστη του εμφανίζονται σε μία λίστα μία κάτω από την άλλη με δυνατότητα φυσικά την ταξινόμηση και φιλτράρισμα των αποτελεσμάτων για καλύτερη εμπειρία χρήστη. Η συγκεκριμένη οθόνη δίνει πολλές επιλογές στον χρήστη που θα αναλυθούν με ακρίβεια στη συνέχεια. Είναι η οθόνη όπου χρήστης ξεκινάει να στήνει σιγά σιγά την παραγγελία του επιλέγοντας αρχικά τον/την μάγειρα/μαγείρισσα.



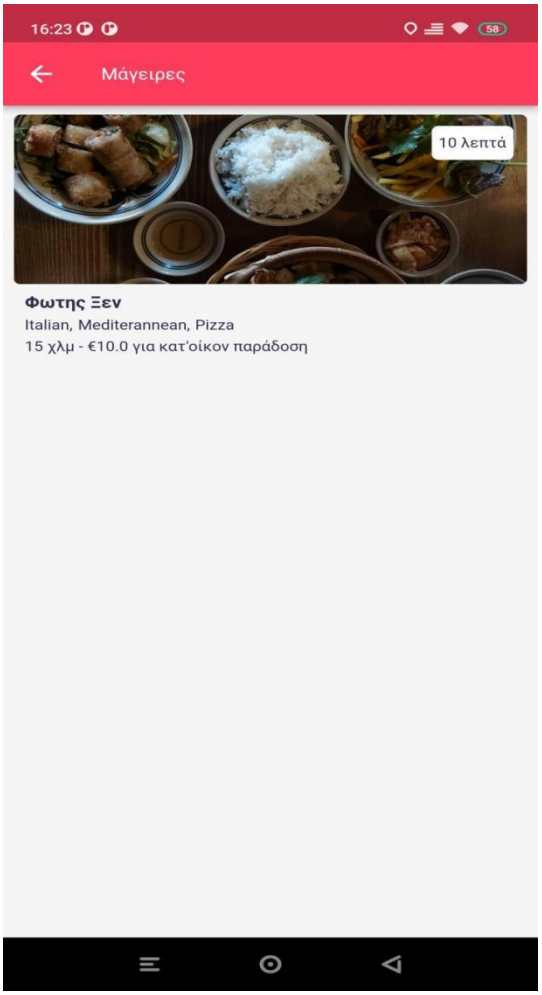
Εικόνα 5:3 Αρχική Οθόνη Πελάτη

Όπως φαίνεται στην εικόνα ηαρχική οθόνη εμπεριέχει αρκετά στοιχεία τα οποία θα αναλύσουμε αμέσως. Αρχικά στο πάνω αριστερά κομμάτι της οθόνης φαίνεται η επιλεγμένη τοποθεσία του χρήστη

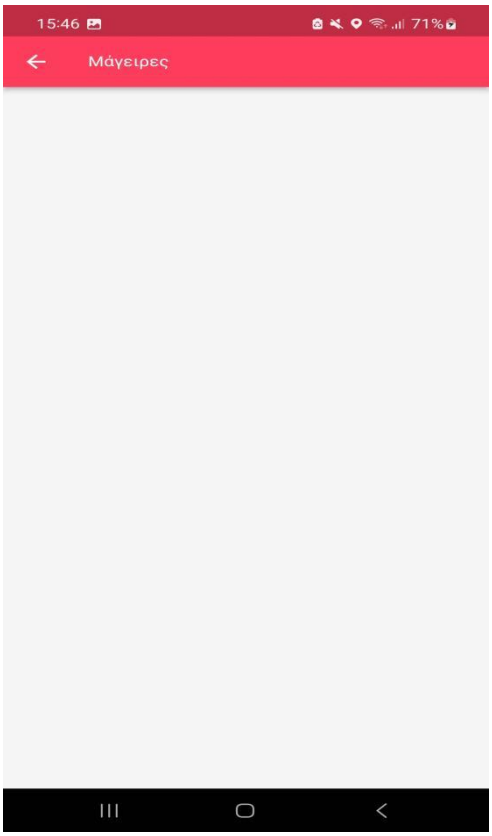
και πατώντας πάνω στο συγκεκριμένο κομμάτι της οθόνης ο χρήστης πηγαίνει στην οθόνη του χάρτη και μπορεί να επιλέξει κάποια άλλη τοποθεσία όπως ακριβώς και στην αρχή.

Από κάτω εμφανίζεται ένα μενού με κατηγορίες τις οποίες ο χρήστης μπορεί μεμονωμένα να επιλέξει ανάλογα με τις προτιμήσεις του. Σε περίπτωση που ο χρήστης επιλέξει οποιαδήποτε κατηγορία θα του εμφανιστούν οι συνεργάτες που ανήκουν σε αυτή την κατηγορία πάντα με το κριτήριο της απόστασης όπως και προηγουμένως. Εάν δεν υπάρχουν συνεργάτες που να πληρούν της προϋποθέσεις τότε θα του εμφανιστεί μία κενή λίστα όπως φαίνεται στις επόμενες φωτογραφίες.

Εικόνα 5:4 Φιλτράρισμα Συνεργατών



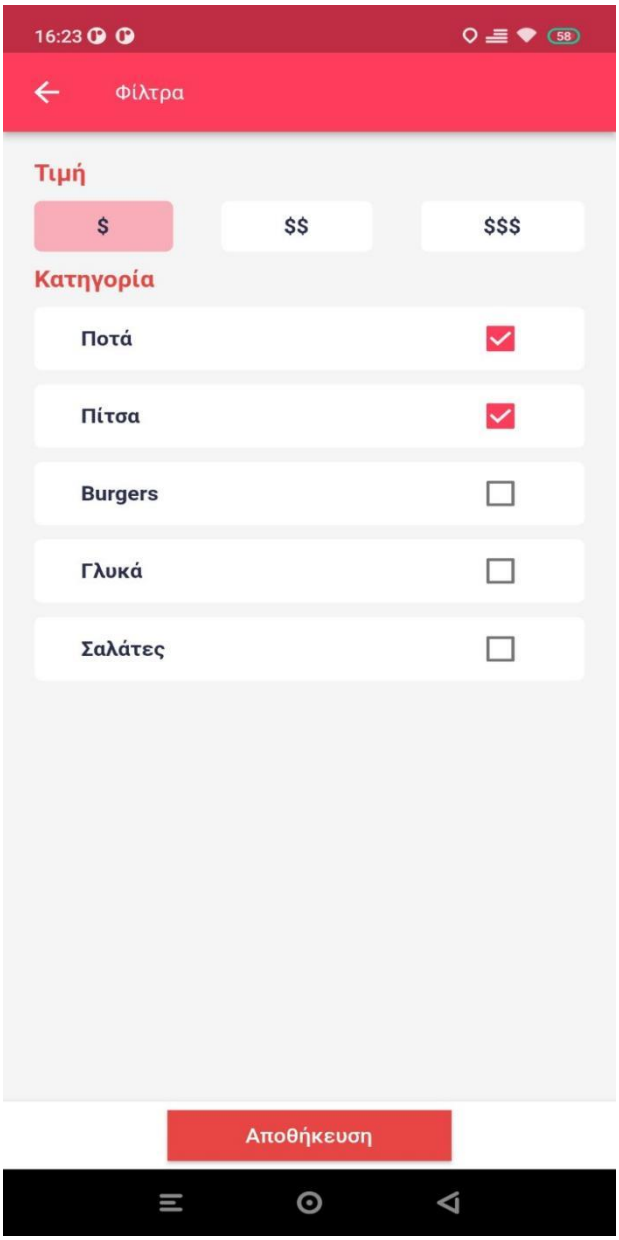
Εικόνα 5:5 Λίστα φιλτραρισμένων συνεργατών



Εικόνα 5:6 Άδεια Λίστα Συνεργατών

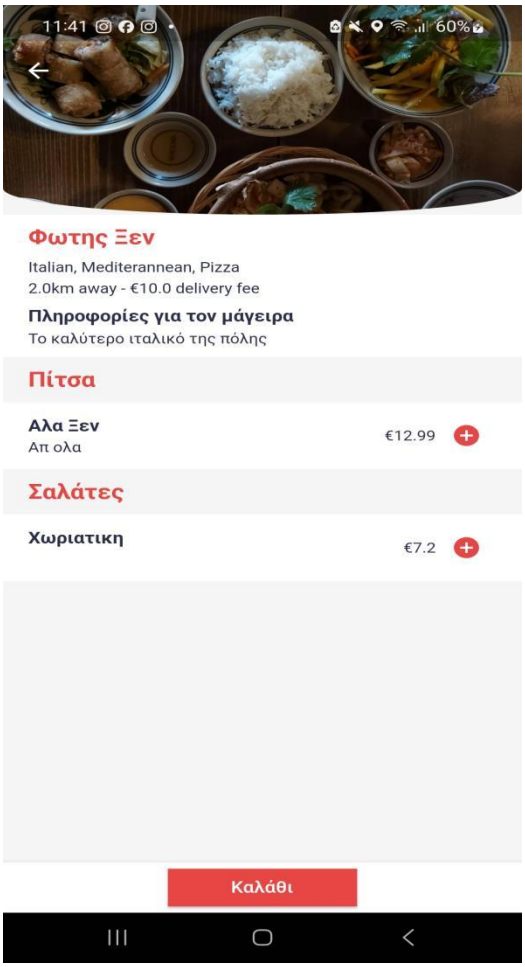
Ακριβώς κάτω από τη λίστα υπάρχει ένα Banner με προσφορές και Voucher όπου ο χρήστης μπορεί να σκρολάρει οριζόντια. Το συγκεκριμένο κομμάτι της οθόνης έχει διαφημιστικό και ενημερωτικό χαρακτήρα και έχει σκοπό να εξυπηρετεί τους συνεργάτες σε συνεννόηση με τον διαχειριστή της εφαρμογής.

Κατά σειρά από πάνω προς τα κάτω ακολουθάει ένα κουμπί για το φιλτράρισμα των συνεργατών - μαγείρων. Πατώντας πάνω στο κομπί αυτό ο χρήστης θα μεταφερθεί στην οθόνη φιλτραρίσματος όπου του δίνεται η επιλογή να φιλτράρει τις διαθέσιμες επιλογές του σύμφωνα με την τιμή (οι τιμές είναι ομαδοποιημένες σε τρείς κατηγορίες, $,$$,$$$ από το φθηνότερο στο ακριβότερο) αλλά και την κατηγορία του φαγητού ή ποτού που θέλει. Μπορεί να επιλέξει ένα η περισσότερα από καθεμία από όλες τις κατηγορίες και μετά πατώντας το κουμπί “Αποθήκευση” στο κάτω μέρος της οθόνης θα μεταφερθεί στην ίδια οθόνη φιλτραρισμένων συνεργατών, όπως στη προηγούμενη περίπτωση από το scrollable menu με τις μεμονωμένες κατηγορίες στο αρχικό Menu, για να δει και να επιλέξει τον συνεργάτη που του ταιριάζει.



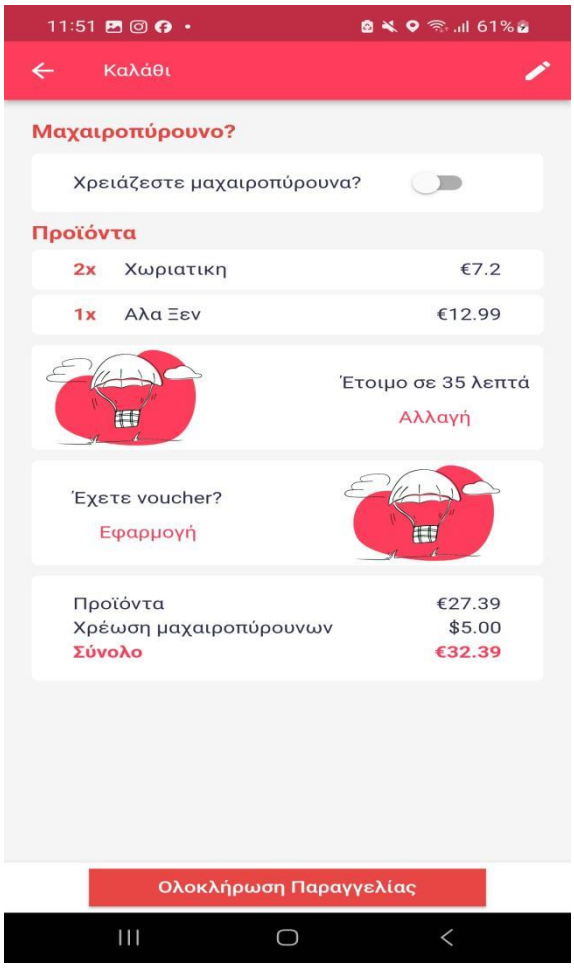
Εικόνα 5:7 Οθόνη Φιλτραρίσματος

Επιλέγοντας κάποιον από τους συνεργάτες μέσω της αρχικής οθόνης ή μέσω των φιλτραρισμένων επιλογών, ο χρήστης μεταφέρεται στην οθόνη με τις πληροφορίες του συνεργάτη που επέλεξε. Εκεί μπορεί να δει μια φωτογραφία του συνεργάτη καθώς και πληροφορίες για την απόσταση και το delivery fee καθώς και μία περιγραφή που έχει γράψει ο ίδιος ο συνεργάτης. Από κάτω φαίνεται και το Menu του συνεργάτη ανά κατηγορίες με το όνομα του προϊόντος καθώς και μία σύντομη περιγραφή από κάτω αλλά και την τιμή. Ο χρήστης μπορεί πατώντας το κουμπί δεξιά από την κάθε επιλογή να την προσθέσει στο καλάθι του μία η περισσότερες φορές, όσες επιθυμεί ο χρήστης. Στο τέλος της οθόνης εμφανίζεται η επιλογή να δεί ο χρήστης το καλάθι του όπου θα είναι οι επιλογές που έχει επιλέξει με το ομώνυμο κουμπί “Καλάθι”.

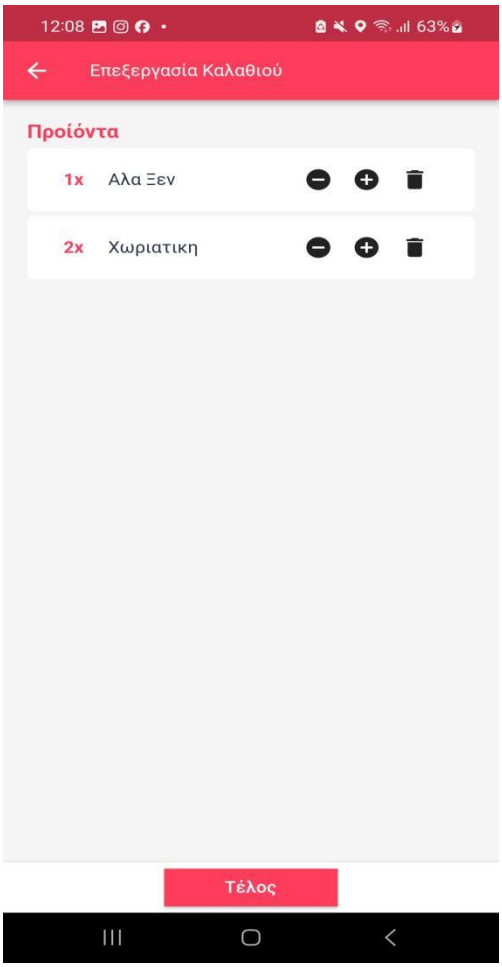


Εικόνα 5:8 Οθόνη συνεργάτη

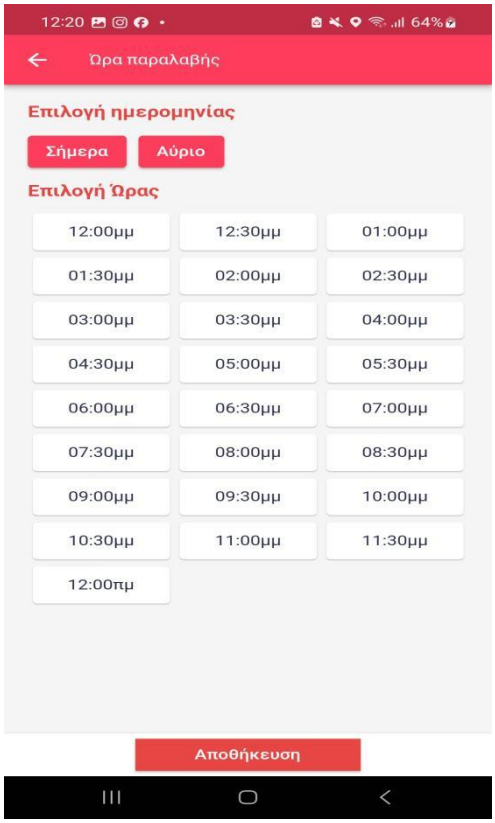
Πατώντας στο καλάθι μπαίνει στην αντίστοιχη οθόνη όπου εμφανίζονται αρκετές πληροφορίες και επιλογές. Αρχικά του δίνεται η επιλογή για μαχαιροπίρουνο η οποία έχει σταθερή χρέωση. Σε περίπτωση που την επιλέξει στο σύνολο της παραγγελίας θα προστεθούν 5€ καθώς και γιατί αυτά προστέθηκαν. Ακριβώς από κάτω εμφανίζεται η λίστα με τα προϊόντα που έχει προσθέσει στο καλάθι και αριστερά τους η ποσότητα του καθενός και στη δεξιά στήλη το ποσό της κάθε μονάδας. Η επεξεργασία της ποσότητας των προϊόντων, είτε σημαίνει περισσότερα, είτε λιγότερα ή και διαγραφή γίνεται από το κουμπί πάνω δεξιά της οθόνης (το μολυβάκι) που οδηγεί τον χρήστη σε καινούργια οθόνη με την λίστα των ήδη επιλεγμένων προϊόντων μαζί με τη ποσότητα του καθενός και τρείς επιλογές με τρία κουμπιά στα δεξιά. Το πρώτο είναι για μείωση της ποσότητας του προϊόντος κατά μία μονάδα ανά πάτημα, το δεύτερο για αύξηση της ποσότητας του προϊόντος ανά πάτημα και το τρίτο για διαγραφή του προϊόντος από τη λίστα. Αφού ο χρήστης τελειώσει με τις αλλαγές πατώντας το κουμπί τέλος στο κάτω μέρος της οθόνης μεταφέρεται στην προηγούμενη οθόνη (Καλάθι) έχοντας πλέον ανανεωμένο το καλάθι του. Περνώντας στο αμέσως επόμενο κομμάτι της οθόνης ο χρήστης βλέπει σε πόση ώρα μπορεί να είναι έτοιμη η παραγγελία του και κάτω από τον χρόνο ένα κουμπί “Αλλαγή”, όπου πατώντας το μεταφέρεται σε μία άλλη οθόνη στην οποία μπορεί να επιλέξει πότε θα γίνει η παραλαβή. Συγκεκριμένα μπορεί να επιλέξει μεταξύ δύο ημερών (σήμερα, αύριο) καθώς και συγκεκριμένης ώρας παράδοσης ανά μισή ώρα από τις ώρες που έχει ο συνεργάτης ότι είναι διαθέσιμος. Επιλέγοντας ημέρα και ώρα και μετά το κουμπί στο τέλος της οθόνης αποθήκευση μεταφέρεται στην οθόνη “Καλάθι” όπου του εμφανίζεται η πληροφορία που έχει επιλέξει καθώς και η δυνατότητα αλλαγής της ώρας και ημέρας ξανά.



