

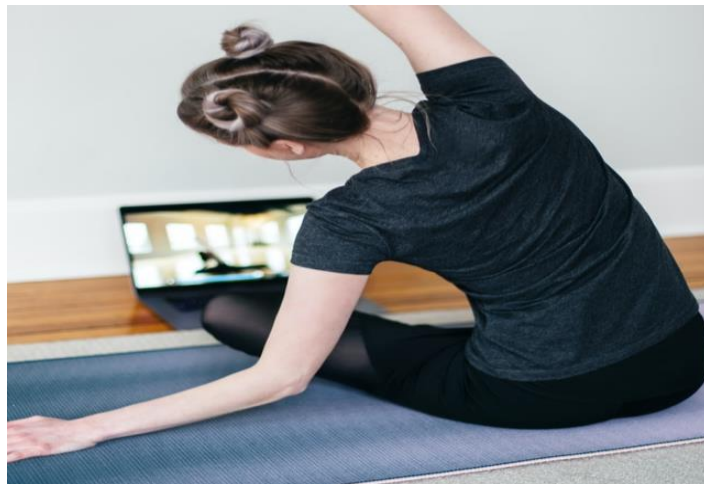


ΔΙΕΘΝΕΣ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Σχεδιασμός και Ανάπτυξη Android  
Εφαρμογής για την Ευεξία»



Της φοιτήτριας  
Κωνσταντίνα Μαραντίδου  
Αρ. Μητρώου: 144339

Επιβλέπων  
Περικλής Χατζημίσιος  
Βαθμίδα: Καθηγητής

Φεβρουάριος 2022

Τίτλος Π.Ε. Σχεδιασμός και ανάπτυξη Android εφαρμογής για την ευεξία

Κωδικός Π.Ε. 20165

Όνοματεπώνυμο φοιτήτριας Κωνσταντίνα Μαραντίδου

Όνοματεπώνυμο εισηγητή Περικλής Χατζημίσιος

Ημερομηνία ανάληψης Π.Ε. 28/04/2020

Ημερομηνία περάτωσης Π.Ε. 06/02/2022

*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 της φοιτήτριας Κωνσταντίνας Μαραντίδου που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, η 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 της 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.

*«Αφιερώνω την παρούσα πτυχιακή εργασία στην αδερφή μου, Μαρία»*



## Πρόλογος

Η παρούσα πτυχιακή εργασία σαν ιδέα ξεκίνησε όταν βρισκόμουν στο 2<sup>ο</sup> έτος της σχολής, δεν είχα καθόλου διάθεση για γυμναστική και είχα μάθει πολύ λίγα πράγματα για τον προγραμματισμό. Με τα χρόνια η ιδέα μου εξελίσσονταν μέχρι που στο 6<sup>ο</sup> έτος βρέθηκα στο γραφείο του κ. Χατζημίσιου με την ιδέα μου ζωγραφισμένη σε μια κόλλα Α4 και πολλή όρεξη να κάνω την ιδέα μου πτυχιακή!

Ο καθηγητής δέχτηκε να αναλάβει την πτυχιακή μου και με παρότρυνε να ξεκινήσω να ψάχνω πληροφορίες σχετικά με τον προγραμματισμό σε Android και να καταγράψω τις δυνατότητές της, πριν ακόμη την αναλάβω επίσημα.

Αρχικά μιλούσαμε για μια εφαρμογή για γυμναστήρια, όχι για έναν μεμονωμένο χρήστη, όπως είναι τώρα. Έπεσε η πανδημία, έκλεισαν τα πάντα και σε μία συνάντησή μας αποφασίσαμε να αλλάξουμε την ιδέα και να κάνουμε έρευνα για εφαρμογές ευεξίας προκειμένου να φτιάξουμε κάτι που ακόμη και σε μία τόσο δύσκολη χρονική στιγμή θα παρέμενε χρήσιμο.

Και κάπως έτσι δημιουργήθηκε η Unicorn. Με τις δυνατότητες που έχει αυτή τη στιγμή καλύπτει τις ανάγκες ενός μέσου χρήστη και αυτό είναι μόνο η αρχή! Στο κείμενο της πτυχιακής μπορεί κανείς να βρει κάποια από τα χαρακτηριστικά που θα έχει στο μέλλον η εφαρμογή.

## Περίληψη

Η εφαρμογή “Unikorn” δημιουργήθηκε τόσο για να παροτρύνει/ενθαρρύνει τους ανθρώπους να βάλουν τη γυμναστική στη ζωή τους, όσο και για να βοηθήσει/οργανώσει τα άτομα που έχουν εντάξει τη γυμναστική στην καθημερινότητά τους και χρειάζονται κάποιου είδους οργάνωση/καταγραφή των προπονήσεών τους.

Η εφαρμογή παρέχει στον χρήστη τη δυνατότητα επιλογής μέσα από μία μεγάλη ποικιλία διαφορετικών ειδών προπονήσεων. Πιο συγκεκριμένα, μπορεί είτε να τις παρακολουθήσει εκείνη την χρονική στιγμή είτε να τις τοποθετήσει στο προσωπικό του πρόγραμμα προπονήσεων για να τις παρακολουθήσει αργότερα. Το εβδομαδιαίο πρόγραμμα του χρήστη καθορίζεται από τον ίδιο και αποθηκεύεται προκειμένου να μην χάνει χρόνο στην αναζήτηση μια προπόνησης που του αρέσει.

