



ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη Android εφαρμογή παραγγελίας σε χώρο
εστίασης



Flutter

Του φοιτητή
Ζήσης Σπάτης
Αρ. Μητρώου: 164748

Επιβλέπων
Ευκλείδης Κεραμόπουλος
Αναπληρωτής Καθηγητής

Σεπτέμβριος 2020

Τίτλος Δ.Ε. Ανάπτυξη Android εφαρμογή παραγγελίας σε χώρο εστίασης

Κωδικός Π.Ε. 20189

Ονοματεπώνυμο φοιτητή/τών Ζήσης Σπάτης

Ονοματεπώνυμο εισηγητή Ευκλείδης Κεραμόπουλος

Ημερομηνία ανάληψης Π.Ε. 17-09-2020

Ημερομηνία περάτωσης Π.Ε. 19-04-2021

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 παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Ζήση Σπάτη που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

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

Πρόλογος

Ο λόγος που επέλεξα να κάνω αυτή την Πτυχιακή Εργασία ήταν μια σκέψη την οποία είχα πριν αρκετό καιρό, σχετικά με το πως θα μπορούσαμε να απλοποιήσουμε μια υπηρεσία, όπως είναι η επίσκεψη σε χώρους εστίασης όσον αφορά στον τρόπο παραγγελίας. Επίσης ένας ακόμη λόγος ήταν να μάθω όλη την διαδικασία ανάπτυξης μιας Android εφαρμογής, ώστε να βελτιώσω τις γνώσεις μου γύρω από αυτόν τον τομέα. Τέλος, με την ολοκλήρωση αυτής της Πτυχιακής Εργασίας θα προσπαθήσω να αξιοποιήσω όλες τις γνώσεις τις οποίες πήρα κατά την περίοδο ολοκλήρωσης της και να μπορέσω να προωθήσω την ιδέα για μελλοντική χρήση σε διάφορα καταστήματα εστίασης.

Περίληψη

Αντικείμενο της παρούσας Πτυχιακής Εργασίας είναι η ανάπτυξη μιας εφαρμογής σε πλατφόρμα Android για αποστολή παραγγελίας από πελάτες σε καταστήματα εστίασης (cafe, beach bar, εστιατόρια κλπ). Με την εφαρμογή αυτή ο πελάτης θα έχει την δυνατότητα κατά την επίσκεψη του σε ένα κατάστημα εστίασης, να μπορεί να παραγγείλει αλλά και να πληρώσει το λογαριασμό μέσα από την εφαρμογή. Η πρόσβαση στο menu και στο σύστημα παραγγελίας θα γίνεται με αναγνώριση του QRCODE που θα υπάρχει στο τραπέζι του αντίστοιχου καταστήματος. Τα πλεονεκτήματα της εφαρμογής αυτής είναι η εύκολη και γρήγορη πρόσβαση στο menu παράλληλα με την εικονική απεικόνιση του, η αποφυγή συνωστισμού και η ταλαιπωρία που θα μπορούσε να δημιουργηθεί λόγω της πολυκοσμίας στο κατάστημα. Για την ανάπτυξη της εφαρμογής αυτής χρησιμοποιήθηκε μια πληθώρα τεχνολογιών αλλά και βιβλιοθηκών ώστε να δοθεί το καλύτερο αποτέλεσμα. Η υλοποίηση της εφαρμογής έγινε με το Flutter το οποίο χρησιμοποιεί τη γλώσσα Dart στο περιβάλλον ανάπτυξης Android Studio. Η αποθήκευση των δεδομένων έγινε στη πλατφόρμα Firebase και όπου χρησιμοποιήθηκαν πολλές υπηρεσίες τις οποίες διαθέτει.

Android order app for eating establishments

Zisis Spatis

Abstract

The aim of this dissertation is to develop an Android application which allows customers of an eating establishment to place an order. Upon arrival at an eating establishment, the customer will be able to place an order and pay the bill using this app. The access to the menu and the ordering environment will be through the scanning of a QR CODE found at each table. The advantages of this application are the easy and fast access to the menu, its visual depiction of the menu, the prevention of overcrowding and the minimization of the waiting time when the eating establishment is busy. To develop this app a range of technologies and libraries were used in order to have the best possible result. The implementation of the app took place using Flutter, which uses the language Dart in an Android Studio environment. The data saving happened in Firebase and a variety of its services was used.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον καθηγητή μου Ευκλείδη Κεραμόπουλο για την εμπιστοσύνη που μου έδειξε από την πρώτη στιγμή που του ζήτησα να αναλάβει την Πτυχιακή Εργασία μου. Επίσης θα ήθελα να ευχαριστήσω την οικογένεια μου για την υποστήριξη που μου έδειξαν κατά την διάρκεια των σπουδών μου τόσο ψυχολογική όσο και οικονομική αλλά και τους φίλους μου.

Περιεχόμενα

Περίληψη	iv
Abstract	v
Ευχαριστίες	vi
Περιεχόμενα	vii
Κατάλογος Σχημάτων	x
Κατάλογος Πινάκων	xi
Συντομογραφίες.....	xii
Κεφάλαιο 1ο: Εισαγωγή.....	1
1.1 Περιγραφή Εργασίας.....	1
1.2 Δομή Εργασίας.....	1
Κεφάλαιο 2ο: Εισαγωγή στο Flutter	3
2.1 Εισαγωγή.....	3
2.2 Τι είναι η Dart	3
2.2.1 Λίγα λόγια για τη Dart	3
2.2.2 Η εξέλιξη της Dart.....	3
2.2.3 Πως λειτουργεί η Dart.....	4
2.2.4 OOP Dart Χαρακτηριστικά	6
2.3 Γιατί το Flutter χρησιμοποιεί το Dart	7
2.4 Τι είναι το flutter	8
2.4.1 Πως ξεκίνησε το Flutter	8
2.4.2 Τι είναι αυτό που ήθελε να λύσει το Flutter	8
2.4.3 Άλλα αντίστοιχα frameworks	9
2.4.4 Διαφορές σε σχέση με τα άλλα frameworks	10
2.4.5 Εισαγωγή στα Widgets.....	12
2.4.6 Ποιος μπορεί να χρησιμοποιήσει το Flutter	17
2.4.7 Εγκατάσταση του Flutter	17
2.5 Επίλογος.....	17
Κεφάλαιο 3ο: Firebase	18
3.1 Εισαγωγή.....	18
3.2 Κατηγορίες προϊόντων.....	18
3.2.1 Ανάπτυξη (Develop)	18
3.2.2 Ποιότητα (Quality)	19

3.2.3	Ανάπτυξη (Grow)	19
3.3	Cloud Firestore.....	20
3.3.1	Κανόνες ασφαλείας Cloud Firestore.....	21
3.4	Firebase Authentication	21
3.4.1	Email/Password	22
3.4.2	Google – Facebook	22
3.5	Επίλογος.....	22
Κεφάλαιο 4ο:	QR Code.....	23
4.1	Εισαγωγή	23
4.2	Λίγα λόγια για το QR Code	23
4.3	Κωδικοί QR Code έναντι γραμμικών κωδικών (Barcode)	23
4.4	Ιστορική αναδρομή.....	24
4.5	Διάφοροι τύποι κωδικών QR	24
4.6	Η δομή ενός QR Code	24
4.7	Στατικοί και δυναμικοί κώδικες QR.....	27
4.7.1	Στατικοί κώδικες.....	27
4.7.2	Δυναμικοί κώδικες.....	28
4.8	Επίλογος.....	28
Κεφάλαιο 5ο:	Υλοποίηση εφαρμογής.....	29
5.1	Εισαγωγή	29
5.2	Δημιουργία Flutter project.....	29
5.3	Δημιουργία Firebase project	31
5.4	Android Manifest, Grandle και pubspec.yaml	32
5.5	Ανάλυση της βάσης δεδομένων της εφαρμογής	34
5.6	Επίλογος.....	36
Κεφάλαιο 6ο:	Παρουσίαση της εφαρμογής και Οδηγίες Χρήσης	37
6.1	Εισαγωγή	37
6.2	Λίγα λόγια για την έναρξη της εφαρμογής	37
6.3	Είσοδος χρήστη και εγγραφή.....	38
6.4	QR Code scanner	39
6.5	Κύριο μενού	40
6.6	Προσφορές.....	41
6.7	Υποκατηγορία του μενού.....	42
6.7.1	Καφές.....	42
6.7.2	Ποτά.....	46

6.7.3	Όλες οι υπόλοιπες υποκατηγορίες	48
6.8	Καλάθι	48
6.9	Προφίλ	52
6.9.1	Ο λογαριασμός μου.....	52
6.9.2	Παραγγελίες	55
6.9.3	Αποθηκευμένες κάρτες	57
6.9.4	Γλώσσα	58
6.9.5	Τρόποι πληρωμής	59
6.10	Μηνύματα Σφάλματος.....	60
Κεφάλαιο 7ο:	Συμπέρασμα.....	63
Βιβλιογραφία.....		64

Κατάλογος Σχημάτων

Σχήμα 1: AOT Compilation	4
Σχήμα 2: Dart2js Compilation.....	5
Σχήμα 3: Develop - Deploy Διαδικασία	5
Σχήμα 4: Τα τέσσερα ποιο διαδεδομένα framework	9
Σχήμα 5: Ένα διάγραμμα Widget Tree	14
Σχήμα 6: Stateless Widget Lifecycle	15
Σχήμα 7: Stateful Widget Lifecycle.....	16
Σχήμα 8: QR Code και Barcode σχήματα.....	23
Σχήμα 9: Δείκτες εντοπισμού θέσης.....	25
Σχήμα 10: Σήματα ευθυγράμμισης.....	25
Σχήμα 11: Σχέδιο χρονισμού.....	26
Σχήμα 12: Πληροφορίες έκδοσης.....	26
Σχήμα 13: Μορφοποίηση πληροφοριών	26
Σχήμα 14: Πλήκτρα διόρθωσης δεδομένων και σφαλμάτων	27
Σχήμα 15: Quit zone	27
Σχήμα 16: Αρχική οθόνη του Android Studio.....	29
Σχήμα 17: Δημιουργία Flutter Project	30
Σχήμα 18: Δομή αρχείων ενός Flutter Project.....	30
Σχήμα 19: AndroidManifest αρχείο.....	32
Σχήμα 20: Αρχείο build.gradle στο φάκελο android/app/	33
Σχήμα 21: Αρχείο build.gradle στο φάκελο android/.....	33
Σχήμα 22: Αρχείο pubspec.yaml 1	33
Σχήμα 23: Αρχείο pubspec.yaml 2	34
Σχήμα 24: Αρχείο pubspec.yaml 3	34
Σχήμα 25: Αρχική οθόνη εφαρμογής.....	37
Σχήμα 26: Είσοδος χρήστη.....	38
Σχήμα 27: Εγγραφή χρήστη	39
Σχήμα 28: Σκανάρισμα του QR Code.....	40
Σχήμα 29: Κύριο Μενού	41
Σχήμα 30: Προσφορές μαγαζιού	42
Σχήμα 31: Κατηγορία καφέ.....	43
Σχήμα 32: Λεπτομέρειες παραγγελίας καφέ	44
Σχήμα 33: Επιλογή Μέγεθος καφέ	45
Σχήμα 34: Επιλογή Ζάχαρης καφέ	45
Σχήμα 35: Μήνυμα επιτυχίας εισόδου στο καλάθι.....	46
Σχήμα 36: Επιλογή αλκοολούχου ποτού.....	46
Σχήμα 37: Επιλογή Χυμού	47
Σχήμα 38: Επιλογή αναψυκτικού	47
Σχήμα 39: Επιλογή προϊόντος μύρα	48
Σχήμα 40: Ένδειξη προϊόντων στο καλάθι.....	49
Σχήμα 41: Καλάθι.....	49
Σχήμα 42: Φόρμα συμπλήρωσης εισόδου κάρτας.....	50
Σχήμα 43: Επιβεβαίωση παραγγελίας.....	51
Σχήμα 44: Μήνυμα επιτυχής παραγγελίας.....	51

Σχήμα 45: Επιλογές προφίλ χρήστη	52
Σχήμα 46: Πληροφορίες χρήστη συνδεδεμένου με Google και Facebook.....	53
Σχήμα 47: Πληροφορίες απλού χρήστη	54
Σχήμα 48: Αλλαγή κωδικού	55
Σχήμα 49: Παραγγελίες.....	56
Σχήμα 50: Πληροφορίες παραγγελίας	57
Σχήμα 51: Κάρτες.....	58
Σχήμα 52: Επιλεγμένη γλώσσα	59
Σχήμα 53: Τρόποι πληρωμής	60
Σχήμα 54: Μήνυμα άδειου καλαθιού.....	61
Σχήμα 55: Μήνυμα λάθους επιλογής 1.....	61
Σχήμα 56: Μήνυμα λάθους επιλογής 2.....	62

Κατάλογος Πινάκων

Πίνακας 1: Μέθοδοι του Stateful Widget	16
Πίνακας 2: Αντιστοιχία SQL Server με Firestore	20
Πίνακας 3: Παράδειγμα καταχώρησης στη βάση Firestore Cloud.....	20
Πίνακας 3: Users Collection Ανάλυση	35
Πίνακας 3: Customers Collection Ανάλυση.....	35

Συντομογραφίες

AOT	Ahead Of Time
CDN	Content Delivery Network
DVM	Dart Virtual Machine
IDE	Integrated Development Environment
JIT	Just In Time
JVM	Java Virtual Machine
OEM	Original Equipment Manufacturer
OOP	Object Oriented Programming
QR	Quick Response
SDK	Software Development Kit
SQL	Structured Query Language
SSL	Secure Sockets Layer
UI	User Interface
VS	Visual Studio

Κεφάλαιο 1ο: Εισαγωγή

1.1 Περιγραφή Εργασίας

Αντικείμενο της πτυχιακής εργασίας είναι η υλοποίηση μιας εφαρμογής σε πλατφόρμα Android. Πιο αναλυτικά, η εφαρμογή έχει να κάνει με την αποστολή παραγγελίας από τους πελάτες σε καταστήματα εστίασης (πχ καφετέριες, εστιατόρια, beach bar κτλπ.). Επομένως, για να χρησιμοποιήσει κανείς αυτή την εφαρμογή θα πρέπει να βρίσκεται σε ένα συγκεκριμένο κατάστημα που συνεργάζεται με αυτή την υπηρεσία. Ο πελάτης θα έχει την δυνατότητα να παραγγείλει μέσω του κινητού του, άμεσα και χωρίς καθυστέρηση. Μέσω ενός αναλυτικού μενού, το οποίο θα έχει τη δυνατότητα εικονικής απεικόνισης και αναλυτικής περιγραφής των προϊόντων, θα μπορεί ο πελάτης να διαλέξει το προϊόν που επιθυμεί και να παραγγείλει στο αντίστοιχο κατάστημα. Όλη αυτή η διαδικασία θα γίνεται μέσω των QR Codes τα οποία θα βρίσκονται στο κάθε τραπέζι του αντίστοιχου καταστήματος. Επομένως, ο πελάτης μέσω της εφαρμογής θα σκανάρει το αντίστοιχο QR Code ώστε να μπορεί να περιηγηθεί στο μενού. Το σημαντικότερο και τελευταίο χαρακτηριστικό είναι η πληρωμή μέσω της εφαρμογής. Αυτό έχει σαν αποτέλεσμα την αποφυγή συνωστισμού και ταλαιπωρίας τόσο στη διάρκεια της παραγγελίας όσο και στη πληρωμή.

Σκοπός της συγκεκριμένης εργασίας είναι να υλοποιηθεί μια εφαρμογή η οποία θα μπορέσει να βελτιώσει μια πτυχή της καθημερινότητας μας, την οποία μπορούμε να αλλάξουμε σε μεγάλο βαθμό. Γι' αυτό το λόγο θέλησα να δημιουργήσω μια εφαρμογή η οποία, να είναι προσιτή προς το κόσμο, με το δεδομένο ότι στις μέρες μας, η τεχνολογία έχει ενταχθεί στην καθημερινότητα μας. Όλα αυτά μου έδωσαν ένα κίνητρο να ασχοληθώ με ένα κλάδο, αυτόν της εστίασης, ο οποίος το τελευταίο διάστημα λόγω της πανδημίας έχει ξεκινήσει να αναβαθμίζεται τεχνολογικά. Τον τελευταίο καιρό, το μενού όλο και περισσότερων καταστημάτων ψηφιοποιείται. Επομένως ο κόσμος άρχισε και συνηθίζει αυτού του είδους την διαδικασία για να ενημερωθεί για τα προϊόντα που διαθέτει ένα κατάστημα. Αυτή είναι η αρχή ώστε ο κόσμος να μπορέσει να απαλλαγεί από την συνηθισμένη παραγγελία ενός προϊόντος στα καταστήματα εστίασης. Πλέον, μέσω της εφαρμογής θα μπορέσει ο πελάτης να αποφασίζει από μόνος του την παραγγελία, που επιθυμεί να κάνει. Όλα αυτά πραγματοποιούνται μέσω μιας εύχρηστης εφαρμογής η οποία πέρα από την παραγγελία, προσφέρει μια αναλυτική περιγραφή του μενού στην οποία ο πελάτης μπορεί να δει εικονικά όλα τα προϊόντα τα οποία υπάρχουν στο κατάστημα και να διαλέξει ένα από αυτά. Με αυτό το τρόπο, η παραγγελία γίνεται πιο διαδραστική, ευχάριστη αλλά και βοηθητική προς τον πελάτη, για το λόγο ότι γνωρίζει ακριβώς αυτό το οποίο θα παραγγείλει.

Όλα αυτά θα φέρουν μια καινούργια εξέλιξη ως προς την παραγγελία στα καταστήματα εστίασης. Πλέον το κάθε κατάστημα θα επενδύει στο να παρουσιάσει ένα όμορφο εικονικά μενού το οποίο θα προσελκύει το κόσμο στο να το εξερευνήσει. Επίσης, ο κόσμος θα γνωρίζει αναλυτικά τα προϊόντα τα οποία παρέχει το κάθε κατάστημα το οποίο θέλει να επισκεφτεί. Επιπλέον, με την παραγγελία και πληρωμή του πελάτη, αποκλείονται τυχόν λάθη είτε από την πλευρά του πελάτη είτε του καταστήματος. Τέλος, ένα από τα σημαντικότερα χαρακτηριστικά τα οποία μπορεί να γλιτώσει η συγκεκριμένη εφαρμογή είναι η μείωση αναμονής για την παραγγελία στο αντίστοιχο κατάστημα.

1.2 Δομή Εργασίας

Η εργασία αποτελείται από 6 κεφάλαια. Στο πρώτο κεφάλαιο παρουσιάζουμε συνοπτικά τη δομή και το περιεχόμενο της εργασίας. Στο δεύτερο, κάνουμε μια εισαγωγή στη τεχνολογία που χρησιμοποιείται στην εφαρμογή. Αναλυτικότερα, βλέπουμε τι είναι το Flutter και αναλύουμε τη γλώσσα που δουλεύει

Κεφάλαιο 1

η οποία είναι η Dart. Ακόμη, αναλύουμε για ποιους λόγους το Flutter έχει επιλέξει τη Dart σαν βασική γλώσσα της και τέλος, βλέπουμε αναλυτικά πως δουλεύει και τι είναι το Flutter. Στη συνέχεια στο τρίτο κεφάλαιο περιγράφουμε τι είναι το Firebase. Αναλύουμε τις υπηρεσίες που διαθέτει και μιλάμε αναλυτικότερα για τις δυο πιο βασικές υπηρεσίες, οι οποίες και χρησιμοποιούνται στην εφαρμογή. Στο τέταρτο κεφάλαιο μιλάμε για το QR Code, κάνουμε μια μικρή ιστορική αναδρομή, βλέπουμε διάφορους τύπους QR Code και τέλος αναλύουμε τη δομή ενός τέτοιου κώδικα. Το πέμπτο κεφάλαιο έχει να κάνει με την υλοποίηση της εφαρμογής όπου αναλύονται κάποια σημαντικά στοιχεία και διάφορες προσθήκες που έγιναν ώστε να υλοποιηθεί η εργασία. Στο έκτο και τελευταίο κεφάλαιο έχουμε μια αναλυτική παρουσίαση της εργασίας με οδηγίες χρήσης, μέσω των οποίων μπορεί κανείς να καταλάβει πως δουλεύει η εφαρμογή.

Κεφάλαιο 2ο: Εισαγωγή στο Flutter

2.1 Εισαγωγή

Στο παρακάτω κεφάλαιο θα αναλύσουμε το Flutter, ξεκινώντας από την γλώσσα την οποία χρησιμοποιεί, που είναι η Dart. Θα εμβαθύνουμε στη συγκεκριμένη γλώσσα, αναλύοντας πως λειτουργεί, την εξέλιξη της αλλά και γιατί την χρησιμοποιεί το Flutter. Στη συνέχεια, θα αναφερθούμε στο Flutter. Συγκεκριμένα, για το πως ξεκίνησε και τι ήταν αυτό που ήθελε να καταφέρει σαν ένα εργαλείο ανάπτυξης εφαρμογών. Ακόμη, θα εμβαθύνουμε στα Widgets τα οποία είναι ένα από τα πολύ σημαντικά χαρακτηριστικά του Flutter και τέλος θα εξηγήσουμε πως μπορεί κάποιος να κάνει εγκατάσταση το συγκεκριμένο εργαλείο.

2.2 Τι είναι η Dart

Η γλώσσα Dart αποτελεί το πυρήνα του Flutter. Μια καινούργια τεχνολογία όπως το Flutter απαιτεί μια σύγχρονη γλώσσα υψηλού επιπέδου για να είναι σε θέση να παρέχει το καλύτερο δυνατό αποτελέσματα και εμπειρία στο προγραμματιστή, καθιστώντας δυνατή τη δημιουργία εκπληκτικών εφαρμογών για κινητά. Ένας προγραμματιστής ο οποίος θέλει να ασχοληθεί με την τεχνολογία Flutter πρέπει να είναι σε θέση να γνωρίζει την προέλευση της γλώσσας Dart, πως δουλεύει, τα πλεονεκτήματα της και γιατί είναι επιλεγμένη γλώσσα προγραμματισμού για την ανάπτυξη του Flutter. Σε αυτή την υποενότητα θα δούμε πως λειτουργεί η γλώσσα Dart και θα κάνουμε μια εισαγωγή του OOP (object-oriented programming) για την Dart.

2.2.1 Λίγα λόγια για τη Dart

Η γλώσσα Dart αναπτύχθηκε από την Google και είναι μια γλώσσα προγραμματισμού που μπορεί να χρησιμοποιηθεί για την ανάπτυξη Web, Mobile, Desktop εφαρμογών. Η συγκεκριμένη γλώσσα χρησιμοποιείται από την Flutter για την ανάπτυξη υψηλού επιπέδου εφαρμογών και για να δίνει την καλύτερη δυνατή εμπειρία στο προγραμματιστή.

2.2.2 Η εξέλιξη της Dart

Η πρώτη φορά που παρουσιάστηκε η γλώσσα ήταν το 2011 και από τότε δεν έχει σταματήσει να εξελίσσεται. Η σταθερή έκδοση της γλώσσας ανακοινώθηκε το 2013 με πολύ σημαντικές αλλαγές που περιλαμβάνουν την κυκλοφορία του Dart 2.0 προς τα τέλη του 2018. Στην αρχή επικεντρώθηκαν στο «web development» για την ανάπτυξη και σύλληψη ιστοσελίδων, με πολύ μεγάλο και κυρίαρχο στόχο την αντικατάσταση της JavaScript. Ωστόσο, η γλώσσα το τελευταίο διάστημα επικεντρώνεται στην ανάπτυξη κινητών εφαρμογών και γι αυτό επιλέχθηκε για τη συγκεκριμένη εργασία. Επομένως, ο σημαντικότερος στόχος της ήταν να λύσει κάποια προβλήματα τα οποία αντιμετώπιζε η JavaScript. Επίσης, έχει στο δυναμικό της καλύτερη απόδοση και καλύτερα εργαλεία για έργα μεγάλης κλίμακας. Η γλώσσα ακόμη, διαθέτει σταθερά και μοντέρνα εργαλεία από τα IDE plugins και έχει σχεδιαστεί για να έχει την καλύτερη δυνατή απόδοση, διατηρώντας την αίσθηση μια δυναμικής γλώσσας. Τέλος, η γλώσσα διαθέτει χαρακτηριστικά OOP και διατηρεί τις επάλξεις για μια γλώσσα στιβαρή και ευέλικτη, η οποία είναι πολλαπλών πλατφορμών και βελτιώνει συνεχώς τα χαρακτηριστικά της.

2.2.3 Πως λειτουργεί η Dart

Η γλώσσα Dart είναι πολύ ευέλικτη λόγω του περιβάλλοντος στο οποίο είναι φτιαγμένη. Ο κώδικας μπορεί να αναπτυχθεί με πολλούς διαφορετικούς τρόπους μόλις υλοποιηθεί και γίνει η δοκιμή.