Εικόνα 5:9 Οθόνη καλαθιού

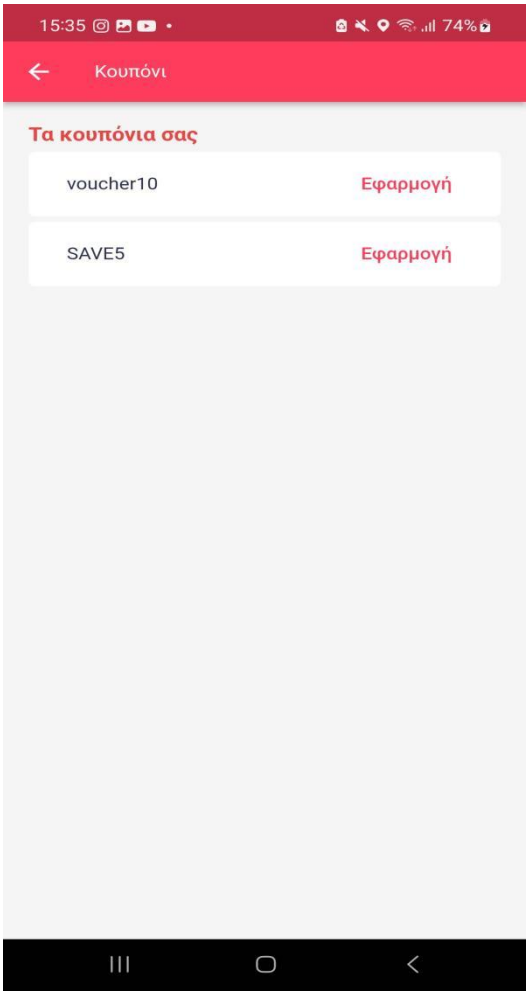


Εικόνα 5:10 Επεξεργασία προϊόντων καλαθιού

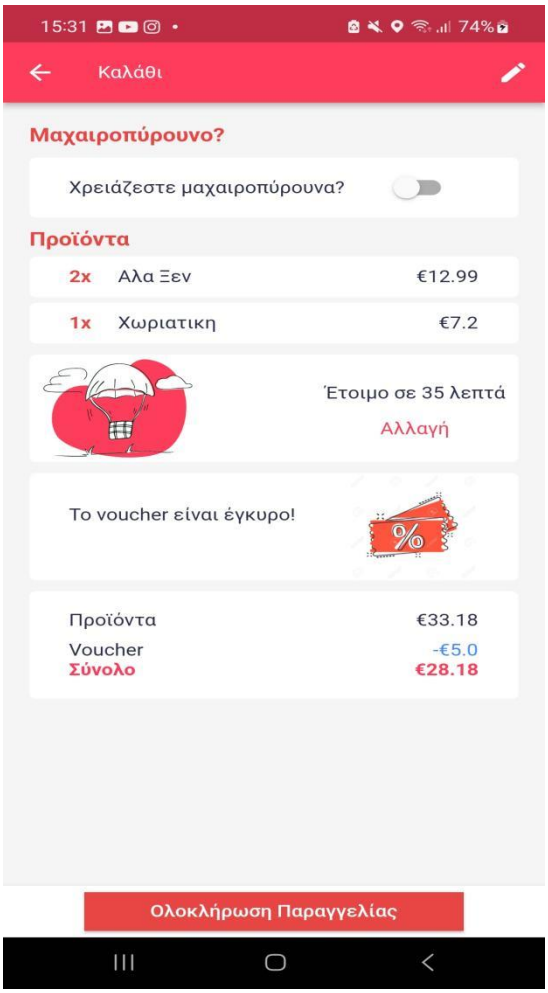


Εικόνα 5:11 Οθόνη ώρας παραλαβής

Καθώς κατεβαίνουμε στην οθόνη το επόμενο τμήμα που βλέπουμε είναι για εκπτωτικό κουπόνι (voucher). Ο χρήστης ερωτάται εάν έχει voucher και πατώντας το κουμπί “Εφαρμογή” του δίνεται η δυνατότητα να προσθέσει το δικό του. Πατώντας το κουμπί πηγαίνει σε καινούργια οθόνη. Στην καινούργια οθόνη εμφανίζονται τα διαθέσιμα κουπόνια με το όνομα του κωδικού καθώς και το κουμπί “Εφαρμογή“ που τα ενεργοποιεί. Μόλις πατηθεί η εφαρμογή, ο χρήστης μεταφέρεται στην οθόνη “Καλάθι” και εκεί του εμφανίζεται το μήνυμα “Το Voucher είναι έγκυρο” καθώς ταυτόχρονα αφαιρείται το αντίστοιχο ποσό από το σύνολο του καλαθιού του.

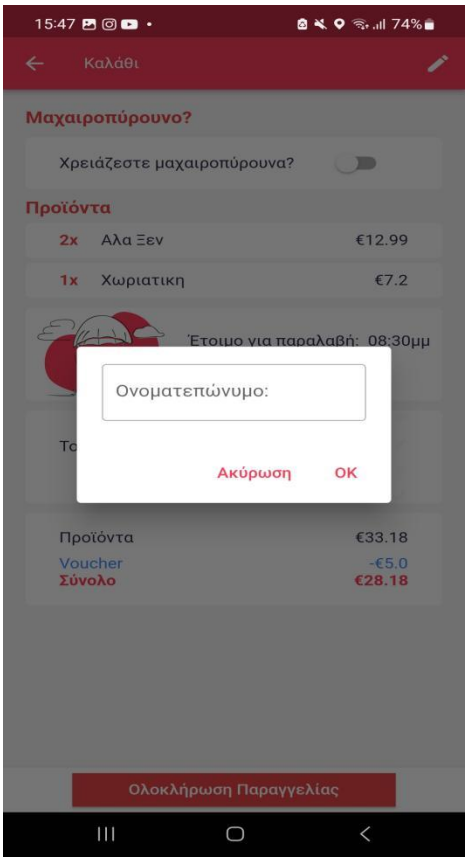


Εικόνα 5:12 Οθόνη voucher

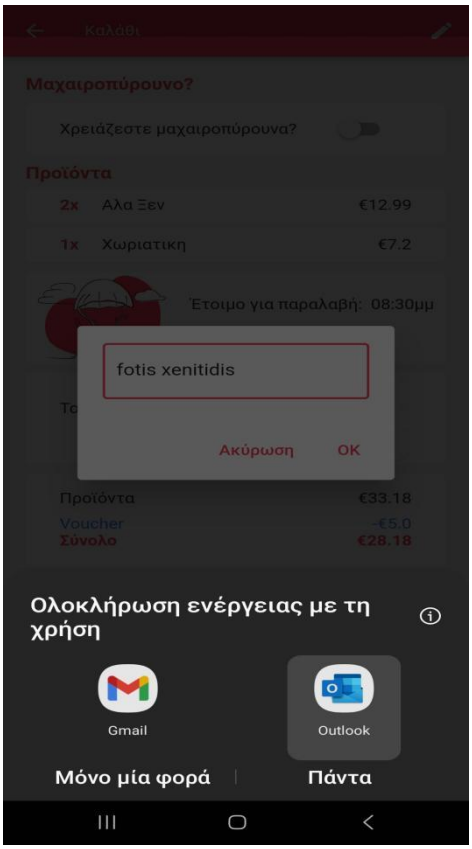


Εικόνα 5:13 Καλάθι με voucher και ώρα

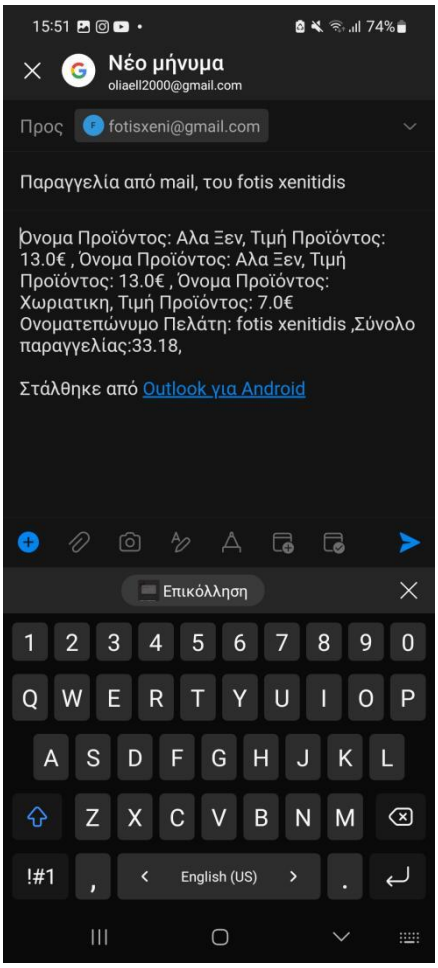
Ο χρήστης πλέον έχει σχεδόν ολοκληρώσει τη παραγγελία του, έχει προσθέσει μαχαιροπίρουνο, προϊόντα, εκπτωτικά κουπόνια και ξέρει και το τελικό ποσό της παραγγελίας ,το μόνο που μένει είναι να την στείλει. Το τελευταίο κουμπί της σελίδας είναι αυτό που θα ολοκληρώσει την παραγγελία όπως γράφει. Πατώντας το ο χρήστης θα εμφανιστεί ένα pop up παράθυρο που του ζητάει ένα ονοματεπώνυμο. Αφού το συμπληρώσει και πατήσει “OK” θα τον ερωτηθεί με ποιόν πάροχο θέλει αν συνεχίσει (Gmail,Outlook). Μόλις επιλέξει τον πάροχο, θα μεταφερθεί στον συγκεκριμένο πάροχο Ηλεκτρονικής Αλληλογραφίας και θα έχει έτοιμο ένα συντεταγμένο mail με παραλήπτη (το email του συγκεκριμένου συνεργάτη), θέμα (“Παραγγελία από Mail, του “ και το όνομα που συμπλήρωσε στο pop up παράθυρο) και σώμα mail τα προϊόντα του καλαθιού, το όνομα του χρήστη αλλά και το συνολικό πόσο παραγγελίας. Το μόνο που μένει είναι να πατήσει το κουμπί αποστολή και φυσικά αν θελήσει μπορεί να στείλει και κάποια σχόλιο σχετικά με τη παραγγελία.



Εικόνα 5:14 Εισαγωγή χρηστη



Εικόνα 5:15 Ολοκλήρωση Ενέργειας με πάροχο



Εικόνα 5:16 Αποστολή Mail

Κεφάλαιο 6ο: Οδηγός λογισμικού για προγραμματιστή

Σε αυτό το κεφάλαιο περιγράφουμε το μοντέλο ανάπτυξης της εφαρμογής από τη μεριά του προγραμματιστή. Το σχεδιαστικό πρότυπο που ακολουθεί η εφαρμογή είναι BLOC(Business Logic Component(Επιειρηματική λογική)) η οποία είναι ένα πρότυπο σχεδίασης που χρησιμοποιείται για τη διαχείριση της κατάστασης και τον διαχωρισμό της επιχειρηματικής λογικής από το επίπεδο UI. Βοηθά στην οργάνωση και τη δόμηση των εφαρμογών Flutter με πιο συντηρήσιμο και επεκτάσιμο τρόπο. Το Bloc είναι ένα συστατικό που είναι υπεύθυνο για τη διαχείριση της επιχειρηματικής λογικής της εφαρμογής. Λαμβάνει συμβάντα εισόδου, τα επεξεργάζεται και εκπέμπει καταστάσεις εξόδου. Είναι ανεξάρτητο από το UI και μπορεί να επαναχρησιμοποιηθεί σε διάφορα μέρη της εφαρμογής.Με την υιοθέτηση της αρχιτεκτονικής του προτύπου Bloc, μπορούμε να επιτύχομε διαχωρισμό των ανησυχιών, να βελτιώσουμε τη συντηρησιμότητα του κώδικα και να διευκολύνουμε τις δοκιμές. Ενθαρρύνει μια προσέγγιση αντιδραστικού προγραμματισμού, όπου οι ενημερώσεις του UI ενεργοποιούνται σε απόκριση σε αλλαγές στην κατάσταση της εφαρμογής, οδηγώντας σε έναν πιο προβλέψιμο και αποτελεσματικό σχεδιασμό εφαρμογών.

6.1 Εκτέλεση των εφαρμογών:

6.1.1 Εφαρμογή Συνεργάτη:

Όπως έχουμε ήδη αναφέρει και οι δύο εφαρμογές έχουν αναπτυχθεί στο Visual Studio Code και για να τρέξουμε την εφαρμογή των συνεργατών θα χρειαστούμε server hosting, με τη εντολή "flutter build web" στον φάκελο build/web, οπότε θα παραχθούν τα αρχεία που θα πρέπει όλα μαζί να "ανέβουν" στον server. Αλλιώς, μπορούμε απλά μέσω του IDE (Visual Studio Code) πατώντας F5 ή Ctrl+ F5 ή μέσω του Μενού που μας δίνεται επιλέγοντας Run και όποια από τις δύο πρώτες επιλογές θέλουμε (Start Debugging/Run without Debugging)για να τρέξουμε σε web browser την εφαρμογή. Τρέχοντας τη συγκεκριμένη εφαρμογή θα γίνει κανονικά η διασύνδεση με την Firebase βάση δεδομένων και ο συνεργάτης θα μπορεί κανονικά να κάνει τις απαραίτητες ενέργειες όπως φαίνονται και περιγράφονται στις φωτογραφίες των προηγούμενων κεφαλαίων.

Εάν θέλουμε να σταματήσουμε να τρέχουμε την εφαρμογή μπορούμε να γράψουμε στο τερματικό απλά την εντολή Ctrl + C.

6.1.2 Εφαρμογή Τελικού Χρήστη:

Για να τρέξει την εφαρμογή στην android συσκευή του, ο χρήστης χρειάζεται να έχει το Gourmeet.apk αρχείο το οποίο παράγεται όταν στο τερματικό, στο φάκελο της εφαρμογής, δώσουμε την εντολή flutter build apk. Το συγκεκριμένο αρχείο εμπεριέχει όλα τα αρχεία και τους συνδέσμους διασύνδεσης με τη βάση και ότι άλλο χρειάζεται η εφαρμογή για να λειτουργήσει κανονικά και πλήρως. Ο χρήστης πατώντας πάνω στο συγκεκριμένο αρχείο από τη android συσκευή του θα του βγάλει το μήνυμα επιβεβαίωσης για την εγκατάσταση της εφαρμογής και αφού εγκατασταθεί και ανοιχτεί την πρώτη φορά θα ζητηθεί δικαίωμα στην τοποθεσία χρήστη. Εφόσον ο χρήστης πατήσει την επιλογή "Κατά τη χρήση της εφαρμογής" δε θα ξαναζητηθεί αυτό το δικαίωμα.

6.2 Δομή των αρχείων και της διαδικτυακής εφαρμογής και του android app

Όλα τα αρχεία βρίσκονται μέσα στον φάκελο του Project flutter_foodapp_backend. Όπως έχουμε αναφέρει το Flutter μας επιτρέπει να δημιουργούμε εφαρμογές για κινητά, web, desktop και embedded συσκευές - όλα από μια ενιαία βάση κώδικα για αυτό τον λόγο υπάρχουν αρχεία για όλες αυτές τις κατηγορίες. Για να γίνει μια εφαρμογή cross-platform θα πρέπει να προστεθούν τα κατάλληλα Plug-ins - βιβλιοθήκες και να γίνουν οι κατάλληλες τροποποιήσεις όσο αναφορά και το responsiveness της εφαρμογής αλλά όχι μόνο. Πολλές βιβλιοθήκες απευθύνονται μόνο σε έναν τύπο συσκευής η λειτουργικού. Η συγκεκριμένη εφαρμογή έχει αναπτυχθεί μόνο σαν web app. Το ίδιο ισχύει και για την android εφαρμογή όπου αναπτύχθηκε αποκλειστικά και μόνο για android συσκευές.