Επιπλέον, κάνοντας αξιολόγηση των βίντεο που παρακολουθεί, ο χρήστης αποκτά πρόσβαση σε μία επιπλέον λειτουργία της εφαρμογής που του προσφέρει - έπειτα από κατηγοριοποίησή του με άλλους χρήστες που φαίνεται να έχουν παρόμοιο προφίλ με βάση τις αξιολογήσεις τους - προτάσεις προπονήσεων. Αυτό επιτυγχάνεται με έναν αλγόριθμο τεχνητής νοημοσύνης, χρησιμοποιώντας κατηγοριοποίηση, με μέτρο την ευκλείδεια απόσταση. Συνεπώς, ο χρήστης δεν είναι απαραίτητο να σκεφτεί τι θα του αρέσει και να ταλαιπωρηθεί ψάχνοντας ανάμεσα σε τόσα βίντεο αν δεν έχει κάτι συγκεκριμένο στο μυαλό του.

Επιπλέον, παρέχεται καταγραφή ιστορικού, με τη δυνατότητα καταγραφής και δραστηριοτήτων που πραγματοποιήθηκαν εκτός εφαρμογής, με στόχο την καλύτερη οργάνωση και προγραμματισμό των προπονήσεων του χρήστη εντός και εκτός της “Unikorn”.

Τέλος, ο χρήστης έχει πρόσβαση σε προκλήσεις που ανεβαίνουν από τον διαχειριστή και τη δυνατότητα να κάνει θετική ή αρνητική αναφορά γενικά για την εφαρμογή. Με αυτό τον τρόπο η “Unikorn” αποτελεί μία συνεχώς εξελισσόμενη εφαρμογή με κέντρο της τον χρήστη και τις ανάγκες του.

# «Design and Development of an Android Application for Well-Being»

«Konstantina Marantidou»

## **Abstract**

Unikorn application was created to help both people that don't feel like training and people that they love training and they need an application to organize/manage their workouts. The most important ability of the app is the provision of a variety of training videos.

If the user wants to have a workout of a specific category, for example, yoga, he/she can search it on a search box and find a lot of workout videos. When the user will choose the video that he/she wants he/she will be able to do it at that moment or add it to his program to do it in another moment. In the case the user decides to put it on his/her program he/she will take advantage of the ability to have a personal organized schedule in his/her mobile phone without having to lose time to find a workout he/she likes in a day that he/she is pressed.

On the other hand, if the user chooses to do the workout at that moment, he/she will be able to rate it and open the ability to have suggestions based on his/her profile. That happens with clustering and allows the user to not lose time neither energy to find what he/she likes and have a workout on a busy day.

Also, the user can find his history in which he/she can see the workouts he did with the app and add the workouts he/she did without the app, so he can organize and manage his/her workouts. To not lose his motivation, the administrator can give him challenges too. If the user wants to send feedback to the administrator he/she can do it with one click on the main menu and like this Unikorn app will be constantly evolving with the user at the center and his/her needs.

## Ευχαριστίες

Θα ήθελα να ευχαριστήσω πρώτα απ' όλα την οικογένειά μου που με στηρίζει όλα αυτά τα χρόνια. Κανένας τους δεν ήξερε τίποτα από πληροφορική όταν μπήκα στη σχολή αλλά βρίσκονταν συνέχεια στο πλάι μου, χωρίς να με πιέζουν και δείχνοντάς μου το πόσο πολύ πιστεύουν σε εμένα και στις ικανότητές μου. Με βοήθησαν να ξεπεράσω όλα τα εμπόδια και να μην εγκαταλείψω ούτε τη σχολή ούτε τις υπόλοιπες δραστηριότητες που παρακολουθούσα ενώ σπούδαζα.

Επίσης, θα ήθελα να ευχαριστήσω τον καθηγητή μου, κύριο Περικλή Χατζημίσιου, ο οποίος με στήριξε από πριν αναλάβει την πτυχιακή μου και σε όλες τις δυσκολίες που αντιμετώπισα τα τελευταία χρόνια των σπουδών μου ήταν δίπλα μου και με ενθάρρυνε να μην τα παρατήσω. Πίστεψε σε εμένα και στην ιδέα μου και με στήριξε για να την υλοποιήσω και τον ευχαριστώ πολύ για αυτό.

# Περιεχόμενα

Πρόλογος.....	v
Περίληψη.....	vi
Abstract .....	vii
Ευχαριστίες .....	viii
Περιεχόμενα .....	ix
Κατάλογος Εικόνων .....	xi
Κατάλογος Πινάκων.....	xiii
Κεφάλαιο 1 <sup>ο</sup> : Εισαγωγή .....	1
1.1 Το πρόβλημα .....	1
1.2 Η ιδέα .....	3
1.3 Σύγκριση με άλλες εφαρμογές .....	4
1.3.1 Εφαρμογές γυμναστικής.....	4
1.3.2 Άλλες σχετικές εφαρμογές .....	4
1.3.3 Σχετικές ιστοσελίδες .....	5
Κεφάλαιο 2 <sup>ο</sup> : Android .....	10
2.1 Εισαγωγή.....	10
2.2 Αρχιτεκτονική του Android .....	11
2.3 Βασικά χαρακτηριστικά .....	13
2.4 Στοιχεία μίας Android εφαρμογής.....	15
2.4.1 Activities .....	15
2.4.2 Fragments .....	16
2.4.3 Layouts .....	17
2.5 Εκδόσεις Android.....	18
Κεφάλαιο 3 <sup>ο</sup> : Προγράμματα & Εργαλεία που χρησιμοποιήθηκαν .....	22
3.1 Android Studio .....	22
3.1.1 Γενικά χαρακτηριστικά .....	22
3.1.2 Το περιβάλλον του Android Studio.....	22
3.2 Apache NetBeans .....	25
3.2.1 Γενικά χαρακτηριστικά .....	25
3.2.2 Το περιβάλλον του Apache NetBeans.....	25
3.3 Visual Studio Code.....	27
3.3.1 Γενικά χαρακτηριστικά .....	27