- **Ανεξαρτητοποίηση**

Όπως το λειτουργικό σύστημα της Java χρησιμοποιεί το Java Virtual Machine (JVM) για να εκτελεστεί, έτσι και η Dart χρησιμοποιεί το Dart Virtual Machine (DVM). Επομένως είναι αναπόφευκτο να χρειάζεται να γίνει η λήψη και η εγκατάσταση του για να μπορεί κάποιος να χρησιμοποιήσει τη Dart. Το Software Development Kit (SDK), πέρα από την μεταγλώττιση και τις βιβλιοθήκες, προσφέρει μια σειρά από άλλα εργαλεία όπως:

- **the pub package manager**
- **dart2js, το οποίο χρησιμοποιείται για να κάνει μεταγλώττιση σε JavaScript**
- **dartdoc, το Dart Documentation generator**
- **dartfmt , είναι ένας μορφοποιητής κώδικα ο οποίος ακολουθεί ένα βασικό σύνολο κανόνων**

Επομένως, κάποιος ο οποίος θέλει να τρέξει κώδικα Dart, χρειάζεται σίγουρα να κάνει εγκατάσταση το DVM. Για την υλοποίηση μιας Flutter εφαρμογής, είτε αυτή είναι για κινητό, για ίντερνετ ή για υπολογιστή, πρέπει να κάνει εγκατάσταση το Flutter, το οποίο περιέχει μέσα όλο το Dart SDK και άλλα Flutter tools.

- **AOT μεταφραστής**

Το Ahead Of Time (AOT) είναι εκτέλεση της μεταγλώττισης μιας υψηλού επιπέδου γλώσσας προγραμματισμού όπως είναι η Dart, σε εγγενή κώδικα μηχανής. Ουσιαστικά, όπως φαίνεται στο Σχήμα 1 όλα ξεκινάνε από ένα Dart αρχείο προέλευσης, το οποίο μπορεί να μεταφραστεί σε ένα δυαδικό αρχείο και να εκτελεστεί σε συγκεκριμένο λειτουργικό σύστημα. Το AOT κάνει το Flutter να είναι γρήγορο και φορητό σε σχέση με άλλες τεχνολογίες.



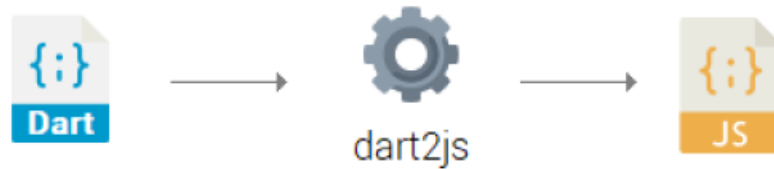
Σχήμα 1: AOT compilation

Με τη βοήθεια του AOT δεν χρειάζεται πλέον να κάνει κάποιος εγκατάσταση το DVM για το λόγο ότι πλέον έχουμε να κάνουμε με ένα δυαδικό αρχείο, το οποίο είτε θέλουμε να το τρέξουμε στο Android όπου έχουμε το .apk, .aab, είτε θέλουμε στο IOS όπου έχουμε το .ipa και τέλος .exe για τα Windows. Πλέον με βάση το Flutter SDK το οποίο περιέχει το AOT compile,

μπορούμε πολύ εύκολα να μεταγλωττίσουμε το Dart κώδικα σε εγγενές δυαδικό για κινητά, ίντερνετ και υπολογιστές. Στην έκδοση Flutter 1.21, το Dart SDK πλέον περιλαμβάνεται στο Flutter SDK, οπότε δεν χρειάζεται να κάνουμε ξεχωριστή εγκατάσταση. Από την έκδοση Flutter 2.6 η `dart2native` εντολή πραγματοποιεί AOT μεταγλώσσες για ένα πρόγραμμα Dart σε εγγενή κώδικα x64 και το αποτέλεσμα που μας δίνει είναι ένα εκτελέσιμο αρχείο.

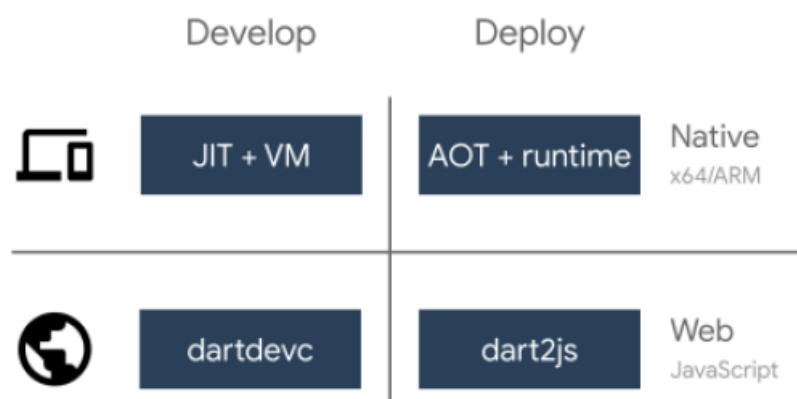
- **Ιστότοπος**

Χάρη στο `dart2js` εργαλείο, ο Dart κώδικας μπορεί πολύ εύκολα και γρήγορα να μεταφραστεί σε JavaScript κώδικα (Σχήμα 2). Το Flutter μπορεί να τρέχει στο Firefox, Chrome και το User Interface UI να είναι ίδιο με άλλες πλατφόρμες. Η `AngularDart` είναι ένα από τα frameworks τα οποία χρησιμοποιήθηκαν από την Google για να φτιαχτούν ορισμένες ιστοσελίδες όπως, «AdSense» και «AdWords».



Σχήμα 2: Dart2js Compilation

Μέχρι στιγμής έχουμε αναφερθεί στην ανάπτυξη (Deployment) του κώδικα Dart. Στη συνέχεια, θα μιλήσουμε αντίστοιχα για την ανάπτυξη (Development) του κώδικα (Σχήμα 3). Εδώ έχουμε να κάνουμε με δύο περιπτώσεις όσον αφορά τον υπολογιστή (desktop) / κινητό (mobile) και τον ιστότοπο (Web).



Σχήμα 3: Develop - Deploy Διαδικασία

- **Υπολογιστής (desktop) / κινητό (mobile) :**

Η τεχνική Just In Time (JIT) μπορεί να θεωρηθεί και ως μια πραγματικού χρόνου συναλλαγή επειδή η συλλογή γίνεται κατά τη διάρκεια της εκτέλεσης του προγράμματος. Ο συνδυασμός αυτών των δύο τεχνολογιών DVM + JIT επιτρέπει την αποστολή του κώδικα δυναμικά χωρίς να το ενδιαφέρει η αρχιτεκτονική του μηχανήματος του χρήστη. Με αυτό το τρόπο, έχουμε ένα πολύ καλό αποτέλεσμα στην εύρεση σφαλμάτων και ακόμα και στην ομαλή εκτέλεση του κώδικα, χωρίς να μας ενδιαφέρει η υποκείμενη αρχιτεκτονική.

- **Ιστότοπος**

Σε αυτή την περίπτωση έχουμε το «dartdevc» εργαλείο, το οποίο μας βοηθάει να τρέχουμε έναν ανιχνευτή λαθών σε εφαρμογές σε ιστοτόπους. Το «dartdevc» το χρησιμοποιούμε καθαρά για ανάπτυξη (Deploy) ενώ το «dart2js» το χρησιμοποιούμε για ανάπτυξη (Deployment). Επίσης έχουμε την δυνατότητα να χρησιμοποιήσουμε ειδικά εργαλεία όπως είναι το «webdev» το οποίο μας βοηθάει στην επεξεργασία Dart αρχείων και στην οπτικοποίηση των αλλαγών στο browser.

2.2.4 OOP Dart Χαρακτηριστικά

Όπως μερικές γλώσσες έτσι και η Dart έχει υποστήριξη σε πρότυπο OOP. Επομένως ακολουθεί πολλές από τις βασικές αρχές της με μια δική της ιδιαιτερότητα. Παρακάτω θα αναλύσουμε τις βασικές αρχές και δομές OOP δίνοντας κάποια αντίστοιχα παραδείγματα στη γλώσσα Dart.

- **Αντικείμενα και κλάσεις**

Μια από τις βασικές αρχές της OOP είναι τα αντικείμενα και οι διάφορες περιπτώσεις καθορισμένων κλάσεων. Συγκεκριμένα, όπως έχουμε προαναφέρει στη Dart όλα είναι ένα αντικείμενο, με άλλα λόγια κάθε αποθήκευση μιας τιμής σε μια μεταβλητή πάντα συνοδεύεται με την παρουσία μιας κλάσης. Στη Dart οι κλάσεις παίζουν πολύ σημαντικό ρόλο, επομένως μπορεί να αποτελούν το ίδιο για τα μέλη παρουσίας δηλαδή για τις μεθόδους και πεδία όσο για τα μέλη κλάσης δηλαδή στατικές μεθόδους και πεδία. Επίσης, μπορεί να χρησιμοποιήσει διαφορετικούς τρόπους για την δημιουργία μια κλάσης χωρίς να γίνεται υπερφόρτωση κατασκευαστή. Αυτό το επιτυγχάνει με την δημιουργία διαφορετικών συναρτήσεων από την γλώσσα και με την βοήθεια ορισμού κατασκευαστών.

- **Ενθυλάκωση**

Το θετικό που έχει η Dart είναι ότι δεν έχει περιορισμούς πρόσβασης σε σχέση με τις γνωστές λέξεις-κλειδιά protected, private και public. Επίσης, μια ακόμη διαφορά είναι ότι η ενθυλάκωση συμβαίνει σε επίπεδο βιβλιοθήκης και όχι σε επίπεδο τάξης όπως γίνεται σε άλλες γλώσσες που χρησιμοποιούν OOP πρότυπα. Ακόμη στην Dart οι διάφορες μέθοδοι getters και setters που δημιουργεί μπορούν να έχουν πρόσβαση από άλλες κλάσεις ώστε να μπορούν ανα πάσα στιγμή να ενημερώνονται άμεσα. Τέλος χρησιμοποιεί το σύμβολο (_) για να ελέγξει την ιδιωτικότητα για τη βιβλιοθήκη του, αυτό συνήθως το συναντάμε σε κλάσεις, μέλη τάξης, συναρτήσεις και μεταβλητές.

- **Κληρονομικότητα**

Με την κληρονομικότητα έχουμε την δυνατότητα της επέκτασης ενός αντικειμένου με σκοπό να χρησιμοποιήσουμε εξειδικευμένες εκδόσεις που μπορεί να μας προσφέρει η γλώσσα και κάποιου αφηρημένου τύπου. Στη Dart αυτή η διαδικασία γίνεται αυτόματα κατά την δήλωση μιας τάξης, δηλαδή γίνεται άμεσα η επέκταση του τύπου του αντικειμένου. Επομένως έχουμε άμεση κληρονομικότητα αλλά

με την λειτουργία mixins που διαθέτει μπορούμε να περιορίζουμε την άμεση κληρονομικότητα ώστε να έχουμε επαναχρησιμοποίηση του κώδικα. Τέλος, έχει μια δυνατότητα με την οποία μπορεί να επεκτείνει πάντα μια κλάση κάτι το οποίο δεν έχουν άλλες γλώσσες.

- **Abstraction**

Με την αφαίρεση μπορούμε να ορίσουμε έναν τύπο και τα βασικά χαρακτηριστικά του, με την βοήθεια μετάβασης σε προχωρημένους τύπους από τους γονείς μιας κλάσης. Το χαρακτηριστικό της Dart είναι ότι δεν ενδιαφέρεται για τον τρόπο εφαρμογής του, αλλά περιέχει τάξεις οι οποίες παρέχουν αυτό που μπορούν να κάνουν. Επίσης, καθιστά κάθε κατηγορία μια διεπαφή και αυτό το πραγματοποιεί με την “σιωπηρή διεπαφή” (silent interface) χωρίς όμως να επεκτείνεται από άλλους.

- **Πολυμορφισμός**

Με τον πολυμορφισμό καταφέρνουμε να έχουμε κληρονομιά. Επίσης, αυτό που αναδεικνύει τον πολυμορφισμό είναι ότι το χρησιμοποιούμε όταν θέλουμε να κάνουμε ένα αντικείμενο να συμπεριφέρεται σαν ένα άλλο με τις ίδιες λειτουργίες. Η Dart το υποστηρίζει αυτό, αλλά σε κάποια κομμάτια διαφέρει σε κάποιες τεχνικές όπως τις γνωρίζουμε από άλλες γλώσσες. [15][7][6]

2.3 Γιατί το Flutter χρησιμοποιεί το Dart

Υπάρχουν πολλοί λόγοι για τους οποίους η Google πήρε την απόφαση να επιλέξει τη γλώσσα προγραμματισμού Dart και παρακάτω θα αναφερθούμε επιγραμματικά σε κάποιους από αυτούς. Ένας από τους πιο διαδεδομένους, είναι η αντικειμενοστρέφεια την οποία περιέχει. Οι περισσότεροι προγραμματιστές στις μέρες μας ξέρουν να γράφουν γλώσσες οι οποίες είναι αντικειμενοστρεφείς, επομένως η γλώσσα Dart θα ήταν εύκολη και γρήγορη στο να την μάθουν. Άρα ο προγραμματιστής δεν χρειάζεται να ασχοληθεί με έναν εντελώς καινούργιο κώδικα, καθώς με τις γνώσεις που έχει είναι σε θέση να διαχειριστεί, όλες τις ιδιαιτερότητες της Dart. Επίσης, ένας πολύ σημαντικός παράγοντας για την παραγωγή και υλοποίηση μιας εφαρμογής είναι η υψηλή επίδοση που μπορεί να σου δώσει η γλώσσα προγραμματισμού με την οποία εργάζεσαι κατά την διάρκεια της εκτέλεσης. Η Dart μπορεί να το εγγυηθεί αυτό, για το λόγο ότι έχει ένα πολύ ισχυρό κατανεμητή μνήμης ο οποίος μπορεί να χειριστεί μικρές κατανομές. Αυτό έχει σαν αποτέλεσμα να αποτελεί την καλύτερη λύση για το λειτουργικό Flutter. Η παραγωγικότητα είναι ένα, πολύ σημαντικό χαρακτηριστικό που έχει κάνει το Dart να είναι επιθυμητό από το Flutter. Το Flutter έχει την δυνατότητα να παράγει περιεχόμενο για Android, Web και Desktop με μια βάση κώδικα, η οποία διατηρεί τις ίδιες παραστάσεις και όψεις για την κάθε πλατφόρμα. Επομένως, μια παραγωγική γλώσσα όπως η Dart επιταχύνει τη διαδικασία κωδικοποίησης και κάνει το framework πιο ελκυστικό. Ένας ακόμη λόγος είναι ότι και οι δύο τεχνολογίες έχουν δημιουργηθεί από την Google και υπάρχει μεγαλύτερη ελευθερία όσον αφορά τον τρόπο διαχείρισης και εξέλιξης, αντίστοιχα και στις δύο.

Επιπλέον, υπάρχουν ακόμη αρκετοί λόγοι που θα μπορούσε κανείς να επιλέξει τη γλώσσα Dart σαν βασική γλώσσα. Πιο συγκεκριμένα, έχουμε να κάνουμε με μια γλώσσα η οποία έχει αυστηρό τυπολόγιο. Επίσης, οι χρόνοι εκτέλεσης είναι πολύ γρήγοροι σε σχέση με άλλες γλώσσες προγραμματισμού και έχει γρήγορο και εύκολο ανιχνευτή λαθών. Επίσης, περιέχει μια πληθώρα εργαλείων όπως έχουμε αναφέρει και προηγουμένως:

- **Tree-shaking optimization**
- **Hot reload feature**
- **DartPad, ένα playground για να μπορέσει κάποιος να εξασκηθεί στη γλώσσα**
- **DevTools, μια συλλογή από ανιχνευτές λαθών**
- **Code documentation generator tool**

- **Υποστηρίζει το JIT και AOT**

2.4 Τι είναι το flutter

Το Flutter είναι ένα σύνολο εργαλείων ανάπτυξης εφαρμογών για κινητά και δημιουργήθηκε από την Google. Επίσης έχουμε να κάνουμε με ένα ανοιχτό κώδικα ο οποίος είναι προσβάσιμος από οποιοδήποτε προγραμματιστή θέλει να ασχοληθεί με το Flutter. Επομένως μιλάμε για ένα εργαλείο με το οποίο μπορεί κανείς να φτιάξει εύκολα και γρήγορα υψηλού επιπέδου εφαρμογές κινητών. Ένα ρητό που έχει το Flutter είναι «write once, and deploy everywhere», έτσι με αυτό το τρόπο μπορεί να δημιουργήσει περιεχόμενο για Android, iOS και ChromeOS. Μελλοντικός στόχος του Flutter είναι να τρέχει σε web εφαρμογές και desktop εφαρμογές σε αρκετά λειτουργικά συστήματα.

Αναλυτικότερα, έχουμε να κάνουμε με ένα ολοκληρωμένο εργαλείο παραγωγής εφαρμογών. Είναι μια πλατφόρμα η οποία σου παρέχει οτιδήποτε θελήσεις για να φτιάξεις μια εφαρμογή όπως: ένας rendering engine, εξαρτήματα διεπαφής χρήστη, testing frameworks, διάφορα εργαλεία και άλλα πολλά χαρακτηριστικά. Επομένως το μοναδικό πράγμα με το οποίο θα ασχοληθεί ο προγραμματιστής είναι καθαρά η λειτουργικότητα της εφαρμογής.

2.4.1 Πως ξεκίνησε το Flutter

Παρόλο που είναι μια καινούργια τεχνολογία, το Flutter τα τελευταία χρόνια έχει γνωρίσει μεγάλη εξέλιξη. Την πρώτη φορά που συναντήσαμε αυτή την τεχνολογία είχε την ονομασία Sky, πρωτοεμφανίστηκε στο Dart Developer Summit 2015 και παρουσιάστηκε από τον Eric Seidel. Αυτό που ήθελε η Google ήταν να δημιουργήσει μια τεχνολογία υψηλού επιπέδου και καλύτερης εμπειρίας χρήσης. Το 2016 παρουσιάστηκε στο κοινό ως Flutter και η πρώτη alpha έκδοση ήταν το Μάιο του 2017 η οποία είχε ήδη κατασκευαστεί για συστήματα Android και iOS. Στα τέλη του 2018 παρουσιάστηκε η σταθερή έκδοση της τεχνολογίας, μετά από μεγάλο ενδιαφέρον της κοινότητας.

2.4.2 Τι είναι αυτό που ήθελε να λύσει το Flutter

Παράλληλα με το Flutter υπάρχουν πολλά άλλα frameworks τα οποία ουσιαστικά κάνουν αυτό που ξεκίνησε να κάνει το Flutter. Όπως έχουμε αναφέρει, προοριζόταν για να προσφέρει μια καλύτερη εμπειρία χρήσης στο χρήστη και ένα υψηλό επίπεδο εκτέλεσης και λειτουργίας. Όμως δεν ήταν μόνο αυτά τα οποία είχε υποσχεθεί. Το κύριο μέλημα ήταν να δημιουργήσουν μια καινοτόμα πλατφόρμα για πολλαπλά συστήματα κινητών συσκευών. Παρακάτω θα αναφερθούμε σε μερικά χαρακτηριστικά που προσπάθησε το Flutter να βελτιώσει και να ενσωματώσει στον εξοπλισμό του.

2.4.2.1 Οι ακριβοί κύκλοι ανάπτυξης

Στο παρελθόν, για να μπορέσει μια ομάδα ή εταιρεία να αντιμετωπίσει την ζήτηση της αγοράς όσον αναφορά την παραγωγή εφαρμογών για κινητές συσκευές, έπρεπε να ξοδέψει αρκετά χρήματα και πολλές εργατοώρες ώστε να καλύψει όλα τα συστήματα. Σε αυτό το κομμάτι επεμβαίνει το Flutter όπου προσφέρει μια ενιαία πλατφόρμα για την παραγωγή πολλαπλών συσκευών και αντίστοιχα λειτουργικών συστημάτων.

2.4.2.2 Αρκετές γλώσσες προγραμματισμού για αντίστοιχα λειτουργικά

Ένα προγραμματιστής ο οποίος θέλει να ασχοληθεί με την ανάπτυξη εφαρμογών για κινητές συσκευές θα πρέπει να γνωρίζει μια πληθώρα γλωσσών προγραμματισμού, ώστε να μπορέσει να καλύψει όλα τα

λειτουργικά συστήματα Switch για iOS, Java/Kotlin για Android. Το Flutter έχει την καινοτομία με την εκμάθηση μιας γλώσσας την Dart να μπορείς να αναπτύξεις εφαρμογές για πολλαπλά συστήματα κινητών συσκευών.

2.4.2.3 Μεγάλος χρόνος μεταγλώττισης

Ένας από τους σημαντικότερους παράγοντες που επηρεάζουν την παραγωγικότητα ενός προγραμματιστή είναι οι μεγάλοι χρόνοι κατασκευής και μεταγλώττισης της εφαρμογής που υλοποιούν. Η υλοποίηση σε λειτουργικό Android είναι ένα γνωστό παράδειγμα όπου οι κύκλοι παραγωγής είναι πολύ μεγάλοι και αυτό έχει σαν αποτελέσματα την ταλαιπωρία των προγραμματιστών που την χρησιμοποιούν. Το Flutter προσπάθησε να μειώσει τους χρόνους μεταγλώττισης ώστε να αυξήσει την συνεχόμενη παραγωγή υλικού και να βοηθήσει τους προγραμματιστές να απαλλαγούν από τις αναμονές που θα μπορούσαν να συναντήσουν. [5][1]

2.4.3 Άλλα αντίστοιχα frameworks

Παράλληλα με την έκδοση του Flutter, υπήρχαν και άλλα αντίστοιχα frameworks τα οποία υλοποίησαν αυτό που ήθελε να κάνει το Flutter (Σχήμα 4). Μερικά από τα πιο διαδεδομένα frameworks είναι τα παρακάτω.

- **Xamarin**
- **React Native**
- **Ionic**
- **Native Script**



Σχήμα 4: Τα τέσσερα πιο διαδεδομένα framework

2.4.3.1 Xamarin

Το Xamarin δημοσιεύτηκε πρώτη φορά το 2011 από την Microsoft. Αναλυτικότερα, έχουμε να κάνουμε με ένα εργαλείο ανάπτυξης κινητών συσκευών το οποίο είναι μια πλατφόρμα ανοιχτού κώδικα. Με το συγκεκριμένο framework μπορούμε να υλοποιήσουμε εφαρμογές για iOS , Android και για Windows με .NET. Οι γλώσσες οι οποίες χρησιμοποιεί είναι HTML, CSS , JavaScript. Το Xamarin έχει βγει ως το δέκατο πιο δημοφιλές framework για ανάπτυξη λογισμικού για πλατφόρμες κινητών συσκευών.

2.4.3.2 React Native

Το React Native δημοσιεύτηκε πρώτη φορά το 2015 είναι ένα open source framework και αυτό. Το React Native ψηφίστηκε ως το όγδοο πιο αγαπημένο framework για cross-platform. Έχουμε να κάνουμε με μια πολύ γρήγορα αναπτυσσόμενη πλατφόρμα. Τα χρησιμοποιούν κυρίως προγραμματιστές που θέλουν να κάνουν γρήγορες και εύκολες εφαρμογές. Οι γλώσσες που χρησιμοποιεί είναι το framework React και JavaScript. Γνωστές εταιρείες που εργάζονται με αυτό το framework είναι η Facebook, Instagram, Tesla και Uber

2.4.3.3 Ionic

Το Ionic ιδρύθηκε το 2012 και θεωρείται από τα πιο δημοφιλή frameworks ανάπτυξης εφαρμογών cross-platform. Πιο συγκεκριμένα, είναι open-source framework από το MIT, το οποίο χρησιμοποιεί σύγχρονες τεχνολογίες όπως HTML5, CSS3 και JavaScript. Ο σκοπός που εξυπηρετεί είναι η δημιουργία υβριδικών και διαδραστικών εφαρμογών για κινητά. Τα διάφορα συστήματα τα οποία μπορεί κάποιος να υλοποιήσει με το συγκεκριμένο framework είναι εφαρμογές για κινητές συσκευές, ιστοτόπους εφαρμογών και υπολογιστικές εφαρμογές.