Πρώτος φάκελος εμφανίζεται ο .dart/tool που είναι μια διεπαφή γραμμής εντολών για το Dart SDK. Το εργαλείο είναι διαθέσιμο ανεξάρτητα από τον τρόπο με τον οποίο αποκτάμε το Dart SDK - είτε κατεβάζουμε ρητά το Dart SDK είτε κατεβάζουμε μόνο το Flutter SDK.

Αυτός ο φάκελος δημιουργείται από το ολοκληρωμένο περιβάλλον ανάπτυξης JetBrains IntelliJ IDEA (ή Visual Studio ή Code Android Studio το οποίο βασίζεται επίσης στο IntelliJ IDEA). Περιέχει αρχεία διαμόρφωσης και ρυθμίσεις ειδικά για το έργο μας Flutter όταν χρησιμοποιούμε αυτά τα IDE για την ανάπτυξη του Flutter.

Έπειτα είναι ο φάκελος android ο οποίος περιέχει τα ειδικά για το Android αρχεία και τους πόρους και τους πόρους για την Flutter App. Περιέχει πράγματα όπως το αρχείο manifest της εφαρμογής, τους πόρους και τα σενάρια κατασκευής.

Οι εφαρμογές Flutter μπορούν να περιλαμβάνουν τόσο κώδικα όσο και στοιχεία ενεργητικού (που μερικές φορές ονομάζονται πόροι(assets). Ένα περιουσιακό στοιχείο είναι ένα αρχείο το οποίο συνδυάζεται και αναπτύσσεται με την εφαρμογή και είναι προσβάσιμο κατά την εκτέλεση. Οι συνήθεις τύποι περιουσιακών στοιχείων περιλαμβάνουν στατικά δεδομένα (για παράδειγμα, αρχεία JSON), αρχεία ρυθμίσεων, εικονίδια και εικόνες (JPEG, WebP, GIF, κινούμενα WebP/GIF, PNG, BMP και WBMP) και ανήκουν στον φάκελο assets. Στη συγκεκριμένη περίπτωση ο φάκελος αυτός εμπεριέχει μόνο φωτογραφίες, σε διάφορες μορφές.

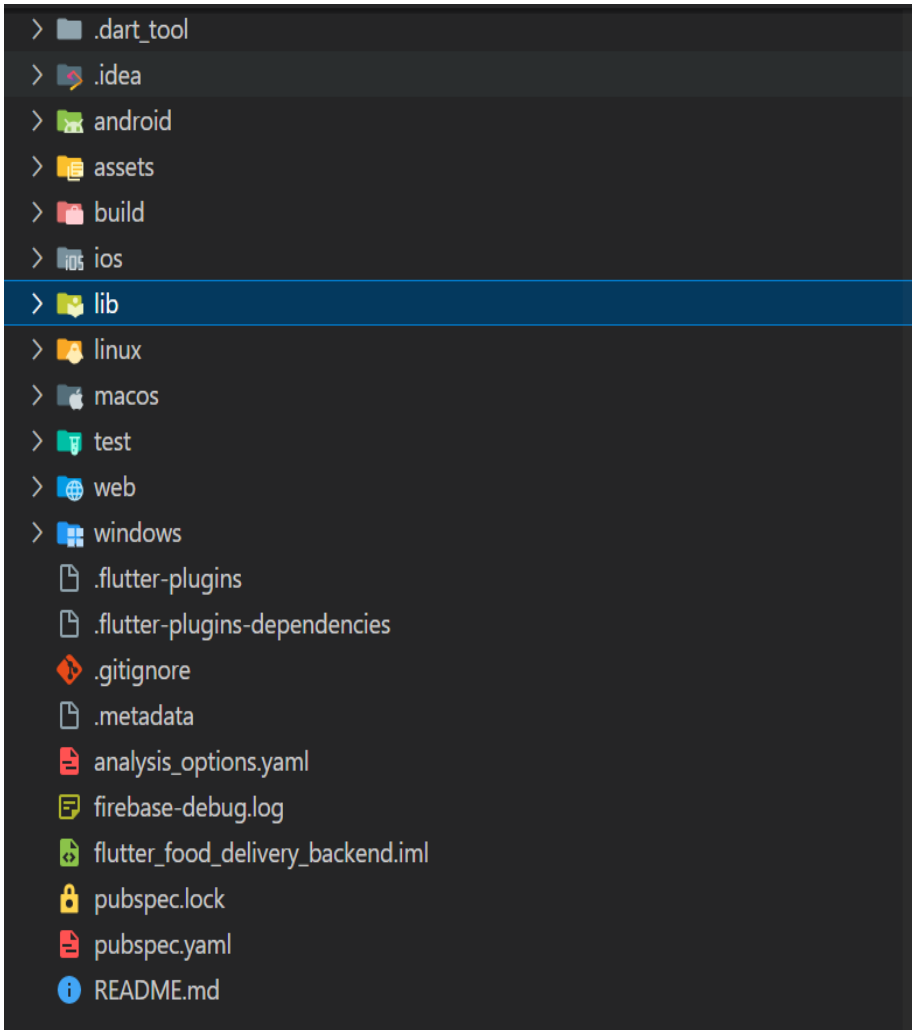
Ο φάκελος linux είναι όπως οι φάκελοι android, web, windows και iOS. Περιέχουν αρχεία ειδικά για την πλατφόρμα, αρχεία που απαιτούνται για τη δημιουργία και την εκτέλεση της εφαρμογής στην απαιτούμενη πλατφόρμα.

Ο φάκελος build θα δημιουργηθεί όταν τα εργαλεία κατασκευής του Flutter εκτελεστούν για πρώτη φορά. Περιέχει τα παραγόμενα αρχεία που απαιτούνται για την εκτέλεση της εφαρμογής στις διάφορες πλατφόρμες. Κάθε πλατφόρμα έχει το δικό της υποφάκελο.

Η γλώσσα dart περιέχει το ίδιο το testing και βρίσκεται μέσα στο φάκελο test. Γενικά, τα αρχεία δοκιμών πρέπει να βρίσκονται μέσα σε ένα φάκελο δοκιμών που βρίσκεται στη ρίζα της εφαρμογής ή του πακέτου Flutter. Τα αρχεία δοκιμών θα πρέπει πάντα να τελειώνουν με _test. dart , αυτή είναι η σύμβαση που χρησιμοποιείται από το test runner κατά την αναζήτηση δοκιμών.

Το README.md είναι το αρχείο οδηγιών χρήσης του κώδικα για προγγραματιστές, σε αυτό το αρχείο οι προγραμματιστές μπορούν να σημειώσουν οτιδήποτε θέλουν να θυμούνται για την εφαρμογή , να ενημερώσουν άλλους συναδέλφους που θα ενασχοληθούν με το Project και γενικά διάφορες πληροφορίες χρήσιμες για την εφαρμογή στο κομμάτι του κώδικα και των εννοιών του.

Τα αρχεία pubspec καθορίζουν εξαρτήσεις που απαιτεί το έργο, όπως συγκεκριμένα πακέτα (και τις εκδόσεις τους), γραμματοσειρές ή αρχεία εικόνας. Καθορίζουν επίσης άλλες απαιτήσεις, όπως εξαρτήσεις από πακέτα προγραμματιστών (όπως πακέτα testing ή mocking) ή συγκεκριμένους περιορισμούς στην έκδοση του Flutter SDK.

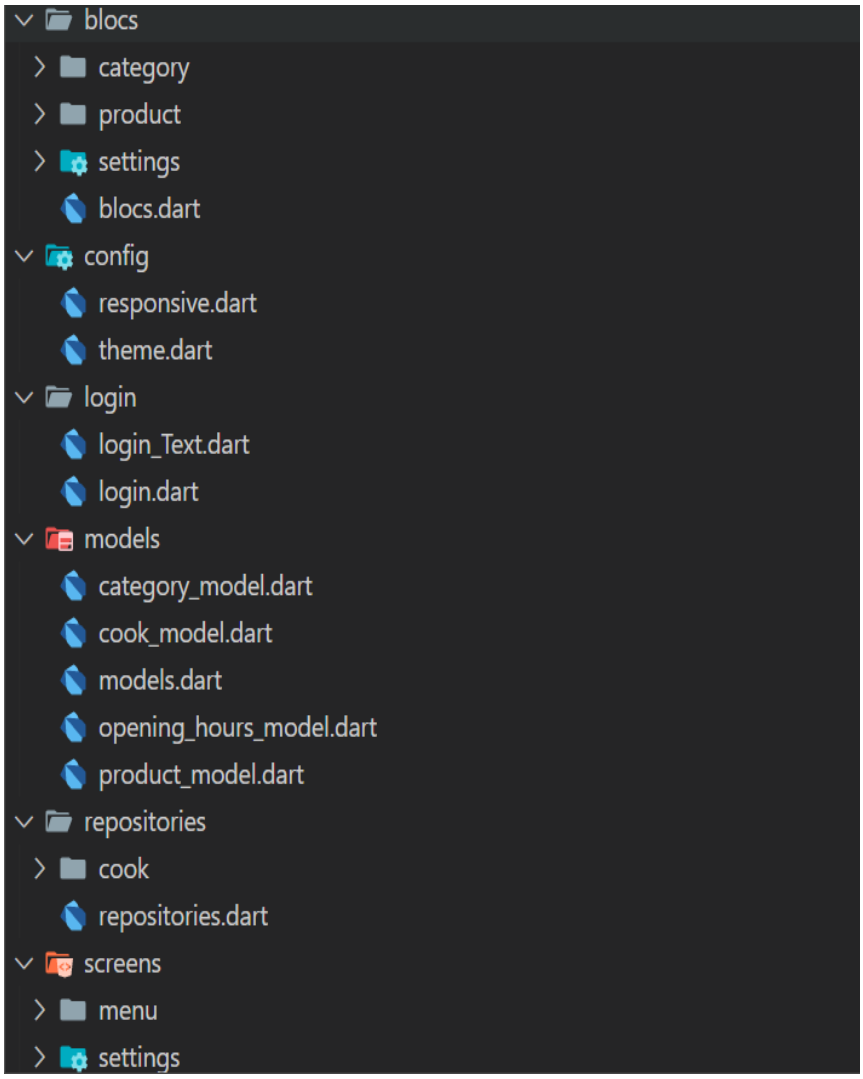


Εικόνα 6:1 Λίστα φακέλων και των δύο εφαρμογών

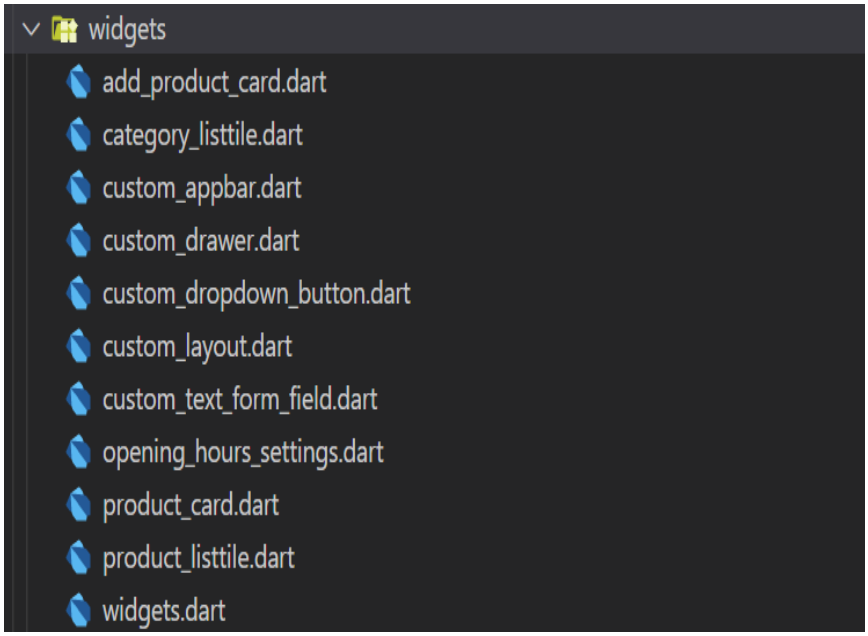
6.2.1 Lib file συνεγάτη

Οι μεγαλύτερες διαφοροποιήσεις μεταξύ των δύο εφαρμογών βρίσκονται στον φάκελο lib καθώς είναι ο κατάλογος που χρησιμεύει ως η ρίζα της βάσης κώδικα της εφαρμογής. Όλος ο κώδικάς βρίσκεται εδώ. Αυτός ο κατάλογος περιέχει όλο τον κώδικα που σχετίζεται με την εφαρμογή. Πάνω σε αυτό τον φάκελο θα μπούμε λίγο πιο βαθιά για να εξηγήσουμε πως είναι δομημένος ο κώδικας στην εφαρμογή. Το main.dart είναι ένα κρίσιμο αρχείο στο έργο , καθώς χρησιμεύει ως σημείο εισόδου για την εφαρμογή. Αυτό το αρχείο περιέχει τη συνάρτηση main(), η οποία είναι υπεύθυνη για την εκκίνηση της εφαρμογής εκτελώντας τον κώδικα που δημιουργεί το UI και αρχικοποιεί οποιαδήποτε άλλα απαραίτητα στοιχεία. Το αρχείο firebase_options περιέχει μεταβλητές και συναρτήσεις για την διασύνδεση με την βάση της εφαρμογής που είναι σε firebase και είναι παραγόμενο αυτόματα.

Στις παρακάτω εικόνες φαίνονται οι υποφάκελοι που περιέχει το Lib αρχείο της εφαρμογής του συνεργάτη καθώς και τι περιέχουν αυτοί.



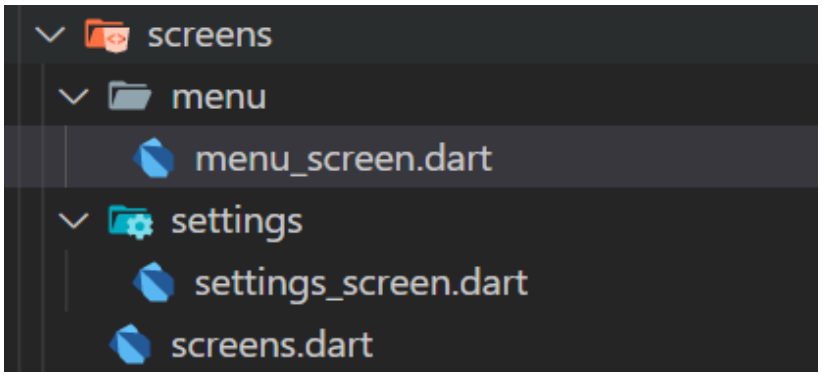
Εικόνα 6:2 Υποφάκελοι Συνεργάτη



Εικόνα 6:3 Widgets φάκελος συνεργάτη

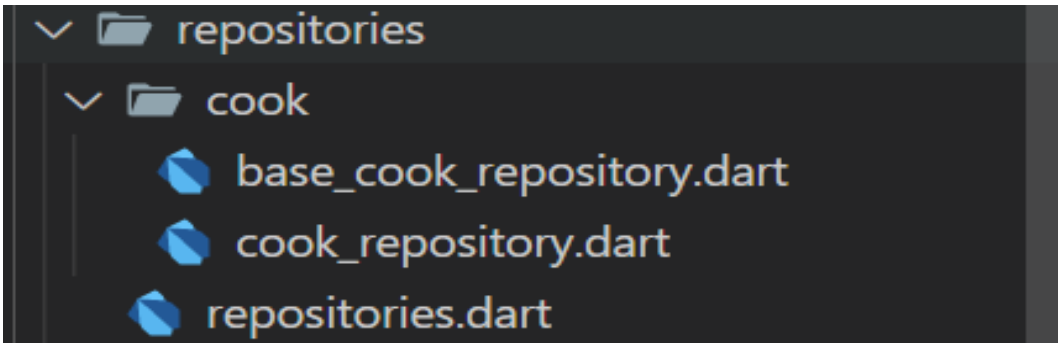
Από κάτω προς τα πάνω θα γίνει η ανάλυση των υποφακέλων. Αρχικά έχουμε το φάκελο widgets όπου περιέχει τα μεμονωμένα widgets που περιέχει η εφαρμογή. Τα widgets είναι τα συστατικά στοιχεία των εφαρμογών flutter που ουσιαστικά αποτελούνται από πολλά widgets το ένα δίπλα στο άλλο αλλά και το ένα μέσα στο άλλο. Για παράδειγμα μια οθόνη ξεκινάει συχνά με το widget Scaffold το οποίο πιάνει όλη την οθόνη και μετά τα υπόλοιπα widget μπαίνουν μέσα σε/πάνω από αυτό. Τα widgets μπαίνουν σε διαφορετικό φάκελο γιατί αυτό κάνει την εφαρμογή πιό εύκολη στην κατανόηση, εύκολη σε μελλοντικές επεκτάσεις και διορθώσεις. Τα ονόματα των widgets αντικατοπτρίζουν τη λειτουργικότητα τους ή την οθόνη στην οποία ανήκουν, συχνά αυτά τα δύο είναι το ίδιο. Τα widgets είναι stateless αρχεία που δεν απαιτούν μεταβλητή κατάσταση. Ένα stateless widget είναι ένα widget που περιγράφει μέρος της διεπαφής χρήστη δημιουργώντας έναν πακέτο άλλων widgets που περιγράφουν τη διεπαφή χρήστη πιο συγκεκριμένα. Η διαδικασία δημιουργίας συνεχίζεται αναδρομικά έως ότου η περιγραφή της διεπαφής χρήστη γίνει πλήρως συγκεκριμένη.