3.3.2	Το περιβάλλον του Visual Studio Code .....	27
3.4	Firebase .....	30
3.4.1	Γενικά χαρακτηριστικά .....	30
3.4.2	Η κονσόλα της Firebase (Firebase Console) .....	31
Κεφάλαιο 4 <sup>ο</sup> :	Υλοποίηση εφαρμογής.....	33
4.1	Εισαγωγή στην εφαρμογή .....	33
4.2	Το κεντρικό μενού.....	34
4.3	Οθόνη προγράμματος χρήστη (Schedule fragment).....	37
4.4	Οθόνη προτεινόμενων προπονήσεων (Training fragment) .....	39
4.5	Οθόνη προπονήσεων (AllVideos fragment).....	40
4.6	Οθόνη ιστορικού (History fragment) .....	42
4.7	DayX activity .....	47
4.8	Send Dialog .....	49
4.9	Rating activity .....	50
4.10	Scheduler activity .....	52
4.11	JSOUP .....	53
4.12	Clustering .....	55
4.13	Επίλογος.....	56
Κεφάλαιο 5ο:	Επίλογος.....	57
5.1	Η Unikorn σήμερα.....	57
5.2	Η Unikorn στο μέλλον.....	57
ΒΙΒΛΙΟΓΡΑΦΙΑ.....		60
ΠΑΡΑΡΤΗΜΑ Α: ΚΩΔΙΚΑΣ ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗΣ .....		63

## Κατάλογος Εικόνων

Εικόνα 1.1: Παράδειγμα εφαρμογής ευεξίας (LG Health).....	2
Εικόνα 1.2: Παράδειγμα εφαρμογής γυμναστικής (Nike Training Club).....	2
Εικόνα 1.3: Παράδειγμα δημιουργίας πλάνου προπονήσεων με συμπλήρωση ερωτηματολογίου (NTC) .....	6
Εικόνα 1.4: Παράδειγμα προπόνησης (PUMATRAC) .....	7
Εικόνα 1.5: Παράδειγμα δημιουργίας προφίλ με συμπλήρωση ερωτηματολογίου (Adidas Training)..	7
Εικόνα 1.6: Παράδειγμα μέτρησης βημάτων, μιλίων και θερμίδων(Βηματομετρητής) .....	8
Εικόνα 1.7: Παράδειγμα καταγραφής ιστορικού (Βηματομετρητής & θερμιδομετρητής).....	8
Εικόνα 1.8: Συνταγές για υγιεινή διατροφή – fitnessblender.....	9
Εικόνα 2.1: Αρχική οθόνη του Android 12 [30] .....	11
Εικόνα 2.2: Η αρχιτεκτονική του Android [7] .....	12
Εικόνα 2.3: Linux Kernel [7] .....	12
Εικόνα 2.4: Hardware Abstraction Layer (HAL) [7] .....	12
Εικόνα 2.5: Native C/C++ Libraries/Android Runtime [7].....	13
Εικόνα 2.6: Java API Framework [7].....	13
Εικόνα 2.7: System Apps [7].....	13
Εικόνα 2.8: Παράδειγμα οθονών στις οποίες προσαρμόζεται το Android [33].....	14
Εικόνα 2.9: Activity Lifecycle [31].....	16
Εικόνα 2.10: Fragment Lifecycle [32] .....	17
Εικόνα 2.11: Android versions [29] .....	20
Εικόνα 3.1: Το περιβάλλον του Android Studio .....	23
Εικόνα 3.2: Android Studio – Toolbar .....	23
Εικόνα 3.3: Android Studio - Navigation Bar.....	23
Εικόνα 3.4: Android Studio - Editor .....	24
Εικόνα 3.5: Android Studio – Tool Windows.....	24
Εικόνα 3.6: Android Studio - Status Bar.....	24
Εικόνα 3.7: Το περιβάλλον του Apache NetBeans .....	26
Εικόνα 3.8: Apache NetBeans- Toolbar.....	26
Εικόνα 3.9: Apache NetBeans – Editor.....	26
Εικόνα 3.10: Apache NetBeans – Tool Windows.....	27
Εικόνα 3.11: Apache NetBeans – Status Bar .....	27
Εικόνα 3.12: Το περιβάλλον του Visual Studio Code.....	28
Εικόνα 3.13: Visual Studio Code – Toolbar.....	29
Εικόνα 3.14: Visual Studio Code – Navigation Bar.....	29
Εικόνα 3.15: Visual Studio Code – Editor .....	29
Εικόνα 3.16: Visual Studio Code – Tool Windows .....	30
Εικόνα 3.17: Visual Studio Code – Status Bar.....	30
Εικόνα 3.18: Firebase Console.....	31
Εικόνα 3.19: Firebase - Realtime Database.....	32
Εικόνα 3.20: Firebase - Realtime Database - Rules .....	32
Εικόνα 3.21: Firebase - Authentication.....	32
Εικόνα 4.1: Αρχική οθόνη [25][26][27].....	34
Εικόνα 4.2: Κεντρικό μενού.....	35