2.4.3.4 Native Script

Το Native Script είναι και αυτό ένα πολύ γνωστό framework με το οποίο μπορεί κανείς να φτιάξει εφαρμογές για πλατφόρμες κινητών συσκευών αλλά και άλλα. Με την βοήθεια γνωστών τεχνολογιών Angular, Vue.js, TypeScript και JavaScript μπορεί κανείς να φτιάξει πολύ ωραίες εφαρμογές διεπαφής χρήστη. Με το συγκεκριμένο framework φτιάχνονται πολύ εύκολα εφαρμογές σε μικρό χρονικό διάστημα. [14]

2.4.4 Διαφορές σε σχέση με τα άλλα frameworks

Επομένως, λόγω της μεγάλης ποικιλίας σε cross-platform frameworks είναι πολύ δύσκολο μια καινούργια τεχνολογία όπως είναι το Flutter να ξεχωρίσει, έχει όμως αρκετές διαφορές που το κάνουν να ξεχωρίζει από τα άλλα frameworks. Παρακάτω θα αναφερθούμε σε ορισμένα από τα σημεία στα οποία διαφέρει:

- **Υψηλή απόδοση**
- **Καλύτερη διαχείριση στη διεπαφή χρήστη**
- **Περιέχει την Dart γλώσσα**
- **Είναι υλοποιημένο από την Google**
- **Είναι open source framework**

2.4.4.1 Υψηλή απόδοση

Η Google έχει δηλώσει ότι η απόδοση του Flutter σε σχέση με τα άλλα frameworks, θα είναι σε καλύτερα επίπεδα. Αυτό όμως είναι πολύ νωρίς για να το επιβεβαιώσει κανείς, μιας και είναι πολύ πρόσφατη τεχνολογία. Το Flutter χρησιμοποιεί μια μηχανή γραφικών που λέγεται Skia η ιδιαιτερότητα της οποίας είναι ότι αποδίδει την εφαρμογή pixel by pixel. Σε σχέση με τα άλλα frameworks, δεν χρειάζεται να καλεί την Original Equipment Manufacturer (OEM) widgets και αυτό έχει σαν αποτέλεσμα να αποφεύγεται οποιαδήποτε συμφόρηση στο σύστημα η οποία θα μπορούσε να καθυστερούσε τον κύκλο παραγωγής της εφαρμογής. Ακόμη, με τον Dart AOT μεταγλωττιστή που χρησιμοποιεί το Flutter είναι σαν να παράγουμε native code. Αυτό δίνει μια πιο ομαλή ροή στην παραγωγή και στην απόδοση του κατά την διαδικασία υλοποίησης μιας εφαρμογής, αλλά και κατά την παραγωγή της. Επομένως, κάνει το Flutter να είναι πιο γρήγορο και αποτελεσματικό.

2.4.4.2 Καλύτερη διαχείριση στη διεπαφή χρήστη

Όσον αφορά στη διεπαφή χρήστη το Flutter δεν περιορίζεται από κανόνες και συμβάσεις. Αυτό το καταφέρνει προβάλλοντας το οπτικό στοιχείο κατευθείαν στον καμβά. Τα περισσότερα frameworks αναπαράγουν την εικόνα που προσφέρει η πλατφόρμα αλλά με διαφορετικό τρόπο. Πολλά είναι τα frameworks που περιορίζονται από τα γραφικά στοιχεία OEM και αυτό έχει σαν αποτέλεσμα να

αναπαράγουν μόνο τις δυνατότητες που τους δίνει η πλατφόρμα και να μην επεμβαίνουν με δικές τους παρεμβάσεις. Επίσης, frameworks που βασίζονται σε τεχνολογίες ιστού, έχουν μεγαλύτερη γκάμα αλλαγών και δεν έχουν τόσους περιορισμούς, αλλά περιορίζονται ανάλογα με την κινητή μηχανή ιστού που θα χρησιμοποιούν, για το λόγο ότι κάθε μηχανή έχει τις δικές της ιδιαιτερότητες. Το Flutter επομένως έχει την δυνατότητα να αφήνει τον προγραμματιστή να επέμβει στη διεπαφή χρήστη. Όλα αυτά γίνονται με βάση ένα πολύ πλούσιο API widgets το οποίο είναι επεκτάσιμο και δεν έχει μικρή απόδοση. Περιέχει ένα μεγάλο αριθμό widget σχεδιασμού πλατφόρμας, το υλικό που έχει οριστεί για το Android και το Cupertino για το λειτουργικό iOS. Επίσης χρησιμοποιεί σημασιολογικούς κανόνες οι οποίοι αποδίδουν πιο αποτελεσματικές και όμορφες διατάξεις σε σχέση με άλλα frameworks που χρησιμοποιούν αρκετά πιο περίπλοκους κανόνες διάταξης CSS.

2.4.4.3 Περιέχει τη γλώσσα Dart

Όπως έχουμε προαναφέρει, ένας από τους πιο σημαντικούς στόχους του Flutter ήταν να φτιάξει μια πιο εξελιγμένη πλατφόρμα σε σχέση με τις ήδη υπάρχουσες. Αυτό το έχει πετύχει μαζί με την γλώσσα Dart. Αναλυτικότερα, με την γλώσσα Dart έχουμε μια πιο ευέλικτη διαδικασία εκτέλεσης κώδικα. Το Flutter χρησιμοποιεί το Dart AOT το οποίο χρησιμοποιείται κατά την έκδοση μιας εφαρμογής και παράλληλα με το JIT σαν δεύτερο μεταγλωττιστή, όλα αυτά μας επιφέρουν ένα ταχύτερο κύκλο εργασίας και πολλές αλλαγές στον κώδικα. Λόγω του AOT μεταγλωττιστή, έχουμε μια γρήγορη μεταφορά από non-native σε native και αυτό μας δίνει την δυνατότητα να έχουμε πιο ικανοποιητικές ταχύτητες. Ακόμη, το γεγονός ότι η Dart είναι μια πολύ γρήγορη γλώσσα εκμάθησης, βοηθάει το Flutter παραπάνω. Παρόλα αυτά είναι μια πολύ μοντέρνα και προχωρημένη γλώσσα η οποία χρησιμοποιεί αντικειμενοστρέφεια με πολλές γνωστές λειτουργίες σε δυναμικές και στατικές γλώσσες. Επίσης έχει μια πολύ ενεργή κοινότητα. Τέλος η σύνταξη του διαφέρει από άλλα frameworks και είναι πιο ευέλικτη και αυτό έχει σαν αποτέλεσμα ο προγραμματιστής να μπορεί, να διαμορφώσει όπως αυτός θέλει το κώδικα του, με πολλαπλά εργαλεία για εντοπισμό σφαλμάτων.

2.4.4.4 Είναι υλοποιημένο από την Google

Το Flutter αν και είναι καινούργιο στη αγορά, μέσα σε μικρό χρονικό διάστημα έχει γίνει πολύ αγαπητό από το κόσμο. Αναλυτικότερα, λόγω του ότι έχει υλοποιηθεί από την Google, έχει μια πολύ μεγάλη κοινότητα να το υποστηρίζει, παράλληλα με διάφορες εκδηλώσεις που πραγματοποιούνται ανά τακτά χρονικά διαστήματα. Όπως έχουμε προαναφέρει, η πρώτη beta έκδοση ανακοινώθηκε στο Google IO 2018 και ύστερα την ίδια χρονία, στο Flutter Live Event, βγήκε η πρώτη σταθερή έκδοση του. Πλέον έχουμε να κάνουμε με πάνω από 200 εκατομμύρια χρήστες σε Flutter εφαρμογές, 3 χιλιάδες εφαρμογές Flutter οι οποίες υπάρχουν στο Play Store και πάνω από 250 χιλιάδες προγραμματιστές που ασχολούνται με τη συγκεκριμένη τεχνολογία.

2.4.4.5 Είναι open source framework

Το γεγονός ότι είναι ένα open source framework βοηθάει πάρα πολύ στην εξέλιξη του. Πιο αναλυτικά, με τη βοήθεια της κοινότητας η Google μπορεί πιο εύκολα να βελτιώσει τυχόν προβλήματα τα οποία μπορεί να εντοπιστούν. Επίσης, με τη συνεργασία της κοινότητας και της Google έχουμε μεγαλύτερη εξέλιξη όσον αφορά, όπως είπαμε και προηγουμένως, τα διάφορα προβλήματα τα οποία μπορεί να εμφανιστούν, υπάρχει μεγαλύτερη προσοχή στο documentation και γενικά στη τεχνολογία και τέλος το σημαντικότερο είναι ότι υπάρχει μια αλυσίδα ανατροφοδότησης απόψεων είτε αρνητικών είτε θετικών, το οποίο βοηθάει αρκετά στην ανάπτυξη του.

2.4.5 Εισαγωγή στα Widgets

Ένας προγραμματιστής ο οποίος θέλει να ασχοληθεί με την τεχνολογία Flutter πρέπει να γνωρίζει πολύ καλά πως λειτουργούν τα Widgets. Αναλυτικότερα, όλη η δομή του κώδικα του Flutter αποτελείται από τα widgets. Μπορούν να θεωρηθούν ως η αναπαράσταση διαφόρων τμημάτων της εφαρμογής και όχι μόνο. Μια εφαρμογή αποτελείται από πολλαπλά widgets, όπου το καθένα αποτελεί ένα τμήμα της. Αυτά, έχουν την δυνατότητα να είναι επεκτάσιμα και εύκολα προσαρμόσιμα και όλα αυτά δίνουν την δυνατότητα στον προγραμματιστή να τα εκμεταλλευτεί όπως αυτός θέλει. Κύριος στόχος των widgets είναι να μπορούν να υλοποιηθούν με το λιγότερο δυνατό κώδικα και να είναι επεκτάσιμα και σωστά δομημένα. Επίσης ένα ακόμη σημαντικό χαρακτηριστικό είναι η συνδεσιμότητα που έχουν μεταξύ τους και οι πολλαπλοί συνδυασμοί ο οποίοι μπορούν να επιτευχθούν.

2.4.5.1 Που συναντάμε widget μέσα σε μια εφαρμογή

Όπως έχουμε αναφέρει προηγουμένως, τα πάντα μέσα σε μια εφαρμογή Flutter είναι Widget. Μπορεί να είναι εικονικά στοιχεία τα οποία είναι εμφανή στο χρήστη και άλλα τα οποία να μην είναι. Αναλυτικότερα, ένα από τα σημαντικότερα κομμάτια σε μια εφαρμογή είναι τα στοιχεία κειμένου ή τα κουμπιά, αυτά επομένως είναι το καθένα ένα widget. Επίσης, η διάταξη δηλαδή εάν τα στοιχεία θα είναι δομημένα σε στήλη ή σε σειρά, αποτελούν ένα widget, το οποίο όπως έχουμε αναφέρει δεν είναι αντιληπτό από το χρήστη. Επιπλέον, η σχεδίαση και η διαμόρφωση κειμένου ή οποιουδήποτε άλλου στοιχείου μέσα στην εφαρμογή. Ακόμη, η ενέργεια που μπορεί να κάνει ο χρήστης πατώντας ένα κουμπί μέσα στην εφαρμογή και γενικά οτιδήποτε έρχεται σε αλληλεπίδραση με το χρήστη αποτελεί ένα widget. Τέλος όλα τα widget μέσα σε μια εφαρμογή αποτελούν τη δομή της εφαρμογής και για να μπορέσουν όλα αυτά να οργανωθούν σωστά χρειάζεται να ομαδοποιηθούν σε ένα widget tree.

2.4.5.1.1 Ορισμένα από τα Widget που συναντάμε

Όπως έχουμε αναφέρει πιο πάνω, υπάρχουν πολλές κατηγορίες στις οποίες μπορούν να χωριστούν τα Widgets, στο σύνολο 160, αλλά για λόγους ευχρηστίας τα χωρίζουμε σε τρεις βασικές κατηγορίες: τα widget τιμής, γραφικά στοιχεία διάταξης και γραφικά στοιχεία πλοήγησης. Παρακάτω θα αναφερθούμε στην κάθε κατηγορία ξεχωριστά.

- **Value widgets**

Στη συγκεκριμένη περίπτωση έχουμε να κάνουμε με την διατήρηση μιας τιμής. Αναλυτικότερα, είναι διάφορα εργαλεία τα οποία τα χρησιμοποιούμε για να δείχνουμε τιμές είτε αυτές προέρχονται από κάποια στάνταρ πλαίσια υπηρεσιών που υπάρχουν στο διαδίκτυο, είτε από τον ίδιο τον χρήστη όπου του έχουμε βάλει ένα συγκεκριμένο εργαλείο μέσα από μια συγκεκριμένη γκάμα εργαλείων, ώστε να κάνουμε λήψη μιας τιμής. Δυο από τα βασικά value widget τα οποία μπορεί κανείς να συναντήσει είναι το Text, μέσω του οποίου είτε μπορείς να εμφανίσεις ένα κείμενο είτε να κάνει λήψη από τον χρήστη μια τιμή κειμένου, οπότε αυτό χρειάζεται στην εφαρμογή. Μερικά από τα υπόλοιπα widget τα οποία μπορεί να συναντήσει κανείς είναι τα παρακάτω: Checkbox, CircularProgressIndicator, Date & Time, Pickers, DataTable, DropdownButton, FlatButton, FloatingActionButton, FlutterLogo, Form, FormField, Icon, IconButton, Image, LinearProgressIndicator, PopupMenuButton, Radio, RaisedButton RawImage, RefreshIndicator, RichText, Slider, Switch, Text, TextField και Tooltip.

- **Layout widgets**

Στα layouts widgets έχουμε τα εργαλεία με τα οποία διαμορφώνουμε το περιβάλλον σε μια εφαρμογή. Πιο αναλυτικά, με τα συγκεκριμένα έχουμε όλο τον έλεγχο για το που θα τοποθετηθούν όλα τα γραφικά πάνω στο περιβάλλον. Ο προγραμματιστής έχει την ευελιξία να φτιάξει όπως έχει φανταστεί το γραφικό

της εφαρμογής χρησιμοποιώντας όλα αυτά τα εργαλεία τα οποία τον βοηθούν στο να στοιχηθεί σωστά το περιβάλλον κάθετα ή οριζόντια. Επιπλέον, μπορεί να δημιουργήσει μια πληθώρα σχεδίων για την διεπαφή χρήστη. Ορισμένα από αυτά τα εργαλεία είναι τα παρακάτω: Align, AppBar, AspectRatio, Baseline, BottomSheet, ButtonBar, Card, Center, Column, ConstrainedBox, Container, CustomMultiChildLayout, Divider, Expanded, ExpansionPanel, FittedBox, Flow, FractionallySizedBox, GridView, IndexedStack, IntrinsicHeight, IntrinsicWidth, LayoutBuilder, LimitedBox, ListBody, ListTile, ListView, MediaQuery, NestedScrollView, OverflowBox, Padding, PageView, Placeholder, Row, Scaffold, Scrollable, Scrollbar, SingleChildScrollView, SizedBox, SizedOverflowBox, SliverAppBar, SnackBar, Stack, Table και Wrap.

- **Navigation widgets**

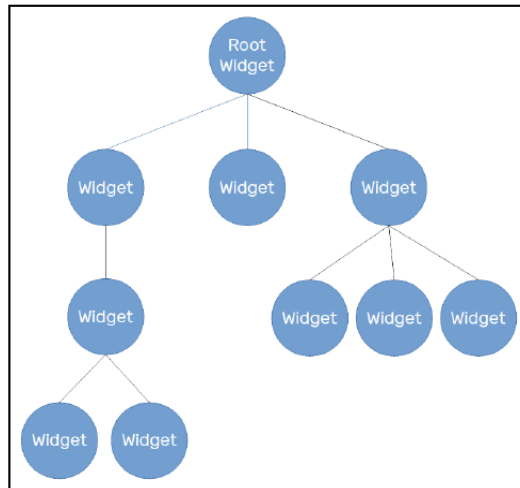
Σε αυτή την περίπτωση τα συγκεκριμένα εργαλεία μας βοηθάνε στη πλοήγηση του χρήστη στην εφαρμογή. Αναλυτικότερα, όταν μια εφαρμογή αποτελείται από αρκετές σελίδες, τότε για την ευκολία του χρήστη, ο προγραμματιστής τοποθετεί ένα μενού μέσου του οποίου γίνεται πιο εύκολη η πρόσβαση σε αυτές τις σελίδες. Τα εργαλεία αυτά αποτελούνται συνήθως από κουμπιά ή μια γραμμή καρτέλων ή ένα είδος «συρταριού» το οποίο ολισθαίνει από την αριστερή προς την δεξιά πλευρά. Ορισμένα από αυτά τα εργαλεία είναι τα παρακάτω: AlertDialog, BottomNavigationBar, Drawer, MaterialApp, Navigator, SimpleDialog, TabBar και TabBarView.

- **Other widgets**

Σε αυτή την κατηγορία έχουμε όλα τα υπόλοιπα widget τα οποία δεν μπορούν να ενταχθούν σε κάποιες από τις παραπάνω κατηγορίες αλλά έχουν την χρησιμότητα τους. Ορισμένα από αυτά είναι τα παρακάτω: GestureDetector, Dismissible, Cupertino, Theme, Transitions και Transforms.

2.4.5.2 Τι είναι το widget tree

Είναι μια πολύ σημαντική έννοια για τις διατάξεις στο Flutter. Στο widget tree έχουμε όλη την διάταξη και την αναπαράσταση των γραφικών στοιχείων και μη, της διεπαφής χρήστη μιας εφαρμογής. Τα γραφικά στοιχεία (widgets) παρουσιάζονται στο δέντρο σαν κόμβοι. Μια αλλαγή που μπορεί να γίνει σε ένα widget επηρεάζει όλο το δέντρο και τους άλλους κόμβους. Δεν υπάρχει σταθερή δομή σε ένα widget tree, πάντα καθορίζεται από το κάθε κόμβο που μπαίνει ή αφαιρείται αντίστοιχα. Επομένως παίζει κύριο ρόλο σε μια εφαρμογή Flutter, γιατί βοηθάει στη χαρτογράφηση στην οθόνη σε ένα δέντρο widget. Στην περίπτωση που θέλουμε να ενημερώσουμε ένα widget, καθορίζει το τρόπο της αναδημιουργίας (Σχήμα 5)



Σχήμα 5: Ένα διάγραμμα Widget Tre

2.4.5.3 Κύκλος ζωής ενός widget

Στο προγραμματισμό έχουμε αρκετούς διαφορετικούς κύκλους ζωής τους οποίους πρέπει να μάθουμε ώστε να μπορέσουμε να χρησιμοποιήσουμε μια γλώσσα όσο το δυνατόν πιο αποδοτικά και σωστά. Οι περισσότεροι κύκλοι ζωής είναι γραμμικού τύπου, ακολουθούν δηλαδή μια σειρά ενεργειών. Σε αυτή την ενότητα θα δούμε τους κύκλους ζωής ενός widget. Για να μπορέσει να χτίσει κάποιος τη διεπαφή χρήστη σε μια εφαρμογή Flutter, χρησιμοποιούνται δύο τύποι widgets, το Stateless Widget και το StatefulWidget. Επιγραμματικά το stateless widget χρησιμοποιείται όταν οι τιμές κατά την διάρκεια κλήσης του δεν αλλάζουν και το stateful widget για το αντίθετο. Ακόμη κάθε widget stateless ή stateful έχει μια build μέθοδο με έναBuildContext, το οποίο καθορίζει την τοποθεσία του κάθε widget μέσα σε ένα widget tree, οπότε ουσιαστικά μιλάμε για αντικείμενα element.

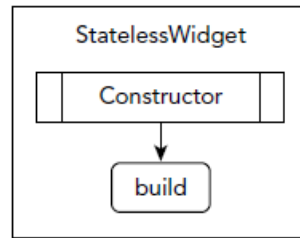
2.4.5.3.1 Stateless Widget

Η δομή ενός Stateless Widget είναι σταθερή. Επομένως δεν έχουμε δυναμική αλλαγή μέσα ένα τέτοιο widget. Ένα παράδειγμα που θα μπορούσαμε να δώσουμε είναι ένα τίτλος μια κατηγορίας σε μια λίστα, έχουμε να κάνουμε με σταθερή δομή και σταθερό περιεχόμενο. Το κάλεσμα ενός stateless widget μπορεί να γίνει με 3 διαφορετικούς τρόπους. Μπορεί να καλεστεί την πρώτη φορά που δημιουργείται, όταν αλλάζει ο γονέας του widget και όταν έχει αλλάξει ένα InheritedWidget. Η βασική δομή που συναντάμε όταν δημιουργούμε για πρώτη φορά ένα τέτοιο widget είναι η παρακάτω (Σχήμα 6).

```

class JournalList extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Container();
  }
}

```



Σχήμα 6: Stateless Widget Lifecycle

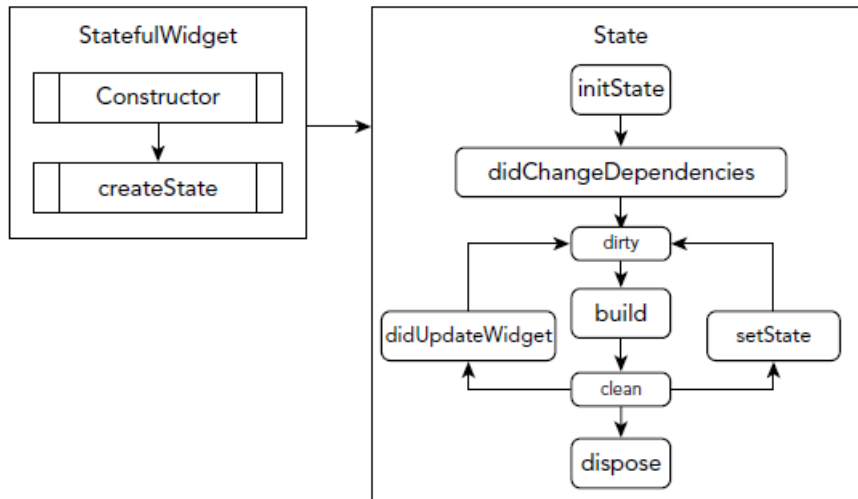
2.4.5.3.2 Stateful Widget

Η δομή ενός Stateful widget αλλάζει δυναμικά κάθε φορά που μπορεί κάποιος να το καλέσει ή να καλεστεί αυτό από μόνο του. Ένα παράδειγμα που θα μπορούσε κάποιος να δώσει είναι ένα ρολόι που μπορεί να εμφανιστεί στην οθόνη το οποίο, αλλάζει συνεχώς η μορφή του, καθώς αλλάζει η ώρα. Επομένως έχουμε να κάνουμε με ένα μεταβλητό widget. Η δήλωση του γίνεται με βάση δυο κατηγορίες, πρώτη είναι η κατηγορία Stateful widget class και η δεύτερη η κατάσταση state class. Στην περίπτωση που αλλάξει ένα τέτοιο widget στην πρώτη κατηγορία έχουμε ολική αλλαγή στο widget, ενώ στη δεύτερη περίπτωση παραμένει σταθερό στο state και αλλάζει κάθε φορά που θα το καλέσουμε. Το κάλεσμα ενός state γίνεται με τη μέθοδο setState(). Η βασική δομή που συναντάμε όταν δημιουργούμε για πρώτη φορά ένα τέτοιο widget είναι η παρακάτω (Σχήμα 7).

```

class JournalEdit extends StatefulWidget {
  @override
  _JournalEditState createState() => _JournalEditState();
}
class _JournalEditState extends State<JournalEdit> {
  @override
  Widget build(BuildContext context) {
    return Container();
  }
}

```



Σχήμα 7: Stateful Widget Lifecycle

Παρακάτω θα δούμε ένα πίνακα με τις διάφορες μεθόδους που συναντάμε σε ένα κύκλο ζωής ενός stateful widget. Πολλές από αυτές τις μεθόδους δεν είναι ανάγκη να χρησιμοποιούνται σε κάθε δημιουργία ενός τέτοιου widget.

Μέθοδοι	Περιγραφή	Κώδικας
initState()	Καλείται μια φορά όταν το αντικείμενο εισάγεται στο δέντρο.	<pre>@override void initState() { super.initState(); print('initState'); }</pre>
dispose()	Καλείται όταν το αντικείμενο αφαιρείται μόνιμα από το δέντρο.	<pre>@override void dispose() { print('dispose'); super.dispose(); }</pre>
didChangeDependencies()	Καλείται όταν το state αντικείμενο αλλάζει.	<pre>@override void didChangeDependencies() { super.didChangeDependencies(); print('didChangeDependencies'); }</pre>
didUpdateWidget(Contacts oldWidget)	Καλείται όταν αλλάζει η διαμόρφωση του widget.	<pre>@override void didUpdateWidget(Contacts oldWidget) { super.didUpdateWidget(oldWidget); print('didUpdateWidget: \$oldWidget'); }</pre>
deactivate()	Καλείται όταν το αντικείμενο αφαιρεθεί από το δέντρο.	<pre>@override void deactivate() { print('deactivate'); super.deactivate(); }</pre>
build(BuildContext context)	Καλείται πολλές φορές για τη δημιουργία της διεπαφής χρήστη και το BuildContext χειρίζεται τη θέση αυτού του γραφικού στοιχείου στο δέντρο widget.	<pre>@override Widget build(BuildContext context) { print('build'); return Container(); }</pre>
setState()	Χρησιμοποιείται όταν θέλουμε να μεταβάλουμε την τιμή.	<pre>setState() { name = _newValue; });</pre>

Πίνακας 1: Μέθοδοι του Stateful Widget

2.4.6 Ποιος μπορεί να χρησιμοποιήσει το Flutter

Κάποιος που θέλει να χρησιμοποιήσει μια τεχνολογία με την οποία θα φτιάξει γρήγορα και εύκολα εφαρμογές σίγουρα θα προτιμήσει το Flutter και το Dart για το επόμενο του πρότζεκτ.