Αμέσως μετά είναι ο φάκελος screens όπου περιέχει το αρχείο screens.dart το οποίο απλά κάνει export τις δύο οθόνες που είναι στους άλλους δύο φακέλους ,menu,settings. Το Menu screen εμπεριέχει την αρχική οθόνη χρήστη με τα προϊόντα του ,τις κατηγορίες του και την προσθήκη ,επεξεργασία και διαγραφή προϊόντων. Μέσα στο αρχείο υπάρχουν πολλά stateless widgets όπου γίνονται όλες οι λειτουργικότητες. Στο αρχείο settings είναι η οθόνη για τις ρυθμίσεις του χρήστη όπου βάζει τα στοιχεία του, το ωράριο και άλλες χρήσιμες πληροφορίες. Και σε αυτό το αρχείο η λειτουργικότητα βρίσκεται στα widgets που αποτελούν το συγκεκριμένο αρχείο σύμφωνα με την αρχιτεκτονική, bloc, που διέπει την/ακολουθεί η εφαρμογή.



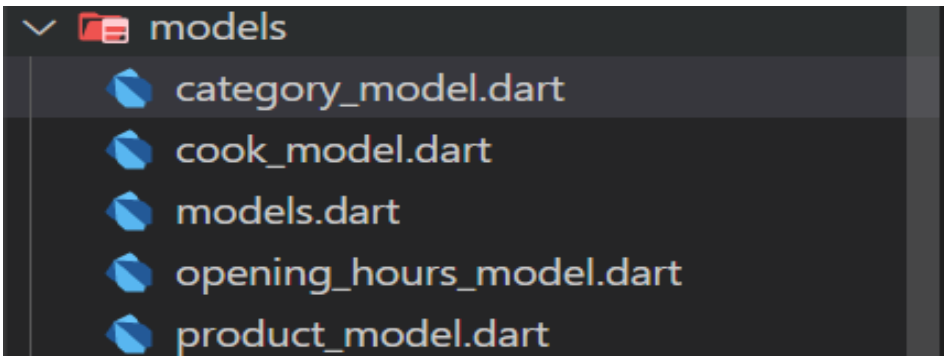
Εικόνα 6:4 Screens folder

Ο φάκελος repositories έχει ένα αρχείο που απλά κάνει export τα δύο άλλα αρχεία. Το αρχείο base_cook_repository είναι abstract και έχει τις μεθόδους που θα χρησιμοποιηθούν στη διασύνδεση με τη βάση, το fetching, το editing, deleting. Το αρχείο cook_repository κάνει extend το base_cook_repository και ουσιαστικά όλη η λειτουργικότητα με τη διασύνδεση με τη βάση και πως θα μεταποιηθούν ή παρθούν τα δεδομένα από αυτή και θα χρησιμοποιηθούν μέσα στη εφαρμογή γίνεται μέσα από αυτό το αρχείο. Τα δεδομένα για τα οποία είναι υπεύθυνα αυτά τα αρχεία είναι μόνο τα δεδομένα που σχετίζονται με τους συνεργάτες δηλαδή τα στοιχεία τους, οι ώρες που είναι διαθέσιμοι, τα προϊόντα που προσφέρουν και γενικά όλα σχεδόν τα δεδομένα εκτός από τα voucher για παράδειγμα.



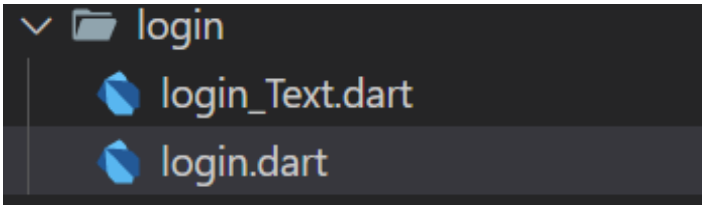
Εικόνα 6:5 Repositories folder

Ο φάκελος models έχει αρχεία που το καθένα αντιπροσωπεύει μία κλάση. Το κάθε αρχείο μέσα στην κλάση έχει τα πρότυπα των μοντέλων που χρησιμοποιεί η κάθε κλάση δηλαδή τα properties αλλά και τις μεθόδους των "κλάσεων". Και σε αυτήν την περίπτωση υπάρχει το ανάλογο αρχείο που κάνει export τα υπόλοιπα αρχεία μοντέλων για να είναι διαθέσιμο σε όποιο αρχείο κάνει import το εκάστοτε μοντέλο κατά μήκος όλης της εφαρμογής.



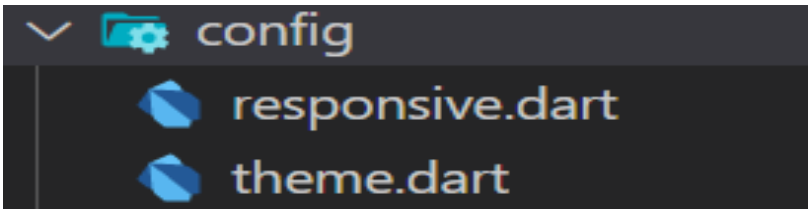
Εικόνα 6:6 Models Folder

Ο φάκελος login σχετίζεται με την πρώτη οθόνη που βλέπει ο χρήστης για την είσοδο του είτε κάνοντας log in είτε δημιουργώντας καινούργιο λογαριασμό μέσω Facebook. Το αρχείο Login είναι stateful και εκεί γίνεται όλη η λειτουργικότητα της εισόδου του συνεργάτη. Υπάρχει η διασύνδεση με το την firebase μέσω της οποίας γίνεται η αυθεντικοποίηση του χρήστη μέσω της διαδικτυακής πλατφόρμας Facebook.



Εικόνα 6:7 Login Folder

Στο φάκελο config γίνεται το configuration για το responsivenesss του UI καθώς εκεί υπάρχουν οι μέθοδοι αναγνώρισης του μεγέθους της οθόνης και προσαρμογής του UI ανάλογα με αυτήν καθώς και ένα αρχείο που ελέγχει κάποιες global μεταβλητές για την εμφάνιση του UI (π.χ. χρώματα, font family).



Εικόνα 6:8 Config folder

Ο τελευταίος φάκελος ,αυτός των bloc, είναι ίσως ο πιό σημαντικός καθώς όλη η λειτουργικότητα της εφαρμογής είναι εκεί. Σε αυτόν τον φάκελο υπάρχουν οι μέθοδοι για την επεξεργασία των δεδομένων αφότου έρθουν από τη βάση αλλά και όσο αυτά χρησιμοποιούνται μέσα στην εφαρμογή. Κάθε υποφάκελος εμπεριέχει τρία αρχεία το bloc, το event και το state για ενότητα του καθενός. Κάθε αρχείο έχει ένα συγκεκριμένο λόγο ύπαρξης.

1. Επιχειρηματική λογική

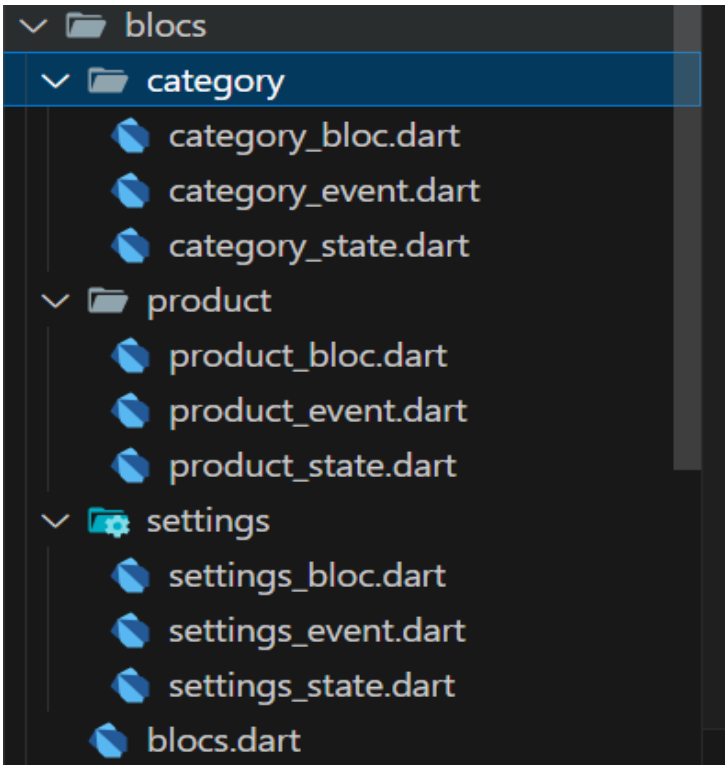
Η βασική λογική και η λειτουργικότητα της εφαρμογής Flutter βρίσκονται εντός των BLoCs. Περιέχει εργασίες όπως η άντληση δεδομένων, ο μετασχηματισμός, η επικύρωση και οποιεσδήποτε άλλες λειτουργίες που δεν σχετίζονται άμεσα με το UI. Με την απομόνωση της επιχειρηματικής λογικής, γίνεται ευκολότερη η επαναχρησιμοποίηση και ο έλεγχος του Flutter.

2. Γεγονότα(events)

Στο BLoC, μπορούμε να ξεκινήσουμε αλλαγές ή ενέργειες στέλνοντας συμβάντα στο BLoC. Τα συμβάντα είναι ουσιαστικά αλληλεπιδράσεις ή εναύσματα του χρήστη, όπως κλικ κουμπιών, εισαγωγή κειμένου ή αιτήματα δικτύου. Αυτά τα Γεγονότα αντιπροσωπεύουν το τι πρέπει να συμβεί στη συνέχεια στην εφαρμογή. Έτσι, βοηθούν στη δημιουργία έξυπνων αλγορίθμων στην εφαρμογή.

3. Καταστάσεις(states)

Το BLoC εκπέμπει καταστάσεις σε απόκριση σε συμβάντα. Αυτά τα States αντιπροσωπεύουν τα διάφορα στιγμιότυπα ή καταστάσεις του UI και των δεδομένων της εφαρμογής . Κάθε κατάσταση αντιπροσωπεύει μια συγκεκριμένη προβολή και οι αλλαγές στην κατάσταση προκαλούν ενημερώσεις του UI.

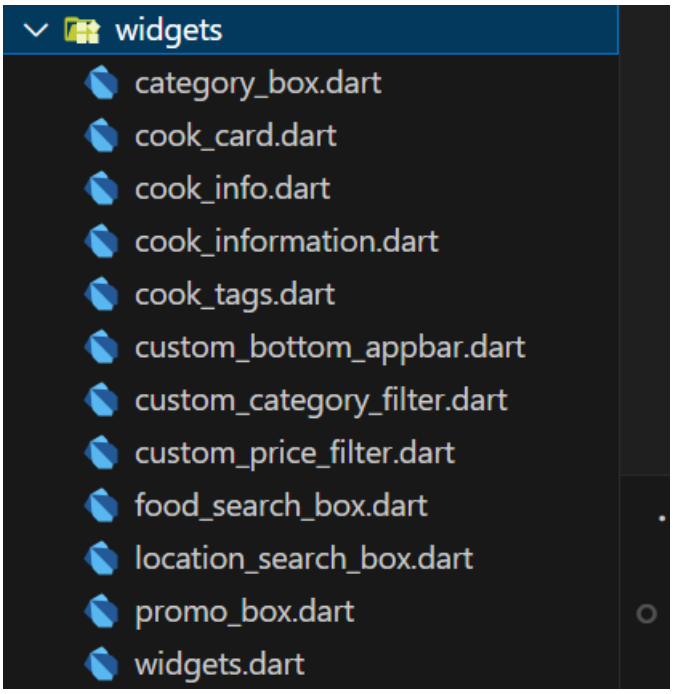


Εικόνα 6:9 Bloc folder

6.2.2 Lib file χρήστη/android app

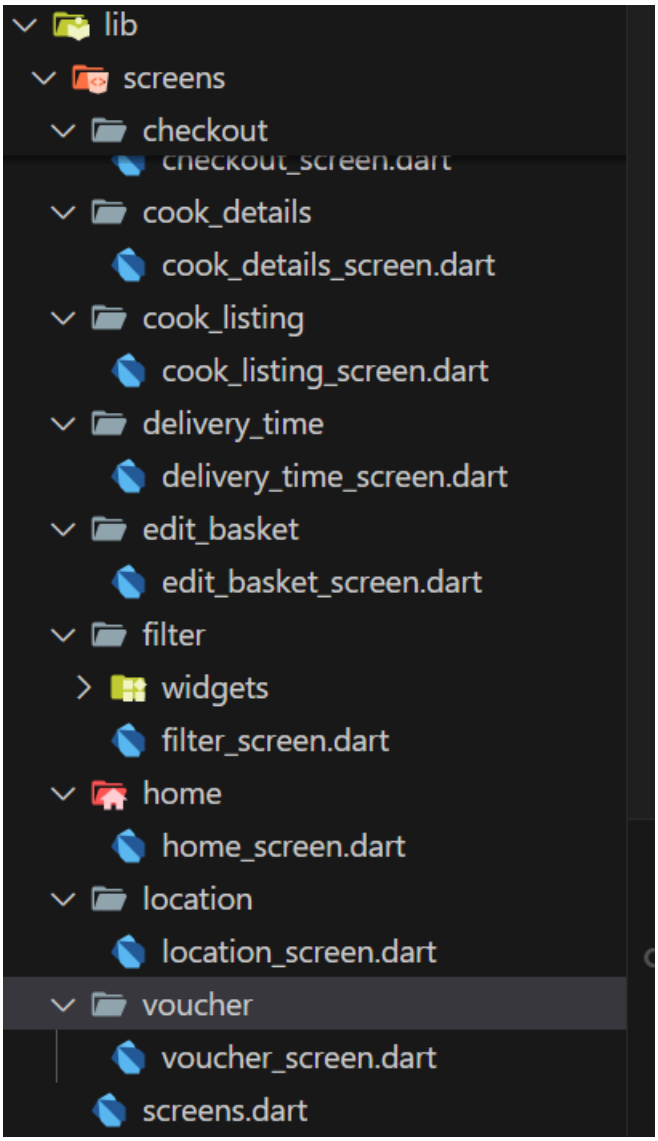
Όπως αναφέραμε και πριν οι φάκελοι του κώδικα και στην εφαρμογή του συνεργάτη αλλά και στην εφαρμογή του χρήστη είναι ίδιοι για αυτό δεν χρειάζεται να τους αναφέρουμε ξανά. Αυτό που αξίζει να αναφέρουμε,εάν και επειδή και αυτή η εφαρμογή είναι δομημένη με την ίδια bloc αρχιτεκτονική οι ομοιότητες θα είναι πολλές, είναι οι δομή των φακέλων καθώς και τα αρχεία μέσα στον φάκελο lib. Ακόμη και οι υποφάκελοι μέσα στον φάκελο lib είναι σχεδόν ίδιοι ,ωστόσο οι αναφορές σε όλους τους υποφακέλους του lib θα γίνουν κανονικά.

Αρχικά έχουμε τον φάκελο widgets όπου αποτελείται από τα widgets αρχεία τα οποία όπως αναφέραμε είναι τα συστατικά στοιχεία των εφαρμογών σε flutter. Η συγκεκριμένη διαφοροποίηση των widgets είναι ίσως το πιό σημαντικό κατόρθωμα της αρχιτεκτονικής σε flutter καθώς μία από τις μεγαλύτερες προκλήσεις μίας τέτοιας εφαρμογής για τον προγραμματιστή είναι η ανταλλαγή των δεδομένων μεταξύ αρχείων δηλαδή το property binding, event binding και το two way binding. Η μεγαλύτερη συμφόρηση κώδικα στα αρχεία γίνεται όταν πολλά widgets είναι στο ίδιο αρχείο και πραγματικά είναι πολύ εύκολο ο κώδικας σε ένα και μόνο αρχείο να γίνει τεράστιος και τρομερά δυσνόητος με αποτέλεσμα η συντήρηση και επέκταση του από άλλους αλλά ακόμα και τον ίδιο τον προγραμματιστή που το έγραψε εξ αρχής να γίνει σχεδόν ακατόρθωτες.



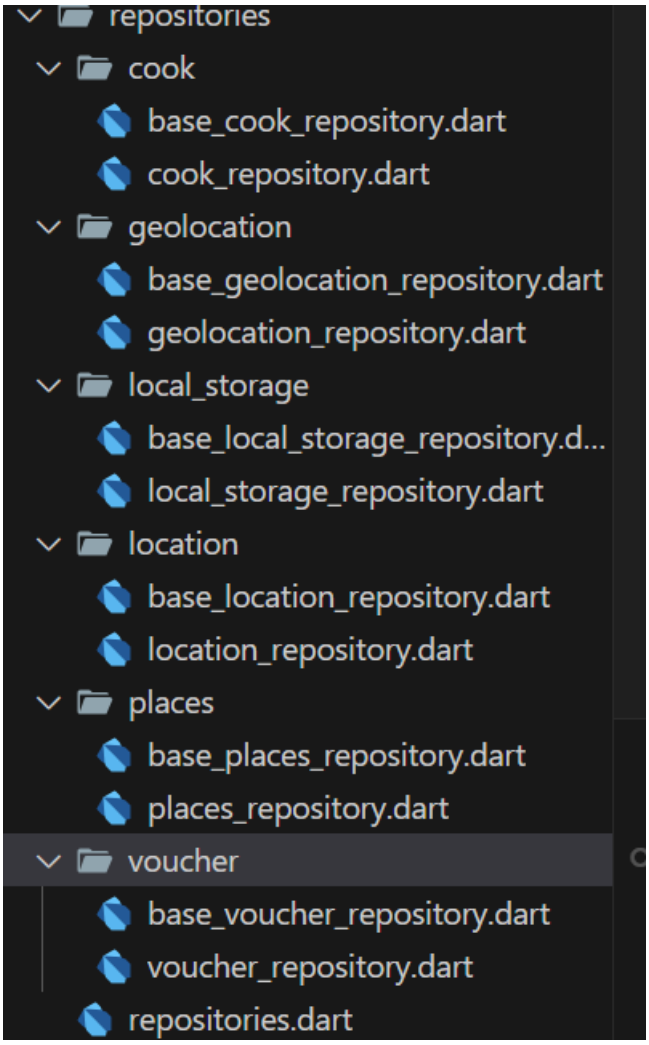
Εικόνα 6:10 User Widgets

Από κάτω προς τα πάνω πάλι, αμέσως μετά είναι ο φάκελος screens όπου ξεχωρίζει τα αρχεία ανάλογα με τις οθόνες ,π.χ. καλάθι, χάρτης. Κάθε οθόνη που θα εμφανιστεί στον χρήστη της android εφαρμογής έχει το δικό της ξεχωριστό αρχείο σε αυτό τον φάκελο το οποίο με τη σειρά του χρησιμοποιεί τα αρχεία του προηγούμενο φακέλου που αναφέραμε, αυτόν των widgets και τα συνθέτει με τρόπο τέτοιο ώστε ο χρήστης να έχει την καλύτερη δυνατή εμπειρία χρήστη. Επίσης να αναφερθεί ότι οι οθόνες είναι χωρισμένες σε ομώνυμους φακέλους που ο καθένας, σχεδόν όλοι, περιέχουν ένα και μόνο αρχείο.