Εικόνα 4.3: Απόσπασμα του “drawer_menu.xml” .....	36
Εικόνα 4.4: Απόσπασμα της “MainActivity.java” – Κεντρικό μενού .....	37
Εικόνα 4.5: Η οθόνη του “My Schedule” .....	37
Εικόνα 4.6: “ScheduleFragment” – Κουμπιά ημέρας .....	38
Εικόνα 4.7: Firebase – Scheduler .....	38
Εικόνα 4.8: Ένα παράδειγμα Challenge .....	39
Εικόνα 4.9: Η οθόνη του “More training” .....	40
Εικόνα 4.10: Λούπα προτάσεων - More training .....	40
Εικόνα 4.11: Η οθόνη του “All videos” .....	41
Εικόνα 4.12: “AllVideosFragment” - HashMap .....	42
Εικόνα 4.13: “AllVideosFragment” – Search option .....	42
Εικόνα 4.14: Η οθόνη του “History” .....	43
Εικόνα 4.15: Η οθόνη του “History” αφού πατηθεί κουμπί χρονολογίας .....	44
Εικόνα 4.16: Η οθόνη του “History” αφού πατηθεί κουμπί μήνα .....	44
Εικόνα 4.17: Η οθόνη του “History” αφού πατηθεί κουμπί ημέρας .....	45
Εικόνα 4.18: Η οθόνη του “History” αφού πατηθεί κουμπί εισαγωγής δραστηριότητας .....	45
Εικόνα 4.19: Μήνυμα λάθους εισαγωγής στοιχείων .....	46
Εικόνα 4.20: “HistoryFragment” – Εμφάνιση δραστηριοτήτων .....	46
Εικόνα 4.21: “HistoryFragment” – Προσθήκη δραστηριότητας .....	47
Εικόνα 4.22: Η οθόνη του “DayX activity” .....	48
Εικόνα 4.23: DayX activity – buttons .....	49
Εικόνα 4.24: Send Dialog .....	49
Εικόνα 4.25: Send Dialog Code .....	50
Εικόνα 4.26: Οθόνη αξιολόγησης βίντεο [5][28] .....	50
Εικόνα 4.27: Rating activity .....	51
Εικόνα 4.28: Firebase – ratings .....	52
Εικόνα 4.29: Οθόνη κατανομής βίντεο σε ημέρες .....	52
Εικόνα 4.30: Scheduler Activity .....	53
Εικόνα 4.31: Στιγμιότυπο από τον κώδικα που φέρνει τα δεδομένα από τη ιστοσελίδα .....	53
Εικόνα 4.32: Στιγμιότυπο της κλάσης VideoData .....	54
Εικόνα 4.33: Αποθήκευση δεδομένων στη βάση (Program.java) .....	54
Εικόνα 4.34: Στιγμιότυπο από τον κώδικα κατηγοριοποίησης .....	55
Εικόνα 4.35: Αντίγραφο πίνακα αξιολογήσεων χρηστών .....	56

## **Κατάλογος Πινάκων**

Πίνακας 2.1: Εκδόσεις Android [11].....	20
---	----



# Κεφάλαιο 1<sup>ο</sup>: Εισαγωγή

## 1.1 Το πρόβλημα

Τα τελευταία χρόνια παρατηρείται ότι όλο και περισσότεροι άνθρωποι απομακρύνονται από τα γυμναστήρια και προτιμούν εναλλακτικές μεθόδους γυμναστικής, είτε πιο γρήγορες είτε πιο ιδιαίτερες. Είναι ευρέως γνωστό ότι στη σημερινή κοινωνία ο μέσος άνθρωπος δεν έχει πάρα πολύ χρόνο και αυτό έχει σαν αποτέλεσμα το να ψάχνει τρόπους να αθλείται για λίγη ώρα και αυτό αν έχει την ενέργεια και τη διάθεση για να το κάνει. Επίσης, πολλοί άνθρωποι νιώθουν κουρασμένοι τα απογεύματα και ότι εάν πάνε π.χ. στο γυμναστήριο ή για τρέξιμο θα χάσουν χρόνο τον οποίο μπορούν να αξιοποιήσουν για να ξεκουραστούν. Αυτή η κατηγορία ανθρώπων είναι πιο πιθανό να επιλέξει έναν ιδιαίτερο-εναλλακτικό τρόπο γυμναστικής, όπως “aqua aerobic”, “tae bo”, κλπ.

Αξίζει να σημειωθεί ότι πολλές φορές ο κόσμος υποτιμά τη σημασία της γυμναστικής και αυτός είναι άλλος ένας λόγος για τον οποίο μπορεί να την αποφεύγει τακτικά. Η αλήθεια, όμως, είναι ότι η γυμναστική είναι πολλή σημαντική για την υγεία των ανθρώπων και για αυτό θα έπρεπε να αποτελεί αναπόσπαστο κομμάτι της καθημερινής ρουτίνας τους.