2.4.6.1 Ομάδες και εταιρείες

Το Flutter έχει αποδείξει ότι σε ομάδες και σε γενικά σε εταιρείες έχει ξεκινήσει να αυξάνει την παραγωγικότητα και την συνεργασία. Στην περίπτωση που μια εταιρεία θέλει να προσθέσει ένα καινούργιο χαρακτηριστικό σε ένα προϊόν το οποίο υλοποιείται από διαφορετικές ομάδες, μια για την κάθε πλατφόρμα, είναι αναμενόμενο να υπάρχει δυσκολία στην επικοινωνία λόγω των διαφορετικών δεξιοτήτων που έχει η κάθε ομάδα. Αυτό το πρόβλημα έλυσε το Flutter. Επομένως οι ομάδες που ήταν χωρισμένες, πλέον έχουν ενωθεί σε μια και αυτό έχει σαν αποτέλεσμα την καλύτερη επικοινωνία και παραγωγικότητα για την εφαρμογή. Έτσι έχουμε ένα πιο ολοκληρωμένο αποτέλεσμα και μια πιο άμεση ανταπόκριση όσον αναφορά στην ενημέρωση ή στο να προστεθεί ένα καινούργιο χαρακτηριστικό.

2.4.6.2 Μεμονωμένους προγραμματιστές

Πολλοί προγραμματιστές έχουν ξεκινήσει να επιλέγουν το Flutter σαν βασική τους ασχολία. Ένας από τους πολλούς λόγους είναι ότι έχουμε να κάνουμε με μια γρήγορη γλώσσα για να την μάθει κάποιος. Επίσης είναι ένα πολύ δυνατό εργαλείο με το οποίο μπορεί κανείς να φτιάξει υψηλού επιπέδου εφαρμογές. Ακόμη για προγραμματιστές οι οποίοι ασχολούνται μόνο με ανοιχτού κώδικα τεχνολογίες το Flutter είναι σίγουρα στις επιλογές τους. Τέλος όλο και περισσότερες είναι οι freelancer δουλειές οι οποίες με τον καιρό αρχίζουν και αυξάνονται.

2.4.7 Εγκατάσταση του Flutter

Η εγκατάσταση του Flutter είναι πολύ απλή και θα την αναλύσουμε παρακάτω. Σε λειτουργικό σύστημα Windows αρχικά πρέπει να κατεβάσουμε το Flutter zip, και μετά να δημιουργήσουμε ένα φάκελο src στο root του δίσκου C:/ πχ src, και να κάνουμε extract εκεί μέσα το zip κατεβάσαμε. Επίσης θα χρειαστεί να κατεβάσουμε και το Node.js και να το κάνουμε εγκατάσταση. Τέλος καλό είναι να κατεβάσουμε και το Git για Windows και να το κάνουμε και αυτό εγκατάσταση. Σε λειτουργικό σύστημα Mac κατεβάζουμε το αρχείο. Κάνουμε extract το αρχείο που κατεβάσαμε και το τοποθετούμε στο σύστημα χρησιμοποιώντας αυτό το path `$ export PATH=`pwd`/flutter/bin:$PATH`. Στη συνέχεια τρέχουμε το `$flutter doctor` για να εξετάσουμε ότι έγινε καλά η διαδικασία της εγκατάστασης αντίστοιχα και στο λειτουργικό windows. Αντίστοιχα και σε αυτό το λειτουργικό κάνουμε εγκατάσταση τα παραπάνω links για το Node.js και Git για Mac.

Για να δουλέψουμε με το Flutter μπορούμε να χρησιμοποιήσουμε είτε το Android Studio/IntelliJ είτε το Visual Studio (VS) code για Windows ή Mac αντίστοιχα. Αυτά τα δυο IDEs είναι τα καλύτερα για να προγραμματίσει κάποιος κινητές συσκευές. Στη συνέχεια θα πρέπει να κάνουμε εγκατάσταση τα αντίστοιχα plugins ανάλογα με το IDE που χρησιμοποιούμε.[8][4][11]

2.5 Επίλογος

Η σημαντικότητα αυτού του κεφαλαίου ήταν να κατανοήσουμε τι είναι η γλώσσα προγραμματισμού που χρησιμοποιεί το Flutter. Ακόμα να δούμε γιατί έχει επιλέξει το Flutter τη γλώσσα αυτή σαν βασικό της εργαλείο και τέλος να εμβαθύνουμε στην έννοια του Flutter. Στο επόμενο κεφάλαιο θα αναφερθούμε στη βάση Firebase και θα αναλύσουμε τις υπηρεσίες τις οποίες διαθέτει.

Κεφάλαιο 3ο: Firebase

3.1 Εισαγωγή

Στο παρακάτω κεφάλαιο θα αναφερθούμε στη βάση δεδομένων (πλατφόρμα) που χρησιμοποιείται στην εφαρμογή που είναι το Firebase. Αναλυτικότερα, θα εξηγήσουμε τη δομή της συγκεκριμένης πλατφόρμας, η οποία αποτελείται από τρεις κατηγορίες και τις οποίες θα αναλύσουμε διεξοδικά. Τέλος, θα εμβαθύνουμε σε δυο από τις πολλές υπηρεσίες που παρέχει το Firebase, οι οποίες έχουν χρησιμοποιηθεί στην εφαρμογή.

3.2 Κατηγορίες προϊόντων

Το Firebase είναι μια πλατφόρμα που αναπτύχθηκε από την Google για να χειρίζεται ολόκληρη την υποδομή διακομιστή βάσης για να συνδέει εφαρμογές όπως iOS, Android και ιστοτόπους. Ουσιαστικά είναι μια πλατφόρμα που περιέχει 18 προϊόντα χωρισμένα σε τρεις κατηγορίες: Ανάπτυξη ή αλλιώς χτίσιμο (Build), Ποιότητα (Quality) και Ανάπτυξη (Grow).

3.2.1 Ανάπτυξη (Develop)

- **Cloud Firestore**

Το Cloud Firestore είναι η αποθήκευση των δεδομένων μεταξύ χρηστών και συσκευών. Επίσης φιλοξενεί μια βάση δεδομένων η οποία είναι NoSQL. Ακόμη, παρέχει γρήγορο και ταυτόχρονο συγχρονισμό και υποστήριξη εκτός σύνδεσης αλλά και ερωτήματα τα οποία μπορούν να γίνουν προς την βάση.

- **Realtime Database**

Η Real-time database είναι η πρώτη βάση της Firebase. Έχουμε να κάνουμε με μια αποτελεσματική λύση για εμφάνιση δεδομένων σε πραγματικό χρόνο και γρήγορα. Ουσιαστικά είναι μια βάση NoSQL στην οποία μπορούμε να αποθηκεύουμε και να συγχρονίζουμε δεδομένα μεταξύ των χρηστών σε πραγματικό χρόνο. Όλο αυτό μας βοηθάει να έχουμε γρήγορη πρόσβαση σε πραγματικό χρόνο, από οποιαδήποτε συσκευή και ιστότοπο.

- **Cloud Storage**

Το Cloud storage το χρησιμοποιεί κανείς για την αποθήκευση περιεχομένου όπως φωτογραφιών, βίντεο και ήχου. Αυτή την λειτουργία μπορεί να την χρησιμοποιήσει κάποιος που θέλει να αποθηκεύσει περιεχόμενο το οποίο μετά μπορεί να το αναπαραστήσει ζωντανά σε μια εφαρμογή. Έχουμε να κάνουμε με ένα ασφαλές και παράλληλα γρήγορο ως προς την ταχύτητα φόρτωσης αποθηκευτικό χώρο, αλλά και αρκετά οικονομικό.

- **Authentication**

Το authentication έχει κάνει με την διαχείριση χρηστών. Αναλυτικότερα, έχουμε να κάνουμε με ένα πολύ απλό και ασφαλή τρόπο ώστε να μπορέσουμε να διαχειριστούμε την ταυτότητα χρηστών στην εφαρμογή μας. Παρέχει μια πληθώρα επιλογής μεθόδων για απλού τύπου πιστοποιήσεις και πιστοποιήσεις τρίτων παρόχων όπως Google, Facebook και άλλα. Επομένως, πολύ εύκολα μπορεί κανείς να εκμεταλλευτεί τις παροχές που προσφέρει το συγκεκριμένο προϊόν και να φτιάξει ένα αποτελεσματικό σύστημα εισόδου και εγγραφής στην εφαρμογή του.

- **Hosting**

Με το συγκεκριμένο προϊόν μπορούμε να φιλοξενήσουμε τον ιστότοπο μας. Παρέχει μια πληθώρα εργαλείων σχεδιασμένα για σύγχρονες εφαρμογές του σήμερα. Επίσης, προσφέρει μια αυτοματοποιημένη εμπειρία όσον αναφορά την προώθηση στο παγκόσμιο Content Delivery

Network (CDN) και δίνει δωρεάν Secure Sockets Layer (SSL) πιστοποιητικά ώστε να παρέχει στο χρήστη ασφάλεια και αξιοπιστία.

3.2.2 Ποιότητα (Quality)

- **Crashlytics**

Το Crashlytics βοηθάει στο να παρακολουθείται η ποιότητα της εφαρμογής, ώστε να δίνεται προτεραιότητα και να διορθώνονται ζητήματα σταθερότητας που διαβρώνουν την ίδια την εφαρμογή. Με το συγκεκριμένο εργαλείο έχουμε ομαδοποίηση και μια ολοκληρωμένη διαχείριση των σφαλμάτων. Με αυτή την διαδικασία έχουμε άμεση πληροφόρηση των σφαλμάτων και γενικές λεπτομέρειες για το τι προκάλεσε το σφάλμα.

- **Performance Monitoring**

Με το Performance Monitoring έχουμε άμεση πληροφόρηση της απόδοσης της εφαρμογής, με βάση τις απόψεις των χρηστών. Επίσης, παρέχει εύκολη αντιμετώπιση των σφαλμάτων, με την χρήση της προσαρμοσμένης ανίχνευσης. Ακόμη, μπορεί κάποιος να κατανοήσει τις καθυστερήσεις του δικτύου που μπορεί να δημιουργηθούν και επηρεάζουν τους χρήστες και να τις εξαλείψει με την αυτόματη παρακολούθηση αιτημάτων HTTP/S.

- **Test Lab**

Το Test Lab δίνει την δυνατότητα να προσομοιώσεις σε πραγματικά περιβάλλοντα χρήσης την εφαρμογή σου, παρέχοντας φυσικές συσκευές. Ακόμη, μέσω της υπηρεσίας Robo, ανιχνεύει αυτόματα τυχόν σφάλματα που μπορούν να προκύψουν πριν εντοπιστούν από τους χρήστες. Όλα αυτά αποθηκεύονται για μια καλύτερη και πιο αναλυτική αντιμετώπιση και διαχείριση των σφαλμάτων.

- **App Distribution**

Το App Distribution παρέχει μια ολική προβολή του προγράμματος δοκιμών beta σε εφαρμογές Android, παρέχοντας πολύτιμα σχόλια πριν την παραγωγή μιας νέας έκδοσης. Παρέχεται ένας κεντρικός πίνακας ελέγχου όπου μέσα από αυτόν μπορεί κανείς να διαχειριστεί πιο εύκολα της πληροφορίες σχετικά με τις εκδόσεις πριν από την κυκλοφορία της εφαρμογής του. Παρέχετε βοήθεια μέσω σχολίων που αναφέρουν τυχόν σφάλματα τα οποία μπορεί να προκύψουν στην εφαρμογή.

3.2.3 Ανάπτυξη (Grow)

- **In-App Messaging**

Το In-App Messaging είναι ακόμα σε beta στάδιο αλλά έχουμε να κάνουμε με ένα εργαλείο το οποίο βοηθάει στο να εμφανίζει μηνύματα επιβεβαίωσης στο χρήστη, όπως για παράδειγμα στο να εγγραφούν στο σύστημα ή να αγοράσουν περιεχόμενο κλπ.

- **Google Analytics**

Ένα από τα βασικά εργαλεία του Firebase είναι το Google Analytics. Αναλυτικότερα, παρέχει δωρεάν απεριόριστες αναφορές σε ξεχωριστά συμβάντα σε μια εφαρμογή. Δίνει την δυνατότητα να μπορείς να επεξεργαστείς ποια δεδομένα θέλεις να εμφανίζεις, τα οποία έχουν ιδιαίτερη σημασία για την εφαρμογή. Εμφανίζει μια πληθώρα δεδομένων όπως, ειδοποιήσεις σφαλμάτων, ποσοστά εισόδου των χρηστών, κινήσεις όσον αναφορά τις πωλήσεις στην εφαρμογή.

- **Predictions**

Το Predictions βοηθάει στο να ενημερώνει τον δημιουργό της εφαρμογής, που να εστιάσει και πως να διαχειριστεί γενικά την εφαρμογή του. Το εργαλείο αυτό δουλεύει με ένα σύστημα

μηχανικής μάθησης, το οποίο βοηθάει να αναγνωρίζει ποια είναι η καλύτερη περίπτωση για την αντίστοιχη εφαρμογή.

- **A/B Testing**

Το A/B Testing βρίσκεται σε beta μορφή και χρησιμοποιείται και αυτό σαν τρόπος βελτισποίησης της εφαρμογής σας. Αναλυτικότερα, διευκολύνει στην εκτέλεση, ανάλυση και κλιμάκωση πειραμάτων και μάρκετινγκ. Ακόμη, ενημερώνει εάν μια πιθανή αλλαγή σε εφαρμογή θα επιφέρει βελτιώσεις και παραπάνω ζήτηση σε κάποια σημεία της.

- **Cloud Messaging**

Το Cloud Messaging έχει να κάνει με την παροχή αξιόπιστης και αποδοτικής σύνδεσης μεταξύ του διακομιστή και των συσκευών. Αυτό έχει σαν αποτέλεσμα να επιτρέπει να λαμβάνονται μηνύματα και ειδοποιήσεις σε iOS , Android και στον ιστότοπο.

- **Remote Config**

Με το Remote Config μπορούμε να επεξεργαστούμε μέσω της κονσόλας Firebase κάποιες αλλαγές στην εφαρμογή μας και παράλληλα να έχουμε άμεση ενημέρωση μέσω του Google Analytics, χωρίς την έγκριση από το App Store.

- **Dynamic Links**

Το Dynamic Links έχει να κάνει με συνδέσμους οι οποίοι μπορούν να χρησιμοποιηθούν μέσα στην εφαρμογή για iOS ή Android.[19]

3.3 Cloud Firestore

Για να καταλάβει κάνεις το Cloud Firestore, θα μπορούσε να το συγκρίνει με το Structured Query Language (SQL) Server Database. Επομένως στη συγκεκριμένη βάση έχουμε να κάνουμε με collections τα οποία περιέχουν έγγραφα (documents). Το έγγραφο είναι ένα ζεύγος κλειδιού-τιμής, το οποίο μπορεί και να υποδείξει υπο-συλλογές. Επίσης, αποθηκεύουν τιμές σαν το JSON και υποστηρίζουν τύπους δεδομένων τα οποία δεν υποστηρίζονται από μόνα τους. Κάθε έγγραφο περιέχει ένα όνομα το οποίο είναι ένα μοναδικό ID που δημιουργείται αυτόμα με εισαγωγή ενός εγγράφου και σε χωρητικότητα πιάνει το πολύ ένα 1MB. Όπως έχουμε αναφέρει, ένα έγγραφο περιέχει πολλούς τύπους δεδομένων όπως: array, Boolean, byte, date and time, floating-point, number, geographical point, integer, map, reference, text string και null.

Μια σύγκριση που θα μπορούσε κάνεις να κάνει μεταξύ του Cloud Firestore και SQL Server Database:

SQL Server Database	Cloud Firestore
Πίνακας (Table)	Συλλογή (Collection)
Γραμμή (Row)	Έγγραφο (Document)
Στήλη (Column)	Δεδομένα

Πίνακας 2: Αντιστοιχία SQL Server με Firestore

Ένα απλό δεδομένο το οποίο συναντάμε στη βάση Cloud Firestore είναι το εξής:

Τύπου	Τιμή
Συλλογή (Collection)	Store
Έγγραφο (Document)	R5NcTWAaWtHTtYtPoOd

Περιεχόμενα εγγράφου	<pre>date: "2019-0202T13:41:12.537285" name: "Coffee Store" tables: 20 uid: "F1GGeKiwp3jRpoCVskdBNmO4GUN4"</pre>
----------------------	--

Πίνακας 3: Παράδειγμα καταχώρησης στη βάση Firestore Cloud

3.3.1 Κανόνες ασφαλείας Cloud Firestore

Είναι πολύ σημαντικό να ασφαλίζει κανείς τα δεδομένα του στη βάση. Το Firestore μέσω των κανόνων ασφαλείας παρέχει αυτή την λειτουργία, η οποία είναι πολύ χρήσιμη. Επομένως, πλέον μπορεί κανείς να ασφαλίσει όλες τις συλλογές, τα έγγραφα και να τα εμφανίζει μόνο στους χρήστες οι οποίοι είναι συνδεδεμένοι ή εγγεγραμμένοι στο σύστημα. Ένα παράδειγμα κανόνα ασφαλείας που μπορεί να δώσει κάποιος είναι το παρακάτω:

```
service cloud.firestore {
  match /databases/{database}/documents {
    match /journals/{document=**} {
      allow read, write: if resource.data.uid == request.auth.uid;
      allow create: if request.auth.uid != null;
    }
  }
}
```

Επομένως μια εξήγηση που θα μπορούσε να δώσεις κάποιος είναι ότι το `match/databases/{database}/documents` σημαίνει ότι δηλώνουμε να ταιριάζουν οι κανόνες με οποιαδήποτε βάση δεδομένων του Cloud Firestore στην εφαρμογή. Στη συνέχεια το `match /store/{document=**}` δηλώνουμε το όνομα της συλλογής (collection) που είναι το `store` και με το `document=**` να διαβάζει όλα τα έγγραφα στη συγκεκριμένη συλλογή (collection). Επομένως αν ισχύουν όλα τα παραπάνω επιτρέπουμε το χρήστη να μπορεί να διαβάσει και να γράφει στα έγγραφα (documents) αν και μόνο αν έχει `uid` και είναι συνδεδεμένος αλλιώς επιστρέφει `null`.

3.4 Firebase Authentication

Το Firebase Authentication παρέχει μια πλήρη αποτελεσματική λύση για την ασφάλεια όσον αναφορά κατά την είσοδο και εγγραφή του χρήστη στην εφαρμογή. Πλέον, είναι απαραίτητο να υπάρχει μια ασφαλής και ιδιωτική συναλλαγή μέσα στις εφαρμογές σε όλες τις πλατφόρμες, και για αυτό το λόγο παρέχει μια backend services λύση. Διάφοροι πάροχοι που υπάρχουν στο Firebase Authentication σύστημα είναι οι εξής: [9]

- **Email/Password**
- **Phone**
- **Google**
- **Play Games (Google)**

- **Game Center (Apple)**
- **Facebook**
- **Twitter**
- **GitHub**
- **Anonymous**

3.4.1 Email/Password

Η συγκεκριμένη υπηρεσία προσφέρει ένα ολοκληρωμένο σύστημα ταυτοποίησης με το email και password. Επομένως, δημιουργεί κάποιος ένα εξωλοκλήρου σύστημα εισόδου-εξόδου με μια απλή ταυτοποίηση που είναι αυτή του email, αλλά και μέσω της Firebase παρέχετε αποθήκευση όλων αυτών των λογαριασμών που συνδέονται με αυτό το τρόπο. Επίσης, προσφέρει και την λειτουργία επαναφοράς κωδικού στην περίπτωση μη υπενθύμισης του.

3.4.2 Google – Facebook

Όπως έχουμε αναφέρει και πιο πάνω το Firebase παρέχει επίσης υπηρεσία με την οποία μπορείς να συνδεθείς μέσω διαφορετικών παρόχων, όπως είναι το Google και το Facebook. Επομένως παρέχει έτοιμες μεθόδους με τις οποίες μπορεί κανείς να έχει πρόσβαση εισόδου-εξόδου χρήστη μέσω των διαφορετικών παρόχων.

3.5 Επίλογος

Σε αυτό το κεφάλαιο αναλύσαμε τι είναι η βάση Firebase και εμβαθύναμε στις υπηρεσίες τις οποίες διαθέτει. Ακόμη, μιλήσαμε πιο διεξοδικά για δύο από τις υπηρεσίες τις οποίες διαθέτει η Firebase τις οποίες έχουμε χρησιμοποιήσει και στην εφαρμογή, συγκεκριμένα τις Authentication και Firestore. Στο επόμενο κεφάλαιο θα μιλήσουμε για την τεχνολογία QR Code και θα δούμε όλες τις πληροφορίες που χρειάζεται κανείς για να κατανοήσει αυτούς τους κώδικες.

Κεφάλαιο 4ο: QR Code

4.1 Εισαγωγή

Στο παρακάτω κεφάλαιο θα αναφερθούμε στο τι είναι το QR Code. Συγκεκριμένα, θα εξηγήσουμε πως ξεκίνησε και πως έγινε τόσο διαδεδομένο. Επίσης, θα κάνουμε μια σύγκριση μεταξύ του QR Code και του Barcode για να δούμε τις διαφορές αλλά και για ποιό λόγο έγινε περισσότερο διαδεδομένο το QR Code από την άλλη τεχνολογία. Ακόμη, θα εξηγήσουμε διαφόρους τύπους αυτής της τεχνολογίας που συναντάμε, αλλά και τις ιδιαιτερότητές τους. Τέλος θα αναφερθούμε στη δομή ενός QR Code, εξηγώντας αναλυτικά το κάθε ένα χαρακτηριστικό του.

4.2 Λίγα λόγια για το QR Code

Στις μέρες μας, το Quick Response (QR) Code είναι πολύ γνωστό για το λόγο ότι το συναντάμε στην καθημερινότητα μας αρκετές φορές. Έχουμε να κάνουμε με ένα κωδικό γρήγορης αναγνώρισης, εξού και το αρχικό της λέξης γρήγορη (Quick) στο όνομα. Η αναγνώριση μπορεί να γίνει από αρκετές συσκευές οι οποίες χρησιμοποιούν έναν συνδυασμό διαστήματος, το λεγόμενο Matrix Barcode (ένα 2-D barcode). Επομένως κατά την μετάδοση του QR Code, λαμβάνουμε ένα μεγάλο φάσμα πληροφοριών. Κάποιες από τις βασικές χρήσεις που μπορεί κανείς να το συναντήσει είναι σε εργοστάσια, πάνω στα μηχανήματα, επομένως στο βιομηχανικό τομέα, σε καταστήματα σούπερ μάρκετ πάνω στα προϊόντα, άρα στο τομέα του λιανικού εμπορίου και ένας από τους κύριους κλάδους που χρησιμοποιείται είναι ο τομέας του μάρκετινγκ, διαφήμισης και γενικά ότι έχει να κάνει με το logistics.

4.3 Κωδικοί QR Code έναντι γραμμικών κωδικών (Barcode)

Αυτές οι δυο τεχνολογίες σε ένα μεγάλο βαθμό είναι πανομοιότυπες ως προς την λογική και τη λειτουργία. Διαφέρουν όμως σε ένα πολύ μεγάλο κριτήριο, το οποίο ανέδειξε τους κωδικούς QR Code παραπάνω από ότι τους Barcode (Σχήμα 8). Ο λόγος είναι ότι οι κωδικοί QR έχουν την δυνατότητα να συγκρατούν περισσότερα δεδομένα. Αυτό το πετυχαίνουν λόγω της συγκράτησης πληροφοριών τόσο οριζόντια όσο και κάθετα σε σχέση με τα Barcodes τα οποία μπορούν να συγκρατούν πληροφορίες μόνο κάθετα. Μια από τις κύριες χρήσεις των γραμμικών κωδικών (Barcode) είναι στο τομέα του λιανικού εμπορίου, όπου η ζήτηση για αποθήκευση δεδομένων είναι πιο απλή και γι αυτό και έχουν επιλεχτεί.



Σχήμα 8: QR Code και Barcode σχήματα

4.4 Ιστορική αναδρομή

Οι δυο τεχνολογίες έχουν ξεκινήσει από την Ιαπωνία. Στη δεκαετία του 1960, ο τομέας του λιανικού εμπορίου γνώρισε μια μεγάλη άνθιση στην Ιαπωνία. Αυτό είχε σαν αποτέλεσμα, λόγω της μεγάλης αύξησης των αγορών αλλά και των προϊόντων που αποθηκεύονταν στα καταστήματα, να αρχίζει να δυσκολεύει η κατάσταση παρακολούθησης τους. Για το λόγο αυτό, δημιουργήθηκε ένα σύστημα POS με το οποίο ξεκίνησε η καταγραφή των προϊόντων από έναν υπολογιστή. Όμως ακόμα και αυτή η εξέλιξη δεν ήταν αρκετή λόγω της μικρής δυνατότητας αποθήκευσης χαρακτήρων, για την ακρίβεια μόλις 20 αλφαριθμητικών.