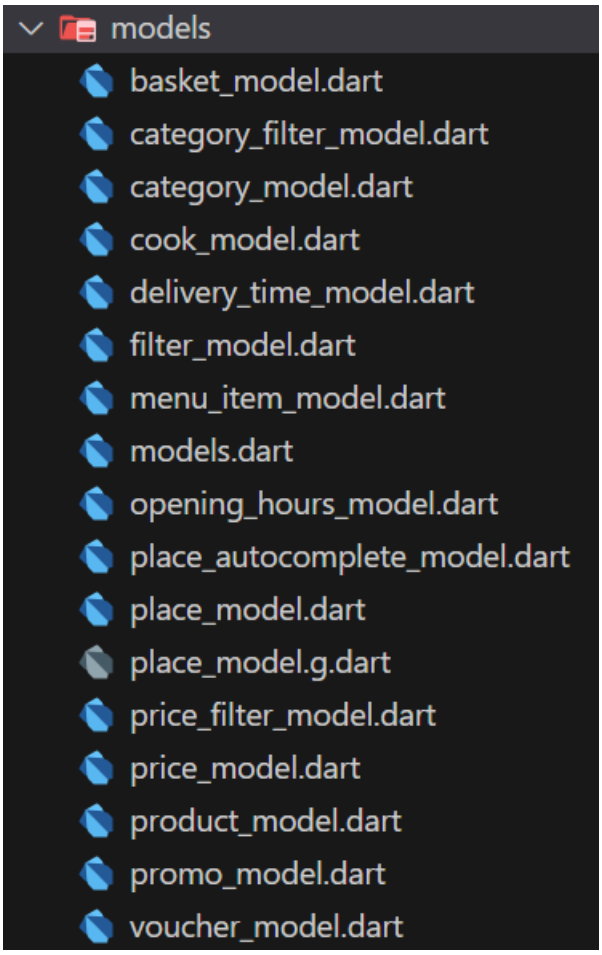
Εικόνα 6:11 User Screens

Ακολουθεί η ανάλυση του φακέλου των repositories όπου κάθε υποφάκελος είναι χωρισμένος ανάλογα με την επιχειρησιακή λογική την οποία καταλαμβάνει και περιέχει δύο αρχεία έκαστος. Το πρώτο αρχείο ξεκινάει με την λέξη base και ουσιαστικά είναι μία abstract κλάση όπου έχει απλά τις μεθόδους τις κλάσεις, χωρίς κώδικα που να τους συνοδεύει και επίσης τους τύπος των δεδομένων που παίρνει, αν χρειάζεται, και επιστρέφει. Το άλλο αρχείο έχει το ίδιο όνομα χωρίς όμως τη λέξη base_ στην αρχή και ουσιαστικά κάνει extend την abstract κλάση και προσθέτει τη λειτουργικότητα ,κάνοντας override, στις μεθόδους της κλάσεις. Το γεγονός ότι υπάρχουν δύο αρχεία ανά φάκελο και το ένα είναι αρκετά "φτωχό" από την άποψη των γραμμών που περιέχει οφείλεται καθαρά σε καλύτερη αρχιτεκτονική και λογική προσέγγιση όσο αναφορά το πως είναι γραμμένη η εφαρμογή καθώς κάνει το πρόγραμμα πιο "καθαρό" και εύληπτο.σ



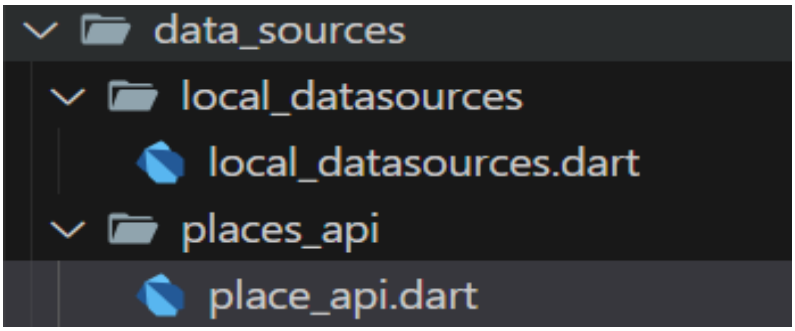
Εικόνα 6:12 User Repositories

Ο φάκελος Models περιέχει τις κλάσεις-μοντέλα που χρησιμοποιούνται στην εφαρμογή , δηλαδή τον τύπο των δεδομένων (map/object) που χρησιμοποιείται μέσα στην εφαρμογή. Κάθε αρχείο έχει το όνομα της κλάσης φυσικά μαζί με τα properties της(πεδία) , τον constructor της (δομητή) αλλά και τις functions της κάθε κλάσης. Το αρχείο place_model.g.dart είναι αυτόματα παραγόμενο από την εντολή “part ‘place_model.g.dart’;” στο αρχείο “place_model.dart” και παράγεται με το πρώτο build που θα κάνει η εφαρμογή αφότου έχει γραφεί η εντολή. Αυτή η διαδικασία ακολουθείται καθώς η εφαρμογή χρησιμοποιεί την δομή δεδομένων hive που είναι πλήρως γραμμένη σε dart και απαιτεί αυτό το αρχείο. Εδώ είναι μία καλή ευκαιρία να αναφέρουμε ότι σε εφαρμογές flutter είναι πιθανόν να συναντήσουμε συχνά αρχεία με κατάληξη .g.dart και αυτό σημαίνει ότι είναι autogenerated, συνήθως έχουν να κάνουν με τη βάση δεδομένων της κινητής συσκευής και γενικά είναι αρχεία τα οποία ο προγραμματιστής ο ίδιος δεν πρέπει να τα πειράξει εκτός και αν ξέρει ακριβώς τι θέλει να κάνει αλλά ακόμα και έτσι επειδή αυτά τα αρχεία παράγονται κάθε φορά που γίνεται ένα καινούργιο build στην εφαρμογή οι αλλαγές που έχει κάνει ουσιαστικά δεν θα υφίστανται πιά μίας και το αρχείο θα είναι ξανά παραγόμενο από την αρχή.



Εικόνα 6:13 User Models

Ο φάκελος data_sources παρουσιάζει μεγάλο ενδιαφέρον εάν και έχει μόλις δύο υποφακέλους με τον καθένα από αυτούς να έχει από ένα αρχείο. Το ενδιαφέρον έγκειται στο γεγονός ότι δεν έχουμε ξανασυναντήσει παρόμοια αρχεία, εάν και είναι dart και αυτά, ούτε σε αυτή την εφαρμογή αλλά ούτε στου συνεργάτη οπότε καλό θα ήταν να εξηγήσουμε ακριβώς τι κάνει το καθένα και γιατί υπάρχει. Αρχικά στον πρώτο υποφάκελο “local_datasources” βρίσκεται το ομώνυμο αρχείο το οποίο έχει μέσα και την abstract κλάση αλλά την κλάση που κάνει extend την abstract. Το συγκεκριμένο αρχείο έχει τις μεθόδους μέσω των οποίων η εφαρμογή έχει πρόσβαση στην τοπική hive βάση της συσκευής αλλά και πρόσβαση στα δεδομένα τοποθεσίας της συσκευής για την χρησιμοποίηση τους κυρίως στην αρχική οθόνη του χάρτη. Οι συγκεκριμένοι μέθοδοι είναι προσβάσιμες μέσω της βιβλιοθήκης-πακέτου της hive που έχουμε εγκαταστήσει στην εφαρμογή. Ο άλλος υποφάκελος με το πολύ αντιπροσωπευτικό όνομα places_api εμπεριέχει αρχικά την abstract κλάση μαζί με αυτή που την “επεκτείνει” (extend) καθώς πεδία και μεθόδους που σχετίζονται με το google api για την τοποθεσία και τους χάρτες.

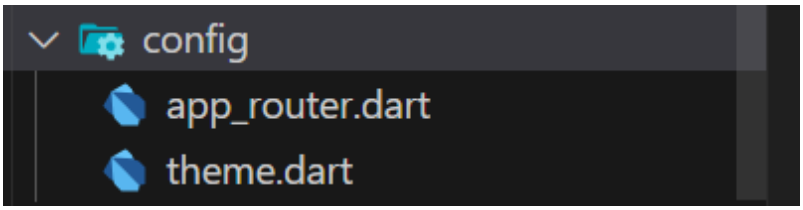


Εικόνα 6:14 User's DataSources Folder

Στον προτελευταίο φάκελο config υπάρχουν οι εντολές για το global UI pattern που χρησιμοποιεί η εφαρμογή όσο αναφορά τα χρώματα, τα font weight, font sizes και άλλα χαρακτηριστικά των γραμμάτων ,πλαισίων και γενικότερα του οπτικού θέματος που χρησιμοποιείται στην εφαρμογή το οποίο φυσικά είναι πολύ παρόμοιο ,σχεδόν ολόιδιο ,με αυτό της εφαρμογής του συνεργάτη καθώς και τα δύο αποτελούν ξεχωριστά μέρη του ίδιου project. Επίσης όμως στο συγκεκριμένο υποφάκελο υπάρχει ένα, καινούργιο αρχείο, το οποίο δεν υπήρχε στη προηγούμενη εφαρμογή, και είναι αυτό του routing (δρομολόγηση) της εφαρμογής. η δρομολόγηση αναφέρεται στον τρόπο με τον οποίο μια εφαρμογή μεταβαίνει μεταξύ οθονών ή σελίδων, ακολουθώντας τις αρχές της πλοήγησης και της δρομολόγησης. Οι διαδρομές αποτελούν αναπόσπαστο μέρος κάθε εφαρμογής για τη δόμηση των σελίδων. Το Flutter χρησιμοποιεί μια στοίβα για τη διαχείριση των διαδρομών, όπου μια νέα διαδρομή ωθείται σε αυτήν κατά την πλοήγηση σε μια νέα οθόνη, και η τρέχουσα διαδρομή αφαιρείται κατά την πλοήγηση πίσω στην προηγούμενη διαδρομή. Αυτή η τεχνική του πλοηγού(router 2.0) δημιουργεί μια ιστορική ακολουθία των σελίδων που επισκέφθηκαν και στις οποίες μπορεί να πλοηγηθεί κανείς αντίστροφα, αποτυπώνοντας την ουσία της χρονολογικής πλοήγησης.

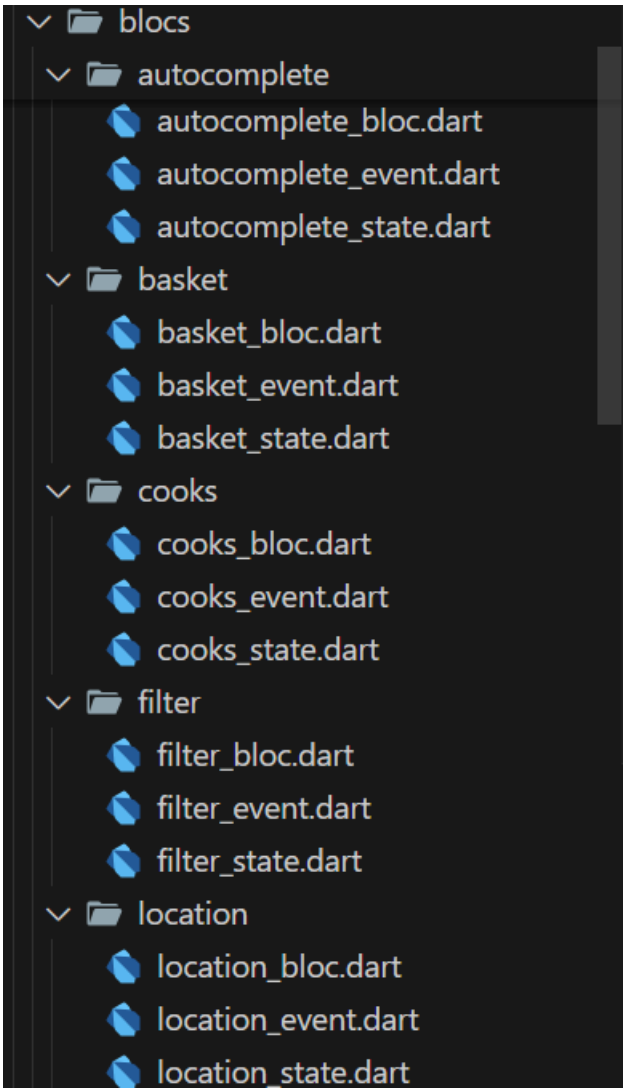
Η πλοήγηση στο Flutter γίνεται με τη χρήση του widget Navigator. Κάθε οθόνη στην εφαρμογή Flutter είναι μια νέα PageRoute και τοποθετείται στη στοίβα που διατηρεί το widget Navigator. Όπως μια στοίβα, η πρώτη διαδρομή τοποθετείται επίσης στην πρώτη, αρχική σελίδα ή πρώτη οθόνη. Στη συνέχεια, καθώς ο χρήστης πλοηγείται στην εφαρμογή , νέες διαδρομές τοποθετούνται στην κορυφή της στοίβας, ενώ η τρέχουσα διαδρομή είναι πάντα το κορυφαίο στοιχείο.

Ο ευκολότερος τρόπος για να απεικονίσουμε αυτό είναι να φανταστούμε την εφαρμογή ως μια στοίβα χαρτιών, όπου κάθε χαρτί αντιπροσωπεύει μια νέα διαδρομή. Καθώς πλοηγούμαστε με το widget δρομολόγησης μέσα στην εφαρμογή, συνεχίζουμε να στοιβάζουμε νέα φύλλα ή νέες οθόνες στην κορυφή. Όταν πρέπει να επιστρέψουμε, αφαιρούμε το πιο πάνω φύλλο, αποκαλύπτοντας το από κάτω. Σε αυτό το αρχείο λοιπόν είναι δηλωμένες όλες οι οθόνες της εφαρμογής καθώς και η static μέθοδο πλοήγησες μεταξύ αυτών.



Εικόνα 6:15 Config user

Τελευταίο αφήνουμε πάλι το φάκελο blocs καθώς είναι από τα πιό σημαντικούς φακέλους για την εφαρμογή αφού έχει να κάνει με την αρχιτεκτονική την ίδια της εφαρμογής. Στον φάκελο αυτό υπάρχουν υποφάκελοι , ο καθένας εκ των οποίων έχει τρία αρχία , ένα που εχει κατάληξει _bloc.dart, ένα με κατάληξη _event.dart και ένα με κατάληξη _state.dart. Κάθε αρχείο έχει να κάνει είτε με την επιχειρηματική λογική, είτε με τα γεγονότα και που πυροδοτούνται κατά την αλληλεπίδραση του χρήστη με την εφαρμογή, είτε με τις καταστάσεις (states) της κάθε οθόνης και των δεδομένων της εφαρμογής. Κάθε κατάσταση αντιπροσωπεύει μία συγκεκριμένη προβολή και οι αλλαγές στην κατάσταση προκαλούν ενημερώσεις του UI.



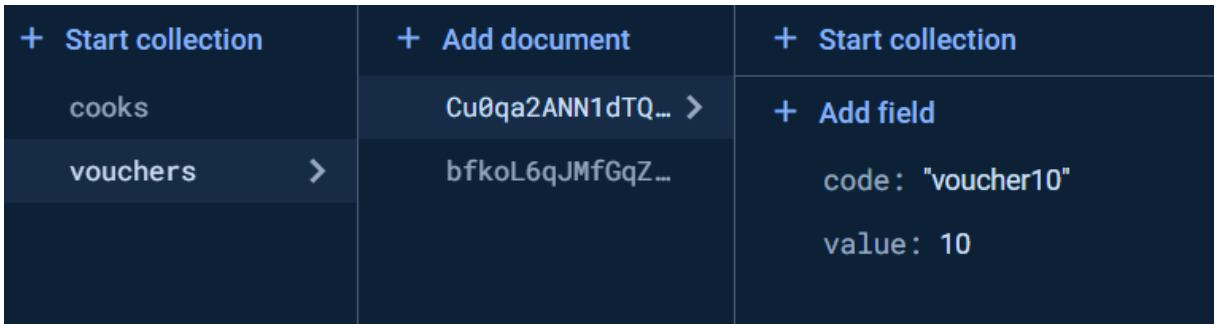
Εικόνα 6:16 User's Bloc Folder

6.3 Δομή της βάσης δεδομένων

Η βάση δεδομένων είναι στη firebase, η Firebase Realtime Database που χρησιμοποιούμε στην εφαρμογή είναι cloud-hosted βάση της οποίας τα δεδομένα είναι αποθηκεύονται σαν JSON και συγχρονίζονται σε πραγματικό χρόνο με τον εκάστοτε συνδεδεμένο πελάτη-χρήστη. Αντί για τυπικά HTTP requests η Firebase Realtime Database χρησιμοποιεί συγχρονισμό δεδομένων κάθε φορά που αλλάζει κάποιο δεδομένο στη βάση, δηλαδή όλες οι συνδεδεμένες συσκευές λαμβάνουν την ενημέρωση-αλλαγή της βάσης μέσα σε εκατομμυριοστά του δευτερόλεπτου. Το cloud Firestore είναι schemaless οπότε δίνει την πλήρη ελευθερία όσο αναφορά τα πεδία που μπορούμε να βάλουμε σε κάθε αρχείο και τι είδους πεδία αποθηκεύουμε στα πεδία αυτά. Αρχικά η βάση έχει δύο collections (συλλογές), η μία για τα vouchers και η άλλη για τους μάγειρες. Οι συλλογές στη βάση ουσιαστικά είναι απλά τα containers για τα documents και δεν μπορούν να περιέχουν απευθείας πεδία με τιμές και δεν μπορούν να περιέχουν άλλες συλλογές.