Τον 21ο αιώνα έχουν δημιουργηθεί πλέον πάρα πολλοί τρόποι γυμναστικής και γίνεται συνεχώς παρότρυνση στους χρήστες να δοκιμάσουν εναλλακτικές προπονήσεις και να τις εντάξουν στο πρόγραμμά τους. Βασικός εχθρός αυτών των προγραμμάτων είναι ο χρόνος των ανθρώπων αλλά και η έλλειψη ενδιαφέροντος μετά από μία χρονική στιγμή που τις περισσότερες φορές δημιουργείται λόγω του ότι ο μέσος άνθρωπος πριν εντάξει στο πρόγραμμά του τη γυμναστική θέτει μη ρεαλιστικούς στόχους και όταν παρατηρήσει ότι δεν ικανοποιούνται στο χρονικό διάστημα που περίμενε απογοητεύεται και τα παρατάει.

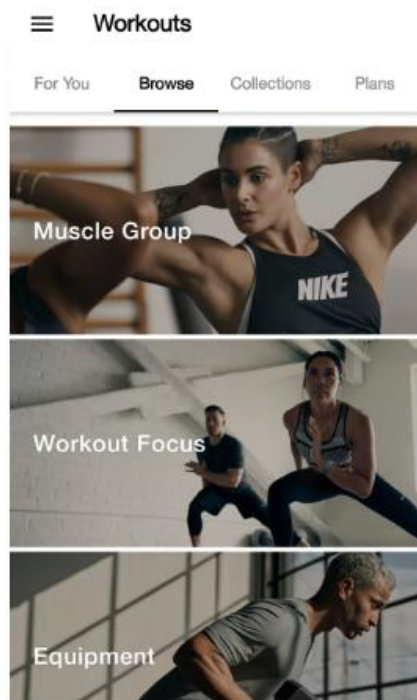
Επιπλέον, τα μέσα κοινωνικής δικτύωσης ασκούν τεράστια επιρροή στους ανθρώπους και συνηθίζουν να προβάλλουν πρότυπα με τα οποία είναι δύσκολο να ταυτιστεί ο μέσος άνθρωπος. Πολλές φορές τα πρότυπα-είδωλα αποτελούν πέρα από διάσημα πρόσωπα, άτομα τα οποία καταναλώνοντας απαγορευμένες ουσίες (αναβολικά) ή/και ακολουθώντας διατροφικά προγράμματα τα οποία βλάπτουν την υγεία καταφέρνουν να κάνουν τους ανθρώπους να νιώθουν μειονεκτικά για το σώμα τους και τις συνήθειές τους, να θέτουν μη ρεαλιστικούς χρόνους, να απογοητεύονται και σε ένα σύντομο χρονικό διάστημα να τα παρατάνε.

Η παραπάνω συνθήκη, έχει ως αποτέλεσμα και το να υπάρχουν πολλοί άνθρωποι με προβλήματα αυτοεκτίμησης ή/και θέματα υγείας τα οποία προκλήθηκαν παίρνοντας σαν σωστό ένα λανθασμένο πρότυπο υγιεινού τρόπου ζωής. Για τους παραπάνω λόγους θα ήταν ωφέλιμο το φιλτράρισμα των ατόμων που παρακολουθούμε στα κοινωνικά δίκτυα καθώς και η επικοινωνία μας με τον προσωπικό γιατρό/διατροφολόγο μας πριν από τη λήψη μιας απόφασης που θα αλλάξει τις καθημερινές διατροφικές μας συνήθειες αλλά και τον τρόπο ζωής μας.

Τέλος, τα τελευταία χρόνια έχει χειροτερέψει η κατάσταση λόγω του Covid-19 και της καραντίνας, η οποία στέρησε τη δυνατότητα πρόσβασης σε χώρους άθλησης ακόμη και στους ανθρώπους που είχαν εντάξει τη γυμναστική στην καθημερινή ρουτίνα τους. Αυτό, σε συνδυασμό και με το ότι ο κόσμος ήταν κλεισμένος σπίτι του οπότε είχε και περισσότερο χρόνο, οδήγησε ένα μεγάλο ποσοστό ανθρώπων στο να εγκαταστήσουν στα κινητά τους και να χρησιμοποιήσουν εφαρμογές υγιεινού τρόπου ζωής όπως εφαρμογές με συμβουλές ευεξίας και εφαρμογές γυμναστικής.



Εικόνα 1.1: Παράδειγμα εφαρμογής ευεξίας (LG Health)



Εικόνα 1.2: Παράδειγμα εφαρμογής γυμναστικής (Nike Training Club)

## 1.2 Η ιδέα

Όπως αναφέρθηκε στην προηγούμενη ενότητα, είναι σημαντικό οι άνθρωποι να εντάξουν τη γυμναστική στην καθημερινή τους ρουτίνα αλλά αυτό δεν είναι πολύ εύκολο δεδομένου των προσωπικών προβλημάτων του κάθε ατόμου με πιο συχνά αναφερόμενα την έλλειψη χρόνου ή/και κινήτρου. Αυτό το πρόβλημα ήρθε να λύσει η παρούσα πτυχιακή εργασία.

Επίσης είναι πολύ σημαντικό να αλλάξει ο τρόπος με τον οποίο οι άνθρωποι βλέπουν και επηρεάζονται από λογαριασμούς στα μέσα κοινωνικής δικτύωσης και να αποκτήσουν περισσότερη αυτοπεποίθηση αρχικά βάζοντας μικρούς στόχους τους οποίους μπορούν να καταφέρουν χωρίς να διαφύγουν σε ακραίους (και βλαβερούς) τρόπους συμπεριφοράς. Σε αυτό θα μπορούσε να βοηθήσει η δημιουργία μιας εφαρμογής εστιασμένη προς τον χρήστη, η οποία θα είναι σε θέση να τον παρακινεί να αποκτήσει έναν πιο υγιεινό τρόπο ζωής θέτοντας ρεαλιστικούς στόχους και σεβόμενος τις ανάγκες και τις δυνατότητές του.