Εδώ έρχεται το Denso Wave με τον προγραμματιστή Masahiro Hara στον οποίο δίνονται τα εύσημα για την εφεύρεση του QR Code. Η κύρια λύση που ανέδειξαν, ήταν η ανάπτυξη ενός μέσου δυο κατευθύνσεων κωδικοποίησης (2-D) , ώστε να κάνουν τους κώδικες πιο ευπροσάρμοστους και να περιέχουν περισσότερες πληροφορίες. Επομένως ασχολήθηκαν αρχικά με το σχήμα το οποίο θα δώσουν και κατέληξαν στο τετράγωνο για το λόγο ότι είναι ένα πολύ συνηθισμένο και εύκολο για ανάγνωση. Στη συνέχεια, είδαν ότι μπορούσαν να τοποθετήσουν περισσότερες πληροφορίες είτε κάθετα είτε οριζόντια. Το σημαντικότερο πλεονέκτημα ήταν η ανάγνωση του συγκεκριμένου σχήματος η οποία ήταν 10 φορές ταχύτερη από τους γραμμικούς κώδικες (barcode). Η επίσημη δημοσιοποίηση έγινε το 1994 και αυτή ήταν η αρχή μιας μεγάλης εξέλιξης. Το 2000 προστέθηκαν οι κωδικοί QR, στα διεθνή πρότυπα ISO και αυτό ήταν το έναυσμα να ξεκινήσει να χρησιμοποιείται σε ολόκληρο το κόσμο. Τέλος, με την αναβάθμιση της τεχνολογίας των κινητών τα QR Codes είδαν μεγάλη εξέλιξη. [16][2]

4.5 Διάφοροι τύποι κωδικών QR

Η δημιουργία και η εξέλιξη των QR Codes έδωσε το έναυσμα να δημιουργηθούν πολλές πανομοιότυπες τεχνολογίες σαν του κωδικά QR. Παρακάτω θα αναφερθούμε σε τρεις:

- **Κωδικοί Micro QR**

Σε αυτούς του κωδικούς το χαρακτηριστικό είναι το μικρό μέγεθος. Λόγω της μεγάλης χρησιμότητας των κωδικών QR, το μέγεθος του σχήματος έπαιξε πολύ μεγάλο ρόλο. Αυτό συνέβει γιατί πολλές εταιρείες ήθελαν να το τοποθετήσουν στα προϊόντα τους και το μέγεθος τους δεν το καθιστούσε εφικτό. Έτσι δημιουργήθηκε το Micro QR το οποίο έδωσε τη λύση σε αυτό το πρόβλημα λόγω του μικρού μεγέθους του. Ωστόσο, έχουν τον περιορισμό ότι μπορούν να καταχωρήσουν μέσα το QR Code μικρό αριθμό πληροφοριών.

- **Κωδικοί iQR**

Εδώ έχουμε να κάνουμε με μια εξέλιξη των κωδικών micro QR. Η διαφορά της συγκεκριμένης τεχνολογίας είναι το ορθογώνιο σχήμα. Το σημαντικό είναι όμως ότι διατηρεί τη μεγάλη ποσότητα δεδομένων που μπορεί να αποθηκεύσει κάνεις μέσα στο QR Code.

- **FrameQR**

Το FrameQR είναι και αυτό μια παραλλαγή του κλασικού κώδικα QR και αναπτύχθηκε το 2014. Η συγκεκριμένη τεχνολογία έχει την ιδιαιτερότητα να μπορεί κανείς να φτιάξει διάφορες παραλλαγές όσον αφορά στο χρώμα, στο σχήμα, στον τύπο, στο logo και σε αρκετά ακόμα χαρακτηριστικά.

4.6 Η δομή ενός QR Code

Όπως έχουμε προαναφέρει, αυτό που κάνει τον κώδικα QR να είναι τόσο προσιτός είναι η γρήγορη απόκριση και ανάγνωση που έχει και γενικά η αποθήκευση πολλών χαρακτήρων. Αναλυτικότερα, έχει

τη δυνατότητα να αποθηκεύσει έως 7089 ψηφία ή 4296 χαρακτήρες, συμπεριλαμβανομένων σημείων στίξης και ειδικών χαρακτήρων. Επίσης μπορεί να κωδικοποιεί εξίσου λέξεις και φράσεις όπως διευθύνσεις διαδικτύου. Η δομή ενός QR Code μπορεί να φαίνεται ότι είναι απλά γραμμές, αλλά όλα αυτά στην πραγματικότητα είναι δείκτες που εξυπηρετούν μια μεγαλύτερη λειτουργία στις δυνατότητες ανταλλαγής πληροφοριών του συγκεκριμένου κώδικα. Επομένως η δομή ενός κώδικα QR αποτελείται από επτά βασικά χαρακτηριστικά:

- **Δείκτες εντοπισμού θέσης**

Όπως βλέπουμε και στη φωτογραφία (Σχήμα 9) το συγκεκριμένο χαρακτηριστικό βρίσκεται στις τρεις γωνίες του κώδικα. Με αυτό το τρόπο ο σαρωτής διαβάζει με ακρίβεια τον κώδικα. Επιπλέον, αυξάνεται η ταχύτητα ανάγνωσης και βοηθάει τον σαρωτή να εντοπίσει το σχήμα και να καταλάβει ότι πρόκειται για κωδικό.



Σχήμα 9: Δείκτες εντοπισμού θέσης

- **Σήματα ευθυγράμμισης**

Τα σήματα ευθυγράμμισης βοηθούν τον κώδικα QR, και σχεδιάζονται σε μια καμπύλη επιφάνεια. Τα μοτίβα ευθυγράμμισης αυξάνονται όλο και περισσότερο με βάση το πόσες πληροφορίες αποθηκεύουμε μέσα στο κώδικα QR (Σχήμα 10).



Σχήμα 10: Σήματα ευθυγράμμισης

- **Σχέδιο χρονισμού**

Το σχέδιο χρονισμού επίσης βοηθάει και αυτό στην διαμόρφωση του πλέγματος δεδομένων. Με την βοήθεια αυτού του χαρακτηριστικού ο σαρωτής καθορίζει πόσο μεγάλη είναι η βάση των δεδομένων ενός κώδικα QR (Σχήμα 11).



Σχήμα 11: Σχέδιο χρονισμού

- **Πληροφορίες έκδοσης**

Γενικά υπάρχουν 40 διαφορετικές εκδόσεις κώδικα QR και οι συγκεκριμένοι δείκτες καθορίζουν αυτήν που χρησιμοποιούμε. Οι πιο διαδεδομένες και γνωστές είναι οι εκδόσεις 1 έως 7 (Σχήμα 12).



Σχήμα 12: Πληροφορίες έκδοσης

- **Μορφοποίηση πληροφοριών**

Το μοτίβο μορφής περιεχομένου διευκολύνει τον σαρωτή να αναγνωρίσει τον κώδικα QR. Επίσης, περιέχει πληροφορίες σε περίπτωση σφάλματος αλλά και το μοτίβο μάσκας δεδομένων (Σχήμα 13).



Σχήμα 13: Μορφοποίηση πληροφοριών

- **Πλήκτρα διόρθωσης δεδομένων και σφαλμάτων**

Στο συγκεκριμένο μηχανισμό περιέχονται όλα τα δεδομένα στο κώδικα QR. Επίσης, περιέχει το χώρο με τα μπλοκ διόρθωσης σφαλμάτων (Σχήμα 14).



Σχήμα 14: Πλήκτρα διόρθωσης δεδομένων και σφαλμάτων

- **Quiet zone**

Το quiet zone παίζει πολύ σημαντικό ρόλο σε ένα κώδικα QR. Προσφέρει δομή και βελτιώνει την κατανόηση του κώδικα (Σχήμα 15). [17]



Σχήμα 15: Quiet zone

4.7 Στατικοί και δυναμικοί κώδικες QR

Οι κώδικες QR πλέον είναι αρκετά ευέλικτοι και μπορεί κανείς να τοποθετήσει ότι πληροφορία θέλει σε αυτούς. Για αυτό το λόγο, ανάλογα την πληροφορία που βάζει κανείς μέσα σε ένα τέτοιο κώδικα, αλλάζει η μορφή του και γίνεται είτε στατικός είτε δυναμικός. Παρακάτω θα εξηγήσουμε επιγραμματικά τις δυο αυτές κατηγορίες.

4.7.1 Στατικοί κώδικες

Στους στατικούς κώδικες QR, συναντάμε πληροφορίες οι οποίες είναι στατικές και δεν μπορούν να επεξεργαστούν μετά την δημιουργία τους. Συνήθως βάζουμε πληροφορίες που ξέρουμε ότι θα παραμείνουν στατικές, οπότε είναι για προσωπική χρήση επαγγελματιών, αναγνωριστικά υπαλλήλων και πολλά άλλα. Αλλά για το τομέα του μάρκετινγκ και γενικά για περιπτώσεις που χρειάζεται να γίνουν κάποιες μετρήσεις με τους συγκεκριμένους κώδικες δεν είναι η καλύτερη λύση. Επομένως, στους στατικούς κώδικες μπορεί να τοποθετηθεί απλό κείμενο, μια ηλεκτρονική διεύθυνση, ένα url, επαγγελματικές κάρτες και πολλά ακόμα.

4.7.2 Δυναμικοί κώδικες

Στην αντίθετη πλευρά, οι δυναμικοί κώδικες QR έχουν τη δυνατότητα να μεταβάλουν το περιεχόμενο τους και να αλλάξουν τον τύπο του κώδικα QR όσες φορές χρειαστεί. Επομένως είναι η καλύτερη περίπτωση για επαγγελματικούς και εμπορικούς σκοπούς. Το περιεχόμενο που συναντάμε κυρίως μέσα σε ένα δυναμικό κώδικα είναι app εφαρμογής, διάφορα pdfs, κουπόνια και διάφορα μέσα κοινωνικής δικτύωσης.[18]

4.8 Επίλογος

Σε αυτό το κεφάλαιο μιλήσαμε για τους κώδικες QR Code και το πως έχουν βοηθήσει σε πολλές πτυχές της ζωής μας. Εμβαθύναμε με μια ιστορική αναδρομή και αναλύσαμε τη δομή ενός κώδικα QR. Σε αυτό το σημείο ολοκληρώσαμε την θεωρητική ανάλυση των τεχνολογιών που χρησιμοποιήθηκαν. Στο επόμενο κεφάλαιο θα μιλήσουμε για την υλοποίηση της εφαρμογής.

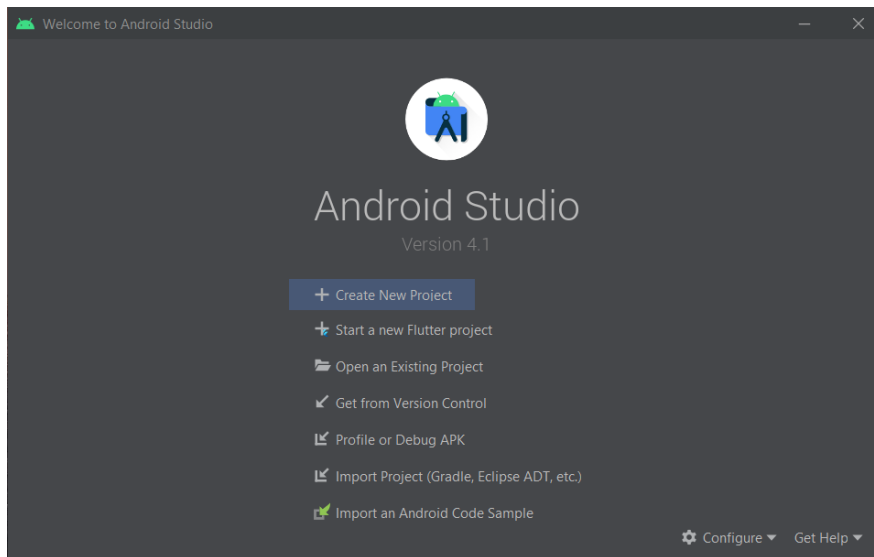
Κεφάλαιο 5ο: Υλοποίηση εφαρμογής

5.1 Εισαγωγή

Στο παρακάτω κεφάλαιο θα αναφερθούμε στα τεχνικά κομμάτια της εργασίας. Πιο αναλυτικά, θα εξηγήσουμε τον τρόπο με τον οποίο μπορεί κανείς να δημιουργήσει ένα καινούργιο Flutter project. Επιπλέον, θα αναλύσουμε το πως γίνεται η δημιουργία ενός Firebase project αλλά και τον τρόπο με τον οποίο θα συνδέσει την πλατφόρμα με την εφαρμογή. Θα αναλύσουμε ορισμένα αρχεία τα οποία χρειάστηκε να επεξεργαστούμε ώστε να γίνουν οι κατάλληλες ρυθμίσεις για την υλοποίηση της εφαρμογής. Τέλος, θα περιγράψουμε διεξοδικά την βάση της εργασίας και θα εξηγήσουμε, την λειτουργία του κάθε πεδίου.

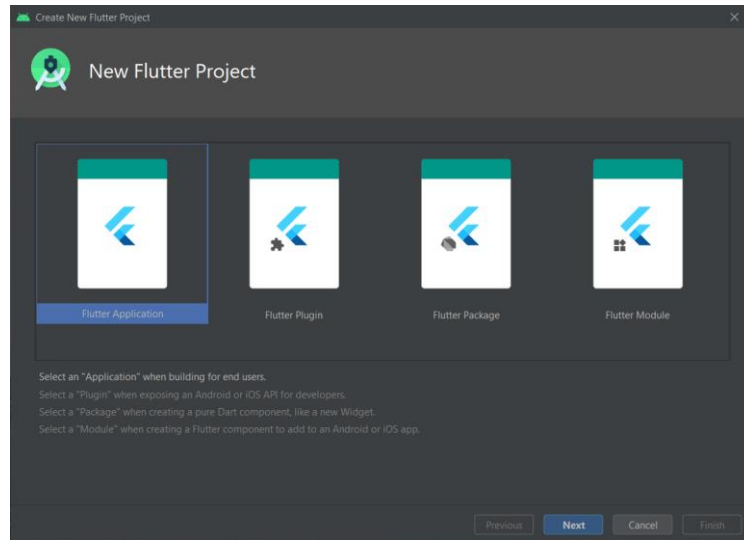
5.2 Δημιουργία Flutter project

Όπως έχουμε αναφέρει στο προηγούμενο κεφάλαιο, η υλοποίηση της συγκεκριμένης εφαρμογής έγινε με το frameword Flutter μέσω του περιβάλλοντος ανάπτυξης λογισμικού Android Studio. Παρακάτω θα αναλύσουμε τη διαδικασία δημιουργίας ενός Flutter project. Αφού ανοίξει το Android Studio και έχουμε κάνει τις απαραίτητες διαδικασίες για την εγκατάσταση του Flutter στον υπολογιστή μας, επιλέγουμε την δημιουργία project με το Flutter (Σχήμα 16).



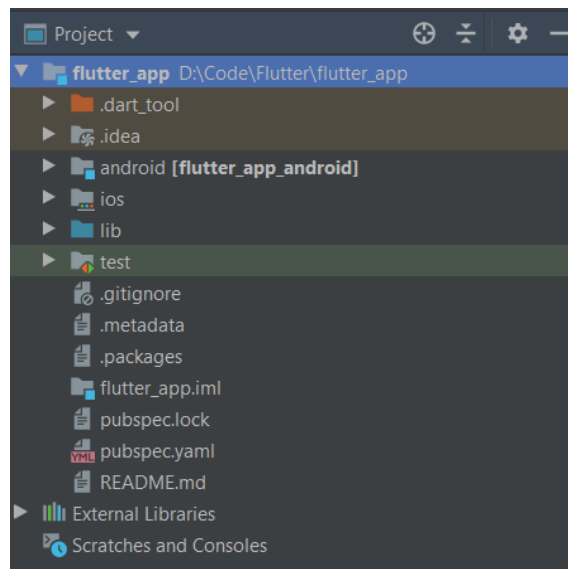
Σχήμα 16: Αρχική οθόνη του Android Studio

Στη συνέχεια μας εμφανίζεται το παράθυρο που βλέπουμε στην παρακάτω φωτογραφία (Σχήμα 17).



Σχήμα 17: Δημιουργία Flutter Project

Παρακάτω, επιλέγουμε όπως φαίνεται και στη φωτογραφία (Σχήμα 17) την πρώτη επιλογή που είναι το Flutter Application για να ξεκινήσουμε την εφαρμογή μας. Στη συνέχεια μας εμφανίζει ένα άλλο παράθυρο που μας ζητάει να δώσουμε όνομα στο project και τοποθεσία για το που θα το αποθηκεύσουμε. Η δομή ενός καινούργιου project με το που το φτιάχνουμε είναι η εξής:



Σχήμα 18: Δομή αρχείων ενός Flutter Project

Παραπάνω (Σχήμα 18) βλέπουμε μια λίστα από τα αρχεία τα οποία δημιουργούνται σε κάθε καινούργιο project. Αναλυτικότερα, οι φάκελοι android και ios, είναι τα ειδικά τμήματα της πλατφόρμας τα οποία περιέχουν πληροφορίες για τις δυο πλατφόρμες αντίστοιχα. Ο φάκελος lib θα περιέχει όλη τη δομή του project, δηλαδή μέσα σε αυτό το φάκελο θα δημιουργηθούν όλα τα αρχεία που χρειάζονται για να φτιάξουμε την εφαρμογή μας. Ο φάκελος test περιέχει όλα τα αρχεία μέσω των οποίων γίνεται η διαδικασία testing της εφαρμογής. Ένα από τα σημαντικότερα αρχεία είναι το pubspec.yaml το οποίο

είναι ουσιαστικά το αρχείο έργου για το Dart. Μέσα σε αυτό μπορεί κανείς να καθορίσει τα assets του project, τα διάφορα dependencies που μπορεί να προσθέσει κανείς στο project του και γενικά ότι έξτρα χρειάζεται να προστεθεί στο project. Τα αρχεία .gitignore και README.md είναι δυο αρχεία τα οποία είναι πολύ γνωστά σε κάποιον προγραμματιστή. Τέλος τα .metadata και .packages είναι δυο αρχεία τα οποία χρειάζεται το Flutter για το χτίσιμο του.

5.3 Δημιουργία Firebase project

Στη συγκεκριμένη ενότητα θα αναφερθούμε στη διαδικασία δημιουργίας της βάσης δεδομένων Firebase. Αναλυτικότερα, για να μπορέσει κανείς να δημιουργήσει ένα project στο Firebase είναι υποχρεωτικό να έχει Google Account. Στη συνέχεια, χρειάζεται να συνδεθεί στο Firebase Console όπου η πρώτη εικόνα που θα αντικρίσει είναι τυχόν ήδη υπάρχοντα projects αλλά και το κουμπί ώστε να δημιουργήσει ένα καινούργιο. Πατώντας το κουμπί θα του εμφανιστεί ένα pop up παράθυρο, όπου θα του ζητηθεί να ονομάσει το project και έπειτα αν θέλει να ενεργοποιήσει τις υπηρεσίες Google Analytics. Αφού περάσει όλα τα βήματα θα του εμφανιστεί το ταμπλό του Firebase Console και όλες οι υπηρεσίες τις οποίες έχει στην διάθεσή του. Με αυτό το τρόπο ο χρήστης δημιουργεί μια διεπαφή ιστού και μια βάση δεδομένων μέσω της οποίας μπορεί να εγκαταστήσει πολλαπλές εφαρμογές είτε είναι εφαρμογή κινητού ή ιστού.

Στη συνέχεια θα αναφερθούμε σε δυο από τις υπηρεσίες τις οποίες διαθέτει το Firebase και τις οποίες έχουμε χρησιμοποιήσει στην εφαρμογή μας, το Cloud Firestore και Authentication. Για να μπορούμε να αλληλεπιδράσουμε με την βάση πρέπει να την ενεργοποιήσουμε. Επομένως, πηγαίνοντας στο Cloud Firestore πατάμε το κουμπί της ενεργοποίησης και μας εμφανίζεται ένα pop up παράθυρο το οποίο μας υποχρεώνει να επιλέξουμε μέσα από δυο επιλογές για το πως θέλουμε να ασφαλήσουμε την βάση μας. Η Firebase μας δίνει το πλεονέκτημα να δημιουργήσουμε τους δικούς μας κανόνες για την βάση. Τέλος αφού ολοκληρώσουμε όλα τα παραπάνω βήματα είμαστε στη θέση να δημιουργήσουμε την βάση.

Για το Authentication λειτουργούμε με τον ίδιο τρόπο. Αναλυτικότερα, πηγαίνουμε στην συγκεκριμένη επιλογή και την ενεργοποιούμε. Στη συνέχεια θα μας εμφανίσει μια πληθώρα επιλογών με τις οποίες μπορούμε να δουλέψουμε ώστε να εγκαταστήσουμε την εφαρμογή μας, για να μπορεί ο χρήστης να έχει μεγαλύτερη ποικιλία σε επιλογές εισόδου στην εφαρμογή. Αυτές που έχουν επιλεγεί στην συγκεκριμένη εφαρμογή είναι το Email/Password, Google και Facebook. Πέρα από το Email/Password για να ενεργοποιηθούν οι άλλες δυο υπηρεσίες θα πρέπει να γίνουν κάποιες επιπρόσθετες διαδικασίες.

Μια τελευταία και πολύ σημαντική εργασία που πρέπει να κάνουμε για να μπορέσουμε να χρησιμοποιήσουμε την συγκεκριμένη βάση στην εφαρμογή μας είναι να δημιουργήσουμε μια σύνδεση μεταξύ της πλατφόρμας για την οποία δημιουργούμε την εφαρμογή και της βάσης. Αναλυτικότερα, στο αρχικό ταμπλό εμφανίζονται κάποιες επιλογές σύνδεσης, μέσω iOS, Android και πλατφόρμα ιστοτόπου. Η συγκεκριμένη εφαρμογή προορίζεται για Android πλατφόρμα επομένως επιλέγουμε αυτήν. Στη συνέχεια μας ζητάει να αναφέρουμε το package name της εφαρμογής μας, ώστε να γίνει η σύνδεση. Τέλος κατεβάζουμε το αρχείο με όνομα google-services.json το οποίο περιέχει όλα τα εργαλεία για να χρησιμοποιήσουμε την βάση και παράλληλα δείχνει που να το αποθηκεύσουμε.

Παρακάτω θα αναφερθούμε αναλυτικά σε κάποιες επιπλέον διαδικασίες που πρέπει να κάνουμε για την εγκατάσταση της βάσης στην εφαρμογή αλλά και γενικά στην εφαρμογή.

5.4 Android Manifest, Grandle και pubspec.yaml

Στη συνέχεια, προστέθηκαν οι κατάλληλες λεπτομέρειες στα συγκεκριμένα αρχεία οι οποίες ήταν απαραίτητες για την εφαρμογή. Συγκεκριμένα, στο AndroidManifest προσθέσαμε το Facebook activity, το οποίο χρειάστηκε να μπει για να μπορέσουμε να ενεργοποιήσουμε την είσοδο στην εφαρμογή μέσω λογαριασμού Facebook (Σχήμα 19).

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.serve_me">
    <uses-permission android:name="android.permission.INTERNET" />
    <application
        android:name="io.flutter.app.FlutterApplication"
        android:label="ServeMe"
        android:icon="@mipmap/ic_launcher">
        <activity...>
        <meta-data android:name="com.facebook.sdk.ApplicationId"
            android:value="@string/facebook_app_id" />

        <activity android:name="com.facebook.FacebookActivity"
            android:configChanges=
                "keyboard|keyboardHidden|screenLayout|screenSize|orientation"
            android:label="@string/app_name" />
        <activity
            android:name="com.facebook.CustomTabActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.VIEW" />
                <category android:name="android.intent.category.DEFAULT" />
                <category android:name="android.intent.category.BROWSABLE" />
                <data android:scheme="@string/fb_login_protocol_scheme" />
            </intent-filter>
        </activity>
        <meta-data
            android:name="flutterEmbedding"
            android:value="2" />
    </application>
</manifest>
```

Σχήμα 19: AndroidManifest αρχείο

Στη συνέχεια, για να μπορέσουμε να έχουμε πρόσβαση στο Firebase και γενικά σε όλες τις υπηρεσίες τις οποίες διαθέτει, αφού κάναμε δημιουργία καινούργιου project στη σελίδα του Firebase με την διαδικασία που έχουμε αναφέρει πιο πάνω, πρέπει να αποθηκεύσουμε ένα αρχείο με όνομα google-services.json. Το αρχείο αυτό το αποθηκεύουμε μέσα στην εφαρμογή μας, συγκεκριμένα στο path: android/app/. Στη συνέχεια, προσθέτουμε τις κατάλληλες βιβλιοθήκες σε δυο διαφορετικά αρχεία grandle ώστε να έχουμε μια ολοκληρωμένη εγκατάσταση της Firebase στην εφαρμογή μας (Σχήμα 20 και 21).