Αρχικά να αναλύσουμε την συλλογή με όνομα vouchers η οποία μαζί με όλα documents και τα δεδομένα τους έχουν γραφεί προγραμματιστικά και όχι από κάποιον χρήστη από οποιαδήποτε από τις δύο εφαρμογές. Υπάρχουν δύο documents που το καθένα έχει ένα μοναδικό όνομα το οποίο είναι αυτόματα δημιουργημένο από την βάση ,από επιλογή, και που το καθένα περιέχει δύο πεδία που

περιέχουν τον ίδιο τύπο δεδομένων. Το πρώτο πεδίο είναι τύπου String και έχει όνομα code. Ουσιαστικά είναι το όνομα του voucher το οποίο εμφανίζεται στην οθόνη του χρήστη. Το δεύτερο πεδίο είναι αυτό της τιμής (value) όπου στην συγκεκριμένη εφαρμογή είναι το ποσό που αφαιρείται από το συνολικό ποσό της παραγγελίας στο καλάθι του χρήστη.



Εικόνα 6:17 Βάση Voucher

Όσο αναφορά την συλλογή cooks (μάγειρες) αυτή τη στιγμή έχουν δύο εγγραφές, δύο documents, με δύο ξεχωριστά ονόματα τα οποία είναι σαν αναγνωριστικά (id) και τα οποία είναι επίσης παραγόμενα αυτόματα κατ' επιλογή. Μόλις γίνει είσοδος του χρήστη μέσω του Facebook authentication γίνεται έλεγχος εάν υπάρχει αυτός ο χρήστης με βάση το όνομα. Εάν δεν υπάρχει, τότε δημιουργείται καινούργιο document με καινούργιο μοναδικό αναγνωριστικό όνομα. Σε κάθε document υπάρχουν τα πεδία που αντιστοιχούν στον συνεργάτη της εφαρμογής δηλαδή τον χρήστη της web app. Το πρώτο πεδίο που φαίνεται στη εικόνα 6.18 είναι το address το οποίο είναι ένα map με τέσσερα πεδία:

- lat: το οποίο είναι το latitude της τοποθεσίας του συνεργάτη και είναι τύπου number (double)
- lon: το οποίο είναι το longitude της τοποθεσίας του συνεργάτη και είναι και αυτό τύπου number (double) και μαζί με το προηγούμενο πεδίο συνθέτουν τη διεύθυνση του συνεργάτη στον χάρτη του χρήστη
- name: είναι ένα string πεδίο που παίρνει το όνομα της περιοχής που ανήκει η διεύθυνση του συνεργάτη
- placeId: είναι ένα string πεδίο που παίρνει ένα ξεχωριστό αναγνωστικό νούμερο

Το επόμενο πεδίο της φωτογραφίας είναι το categories το οποίο είναι ένας πίνακας από Map και περιέχει μόνο τις κατηγορίες στις οποίες ο εκάστοτε μάγειρας έχει προϊόντα. Το κάθε Map έχει τα εξής πεδία:

- description: που περιέχει απλά τη περιγραφή-όνομα της κατηγορίας
- id: είναι ένα ξεχωριστός αναγνωριστικός αριθμός σε String
- ΕικόναUrl: αυτό το πεδίο είναι String και περιέχει τη διαδρομή για τη φωτογραφία της κατηγορίας
- index: περιέχει το index του πίνακα, είναι αριθμός και ξεκινάει από το μήδεν φυσικά
- name: είναι String και περιέχει το όνομα της κατηγορίας, συνήθως ταυτίζεται με το description(περιγραφή)

Μετά είναι το πεδίο deliveryFee που είναι τύπου number (double) και είναι το έξτρα ποσό που χρειάζεται ο συνεργάτης για να κάνει delivery τη παραγγελία.

Το πεδίο deliveryTime είναι του ίδιου τύπου, number (double), και είναι ο χρόνος για την εκτέλεση της παραγγελίας σε περίπτωση που ο χρήστης δεν διαλέξει συγκεκριμένη ώρα.

Το πεδίο description είναι ένα πεδίο String όπου ο συνεργάτης συμπληρώνει για να περιγράψει τον εαυτό του/της ή να πει λίγα λόγια για τις υπηρεσίες που προσφέρει αλλά και για την εμπειρία του πάνω στον τομέα της μαγειρικής.

Έπειτα υπάρχει το πεδίο email για το email του χρήστη, είναι σημαντικό πεδίο καθώς εκεί θα σταλεί η παραγγελία του χρήστη. Το πεδίο id είναι ίδιο με το ξεχωριστό αναγνωριστικό όνομα του document του κάθε συνεργάτη-μάγειρα.

Το ΕικόναUrl είναι ένα String πεδίο για την διεύθυνση της φωτογραφίας του συνεργάτη και μετά το name είναι String πεδίο για το όνομα του χρήστη το οποίο το παίρνει αυτόματα η εφαρμογή κατά την είσοδο του συνεργάτη στην εφαρμογή μέσω Facebook. Το πεδίο priceCategory είναι ένα String το οποίο περιέχει μία από τις τρείς τιμές €,€€,€€€ που αντιστοιχούν στις τρείς κατηγορίες στις τιμές των προϊόντων που προσφέρει ο κάθε συνεργάτης.

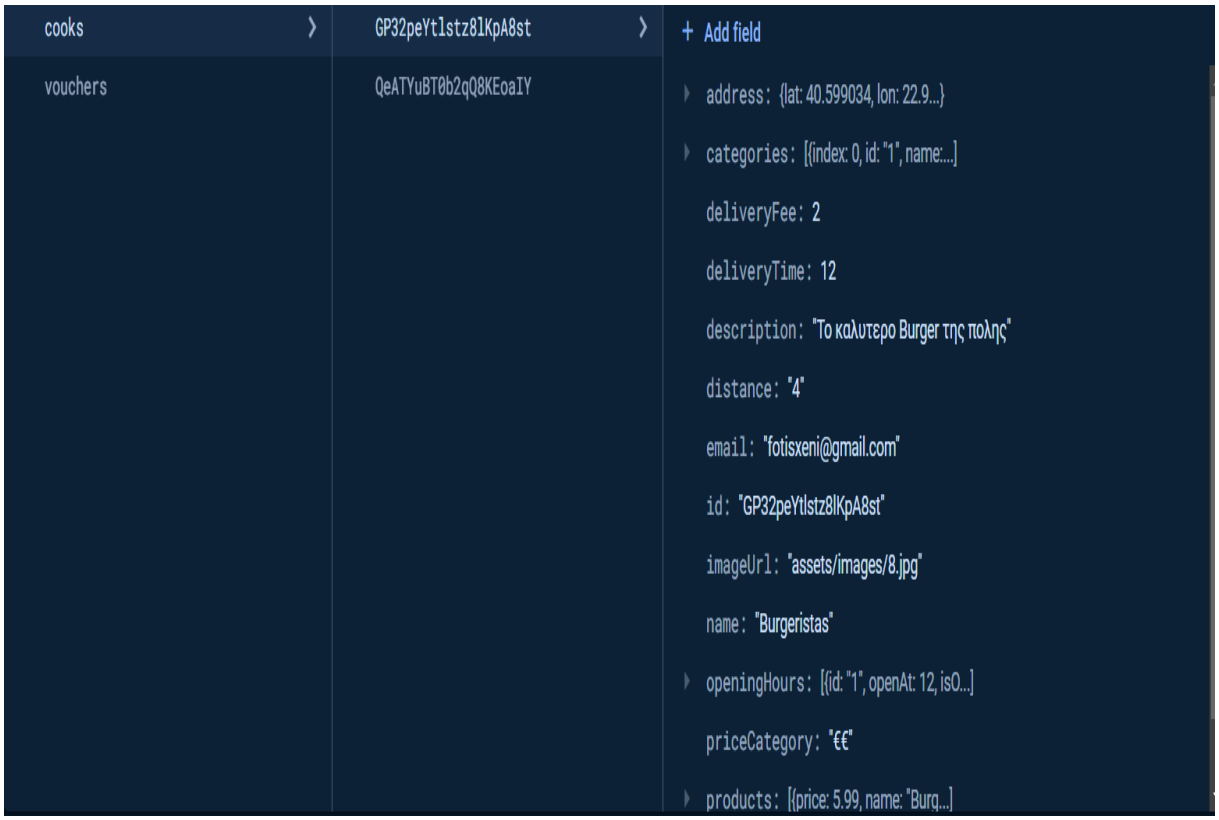
Το πεδίο openingHours είναι ένας πίνακας από 7 Maps ,ένα στοιχείο πίνακα για κάθε μέρα της εβδομάδας, τα οποία περιέχουν τα εξής ζευγάρια name-value:

- closeAt: είναι ένα number πεδίο που περιέχει το ανώτερο όριο σε εικοσιτετράωρο ρολόι που ο συνεργάτης εξυπηρετεί παραγγελίες.
- day: Το όνομα της μέρας που αντιστοιχεί το ωράριο που αναφέρεται
- id: ένας ξεχωριστός αριθμός από το 1-7 για κάθε μέρα της εβδομάδας σε String
- isOpen: είναι ένα boolean πεδίο που συμβολίζει εάν ο συνεργάτης εξυπηρετεί παραγγελίες τη συγκεκριμένη μέρα
- openAt: είναι ένα πεδίο number όπου περιέχει σε εικοσιτετράωρο ρολόι τον αριθμό της ώρας που ο χρήστης ξεκινάει να εξυπηρετεί παραγγελίες

Το πεδίο tags είναι ένας πίνακας που περιέχει τα είδη των κατηγοριών που προσφέρει ο συνεργάτης . Το κάθε στοιχείο του πίνακα είναι ένα String με το όνομα της κατηγορίας. Το συγκεκριμένο πεδίο-πίνακας βοηθάει στο φιλτράρισμα των συνεργατών από τον χρήστη.

Τελευταίο αφήσαμε το πιο σημαντικό πεδίο και είναι το πεδίο products που είναι ένας πίνακας από Maps και είναι τα προϊόντα που προσφέρει ο κάθε συνεργάτης στους χρήστες. Το κάθε map περιέχει τα παρακάτω πεδία:

- category: είναι String πεδίο που περιέχει το όνομα της κατηγορίας που ανήκει το κάθε προϊόν π.χ. Burger, Σαλάτα, Pizza
- cookId: είναι το μοναδικό αναγνωριστικό όνομα του document του κάθε συνεργάτη-μάγειρα
- description: είναι η περιγραφή που βάζει ο κάθε συνεργάτης για το κάθε προϊόν του, π.χ. τα υλικά που περιέχει το κάθε προϊόν, πιάτο κ.λπ.
- id: είναι String πεδίο που περιέχει έναν αύξοντα μοναδικό αριθμό ξεκινώντας από το 1 για κάθε προϊόν
- ΕικόναUrl: είναι το path για τη φωτογραφία του κάθε προϊόντος που βάζει ο συνεργάτης
- name: το όνομα που δίνει ο συνεργάτης στο κάθε προϊόν που προσφέρει
- price: είναι αριθμητικό πεδίο, double, και είναι η τιμή σε ευρώ (€) του προϊόντος



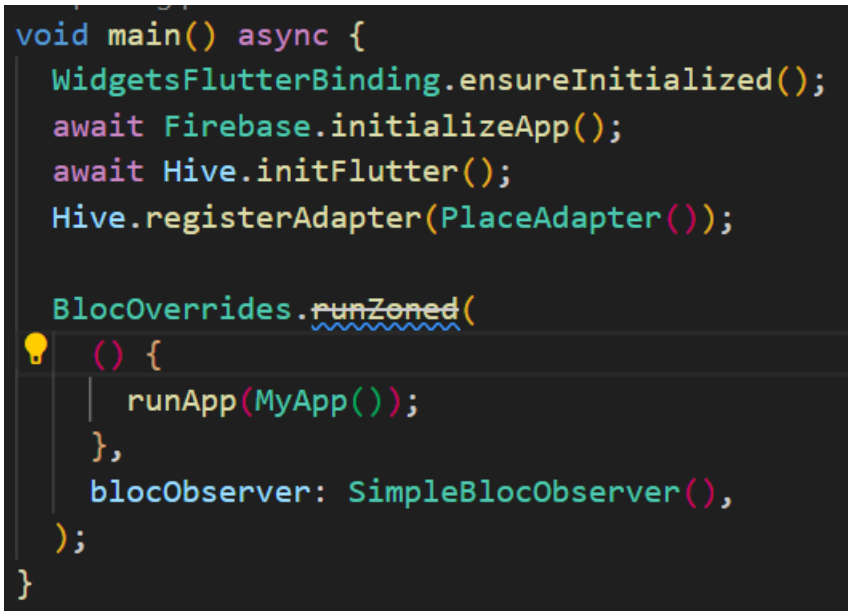
Εικόνα 6:18 Βάση Συνεργατών

6.4 Σύντομη παρουσίαση της λογικής της εφαρμογής

Σε αυτή τηην ενότητα θα αναφερθούμε σε κομμάτια κώδικα που υλοποιούν κάποιες από τις βασικές λειτουργίες της εφαρμογής.

6.4.1 Βασικά χαρακτηριστικά

Κοιτώντας το αρχείο main.dart βλέπουμε αρχικά την ασύχρονη μέθοδο main() όπου είναι η κύρια συνάρτηση στη Flutter και είναι υπεύθυνη για την αρχικοποίηση της εφαρμογής και την έναρξη της εκτέλεσης της εφαρμογής. Μέσα σε αυτήν καλούνται κάποιες μέθοδοι για την αρχικοποίηση των εσωτερικών μεθόδων που αφορούν το binding τη εφαρμογής δηλαδή το πως δεδομένα ή γεγονότα περνάνε από τη μια κλάση στην άλλη.Επίσης εκτελούνται μέθοδοι για την αρχικοποίηση της βάσης των δεδομένων (Hive) της κινητής συσκευής αλλά και της αρχικοποίησης της εφαρμογής σύμφωνα με την Firebase Realtime Database. Έπειτα η κύρια μέθοδο καλέι την runApp μέθοδο με ένα στιγμιότυπο της εφαρμογής το οποίο είναι ένα προσαρμοσμένο widget που ορίζει τη συμπεριφορά της εφαρμογής στα όριο που ορίζει η κάθε οθόνη.



```
void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  await Hive.initFlutter();
  Hive.registerAdapter(PlaceAdapter());

  BlocOverrides.runZoned(
    () {
      runApp(MyApp());
    },
    blocObserver: SimpleBlocObserver(),
  );
}
```

Εικόνα 6:19 Κώδικας κύριας μεθόδου

Αναλύοντας την κλάση MyApp το πρώτο πράγμα που παρατηρούμε ότι πρόκειται για ένα Stateless Widget το οποίο υλοποιεί την μέθοδο Build για το χτήσιμο των widget και διαδικασιών για την δημιουργία της οθόνης του χρήστη. Αρχικά δηλώνονται οι μέθοδοι(providers) της εφαρμογής όπως αυτή που χρησιμεύει στην εύρεση της τοποθεσίας για την χρησιμοποίηση του χάρτη καθώς και αυτή που κάνει τις κλήσεις στη βάση των μαγείρων για την απόκτηση δεδομένων από τη συγκεκριμένη βάση. Με την ίδια λογική γίνεται η δήλωση των μεθόδων για το Bloc pattern το οποίο κάνει τη διαχείριση των διαφόρων γεγονότων που προκαλούνται από τις ενέργειες του χρήστη.

```
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MultiRepositoryProvider(
      providers: [
        RepositoryProvider<GeolocationRepository>(
          create: (_) => GeolocationRepository(),
        ), // RepositoryProvider

        RepositoryProvider<CookRepository>(
          create: (_) => CookRepository(),
        ), // RepositoryProvider
        RepositoryProvider<LocationRepository>(
          create: (_) => LocationRepository(
              placesApi: PlacesApiImpl(),
              localDatasource: LocalDatasourceImpl()), // Lo
        ) // RepositoryProvider
      ],
      child: MultiBlocProvider(
        providers: [
          BlocProvider(
            create: (context) => LocationBloc(
              geolocationRepository: context.read<Geolocatio
              locationRepository: context.read<LocationRepos
```

Εικόνα 6:20 MyApp κώδικας (1)

```
    cookRepository: context.read<CookRepository>(),
  )..add(LoadMap()), // LocationBloc
), // BlocProvider
BlocProvider(
  create: (context) => CooksBloc(
      cookRepository: context.read<CookRepository>(),
      locationBloc: context.read<LocationBloc>()), // Coo
), // BlocProvider
BlocProvider(
  create: (context) => AutocompleteBloc(
    locationRepository: context.read<LocationRepository>(
  )..add(LoadAutocomplete()), // AutocompleteBloc
), // BlocProvider
BlocProvider(
  create: (context) => FilterBloc(
    cooksBloc: context.read<CooksBloc>(),
  )..add(LoadFilter()), // FilterBloc
), // BlocProvider
BlocProvider(
  create: (context) => VoucherBloc(
    voucherRepository: VoucherRepository(),
  )..add(LoadVouchers()), // VoucherBloc
), // BlocProvider
BlocProvider(
  create: (context) => BasketBloc(
    voucherBloc: BlocProvider.of<VoucherBloc>(context),
  )..add(StartBasket()), // BasketBloc
```