Η εφαρμογή Unikorn, η οποία αναπτύχθηκε στα πλαίσια της παρούσης πτυχιακής εργασίας, εστιάζει στο να παρακινήσει τον χρήστη μειώνοντας τον χρόνο που χρειάζεται για να γυμναστεί και δίνοντάς του τη δυνατότητα της εναλλακτικής γυμναστικής (όπως “yoga”) την ώρα και τη στιγμή που ο ίδιος θέλει. Ταυτόχρονα, με τις προκλήσεις επιτυγχάνεται να υπάρχει ένα κίνητρο στο να χρησιμοποιήσει ο χρήστης μία μέρα την εφαρμογή και να πατήσει. έπειτα από το “challenge”. το “more training” όπου θα δει προγράμματα που σύμφωνα με έναν κώδικα ευφυΐας ταιριάζουν με το προφίλ του, άρα υπάρχουν πολλές πιθανότητες να του αρέσουν και να τον παρακινήσουν.

Επίσης, το ότι μπορεί να φτιάξει το δικό του προσωπικό εβδομαδιαίο πρόγραμμα γυμναστικής από τη μία συνεισφέρει στο να έχει κάθε μέρα στο πρόγραμμά του βίντεο που του αρέσουν (τα έχει αποθηκευμένα) και από την άλλη να μην σπάει τη ρουτίνα του και να νιώθει υπεύθυνος για την τήρηση του προγράμματος που ο ίδιος καθόρισε για τον εαυτό του. Το αίσθημα υπευθυνότητας και το ότι όσο περνάει ο χρόνος παρατηρεί ότι όντως τηρεί το όποιο πρόγραμμα βάζει ο ίδιος στον εαυτό του, είναι αυτό που θα τον οδηγήσει στο να κάνει κι άλλα πράγματα ακόμα και εκτός εφαρμογής και να μην χάσει τη διάθεσή του για έναν πιο υγιεινό τρόπο ζωής.

Τέλος, το ότι στο ιστορικό μπορεί πέραν του να παρατηρήσει τις προπονήσεις του, με αποτέλεσμα να είναι πιο οργανωμένος, να προσθέσει και δραστηριότητες που πραγματοποίησε εκτός εφαρμογής, μπορεί να προσελκύσει και άτομα της κατηγορίας των ανθρώπων που ήδη έχουν εντάξει τη γυμναστική στη ζωή τους και χρειάζονται ένα πλάνο προπόνησης. Για παράδειγμα, ένας αθλητής “powerlifting” χρειάζεται να έχει και να τηρεί ένα εβδομαδιαίο πλάνο προπονήσεων. Παρ’ όλο που δεν χρειάζεται προτάσεις προπονήσεων χρειάζεται την καταγραφή της καθημερινής του ασχολίας με το άθλημα προκειμένου να είναι σε θέση να φτιάξει το πλάνο προπονήσεών του για την επόμενη εβδομάδα αλλά και να παρατηρήσει την εξέλιξή του στο άθλημα (π.χ. ότι πριν από 2 μέρες έκανε άρση θανάτου με 95 κιλά και σήμερα με 100).

Συνεπώς, η γενική ιδέα της εφαρμογής είναι τόσο το να προσελκύσει σε έναν πιο υγιεινό τρόπο ζωής άτομα που για οποιονδήποτε προσωπικό λόγο δεν τα καταφέρνουν όσο και να βοηθήσει κατά κάποιο τρόπο τα άτομα που ήδη ασχολούνται με τη γυμναστική και ζητούν οργάνωση και καταγραφή του ιστορικού τους.

### 1.3 Σύγκριση με άλλες εφαρμογές

#### 1.3.1 Εφαρμογές γυμναστικής

Την τελευταία δεκαετία οι εφαρμογές ευεξίας και οι εφαρμογές για γυμναστική από το σπίτι έχουν αυξηθεί σημαντικά, με αποτέλεσμα ο ανταγωνισμός μεταξύ τους να είναι μεγάλος. Κάποιες από τις εφαρμογές που υπάρχουν αυτή τη στιγμή στην αγορά και θα μελετήσουμε παρακάτω είναι οι εξής: NTC (Nike Training Club), PUMATRAC, Adidas Training by Runtastic.

(α) Στην εφαρμογή Nike Training Club (NTC) αρχικά ο χρήστης φτιάχνει το προφίλ του προκειμένου να έχει προτάσεις για γυμναστική (από την εφαρμογή). Έπειτα, αν θέλει μπορεί να επιλέξει μία από τις έτοιμες προτάσεις (ανάλογα με το είδος γυμναστική που προτιμά για εκείνη την ημέρα) ή να επιλέξει τη δημιουργία ενός πλάνου πιο εξειδικευμένο με βάση τις ανάγκες και τους στόχους του. Εάν έχει δημιουργήσει το δικό του πλάνο μπορεί να δει πως τα πηγαίνει εβδομαδιαία, η εφαρμογή του πετάει ειδοποίηση κάθε φορά που είναι η μέρα που έχει προπόνηση, μπορεί να δει τις ασκήσεις σε βίντεο και στο τέλος κάθε προπόνησης δίνει στην εφαρμογή πληροφορίες σχετικά με το πόσο δυσκολεύτηκε να πραγματοποιήσει την προπόνηση.