```

apply plugin: 'com.android.application'
apply from: "$flutterRoot/packages/flutter_tools/gradle/flutter.gradle"
apply plugin: 'com.google.gms.google-services'

```

Σχήμα 20: Αρχείο build.gradle στο φάκελο android/app/

```

dependencies {
  classpath 'com.android.tools.build:gradle:3.5.0'
  classpath 'com.google.gms:google-services:4.3.3'
}

```

Σχήμα 21: Αρχείο build.gradle στο φάκελο android/

Στο αρχείο έχουμε προσθέσει όλες τις βιβλιοθήκες που μπορεί κανείς να βρει από την ιστοσελίδα τους. Το συγκεκριμένο, είναι ένα επίσημο ηλεκτρονικό αποθετήριο το οποίο περιέχει μια πληθώρα βιβλιοθηκών δημιουργημένες από άτομα από την ομάδα του Flutter είτε από την κοινότητα. Συγκεκριμένα, στην εφαρμογή έχουν προστεθεί οι παρακάτω βιβλιοθήκες (Σχήμα 22, 23 και 24).

```

#Firebase
firebase_core: "0.5.0"
firebase_auth: "^0.18.0+1"
cloud_firestore: "^0.14.0+2"
flutter_facebook_auth: ^0.3.1+1
flutter_facebook_login: ^3.0.0
google_sign_in: ^4.5.4

#Google Fonts
google_fonts: ^1.1.0

#Flutter Toast
fluttertoast: ^7.1.1

#QR Code
barcode_scan: ^1.0.0

```

Σχήμα 22: Αρχείο pubspec.yaml 1

```
flutter_slidable: ^0.5.7
credit_card: 0.1.6
modal_bottom_sheet: ^0.2.2
string_validator: ^0.1.4
flutter_masked_text: ^0.8.0
carousel_slider: ^2.3.1
sweetalert: ^0.0.1
fancy_dialog: ^1.0.4
```

Σχήμα 23: Αρχείο pubspec.yaml 2

```
flutter_icons:
  android: true
  ios: true
  image_path_android: "assets/icons/serve.png"
  image_path_ios: "assets/icons/serve.png"

flutter:

  uses-material-design: true

  assets:
    - assets/image/
    - assets/icons/
    - assets/credit_card/
```

Σχήμα 24: Αρχείο pubspec.yaml 3

Στα σχήματα 22 και 23 περιέχονται όλες οι βιβλιοθήκες που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής. Ορισμένες από αυτές βοήθησαν στην υλοποίηση της βάσης με την εφαρμογή και άλλες στην διεπαφή χρήστη της εφαρμογής. Στο σχήμα 24 περιέχεται η υλοποίηση επιλογής του εικονιδίου της εφαρμογής και η προσθήκη φωτογραφιών.

5.5 Ανάλυση της βάσης δεδομένων της εφαρμογής

Παρακάτω θα αναφερθούμε επιγραμματικά στη βάση και στο πως είναι δομημένη. Τα δυο βασικά collections είναι το customers και το users. Στο customers αποθηκεύουμε τους πελάτες δηλαδή τα μαγαζιά τα οποία συνεργάζονται με την εφαρμογή ώστε να απεικονίζουν το μενού τους. Στο users έχουμε αποθηκευμένους όλους τους χρήστες – πελάτες που χρησιμοποιούν την εφαρμογή για να παραγγέλνουν. Ο κύριος λόγος που γίνεται η αποθήκευση των πελατών στην εφαρμογή είναι για την ασφαλή λειτουργία της. Αναλυτικότερα, θα μιλήσουμε αρχικά για το collection **Users**. Το συγκεκριμένο collection αποτελείται από πολλά documents τα οποία έχουν σαν **id** το user_id του χρήστη. Επιπλέον, στο εσωτερικό τους αποτελούνται από δυο collections **credit_card** και **orders** και τα πεδία **email**, **fullname**, **password**. Στη συνέχεια το collection **credit_card** αποτελεί τις αποθηκευμένες κάρτες του χρήστη στην εφαρμογή επομένως έχουμε πολλαπλά documents τα οποία έχουν σαν **id** το credit_card_id.

Το συγκεκριμένο document με την σειρά τους αποτελείται από τα παρακάτω πεδία **credit_card_cvv**, **credit_card_expiry_data**, **credit_card_holder_name** και **credit_card_number**. Το άλλο collection **Orders** αποτελεί τις παραγγελίες που έχει κάνει ο χρήστης στα αντίστοιχα μαγαζιά και έχει σαν id το **order_id**. Το συγκεκριμένο document αποτελείται από τα πεδία **order_date**, **order_details**, **order_paymethod**, **order_store**, **order_store_address**, **order_store_img**, **order_store_table** και **order_total**.

Users (collection)		
Id (document)		
Email (string)	Credit_card (collection)	Orders (collection)
FullName (string)	Id (document)	Id (document)
Password (string)	credit_card_cvv (string)	order_date (string)
	credit_card_expiry_data (string)	order_details (array)
	credit_card_holder_name (string)	order_paymethod (string)
	credit_card_number (string)	order_store (string)
		order_store_address (string)
		order_store_img (string)
		order_store_table (string)
		order_total (number)

Πίνακας 4: Users Collection Ανάλυση

Παρακάτω θα αναφερθούμε στο collection customers. Πιο αναλυτικά, έχουμε να κάνουμε με όλα τα μαγαζιά τα οποία χρησιμοποιούν αυτή την εφαρμογή για την διαδικασία των παραγγελιών. Επομένως, όλα τα documents είναι το κάθε μαγαζί με το αντίστοιχο id του **store_id**. Ένα document – μαγαζί αποτελείται από τα πεδία **store_address**, **store_image**, **store_name** και **store_tables** αλλά και από δυο collection **categories** και **offers**. Αναλυτικότερα, το collection **offers** έχει να κάνει με τις προσφορές που μπορεί να έχει το κάθε κατάστημα. Επομένως τα documents που το αποτελούν είναι οι διάφορες προσφορές με το αντίστοιχο id τους **offer_id** και τα πεδία **store_offer_desc**, **store_offer_image**, **store_offer_price** και **store_offer_title**. Το άλλο collection **categories** αποτελεί όλες τις κατηγορίες που έχει το κάθε κατάστημα σε τρόφιμα, ροφήματα κτλπ. Επομένως αποτελείται από τα αντίστοιχα documents με id το **categorie_id** και τα αντίστοιχα πεδία **categorie_image**, **categorie_name**, **products**. Το πεδίο **products** αποτελεί ένα array το οποίο περιέχει αναλυτικά όλα τα προϊόντα της συγκεκριμένης κατηγορίας.

Customers (collection)		
Id (document)		
store_address (string)	Categories (collection)	Offers (collection)
store_image (string)	Id (document)	Id (document)
store_name (string)	categorie_image (string)	store_offer_desc (string)
store_tables (array)	categorie_name (string)	store_offer_image (array)
	products (array)	store_offer_price (number)
		store_offer_title (string)

Πίνακας 5: Customers Collection Ανάλυση

5.6 Επίλογος

Σε αυτό το κεφάλαιο αναλύσαμε κάποιες πτυχές της εφαρμογής. Στη συνέχεια δώσαμε κάποιες επιπλέον πληροφορίες για κάποια πιο πρακτικά θέματα και το πως υλοποιήθηκαν. Στο επόμενο κεφάλαιο θα εξηγήσουμε αναλυτικά πως δουλεύει η εφαρμογή και θα δώσουμε οδηγίες χρήσης.

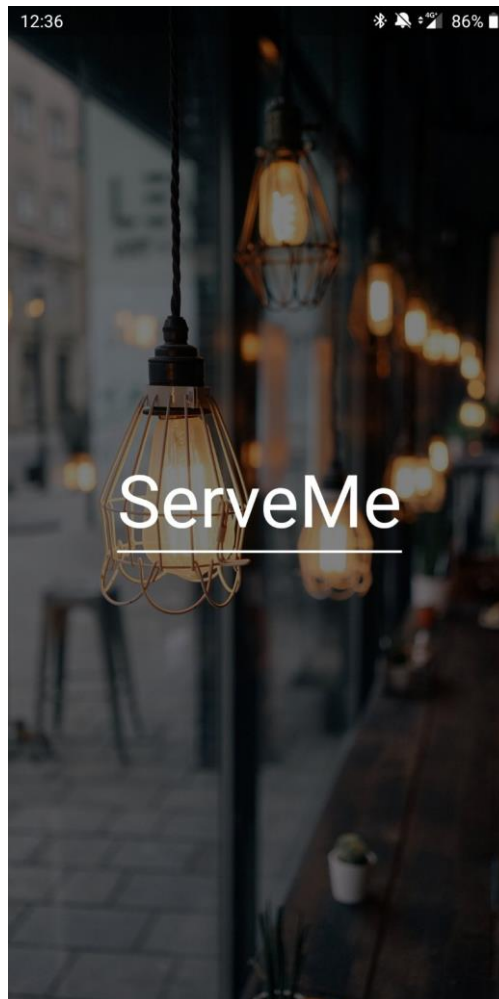
Κεφάλαιο 6ο: Παρουσίαση της εφαρμογής και Οδηγίες Χρήσης

6.1 Εισαγωγή

Στο παρακάτω κεφάλαιο θα εξηγήσουμε πως λειτουργεί η εφαρμογή και τρόπο με τον οποίο μπορεί κανείς να την χρησιμοποιήσει. Ακόμη, θα αναφερθούμε σε όλα τα σημεία της και τρόπο με τον οποίο λειτουργούν με την αναπαράσταση σχημάτων.

6.2 Λίγα λόγια για την έναρξη της εφαρμογής

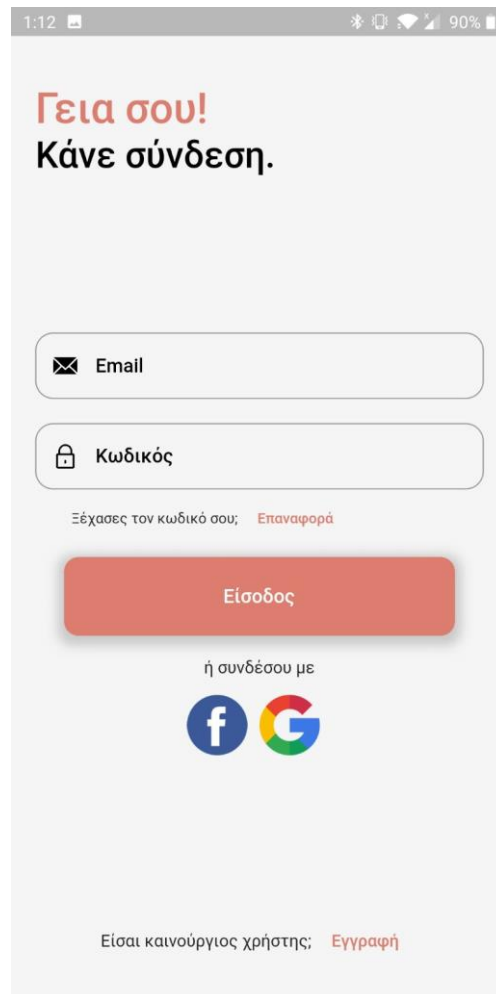
Αρχικά για να χρησιμοποιήσει κανείς την εφαρμογή θα πρέπει να βρίσκεται στο αντίστοιχο κατάσταση το οποίο υποστηρίζει τη συγκεκριμένη υπηρεσία παραγγελίας μέσω κινητού. Αναλυτικότερα, για να μπορέσει κάποιος να παραγγείλει μέσω της εφαρμογής, θα πρέπει να την κατεβάσει στο κινητό του. Στη συνέχεια να πάει σε κάποιο χώρο εστίασης που χρησιμοποιεί τη συγκεκριμένη υπηρεσία και να κάτσει σε ένα τραπέζι το οποίο έχει πάνω ένα QR Code ώστε να μπορέσει να το σκανάρει και να παραγγείλει. Αρχικά ανοίγοντας κάποιος την εφαρμογή θα δει την αρχική οθόνη η οποία είναι η παρακάτω (Σχήμα 25).



Σχήμα 25: Αρχική οθόνη εφαρμογής

6.3 Είσοδος χρήστη και εγγραφή

Στη συνέχεια για να μπορέσει κάποιος να χρησιμοποιήσει την εφαρμογή πρέπει να γίνει χρήστης. Επομένως, θα πρέπει να δημιουργήσει έναν λογαριασμό. Το συγκεκριμένο λογαριασμό θα μπορέσει να τον δημιουργήσει με τρεις επιλογές: 1. με ονοματεπώνυμο, email και κωδικό, 2. σύνδεση με λογαριασμό Facebook, ή 3. με λογαριασμό Google. Στην φωτογραφία παρακάτω (Σχήμα 26) θα δούμε το πρώτο παράθυρο που συναντάει ο χρήστης. Σε περίπτωση που έχει συνδεθεί με απλό λογαριασμό θα του ζητηθεί να εισάγει το Email και Κωδικό, διαφορετικά να συνδεθεί κατευθείαν με τις δυο επιλογές Facebook ή Google.



Σχήμα 26: Είσοδος χρήστη

Εάν δεν διαθέτει λογαριασμό και δε θέλει να συνδεθεί κατευθείαν μέσω Facebook και Google, τότε πρέπει να πατήσει Εγγραφή, όπου θα του εμφανιστεί η παρακάτω φωτογραφία (Σχήμα 27).

1:12

Καλώς όρισες!
Κάνε εγγραφή.

Όνοματεπώνυμο

Email

Κωδικός

Ο κωδικός σου πρέπει να περιέχει τουλάχιστον ένα κεφαλαίο και ένα μικρό γράμμα, ένα ψηφίο και ένα special character (πχ. ! @ # \$ % ^ * ~)

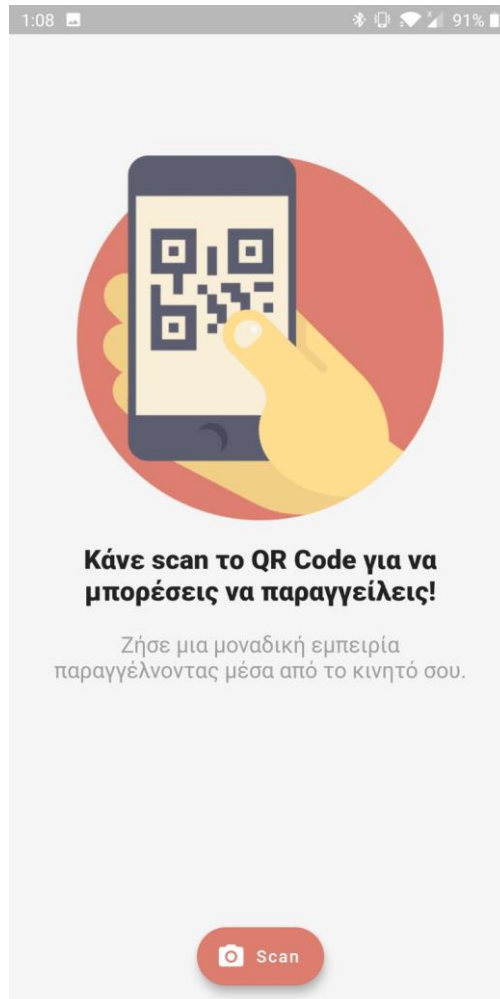
Εγγραφή

Πίσω στην είσοδο χρήστη

Σχήμα 27: Εγγραφή χρήστη

6.4 QR Code scanner

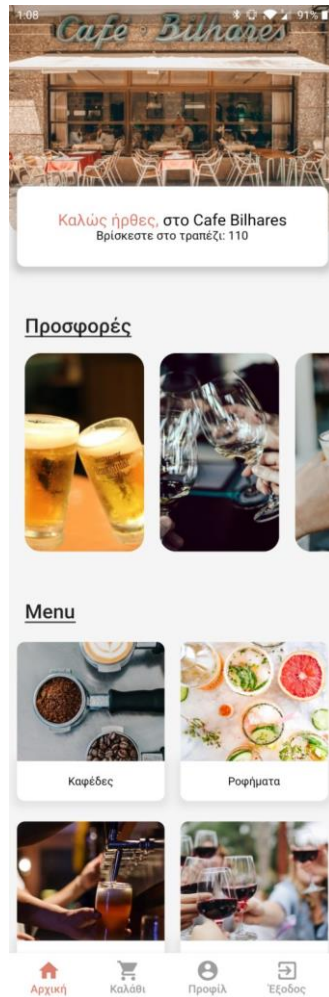
Αφού ο χρήστης περάσει τη διαδικασία της εγγραφής του ή της εισόδου του, θα του εμφανιστεί το παρακάτω παράθυρο (Σχήμα 28). Αναλυτικότερα, σε περίπτωση που ο χρήστης έχει ήδη λογαριασμό στην εφαρμογή και δεν κάνει αποσύνδεση, όταν θα πάει να τη λειτουργήσει ξανά κατά την είσοδο του στην εφαρμογή θα του εμφανιστεί αμέσως η παρακάτω σελίδα (Σχήμα 28). Στη συγκεκριμένη σελίδα ο χρήστης το μόνο που έχει να κάνει είναι να πατήσει το κουμπί scan ώστε να του ανοίξει η κάμερα για να σκανάρει το QR Code που έχει πάνω στο τραπέζι. Στη περίπτωση που για κάποιο λόγο κάτι πάει στραβά θα του εμφανιστούν τα κατάλληλα μηνύματα για να μπορέσει να συνεχίσει τη διαδικασία.



Σχήμα 28: Σκανάρισμα του QR Code

6.5 Κύριο μενού

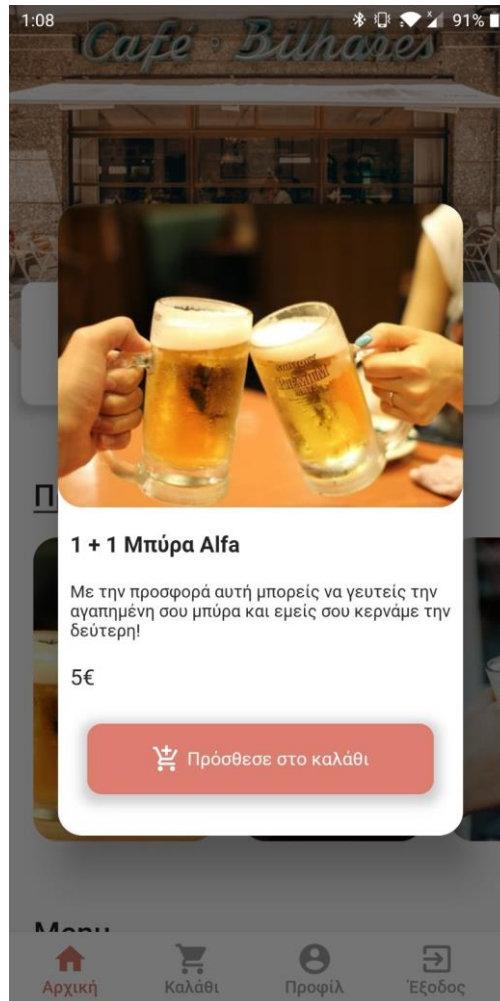
Το συγκεκριμένο παράθυρο θα εμφανιστεί αφού έχουν πάει όλα καλά. Επομένως ο χρήστης πλέον βλέπει την φωτογραφία του καταστήματος, το όνομα και σε πιο τραπέζι βρίσκεται (Σχήμα 29). Στη συνέχεια μπορεί να περιηγηθεί στο μενού του καταστήματος αλλά και να δει τις διάφορες προσφορές που μπορεί να έχει το συγκεκριμένο κατάστημα που βρίσκεται. Στο κάτω μέρος της οθόνης θα δει τέσσερις επιλογές, τις οποίες θα αναλύσουμε παρακάτω την καθεμία ξεχωριστά. Η πρώτη είναι η Αρχική οθόνη (η οθόνη στην οποία βρισκόμαστε) η δεύτερη και η Τρίτη θα αναλυθούν παρακάτω και η τέταρτη επιλογή είναι η Έξοδος στην οποία πατώντας την κανείς μπορεί να φύγει από την εφαρμογή.



Σχήμα 29: Κύριο Μενού

6.6 Προσφορές

Η συγκεκριμένη κατηγορία στην εφαρμογή έχει να κάνει με το κάθε κατάστημα. Πιο αναλυτικά, το κάθε κατάστημα μπορεί να έχει την επιλογή να διαμορφώσει όπως θέλει το menu και αντίστοιχα να προσθέσει ή να αφαιρέσει διάφορες προσφορές που μπορεί να έχει. Επομένως όπως βλέπουμε και παραπάνω στη φωτογραφία (Σχήμα 29) στο συγκεκριμένο κατάστημα η πρώτη επιλογή προσφοράς έχει να κάνει με μπίρα και πατώντας πάνω της θα εμφανιστεί το παρακάτω pop up παράθυρο (Σχήμα 30) όπου και θα δει παραπάνω λεπτομέρειες για την συγκεκριμένη προσφορά.



Σχήμα 30: Προσφορές μαγαζιού

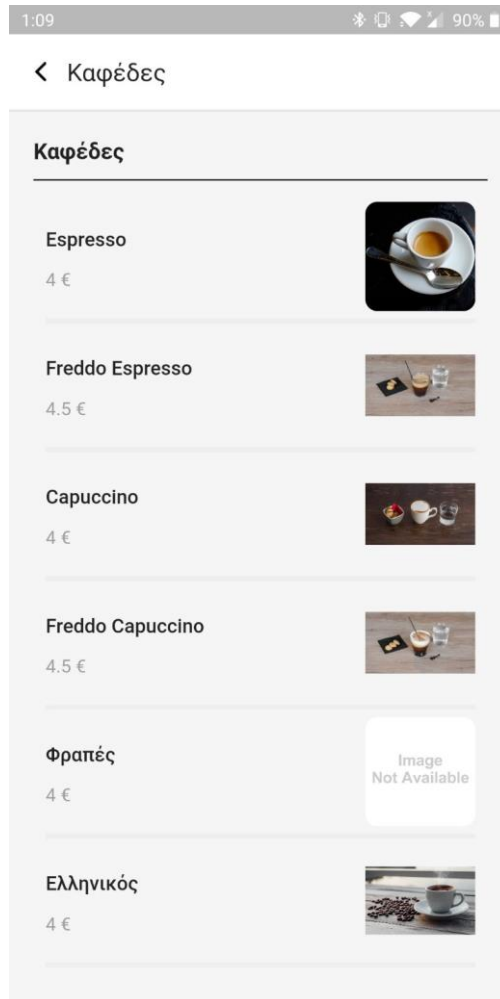
Εάν επιλέξει να πάρει τη συγκεκριμένη προσφορά, θα πρέπει να πατήσει το κουμπί «Πρόσθεσε στο καλάθι σου», διαφορετικά θα πατήσει κάπου στο κενό για να συνεχίσει την περιήγηση του.

6.7 Υποκατηγορία του μενού

Στη συγκεκριμένη ενότητα θα αναφερθούμε στις υποκατηγορίες του μενού. Επομένως θα δούμε 3 διαφορετικές διεπαφές χρήστη, οι οποίες είναι ο Καφές, το Ποτό και όλα τα υπόλοιπα τα οποία συγκαταλέγονται σε μια ενότητα.

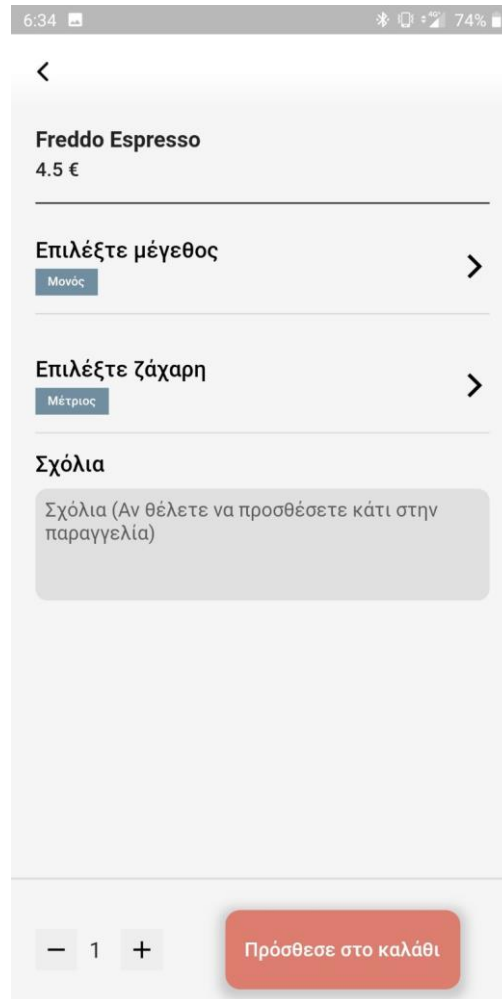
6.7.1 Καφές

Εάν ο χρήστης πατήσει πάνω στην κατηγορία καφέ θα εμφανιστεί το παρακάτω παράθυρο (Σχήμα 31).



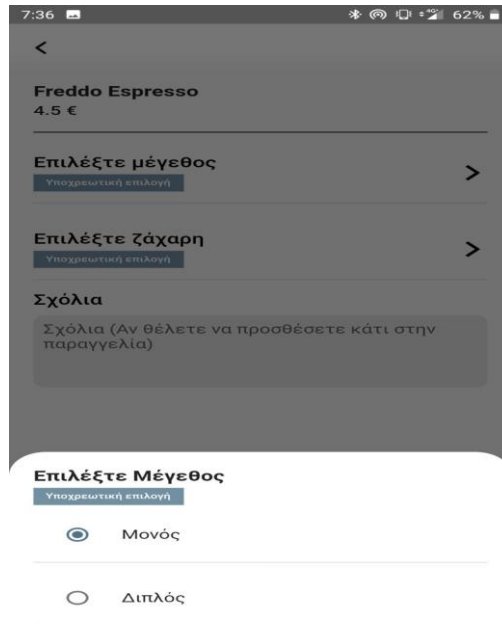
Σχήμα 31: Κατηγορία καφέ

Εδώ θα συναντήσει μια πληθώρα επιλογών από καφέδες. Η δομή εμφάνισης ενός συγκεκριμένου καφέ είναι όπως φαίνεται στο Σχήμα 31 το όνομα του καφέ, η τιμή από κάτω και στα δεξιά η φωτογραφία του αντίστοιχου καφέ. Στη περίπτωση που δεν υπάρχει φωτογραφία θα εμφανιστεί το μήνυμα Image Not Available. Επομένως πατώντας ο χρήστης πάνω σε ένα καφέ που θέλει, θα του εμφανιστεί η παρακάτω φωτογραφία (Σχήμα 32).

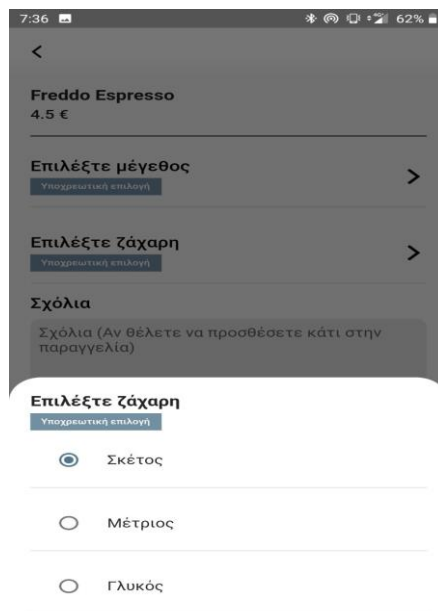


Σχήμα 32: Λεπτομέρειες παραγγελίας καφέ

Στην συγκεκριμένη οθόνη (Σχήμα 32) ζητείται από τον χρήστη να επιλέξει Μέγεθος και Ζάχαρη για το συγκεκριμένο καφέ που επέλεξε. Επομένως πατώντας στις δυο αυτές επιλογές του εμφανίζεται κάτω στην οθόνη να διαλέξει μέσα από μια λίστα τις διαθέσιμες επιλογές που υπάρχουν. Στη φωτογραφία βλέπουμε ότι ο χρήστης έχει επιλέξει στην επιλογή «Επιλέξτε μέγεθος» «Μονός» (Σχήμα 33) και στην επιλογή «Επιλέξτε ζάχαρη» «Μέτριος» (Σχήμα 34). Επίσης έχει τις επιλογές να αφήσει κάποιο σχόλιο που θέλει για το καφέ και να προσθέσει ή να αφαιρέσει ποσότητα.

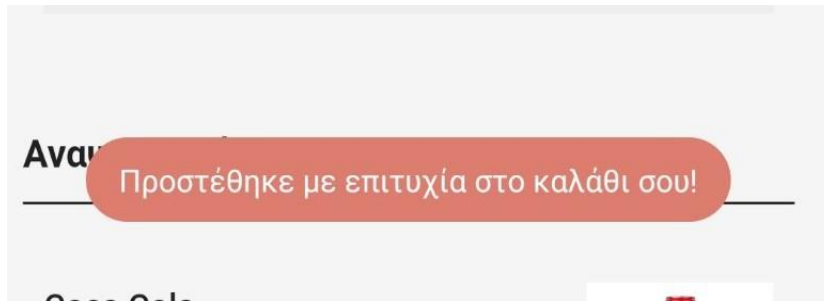


Σχήμα 33: Επιλογή Μέγεθος καφέ



Σχήμα 34: Επιλογή Ζάχαρης καφέ

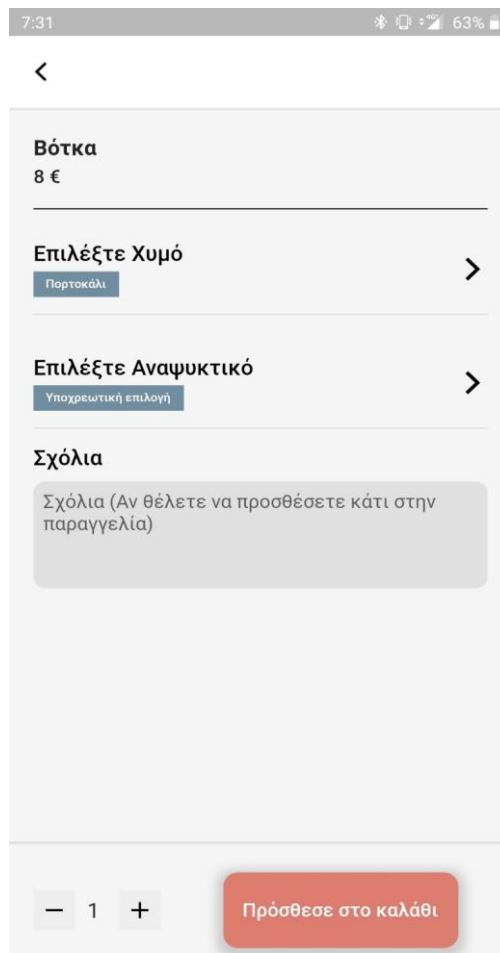
Τέλος εάν έχει ολοκληρώσει όλες τις επιλογές πατάει «Πρόσθεσε στο καλάθι» και του εμφανίζεται το αντίστοιχο μήνυμα (Σχήμα 35).



Σχήμα 35: Μήνυμα επιτυχίας εισόδου στο καλάθι

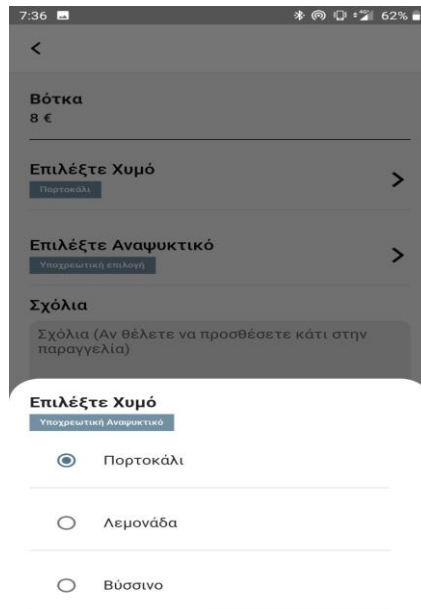
6.7.2 Ποτά

Στη υποκατηγορία αυτή ο χρήστης έχει την επιλογή να επιλέξει ένα ποτό από αυτά που υπάρχουν στο μενού. Επομένως με το που κάνει την επιλογή του, εμφανίζεται ένα αντίστοιχο παράθυρο όπως στην κατηγορία καφέ, όπου πρέπει να επιλέξει μέσα από δυο επιλογές αυτό που θέλει. Όπως βλέπουμε στο παρακάτω σχήμα (Σχήμα 36) ο χρήστης έχει επιλέξει μόνο την επιλογή «Επιλέξτε Χυμό» και συγκεκριμένα το Πορτοκάλι. Στη συγκεκριμένη περίπτωση μια από τις δυο επιλογές είναι υποχρεωτική.

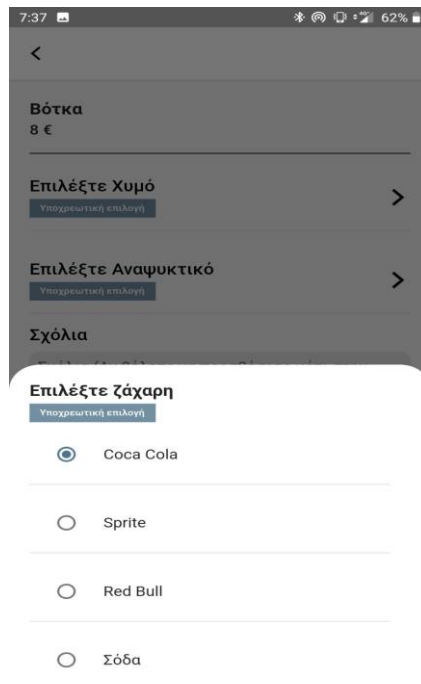


Σχήμα 36: Επιλογή αλκοολούχου ποτού

Επομένως, πατώντας στις δυο επιλογές αντίστοιχα και εδώ του εμφανίζονται στο κάτω μέρος να επιλέξει μια από την λίστα επιλογών που υπάρχουν (Σχήμα 37).



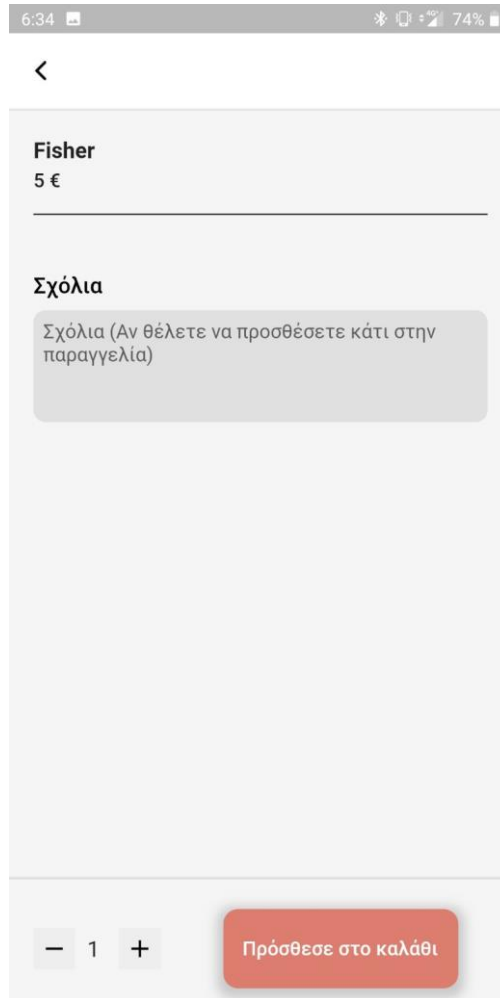
Σχήμα 37: Επιλογή Χυμού



Σχήμα 38: Επιλογή αναψυκτικού

6.7.3 Όλες οι υπόλοιπες υποκατηγορίες

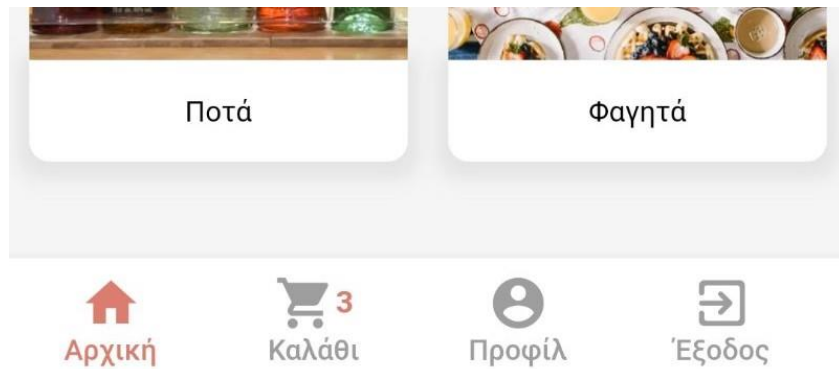
Σε αυτές τις υποκατηγορίες ανήκουν τα προϊόντα τα οποία δεν έχουν κάποιες επιπλέον επιλογές να προσθέσει κανείς. Έχουμε να κάνουμε με μια απλή προσθήκη με ένα προερατικό σχόλιο (Σχήμα 39).



Σχήμα 39: Επιλογή προϊόντος μύρα

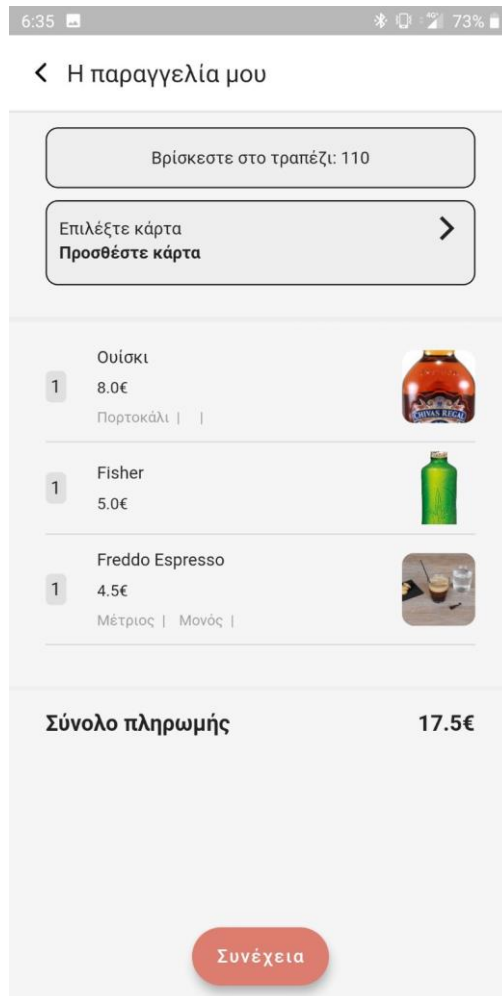
6.8 Καλάθι