Εικόνα 6:21 MyApp κώδικας (2)

6.4.2 Επιλογή τοποθεσίας χρήστη

Όπως έχουμε ξαναπεί κάθε οθόνη αποτελείται από πολλά widgets το ένα μέσα στο άλλο, συνήθως το πρώτο είναι το Scaffold ,το ίδιο ισχύει και για την πρώτη οθόνη του χρήστη για την επιλογή της τοποθεσίας του. Έπειτα υπάρχει το Widget BlocBuilder μέσα στο οποίο υπάρχουν τα υπόλοιπα συστατικά κομμάτια της οθόνης (widgets) καθώς και η διαχείριση των γεγονότων που προκύπτουν από τις ενέργειες του χρήστη αλλά και την κατάσταση της εφαρμογής και των δεδομένων της. Το κύριο συστατικό κομμάτι της οθόνης είναι το GoogleMap το οποίο περιέχει τον χάρτη καθώς και το πεδίο όπου ο χρήστης συμπληρώνει τη διεύθυνση του. Με την αποθήκευση ο χρήτης ανακατευθύνεται στην αρχική οθόνη της εφαρμογής με το Navigator.pushNamed(context, ‘/’) όπου ουσιαστικά πηγαίνει στην οθόνη με πεδίο routename το “/”.

```dart
class LocationScreen extends StatelessWidget {
  static const String routeName = '/location';

  static Route route() {
    return MaterialPageRoute(
      builder: (_) => LocationScreen(),
      settings: RouteSettings(name: routeName),
    ); // MaterialPageRoute
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: BlocBuilder<LocationBloc, LocationState>(
        builder: (context, state) {
          if (state is LocationLoading) {
            return Center(
              child: CircularProgressIndicator(),
            ); // Center
          }
          if (state is LocationLoaded) {
            Set<Marker> markers = state.cooks!.map((cook) {
              return Marker(
                markerId: MarkerId(cook.id),
                infoWindow: InfoWindow(
                  title: cook.name,
                  snippet: cook.description,
                  onTap: () {
                    Navigator.pushNamed(
```

Εικόνα 6:22 location screen code1

```
return Stack(
  children: [
    GoogleMap(
      myLocationEnabled: true,
      buildingsEnabled: false,
      onMapCreated: (GoogleMapController controller) {
        context.read<LocationBloc>().add(
              LoadMap(controller: controller),
            );
      },
      markers: markers,
      initialCameraPosition: CameraPosition(
        target: LatLng(
          state.place.lat,
          state.place.lon,
        ), // LatLng
        zoom: 15,
      ), // CameraPosition
    ), // GoogleMap
```

Εικόνα 6:23 location screen code2

```
class _SearchBoxSuggestions extends StatelessWidget {
  const _SearchBoxSuggestions({
    Key? key,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return BlocBuilder<AutocompleteBloc, AutocompleteState>(
      builder: (context, state) {
        if (state is AutocompleteLoading) {
          return SizedBox();
        }
        if (state is AutocompleteLoaded) {
          return ListView.builder(
            padding: EdgeInsets.zero,
            shrinkWrap: true,
            itemCount: state.autocomplete.length,
            itemBuilder: (context, index) {
              return Container(
                color: Colors.black.withOpacity(0.6),
                child: ListTile(
                  title: Text(
                    state.autocomplete[index].name,
                    style: Theme.of(context)
                        .textTheme
                        .headline6!
                        .copyWith(color: Colors.white),
                  ), // Text
                  onTap: () {
```

Εικόνα 6:24 location screen searchbox

```
onTap: () {
  context.read<LocationBloc>().add(
        SearchLocation(
          placeId: state.autocomplete[index].placeId,
        ), // SearchLocation
      );
  context.read<AutocompleteBloc>().add(ClearAutocomplete());
```

Εικόνα 6:25 location screen searcbox2

6.4.3 Λίστα των συνεργατών

Στην επόμενη εικόνα φαίνεται ο κώδικας της αρχικής οθόνης της εφαρμογής όπου φαίνεται η λίστα των συνεργατών. Το ListView.builder χρησιμοποιείται καθώς θέλουμε να εμφανίσουμε μία λίστα της οποίας το μέγεθος δεν γνωρίζουμε και επίσης είναι μεταβαλλόμενη καθώς εξαρτάται από τα φίλτρα που θα βάλει ο χρήστης. Το widget CookCard εμπεριέχει την εμφάνιση του κάθε συνεργάτη δηλαδή τη φωτογραφία μαζί με τις πληροφορίες από κάτω.

```
BlocBuilder<CooksBloc, CooksState>(
  builder: (context, state) {
    if (state is CooksLoading) {
      return Center(
        child: CircularProgressIndicator(),
      ); // Center
    }
    if (state is CooksLoaded) {
      return Padding(
        padding: const EdgeInsets.all(8.0),
        child: ListView.builder(
          physics: NeverScrollableScrollPhysics(),
          shrinkWrap: true,
          itemCount: state.cooks.length,
          itemBuilder: (context, index) {
            cooksToSearch = state.cooks;
            return CookCard(
              cook: state.cooks[index],
            ); // CookCard
          },
        ), // ListView.builder
      ); // Padding
    } else {
      return Text('Something went wrong');
    }
  },
```

Εικόνα 6:26 κώδικας λίστα συνεργατών

6.4.4 Οθόνη φιλτραρίσματος χρήστη

Στην οθόνη υπάρχουν δύο κυρίαρχα widgets τα οποία το ένα αφορά την τιμή φιλτραρίσματος και το άλλο την κατηγορία φαγητού, στο καθένα υπάρχει η αντίστοιχη λειτουργικότητα για το φιλτράρισμα η οποία επί της ουσίας ενεργοποιείται όταν πατηθεί το κουμπί ‘Αποθήκευση’.

```
body: Padding(
  padding: const EdgeInsets.all(20.0),
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Text(
        'Τιμή',
        style: Theme.of(context).textTheme.headline4!.copyW
            color: Theme.of(context).colorScheme.secondar
          ),
      ), // Text
      CustomPriceFilter(),
      Text(
        'Κατηγορία',
        style: Theme.of(context).textTheme.headline4!.copyW
            color: Theme.of(context).colorScheme.secondar
          ),
      ), // Text
      CustomCategoryFilter(),
    ],
  ), // Column
), // Padding
```

Εικόνα 6:27 Κώδικας για φιλτράρισμα χρήστη

```
onPressed: () {
  List categories = [];
  List prices = [];
  List<Cook> filteredCooks = [];
  for (var s in state.filter.categoryFilters) {
    if (s.value) {
      categories.add(s.category.name);
    }
  }
  for (var s in state.filter.priceFilters) {
    if (s.value) {
      prices.add(s.price.price);
    }
  }
  for (var cook in cooks) {
    for (var cat in cook.categories) {
      for (var price in prices) {
        for (var category in categories) {
          if (cat.name == category &&
              cook.priceCategory == price) {
            filteredCooks.add(cook);
          }
        }
      }
    }
  }
  filteredCooks = filteredCooks.toSet().toList();
```

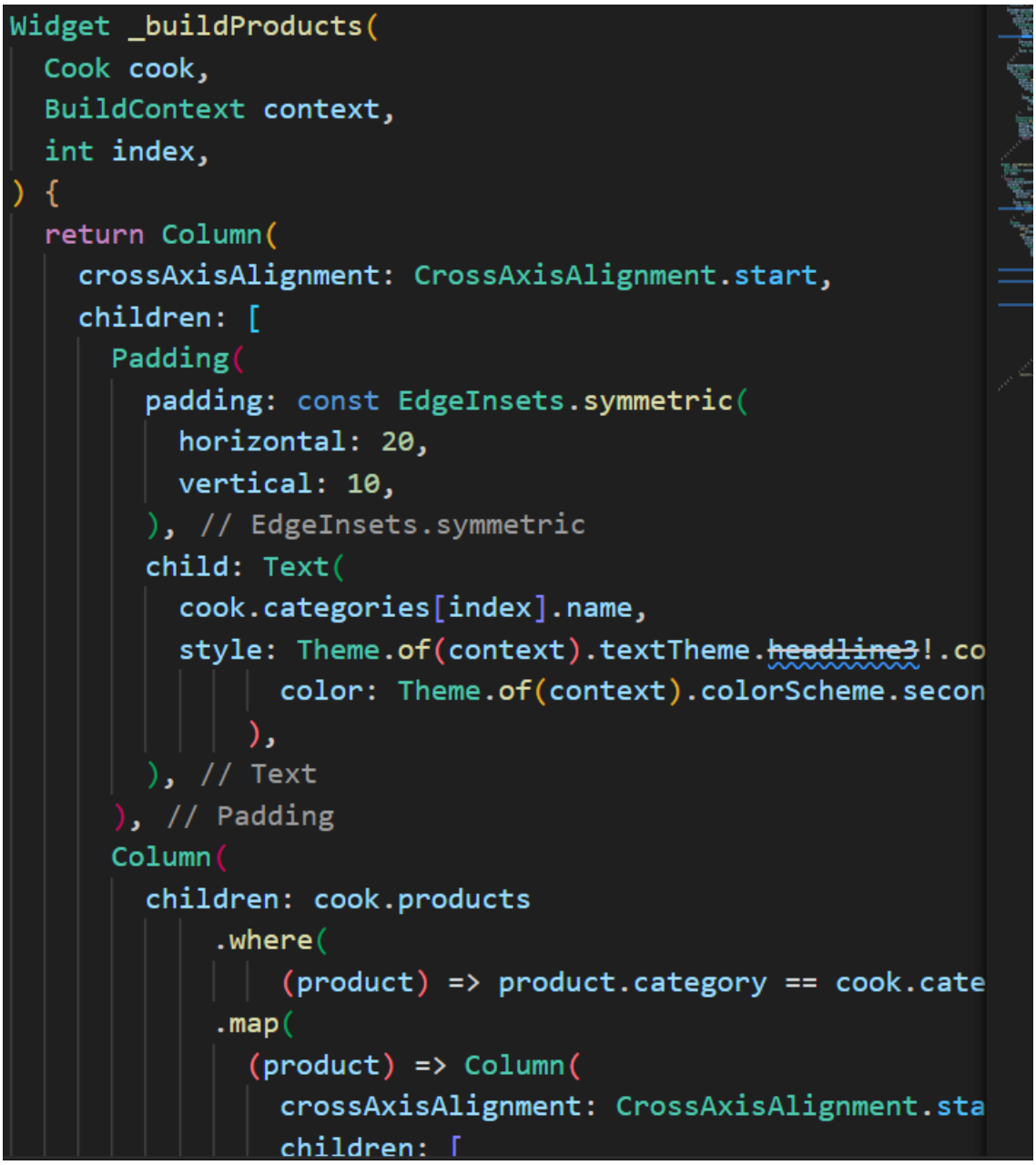
Εικόνα 6:28 Κώδικας κουμπιού φιλτραρίσματος χρήστη

6.4.5 Οθόνη συνεργάτη-μάγειρα

Ο κώδικας της οθόνης του συνεργάτη εμπεριέχει πληροφορίες που απλά ανυπαραστώνται στην οθόνη για τον εκάστοτε επιλεγμένο συνεργάτη το οποίο συμβαίνει αφού το αρχείο CookInformation της συγκερκιμένης οθόνης,το οποίο είναι ένα από τα βασικά widgets του CookDetailsScreen αρχείου για το οποίο μιλάμε τώρα, έχει ως πεδίο ένα αντικείμενο τύπου Cook(συνεργάτη-μάγειρα) όπου έχει όλα τα στοιχεία. Έπειτα έχει μία λίστα από widgets τύπου _buildProducts όπου εκεί γίνεται η προσθήκη στο καλάθι του εκάστοτε προϊόντος εάν αυτό θελήσει και πράξει ο χρήστης.

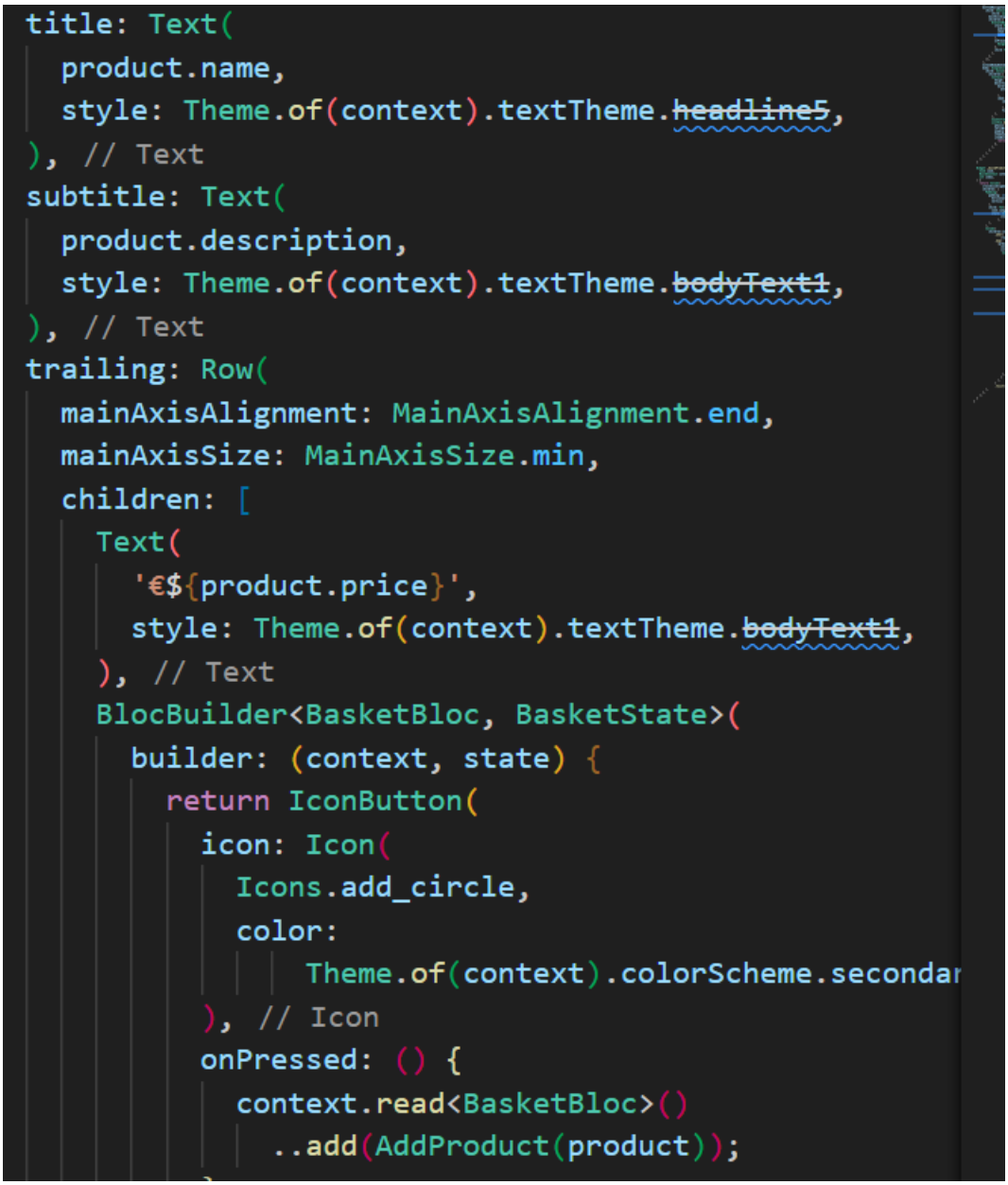
```
body: SingleChildScrollView(
  child: Column(
    children: [
      Container(
        height: 200,
        decoration: BoxDecoration(
          borderRadius: BorderRadius.vertical(
            bottom:
                Radius.elliptical(MediaQuery.of(contex
          ), // BorderRadius.vertical
          image: DecorationImage(
              image: AssetImage(
                cook.imageUrl,
              ), // AssetImage
              fit: BoxFit.cover), // DecorationImage
        ), // BoxDecoration
      ), // Container
      CookInformation(cook: cook),
      ListView.builder(
        physics: NeverScrollableScrollPhysics(),
        shrinkWrap: true,
        padding: EdgeInsets.zero,
        itemCount: cook.categories.length,
        itemBuilder: (context, index) {
          return _buildProducts(cook, context, index);
        },
      ), // ListView.builder
```

Εικόνα 6:29 Code cook details screen



```
Widget _buildProducts(
  Cook cook,
  BuildContext context,
  int index,
) {
  return Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Padding(
        padding: const EdgeInsets.symmetric(
          horizontal: 20,
          vertical: 10,
        ), // EdgeInsets.symmetric
        child: Text(
          cook.categories[index].name,
          style: Theme.of(context).textTheme.headline3!.co
                color: Theme.of(context).colorScheme.secon
              ),
        ), // Text
      ), // Padding
      Column(
        children: cook.products
            .where(
              (product) => product.category == cook.cate
            .map(
              (product) => Column(
                crossAxisAlignment: CrossAxisAlignment.sta
                children: [
```