Επίσης, κατά την διάρκεια της προπόνησης παρέχεται στον χρήστη η δυνατότητα να ακούει μουσική από το κινητό του. Επιπλέον, παρέχεται στον χρήστη η δυνατότητα να δει προτάσεις προπονήσεων με βάση τις πληροφορίες που έχει δώσει κατά τη δημιουργία του προφίλ του, να δει αποθηκευμένα βίντεο καθώς και να παρακολουθήσει το ιστορικό των προπονήσεών του.

(β) Η εφαρμογή “PUMATRAC” είναι μια εφαρμογή αντίστοιχη της “NTC”, στην οποία ο χρήστης δημιουργεί το προφίλ του και θέτει στόχους. Έπειτα η εφαρμογή του εμφανίζει προτεινόμενες προπονήσεις με βάση το προφίλ του και παρέχεται στον χρήστη η δυνατότητα αν γνωρίζει τι προπόνηση προτιμά για την παρούσα χρονική στιγμή, να ψάξει μέσω του “search box” και να του εμφανιστούν προπονήσεις με βάση την αναζήτησή του.

Οι ασκήσεις στις προπονήσεις γίνονται με παρακολούθηση βίντεο και καταγράφονται σε ένα ημερολόγιο προπονήσεων. Επίσης, αν ο χρήστης θέλει να πάει για τρέξιμο, το επιλέγει στην εφαρμογή και έτσι μπορεί να έχει χρονόμετρο, μουσική και αναγράφεται στο εβδομαδιαίο πρόγραμμά του.

(γ) Η εφαρμογή “Adidas Training by Runtastic” είναι αντίστοιχη των 2 παραπάνω εφαρμογών. Ο χρήστης δημιουργεί το προφίλ του και θέτει κάποιους στόχους. Στη συνέχεια η εφαρμογή εμφανίζει προτάσεις για προγράμματα ανάλογα με το προφίλ του χρήστη. Όπως και στην “NTC”, ο χρήστης μπορεί να δημιουργήσει ένα πλάνο προπονήσεων αντίστοιχα με τις απαιτήσεις και τις ανάγκες του. Αφού διαλέξει ένα από τα προτεινόμενα προγράμματα προπονήσεων, μπορεί να παρακολουθήσει τις ασκήσεις σε βίντεο. Κρατείται ένα αντίστοιχο ιστορικό (και πρόγραμμα) προπονήσεων. Κάτι διαφορετικό από τις 2 παραπάνω εφαρμογές είναι οι προκλήσεις που περιέχει, με στόχο τη διαρκή παρακίνηση του χρήστη.

#### 1.3.2 Άλλες σχετικές εφαρμογές

Επίσης, υπάρχουν πολλές εφαρμογές με βηματομετρητές ή/και θερμοδομετρητές, που αν και δεν έχουν κοινές δυνατότητες με τη Unikorn αποτελούν και αυτές εφαρμογές με στόχο την καλύτερη ποιότητα ζωής των χρηστών τους και δεν θα μπορούσαν να λείπουν από αυτή την ενότητα. Κάποιες από αυτές τις εφαρμογές, στις οποίες θα αναφερθούμε παρακάτω είναι οι “Βηματομετρητής” και “Βηματομετρητής & θερμοδομετρητής χωρίς χρέωση”.

(α) Ο “Βηματομετρητής” είναι μια εφαρμογή που μετράει βήματα, θερμίδες, λεπτά (που βρίσκεται ο χρήστης εν κινήσει) και μίλια. Επιπλέον, κρατάει ιστορικό για να μπορεί να δει ο χρήστης την εβδομαδιαία και μηνιαία επίδοσή του. Κάτι εντυπωσιακό είναι ότι Ο αριθμός των βημάτων του χρήστη εμφανίζεται στην αρχική του οθόνη ακόμη και όταν δεν έχει ανοιχτή την εφαρμογή. Έτσι έχει καλύτερο έλεγχο της φυσικής του κατάστασης.

(β) Ο “Βηματομετρητής & θερμιδομετρητής χωρίς χρέωση”, είναι μία εφαρμογή αντίστοιχη του “Βηματομετρητή”. Μετράει θερμίδες, βήματα και μίλια και για να το επιτύχει αυτό, έχει σύνδεση με gps. Δίνεται στο χρήστη η δυνατότητα να θέσει στόχους και να παρακολουθεί την πορεία του στο να τους πετύχει (π.χ. 2986 βήματα). Επίσης, μπορεί κανείς να παρακολουθήσει το εβδομαδιαίο και μηνιαίο ιστορικό του, όπως και στις προηγούμενες εφαρμογές που αναφέρθηκαν. Τέλος, όπως και στην άλλη εφαρμογή, τα βήματά σου εμφανίζονται και στην οθόνη του κινητού.

### 1.3.3 Σχετικές ιστοσελίδες

Υπάρχουν και κάποιες ιστοσελίδες με δυνατότητες παρόμοιες με την εφαρμογή μας. Αξίζει να σημειωθεί ότι χρησιμοποιήθηκε μία από αυτές προκειμένου να αποσπαστούν τα βίντεο και να έχουμε κάποια χαρακτηριστικά τους προκειμένου να τα προτείνουμε στους χρήστες ανάλογα με τις ανάγκες του. Αυτή η ιστοσελίδα ονομάζεται “fitnessblender.com”.