Στο συγκεκριμένη ενότητα θα μιλήσουμε για το παράθυρο στο οποίο απεικονίζεται το καλάθι. Αφού ο χρήστης επιλέξει ότι θέλει από το μενού τότε στο κάτω μέρος της οθόνης θα εμφανιστούν δίπλα από το εικονίδιο του καλαθιού το πόσα προϊόντα έχουν τοποθετηθεί σε αυτό (Σχήμα 40).



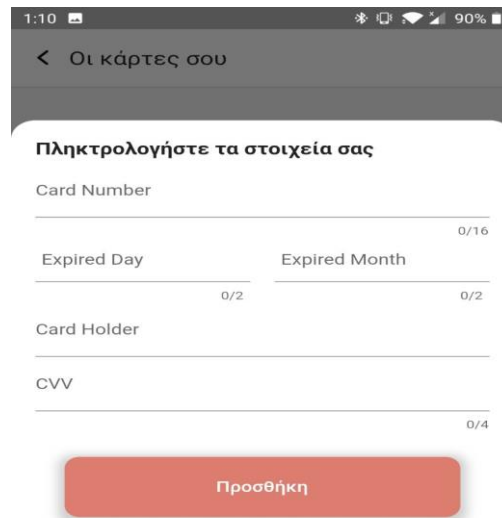
Σχήμα 40: Ένδειξη προϊόντων στο καλάθι

Επομένως πατώντας πάνω στην επιλογή «καλάθι» θα του εμφανιστεί η παρακάτω φωτογραφία με την αναλυτική περιγραφή της παραγγελίας του (Σχήμα 41).



Σχήμα 41: Καλάθι

Θα δει δηλαδή το τραπέζι στο οποίο βρίσκεται, τον τρόπο πληρωμής (που είναι μόνο με την κάρτα του), τα προϊόντα που έχει τοποθετήσει στο καλάθι και από κάτω το σύνολο πληρωμής. Στην επιλογή τρόπου πληρωμής θα πρέπει ο χρήστης να επιλέξει μια από τις κάρτες που έχει αποθηκευμένες στην εφαρμογή. Στη περίπτωση που δεν έχει κάποια κάρτα αποθηκευμένη, θα του εμφανιστεί η αντίστοιχη φόρμα για να συμπληρώσει την κάρτα (Σχήμα 42).



1:10 90%

< Οι κάρτες σου

Πληκτρολογήστε τα στοιχεία σας

Card Number 0/16

Expired Day 0/2 Expired Month 0/2

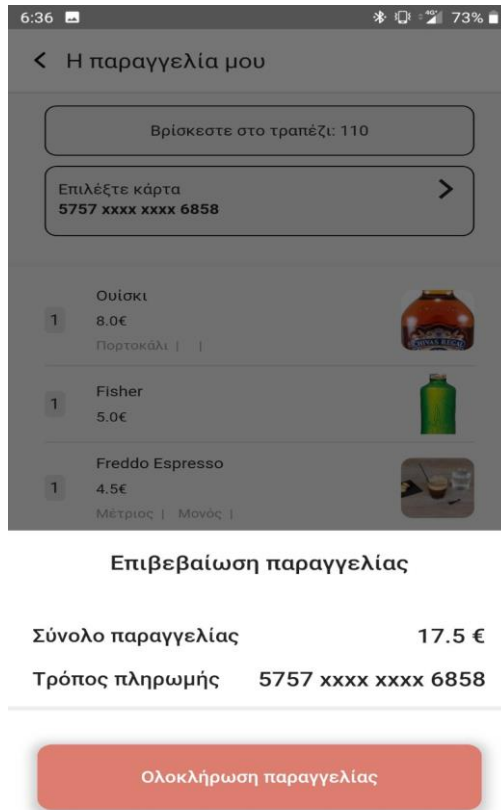
Card Holder

CVV 0/4

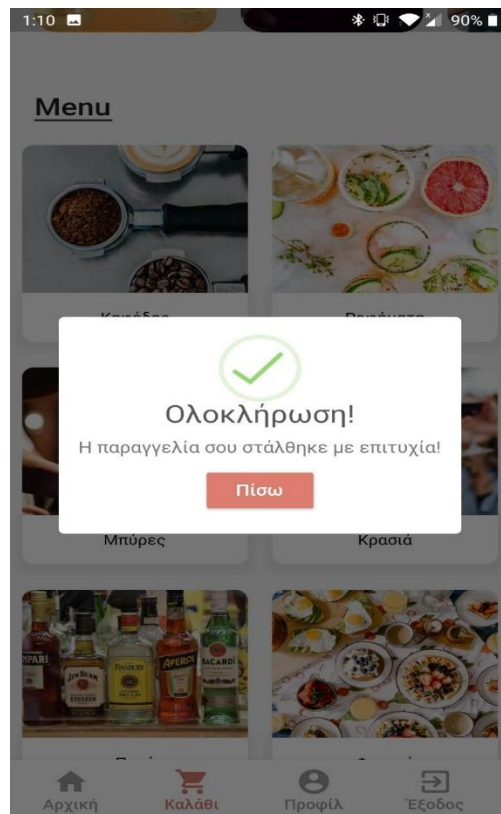
Προσθήκη

Σχήμα 42: Φόρμα συμπλήρωσης εισόδου κάρτας

Στη συνέχεια αφού έχει επιλέξει και τρόπο πληρωμής, θα μπορέσει να ολοκληρώσει την παραγγελία του. Ακόμη, πριν ολοκληρώσει την παραγγελία του θα μπορεί σέρνοντας ένα προϊόν προς τα αριστερά να το διαγράψει από το καλάθι. Πατώντας το «συνέχεια» θα του εμφανιστεί μια σύνοψη με το συνολικό ποσό της παραγγελίας του και την κάρτα που έχει επιλέξει για την συγκεκριμένη συναλλαγή, ώστε να κάνει μια επαλήθευση (Σχήμα 43). Τέλος, πατώντας το «Ολοκλήρωση παραγγελίας» στέλνει την παραγγελία του και του εμφανίζεται μήνυμα ότι όλα έγιναν σωστά (Σχήμα 44).



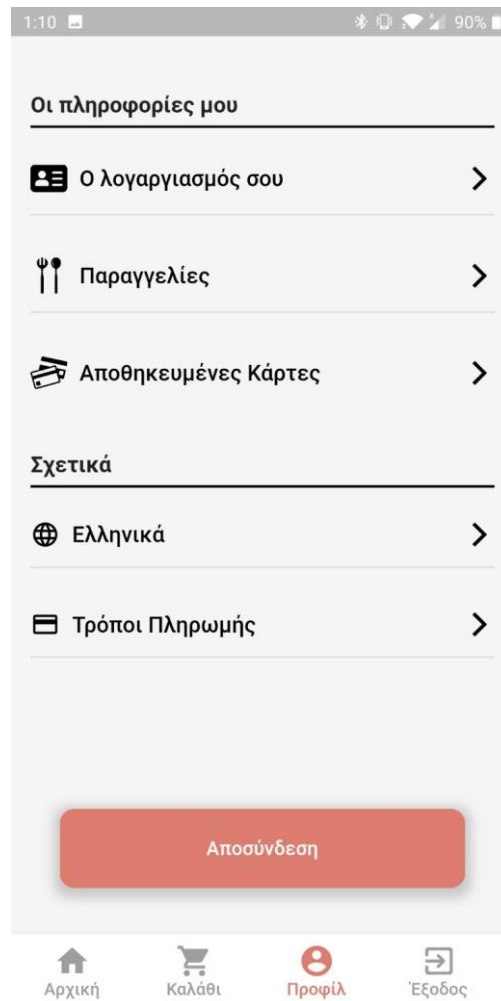
Σχήμα 43: Επιβεβαίωση παραγγελίας



Σχήμα 44: Μήνυμα επιτυχής παραγγελίας

6.9 Προφίλ

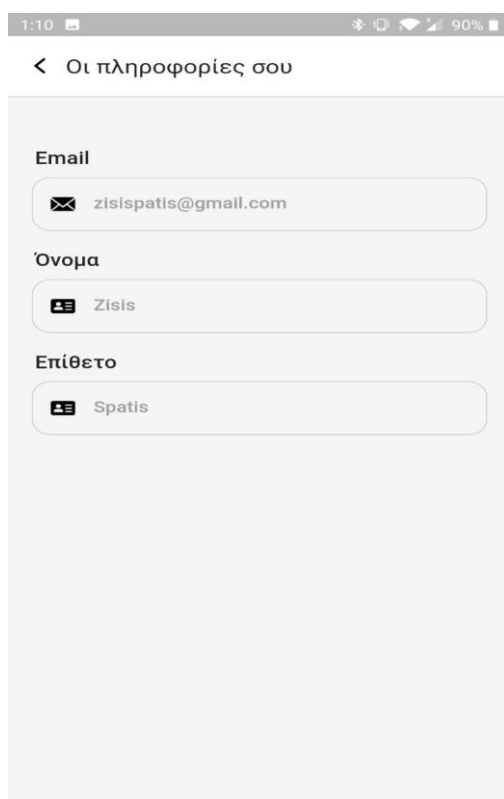
Στην συγκεκριμένη επιλογή την οποία μπορεί να την βρει στο μενού πλοήγησης της εφαρμογής μπορεί κανείς να διαχειριστεί το προφίλ του, να δει τις παραγγελίες που έχει κάνει χρησιμοποιώντας την εφαρμογή, καθώς και να διαχειριστεί τις κάρτες που έχει αποθηκευμένες στην εφαρμογή. Επίσης θα δει κάποιες πληροφορίες σχετικά με την εφαρμογή, όπως τη γλώσσα και τους διάφορους τρόπους πληρωμής (Σχήμα 45).



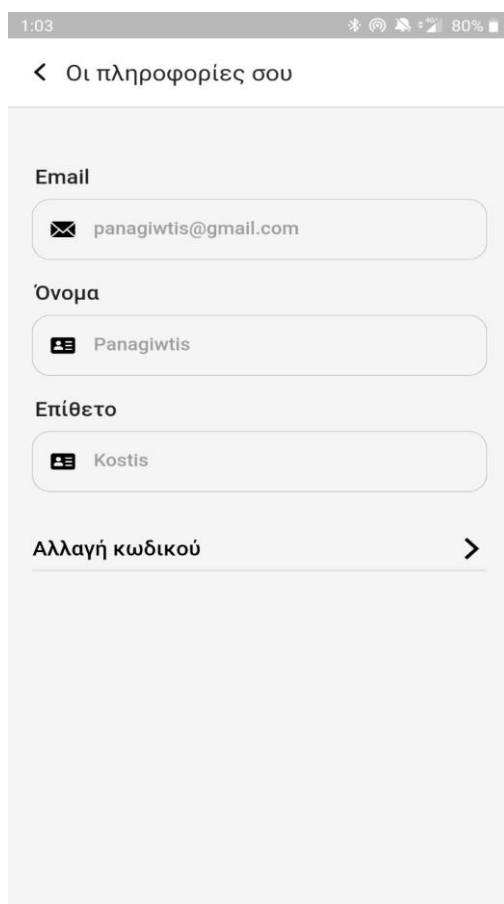
Σχήμα 45: Επιλογές προφίλ χρήστη

6.9.1 Ο λογαριασμός μου

Η οθόνη αυτή δίνει τη δυνατότητα στο χρήστη να μπορεί να δει τα προσωπικά στοιχεία του, Όνομα, Επίθετο και Email και Κωδικό. Στην περίπτωση που ο χρήστης έχει συνδεθεί με ένα απλό λογαριασμό θα έχει την επιλογή να επεξεργαστεί τα δεδομένα του και το κωδικό του, όταν όμως έχει συνδεθεί με Facebook και Google τότε το μόνο που μπορεί να κάνει είναι να δει τα στοιχεία με τα οποία έχει συνδεθεί (Σχήμα 46, Σχήμα 47).

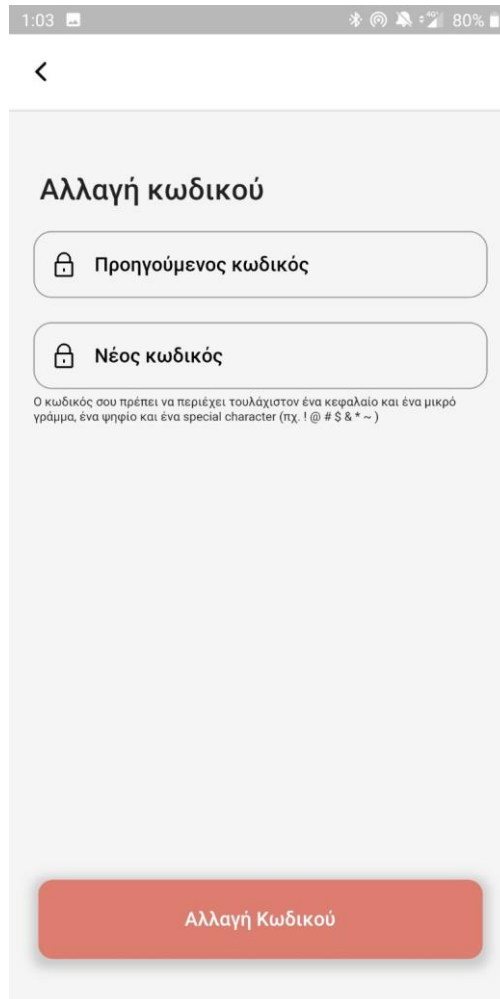


Σχήμα 46: Πληροφορίες χρήστη συνδεδεμένου με Google και Facebook



Σχήμα 47: Πληροφορίες απλού χρήστη

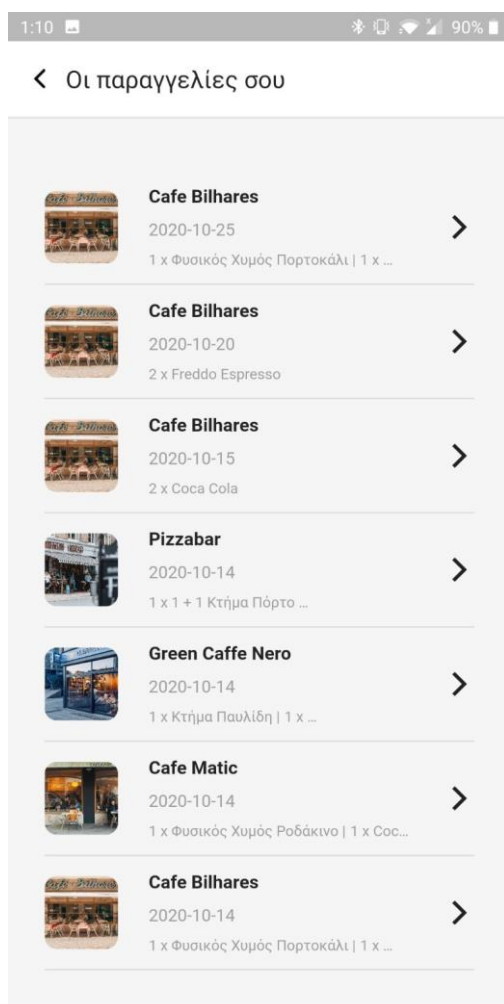
Στη περίπτωση που ο χρήστης συνδεθεί με απλό λογαριασμό, όπως φαίνεται στο Σχήμα 47, εμφανίζεται η επιλογή αλλαγή κωδικού. Πατώντας πάνω θα του εμφανιστεί ένα άλλο παράθυρο, όπου θα του ζητηθεί να βάλει το παλιό κωδικό του και μετά το καινούργιο ώστε να ολοκληρωθεί η διαδικασία (Σχήμα 48).



Σχήμα 48: Αλλαγή κωδικού

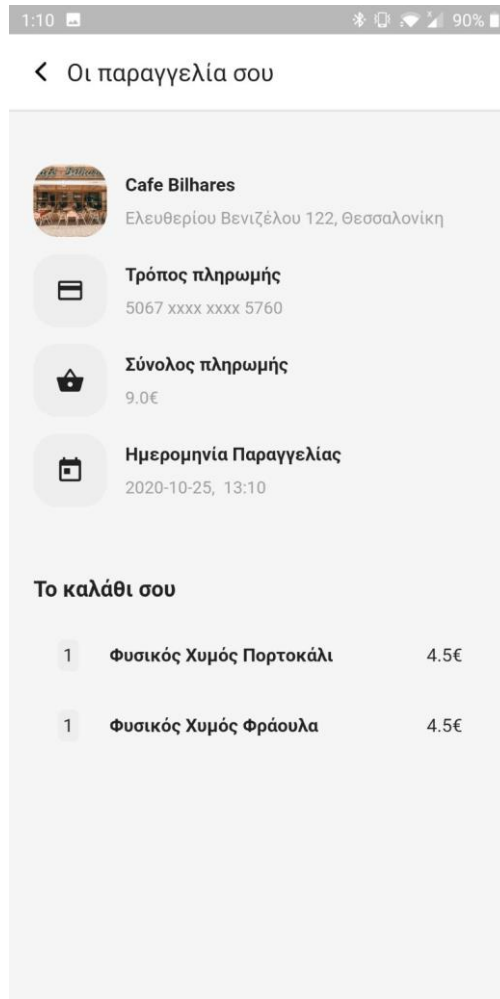
6.9.2 Παραγγελίες

Στη επιλογή αυτή ο χρήστης μπορεί να δει όλες τις παραγγελίες που έχει κάνει με το συγκεκριμένο λογαριασμό. Επομένως εμφανίζεται η παρακάτω οθόνη (Σχήμα 49).



Σχήμα 49: Παραγγελίες

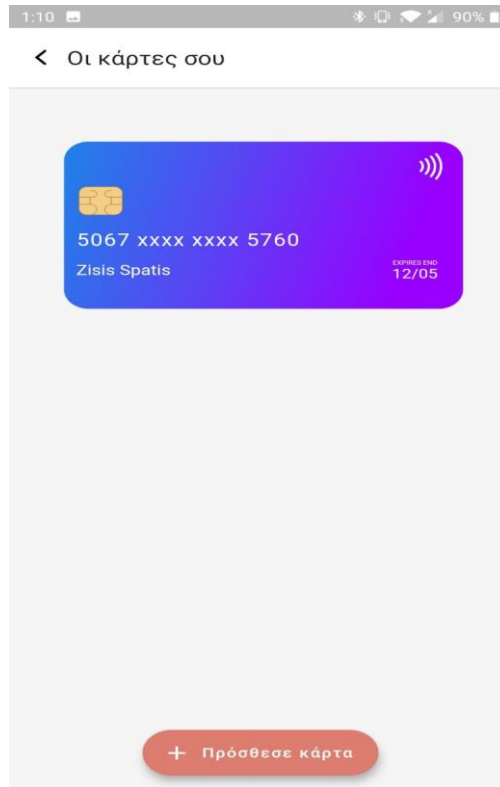
Επομένως πατώντας πάνω σε μια παραγγελία μπορεί να δει αναλυτικά το μαγαζί στο οποίο έγινε και τη διεύθυνση του, τον τρόπο πληρωμής με την κάρτα που χρησιμοποίησε, το σύνολο της πληρωμής που είχε κάνει και την ημερομηνία της παραγγελίας. Τέλος στο κάτω μέρος μπορεί να δει αναλυτικά την παραγγελία που είχε κάνει, δηλαδή τα προϊόντα που είχε παραγγείλει (Σχήμα 50).



Σχήμα 50: Πληροφορίες παραγγελίας

6.9.3 Αποθηκευμένες κάρτες

Σε αυτή την οθόνη μπορεί ο χρήστης να δει τις κάρτες που έχει αποθηκευμένες στην εφαρμογή. Εάν δεν έχει καμία κάρτα μπορεί να πατήσει το κουμπί «Πρόσθεσε κάρτα» (Σχήμα 51).

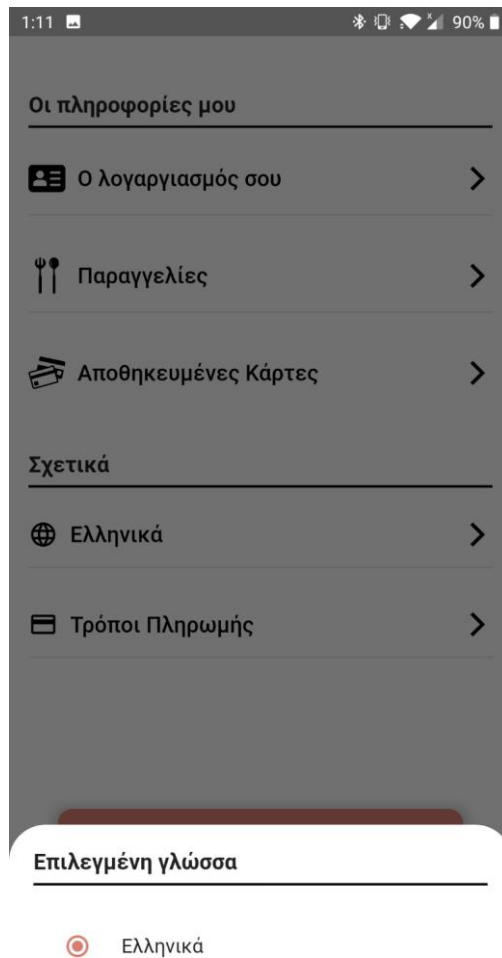


Σχήμα 51: Κάρτες

Με το που πατάει το πρόσθεσε κάρτα του εμφανίζεται η φόρμα όπου μπορεί να προσθέσει την κάρτα του, ώστε να την χρησιμοποιήσει σε μια μελλοντική συναλλαγή σε κάποιο κατάστημα.

6.9.4 Γλώσσα

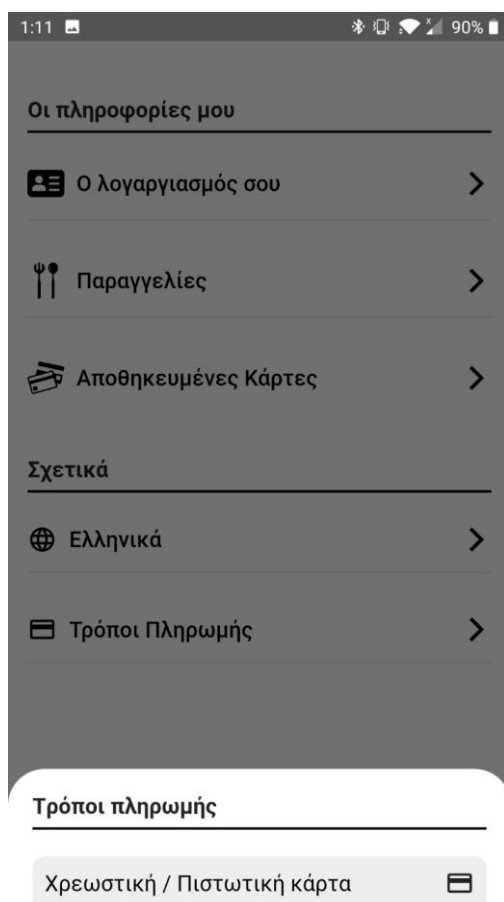
Στη κατηγορία «Σχετικά» η επιλογή γλώσσα έχει να κάνει με τη γλώσσα που θα εμφανίζεται γενικά η εφαρμογή (Σχήμα 52). Μελλοντικά θα υπάρχουν και άλλες γλώσσες. Επομένως, πατώντας κάποιος πάνω εμφανίζονται στο κάτω μέρος οι αντίστοιχες γλώσσες. Τη συγκεκριμένη στιγμή η μόνη επιλεγμένη γλώσσα που υπάρχει είναι τα Ελληνικά.



Σχήμα 52: Επιλεγμένη γλώσσα

6.9.5 Τρόποι πληρωμής

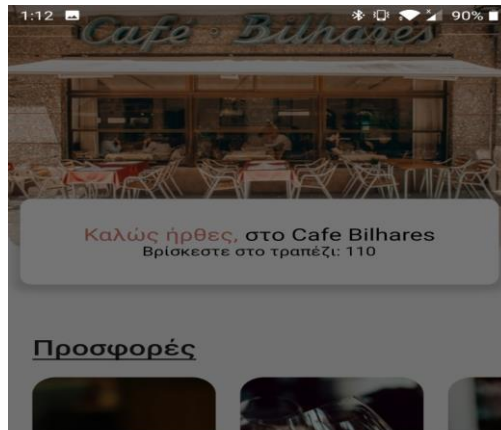
Στην επιλογή «Τρόποι πληρωμής» μπορεί κανείς να δει τους τρόπους με τους οποίους μπορεί να πληρώσει στην συγκεκριμένη εφαρμογή. Επομένως πατώντας πάνω εμφανίζονται στο κάτω μέρος οι επιλογές (Σχήμα 53).



Σχήμα 53: Τρόποι πληρωμής

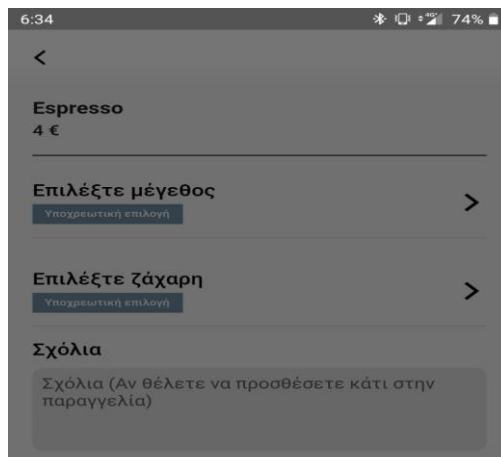
6.10 Μηνύματα Σφάλματος

Στην περίπτωση που κάτι γίνει λάθος από την πλευρά του χρήστη εμφανίζονται τα κατάλληλα μηνύματα. Ένα πιθανό σφάλμα είναι να πάει ο χρήστης να πατήσει την επιλογή «Καλάθι» χωρίς να έχει προσθέσει κάποιο προϊόν και τότε του εμφανίζεται το αντίστοιχο μήνυμα (Σχήμα 54).



Σχήμα 54: Μήνυμα άδειου καλαθιού

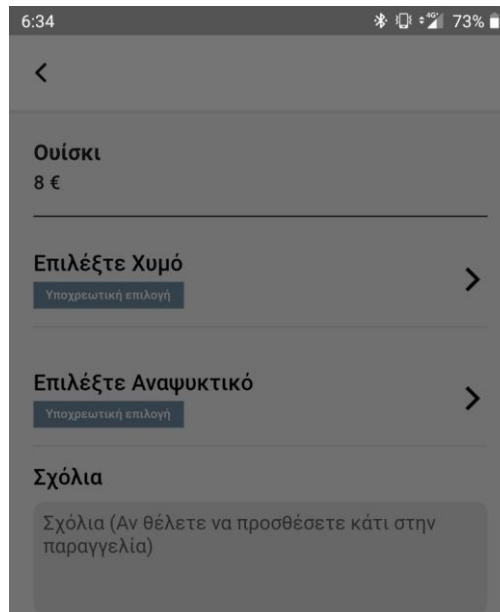
Στην συνέχεια, εάν ο χρήστης στην επιλογή καφέδες πάει να προσθέσει έναν καφέ χωρίς να επιλέξει τις υποχρεωτικές επιλογές οι οποίες του ζητούνται για να συνεχίσει, θα του εμφανιστεί το κατάλληλο μήνυμα (Σχήμα 55).



Σχήμα 55: Μήνυμα λάθους επιλογής 1

Κεφάλαιο 6

Επίσης, όταν πάει να επιλέξει κάποιο ποτό θα του εμφανιστεί κατάλληλο μήνυμα για να επιλέξει χυμό ή αναψυκτικό ώστε να ολοκληρωθεί η παραγγελία (Σχήμα 56).



Προσοχή!

Πρέπει να επιλέξεις μια από τις δύο επιλογές χυμό ή αναψυκτικό.

Κλείσιμο

Σχήμα 56: Μήνυμα λάθους επιλογής 2

Κεφάλαιο 7ο: Συμπέρασμα

Στη συγκεκριμένη εργασία έχει δημιουργηθεί μια εφαρμογή για κινητά τηλέφωνα η οποία θα μπορέσει να διευκολύνει μια πτυχή της καθημερινότητας μας, όπως είναι η παραγγελία σε διάφορους χώρους εστίασης. Ήταν μια ιδέα η οποία ήθελα να υλοποιήσω εδώ και καιρό και μετά τα γεγονότα που βιώνουμε στις μέρες μας με την εξάπλωση της πανδημίας, πιστεύω ότι θα είναι μια πολύ πρωτοπόρα και καινοτόμα υπηρεσία η οποία θα ξεκινήσει να εμφανίζεται και στη χώρα μας. Στο εξωτερικό, η ιδέα του να παραγγέλνει κάποιος από το κινητό του αλλά και να πληρώνει έχει αναπτυχθεί τα τελευταία χρόνια σε κάποιες χώρες. Επομένως, ο κόσμος στη χώρα μας, με την εξοικείωση που έχει πλέον με τις εφαρμογές κινητών συσκευών, δεν θα δυσκολευτεί να υιοθετήσει μια τέτοια υπηρεσία. Η συγκεκριμένη υπηρεσία προσφέρει μια πιο γρήγορη και εύκολη πρόσβαση στο μενού του αντίστοιχου καταστήματος αλλά και γενικά σε όλη τη διαδικασία της παραγγελίας και πληρωμής σε ένα κατάστημα εστίασης.

Για την ανάπτυξη της εφαρμογής ήθελα να χρησιμοποιήσω ένα καινούργιο εργαλείο, το οποίο θα με βοηθούσε τόσο στο να φτιάξω εύκολα και γρήγορα μια πολύ ωραία διεπαφή χρήστη, αλλά και στο να φτιάξω μια εφαρμογή η οποία θα είναι cross platform. Γι' αυτό το λόγο χρησιμοποίησα το Flutter που μου παρέχει όλες αυτές τις δυνατότητες και με μια μεγάλη γκάμα βιβλιοθηκών. Επίσης η βάση παίζει πολύ σημαντικό ρόλο σε μια εφαρμογή. Επομένως, σε αυτή τη περίπτωση χρησιμοποίησα μια καινοτόμα βάση όπως είναι η Firebase, η οποία παρέχει ένα μεγάλο όγκο υπηρεσιών για μπορεί κάποιος να φτιάξει το ιδανικό προφίλ για τη βάση του. Ακόμη, η χρησιμότητα της τεχνολογίας QR Code βοήθησε στο να μπορέσω να διαχειριστώ ένα κομμάτι της εφαρμογής. Όλα αυτά μου έδωσαν γνώσεις τις οποίες θα αξιοποιήσω μελλοντικά, στο επάγγελμά μου.

Η εφαρμογή σίγουρα μπορεί να αναβαθμιστεί μελλοντικά σε πολλά κομμάτια της. Ένα από αυτά μπορεί να είναι στο διαφημιστικό κομμάτι το οποίο θα μπορούν να έχουν τα μαγαζιά μέσα από την εφαρμογή. Επομένως, το κάθε μαγαζί που θα συνεργάζεται με την εφαρμογή θα έχει την δυνατότητα πέρα από τις προσφορές που θα μπορεί να έχει στο μενού του, να μπορεί να εμφανίζεται σαν διαφήμιση κατά την είσοδο του πελάτη στην εφαρμογή ή κατά το τέλος της παραγγελίας του. Άρα ο πελάτης που μπορεί να βρίσκεται σε κοντινή περιοχή με το μαγαζί που διαφημίζεται να έχει την δυνατότητα να ενημερωθεί για αυτό με ένα γρήγορο διαφημιστικό μήνυμα. Επιπλέον, μια ακόμη μελλοντική επέκταση που μπορεί να γνωρίσει η εφαρμογή, είναι η δημιουργία μιας κοινότητας. Με άλλα λόγια, ο κόσμος που χρησιμοποιεί την εφαρμογή θα μπορεί να επιβάλει κριτική σε ένα κατάστημα που πήγε, αφήνοντας ένα σχόλιο και like. Αυτό θα έχει σαν αποτέλεσμα να γνωρίζει κάποιος πριν επισκεφτεί ένα κατάστημα εαν αξίζει να πάει. Ακόμη, μέσα σε αυτή την κοινότητα κάθε μαγαζί θα έχει το δικό της προφίλ το οποίο θα ενημερώνει με φωτογραφίες, έτσι ώστε πριν ένας πελάτης επισκεφτεί το μαγαζί να έχει μια εικονική άποψη από το μενού και από το χώρο του καταστήματος. Με αυτό το τρόπο, ο επαγγελματίας μπορεί να κερδίσει έναν πελάτη πριν καν επισκεφτεί το κατάστημα. Τέλος μια ακόμη αναβάθμιση που μπορεί να γίνει είναι ως προς την επισκεψιμότητα που έχει το κάθε κατάστημα. Με άλλα λόγια, πριν ένας πελάτης επισκεφτεί ένα κατάστημα, να γνωρίζει την διαθεσιμότητα σε τραπέζια που έχει το κατάστημα που θέλει να πάει. Με αυτό το τρόπο γλιτώνει ταλαιπωρία αναμονής και μπορεί να κανονίσει την έξοδο και το μαγαζί που θα επισκεφτεί με αυτό το τρόπο.

Τέλος, πρόκειται για μια ιδέα που θέλω να την κυνηγήσω μελλοντικά και να προσπαθήσω να την περάσω στην καθημερινότητα του κόσμου ώστε να γίνει ο νέος τρόπος παραγγελίας σε καταστήματα εστίασης.

Βιβλιογραφία

- [1] A. Biessek, Flutter for Beginners: An introductory guide to building cross-platform mobile applications with Flutter and Dart 2. UK: Packt Publishing, 2019
- [2] Boyles, A. The Complete Guide to QR Codes. QR-Codes.com, 2012
- [3] Clow, M. ,Learn Google Flutter Fast: 65 Example Apps. Independently published, 2019
- [4] Prajyot M. , Salvatore G. , Google Flutter Mobile Development Quick Start Guide: Get up and running with IOS and Android mobile app development. Packt Publishing, 2019
- [5] Miola, A. , Flutter Complete Reference: Create beautiful, fast and native apps for any device. Independently published, 2020
- [6] Napoli, M. L. , Beginning Flutter: A Hands On Guide to App Development (1st ed.). Indianapolis: John Wiley & Sons, 201
- [7] Payne, R. , Beginning App Development with Flutter: Create Cross-Platform Mobile Apps (1st ed.). USA:Apress, 2019
- [8] Windmill, E, Flutter in Action (1st ed.). Manning Publications, 2019
- [9] “Firebase Products,” Google. [Online]. Available: <https://firebase.google.com/products-build>. [Accessed: 30-Oct-2020].
- [10] Yahiaoui, H. , Firebase Cookbook: Over 70 recipes to help you create real-time web and mobile applications with Firebase. Packt Publishing, 2017
- [11] Zaccagnino, C. , Programming Flutter: Native, Cross-Platform Apps the Easy Way (The Pragmatic Programmers) (1st ed.). Pragmatic Bookshelf, 2020
- [12] “Add Firebase to your Android project,” Google.com. [Online]. Available: <https://firebase.google.com/docs/android/setup>. [Accessed: 30-Oct-2020].
- [13] “Flutter - Beautiful native apps in record time,” Flutter.dev. [Online]. Available: <https://flutter.dev/>. [Accessed: 30-Oct-2020].
- [14] Devathon, “10 best Android frameworks for app development in 2020,” Medium, 20-Feb-2020. [Online]. Available: https://medium.com/@devathon_/10-best-android-frameworks-for-app-development-in-2020-98f5afb300e9. [Accessed: 30-Oct-2020].
- [15] “Dart programming language,” Dart.dev. [Online]. Available: <https://dart.dev/>. [Accessed: 30-Oct-2020].
- [16] “QR Code Basics,” Qr-code-generator.com. [Online]. Available: <https://www.qr-code-generator.com/qr-code-marketing/qr-codes-basics/>. [Accessed: 30-Oct-2020].
- [17] “History of QR Code,” Qrcode.com. [Online]. Available: <https://www.qrcode.com/en/history/>. [Accessed: 30-Oct-2020].