Εικόνα 6:30 Code build products1



```
title: Text(
  product.name,
  style: Theme.of(context).textTheme.headline5,
), // Text
subtitle: Text(
  product.description,
  style: Theme.of(context).textTheme.bodyText1,
), // Text
trailing: Row(
  mainAxisAlignment: MainAxisAlignment.end,
  mainAxisSize: MainAxisSize.min,
  children: [
    Text(
      '€${product.price}',
      style: Theme.of(context).textTheme.bodyText1,
    ), // Text
    BlocBuilder<BasketBloc, BasketState>(
      builder: (context, state) {
        return IconButton(
          icon: Icon(
            Icons.add_circle,
            color:
                Theme.of(context).colorScheme.secondar
          ), // Icon
          onPressed: () {
            context.read<BasketBloc>()
              ..add(AddProduct(product));
```

Εικόνα 6:31 Code build products2

6.4.6 Καλάθι

Στην οθόνη του καλαθιού, η οποία και αυτή με τη σειρά της περιέχει πολλά widgets, δύο widgets είναι αυτά που ξεχωρίζουν, το ένα είναι αυτό με την λίστα των προϊόντων και το δεύτερο αυτό για το συνολικό ποσό της παραγγελίας. Το δεύτερο ουσιαστικά προσθέτει την τιμή του κάθε προϊόντος πολλαπλασιασμένο με την ποσότητα του κάθε προϊόντος και επίσης επιβάλλει τις εξτρά χρεώσεις όπως αυτή του μαχαιροπίρουνου αλλά επίσης αφαιρεί ποσό σε περίπτωση voucher. Η επόμενη εικόνα δείχνει τον κώδικα για την λίστα των προϊόντων που έχει επιλέξει ο χρήστης την οποία έχει αποθηκεύσει η εφαρμογή στην τοπική βάση της και με τη βοήθεια του Bloc τις ανακτά.

```
ListView.builder(
  shrinkWrap: true,
  itemCount: state.basket
      .itemQuantity(state.basket.products)
      .keys
      .length,
  itemBuilder: (context, index) {
    return Container(
      width: double.infinity,
      margin: const EdgeInsets.only(top: 5),
      padding: const EdgeInsets.symmetric(
        horizontal: 30,
        vertical: 10,
      ), // EdgeInsets.symmetric
      decoration: BoxDecoration(
        color: Colors.white,
        borderRadius: BorderRadius.circular(5.0),
      ), // BoxDecoration
      child: Row(
        mainAxisAlignment:
            MainAxisAlignment.spaceBetween,
        children: [
```

Εικόνα 6:32 Κώδικας για καλάθι1

```
Text(
  '${state.basket.itemQuantity(state.basket.products).entries.elementAt(index).value}x',
  style: Theme.of(context)
      .textTheme
      .headline5!
      .copyWith(
        color: Theme.of(context)
            .colorScheme
            .secondary,
      ),
), // Text
SizedBox(
  width: 20,
), // SizedBox
Expanded(
  child: Text(
    '${state.basket.itemQuantity(state.basket.products).keys.elementAt(index).name}',
    textAlign: TextAlign.left,
    style:
        Theme.of(context).textTheme.headline6,
  ), // Text
), // Expanded
Text(
  '€${state.basket.itemQuantity(state.basket.products).keys.elementAt(index).price}',
  style:
      Theme.of(context).textTheme.headline6,
), // Text
```

Εικόνα 6:33 Κώδικας για καλάθι2

```
Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: [
    Text(
      'Σύνολο',
      style: Theme.of(context)
          .textTheme
          .headline5!
          .copyWith(
              color: Theme.of(context)
                  .colorScheme
                  .primary),
    ), // Text
    Text(
      state.basket.cutlery
          ? '€${state.basket.totalString}'
          : '€${double.parse(state.basket.totalString) - 5
      style: Theme.of(context)
          .textTheme
          .headline5!
          .copyWith(
              color: Theme.of(context)
                  .colorScheme
                  .primary),
    ), // Text
  ],
), // Row
```

ενο

Εικόνα 6:34 Κώδικας για το σύνολο καλαθιού

6.4.7 Ολοκλήρωση παραγγελίας

Στην οθόνη του καλαθιού πατώντας το κουμπί “Ολοκλήρωση παραγγελίας” εκτελείται ο παρακάτω κώδικας ο οποίος ουσιαστικά παίρνει πληροφορίες της παραγγελίας του χρήστη ,τις βάζει σε κείμενο και τις περνάει έτοιμες στο σώμα του mail που είναι έτοιμος να στείλει ο χρήστης.

```
child: Text('Ολοκλήρωση Παραγγελίας'),
onPressed: () async {
  showDialog(
      context: context,
      builder: (builder) {
        return AlertDialog(
          actions: [
            TextButton(
                onPressed: () {
                  Navigator.pop(context);
                },
                child: Text('Ακύρωση')), // TextButton
            TextButton(
                onPressed: () async {
                  final s = BlocProvider.of<CooksBloc>(context)
                      .state
                      .props
                      .asMap();
                  List temp = s.values.toList();
                  List<Cook> cooks = temp[0];
                  // var t = temp[0];

                  final state =
                      BlocProvider.of<BasketBloc>(context)
                          .state
                          .props
                          .asMap();
```

Εικόνα 6:35 Κώδικας ολοκλήρωσης παραγγελίας1

```
  body = l
          .toString()
          .replaceAll('[', '')
          .replaceAll('{', '')
          .replaceAll(']', '')
          .replaceAll('}', '') +
      "Ονοματεπώνυμο Πελάτη: ${stoixeia.text} " +
      ' ,Σύνολο παραγγελίας:$sum, ';
  final Email email = Email(
    body: body,
    subject:
        'Παραγγελία από mail, του ${stoixeia.text}',
    recipients: [cooks[0].email],
    isHTML: true,
  ); // Email

  String platformResponse;
  try {
    await FlutterEmailSender.send(email);
    platformResponse = 'Επιτυχής Αποστολή mail';
  } catch (error) {
    print(error);
    platformResponse = error.toString();
  }
  Navigator.pop(context);
},
child: Text('OK')) // TextButton
```

Εικόνα 6:36 Κώδικας ολοκλήρωσης παραγγελίας2

Κεφάλαιο 7ο: Συμπεράσματα ή/και προτάσεις βελτίωσης

Οι διαδικτυακές εφαρμογές ηλεκτρονικού εμπορίου στον κλάδο της εστίασης έχουν αλλάξει συθέμελα το πως λειτουργεί σήμερα όχι μόνο ο ίδιος ο κλάδος αλλά και ολόκληρο το οικονομικό σύστημα των επιχειρήσεων ανά τις κοινωνίες καθώς άνοιξαν πολλές νέες ευκαιρίες στις επιχειρήσεις αλλά και στους ιδιώτες όσο αναφορά ακόμα και τη παράδοση των προϊόντων στον καταναλωτή. Η δυναμική που αποκτά ένα κατάστημα στον κλάδο της εστίασης μέσω μιάς διαδικτυακής εφαρμογής είναι πολύ μεγαλύτερη των παραδοσιακών μεθόδων και αυτό θεμελιώθηκε πλήρως στην πανδημία του κορονοϊού όπου επιχειρήσεις βγάζαν κέρδη όπως ποτέ άλλοτε αποκλειστηκά και μόνο εξυπηρετώντας κατ'οίκον παραγγελίες. Φυσικά μετά την περίοδο αυτή ο μέσος καταναλωτής έχοντας πλέον γνωρίσει την ευκολία της διαδικασίας παραγγελίας και παράδοσης μέσω αυτών των εφαρμογών συνεχίζει να τροφοδωτεί το συγκεκριμένο σύστημα με μεγάλη ευχαρίστηση και συνέπεια. Αυτό το συμπέρασμα φυσικά καθιστά τέτοιες εφαρμογές ζωτικής σημασίας για τον ίδιο τον κλάδο καθώς δεν απευθύνεται σε επιχειρήσεις αλλά σε ιδιώτες,όσο αναφορά τους καταναλωτές όμως υπάρχει το ίδιο μοτίβο με τις υπόλοιπες εφαρμογές του εμπορίου όπου απευθύνονται σε επιχειρήσεις. Η συγκεκριμένη εφαρμογή όμως στοχεύοντας στην συνεργασία και ανάδειξη επαγγελματιών και μη του χώρου προσδίδει μία έξτρα λειτουργικότητα και μία διαφορετική μάτια στον χώρο του κλάδου της εστίασης που δεν είχαμε ξανασυναντήσει στον συγκεκριμένο κλάδο, σε άλλους κλάδους όμως όπως αυτός της μεταφοράς είχαμε παρόμοια παραδείγματα όπως αυτό της Uber όπου ένας απλός πολίτης μπορεί να προσφέρει υπηρεσίες ταξί/μεταφοράς σε έναν πελάτη της εφαρμογής. Η διαφορά με τη δικιά μας εφαρμογή(Gourmeet) είναι ότι το Uber δεν στοχεύει στην ανάδειξη την συνεργατών της στο χώρο.Ο κλάδος έχει ανάγκη από διαφορετικές προσεγγίσεις και οι επιχειρήσεις-συνεργάτες είναι οφέλιμο για το κοινωνικό σύνολο να εκμεταλλεύονται όλες τις νέες τεχνολογίες που είναι διαθέσιμες ώστε να έχουν το καλύτερο δυνατό αποτέλεσμα όσο αναφορά την αύξηση της ποιότητας των ηλεκτρονικών υπηρεσιών που παρέχουν προς τους χρήστες-πελάτες.

Από την άλλη μεριά, οι προγραμματιστές τέτοιων εφαρμογών πρέπει να παρακολουθούν όλες τις τελευταίες τεχνολογικές εξελίξεις, ώστε να είναι σε θέση να παρέχουν ένα ασφαλές και αποδοτικό περιβάλλον για τον εκάστοτε συνεργάτη μιας τέτοιας εφαρμογής.

Ως προς την ανάπτυξη μιας διαδικτυακής ή κινητής εφαρμογής για τον κλάδο της εστίασης, πρέπει να γίνεται καλός σχεδιασμός ως προς το τι δεδομένα θα επεξεργάζεται η εφαρμογή, άρα το πως θα σχεδιαστεί η βάση δεδομένων , το τι λειτουργίες θέλουμε να έχει η εφαρμογή και τι τεχνολογίες θα χρησιμοποιήσουμε για να φτάσουμε στο επιθυμητό αποτέλεσμα.

Παρόμοιες εφαρμογές με τη Gourmeet πραγματοποιούν αλλαγές και βελτιώσεις ανά πολύ συχνά διαστήματα προσθέτοντας συχνά καινούργιες λειτουργίες προσφορές και υπηρεσίες με γνώμονα πάντα τις ανάγκες των καταστημάτων που συνεργάζονται, τους εξωτερικούς συνεργάτες (διανομείς) αλλά και φυσικά τους ίδιους τους πελάτες-χρήστες των εφαρμογών. Οι εφαρμογές της πτυχιακής εργασίας εξυπηρετούν με παρόμοιο τρόπο τον χρήστη οπότε όσο αναφορά αυτό το κομμάτι μπορούν να προβλεθούν κάποιες βελτιώσεις που μπορούν να γίνουν στο εγγύς μέλλον αλλά όσο αναφορά το κομμάτι των συνεργατών αποτελεί δυσκολία να προβλεθούν τυχόν αλλαγές ή βελτιώσεις καθώς είναι ένας τομέας που δεν έχει εξερευνηθεί ιδιαίτερα και ειδικά στον ελληνικό χώρο οπότε σίγουρα όταν ή αν βγεί παραγωγικά οι συγκεκριμένες εφαρμογές θα πρέπει να δωθεί μεγάλη σημασία στα σχόλια των συνεργατών καθώς και, όπως γίνεται με όλες τις καινούργιες κυρίως κινητές εφαρμογές, η εμπειρία χρήστη θα καθορίσει πολλές από αυτές τις αλλαγές και τις βελτιώσεις.

Επεκτάσεις που μπορούν να γίνουν αρχικά έχουν να κάνουν με έναν τομέα που δεν έχει εμβαθύνει ιδιαίτερα η συγκεκριμένη εφαρμογή-πτυχιακή και είναι η παράδοση των προϊόντων στον χρήστη καθώς η εφαρμογή έχει σαν επιλογή ο ίδιος ο συνεργάτης να παραδίδει την παραγγελιά ή, το πιο σύνηθες μάλλον στη συγκεκριμένη περίπτωση, θα είναι ένας δεύτερος εξωτερικός συνεργάτης-

διανομέας να κάνει τη παράδοση, σε αυτή τη περίπτωση θα πρέπει να δημιουργηθεί και μία ακόμα εφαρμογή που να εξυπηρετεί αυτό το σκοπό.

Βελτιώσεις μπορούν να γίνουν και όσο αναφορά την εφαρμογή του συνεργάτη της πλατφόρμας ο οποίος δημιουργεί ένα προφίλ με εβδομαδιαίο ωράριο,λίστα προϊόντων αλλά και κάποια σχόλαι για τον εαυτό του αλλά θα μπορούσε να είναι διαθέσιμες κι άλλες πληρφορορίες όπως προϋπηρεσία χρήστη ή κάποιου είδους πιστοποίηση.

Όσο αναφορά την κινητή εφαρμογή του χρήστη θα μπορούσαν στο μέλλον να γίνουν κάποιες βελτιώσεις που όπως είπαμε θα ακολουθούν το γενικότερο κλίμα που επικρατεί και μεταβάλλεται παρόμοιων εφαρμογών του ίδιου κλάδου. Άλλες χρήσιμες υπηρεσίες που θα μπορούσαν να προστεθούν είναι ένα ιστορικό παραγγελιών χρήστη, η δυνατότητα βαθμολόγησης συνεργάτη ή ακόμα και η δυνατότητα προσθήκης λίστας των αγαπημένων συνεργατών κάθε χρήστη. Αργότερα όσο αναφορά τη διανομή ίσως ήταν χρήσιμη η προσθήκη ζωντανής παρακολούθησης της παραγγελίας όπως έχουν ήδη κάποιες εφαρμογές του κλάδου.Τελευταίο αλλά εξίσου σημαντικό να μπορεί ο χρήστης να πληρώσει με κάρτα,βέβαια αυτό θα πρέπει να είναι στα πλαίσια του νομικού πλαισίου της χώρας και ίσως είναι μια λειτουργία που χρειάζεται μεγάλη προσοχή καθώς μέχρι στιγμής τέτοιο νομικό πλαίσιο δεν υπάρχει αυτή τη στιγμή.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Βιβλία

[1] Rap Payne, Beginning App Development with Flutter: Create Cross-Platform Mobile Apps, 1st edition, 2019.

[2] Priyanka Tyagi, Prgmatic Flutter: Building Cross-Platform Mobile Apps for Android, iOS, Web & Desktop, 1st edition, 2021.

Internet Site

[3] Flutter Awesome, a curated collection of high-quality Flutter packages and resources in other words a community-driven platform where developers can discover and share Flutter libraries, tools, tutorials, and articles. It helps developers to find reusable components and solutions that can be integrated into their Flutter projects to enhance development efficiency and productivity. Available: https://flutterawesome.com/tag/login-screen/

[4] Flutter offical web page, the official web page for Flutter where developers can be helped by the official Google Documentation. Available: https://docs.flutter.dev/resources/inside-flutter

[5] Discover Firebase for Flutter, A guide for developers on how to implement Firebase in their Flutter project. Available: https://firebase.google.com/docs/flutter

[6] Codelab: Get to know Firebawse for Flutter, the first guide to get to know how firebase works and how it can be implemented in the Flutter project.Available: https://www.youtube.com/watch?v=wUSkeTaBonA

[7] A comprehensive guide to Flutter routing, get to know how Flutter routing works and how it should be used. Available: https://www.dhiwise.com/post/deep-dive-into-flutter-routing-everything-you-need-to-know

[8] Hive package, how to use Hive package in Flutter app to store data. Available: https://pub.dev/packages/hive

[9] Box class Hive, what is box class that hiive package uses. Available: https://pub.dev/documentation/hive/latest/hive/Box-class.html

[10] Flutter navigation between pages website, How to navigate between pages. Available: https://docs.flutter.dev/ui/navigation

[11] Flutter named routing, how to use named routes to navigate through pages. Available: https://docs.flutter.dev/cookbook/navigation/named-routes

[12] How to use Flutter Bloc Architecture to build High Performance App, describes bloc pattern as well as how and why to use it in your Flutter app. Available: https://ripenapps.com/blog/how-to-use-flutter-bloc-architecture-to-build-high-performance-apps/

[13] Widgets and Helper methods, a youtube video that explains basic concepts about Flutter's widgets. Available: https://www.youtube.com/embed/IOyq-eTRhvo?rel=0&enablejsapi=1&origin=https%3A%2F%2Fapi.flutter.dev

[14] How to create Stateless WIdgets- Flutter Widgets Youtube Video, a guide on how to creaate stateless widgets by google developers team. Available: https://www.youtube.com/embed/wE7khGHVkYY?rel=0&enablejsapi=1&origin=https%3A%2F%2Fapi.flutter.dev

[15] How to create stateless widgets guide. Available: https://api.flutter.dev/flutter/widgets/StatelessWidget-class.html

[16] Adding assets and Εικόνας in Flutter project, Flutter's official documents page on how to add assets and Εικόνας into your project. Available: https://docs.flutter.dev/ui/assets/assets-and-Εικόνας

[17] Intergrate Flutter Web, how to deploy your Flutter app to Firebase. Available: https://firebase.google.com/docs/hosting/frameworks/flutter

[18] Dhttpd, a simple Http server that can serve up any directory, built with Dart. Available: https://pub.dev/packages/dhttpd

[19] Web renderers. Available: https://docs.flutter.dev/platform-integration/web/renderers

[20] Bloc library, A predictable state management librady for Dart. Available: https://bloclibrary.dev/

[21] Bloc Tutorial, How Bloc pattern works and how to user it complete guide. Availaible: https://www.dhiwise.com/post/flutter-bloc-tutorial-understanding-state-management