Η ιστοσελίδα “fitnessblender” παρέχει στο χρήστη τη δυνατότητα παρακολούθησης βίντεο προσαρμοσμένα στις ανάγκες του. Ο χρήστης δεν απαιτείται να δημιουργήσει προφίλ, μόνο στην περίπτωση που ζητήσει κάτι για το οποίο πρέπει να πληρώσει, όπως για παράδειγμα ένα πρόγραμμα διατροφής. Τα βίντεο προπονήσεων κατατάσσονται σε διαφορετικές κατηγορίες ανάλογα με τον βαθμό δυσκολίας τους, πόσες θερμίδες καίει ο χρήστης εκτελώντας τα, πόση ώρα διαρκούν και αν χρειάζονται ή όχι εξοπλισμό.

Επίσης, στην ιστοσελίδα μπορεί κανείς να βρει συνταγές και άρθρα ειδικών σχετικά με την απόκτηση μιας καλής ποιότητας ζωής. Οι χρήστες έχουν τη δυνατότητα να συμμετέχουν σε δημόσιες συζητήσεις και εάν το επιθυμούν, να εγγραφούν και να πληρώσουν για να αποκτήσουν επιπλέον παροχές, όπως προγράμματα προπονήσεων, προγράμματα διατροφής, κλπ.

Επομένως, κάποια κοινά της εφαρμογής Unicorn με άλλες εφαρμογές της ίδιας κατηγορίας είναι:

- Δυνατότητα χρήσης της εφαρμογής μόνο από αυθεντικοποιημένους χρήστες.
- Παροχή ενός συνόλου προπονήσεων.
- Παροχή προτάσεων προπονήσεων με βάση το προφίλ του χρήστη
- Καταγραφή ιστορικού προπονήσεων.
- Προκλήσεις.

Κάποιες δυνατότητες που κάνουν τη Unicorn να ξεχωρίζει είναι οι:

- Δυνατότητα καταγραφής προπονήσεων και δραστηριοτήτων που εκτέλεσε ο χρήστης εκτός εφαρμογής.
- Δημιουργία του προφίλ του χρήστη με βάση την αξιολόγηση που πραγματοποιεί σε προηγούμενες προπονήσεις που έχει παρακολουθήσει και όχι με βάση το προφίλ που έχει ο ίδιος δημιουργήσει (οπότε είναι πιο ενημερωμένο).
- Δυνατότητα απ’ ευθείας επικοινωνίας με τον διαχειριστή της εφαρμογής σε περίπτωση που ο χρήστης θέλει κάτι να βελτιωθεί/αλλάξει.

- Δυνατότητα ο χρήστης να φτιάξει μόνος του το εβδομαδιαίο πλάνο προπονήσεών του (Δεν παρέχονται έτοιμα πλάνα εβδομαδιαίων προπονήσεων όπως σε άλλες εφαρμογές)

Με αυτές τις δυνατότητες η *Unikorn* αποτελεί μία εφαρμογή χρήσιμη και για τους ανθρώπους που λατρεύουν τη γυμναστική αλλά και για τα άτομα που για ξεχωριστούς λόγους το καθένα, δεν καταφέρνουν να ασχοληθούν με αυτήν. Τα άτομα που λατρεύουν τη γυμναστική είναι αυτά που θα προτιμήσουν την εφαρμογή για να έχουν ένα οργανωμένο πλάνο προπονήσεων. Από την άλλη μεριά τα άτομα που δεν καταφέρνουν να εντάξουν τη γυμναστική στη ζωή τους θα επιλέξουν τη χρήση της συγκεκριμένης εφαρμογής λόγω του της ταχύτητας με την οποία μπορούν να βρουν μία προπόνηση που να ταιριάζει στο προφίλ τους και της οργάνωσης που τους παρέχει.

← **Plan setup**

---

Everything you tell us here helps your plan be right for you.

**1 EQUIPMENT AVAILABLE**

We'll build your plan around equipment you typically have.

---

No Equipment

---

Select Equipment

---

**2 WORKOUTS PER WEEK**

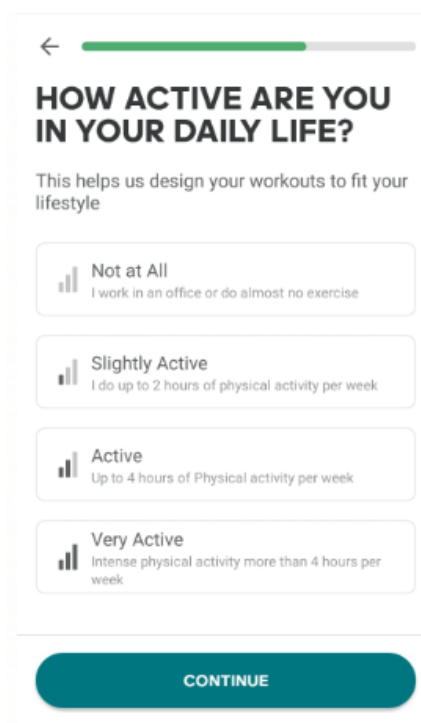
**3 INCLUDE RUNNING**

**4 YOUR ACTIVITY LEVEL**

Εικόνα 1.3: Παράδειγμα δημιουργίας πλάνου προπονήσεων με συμπλήρωση ερωτηματολογίου (NTC)



Εικόνα 1.4: Παράδειγμα προπόνησης (PUMATRAC)



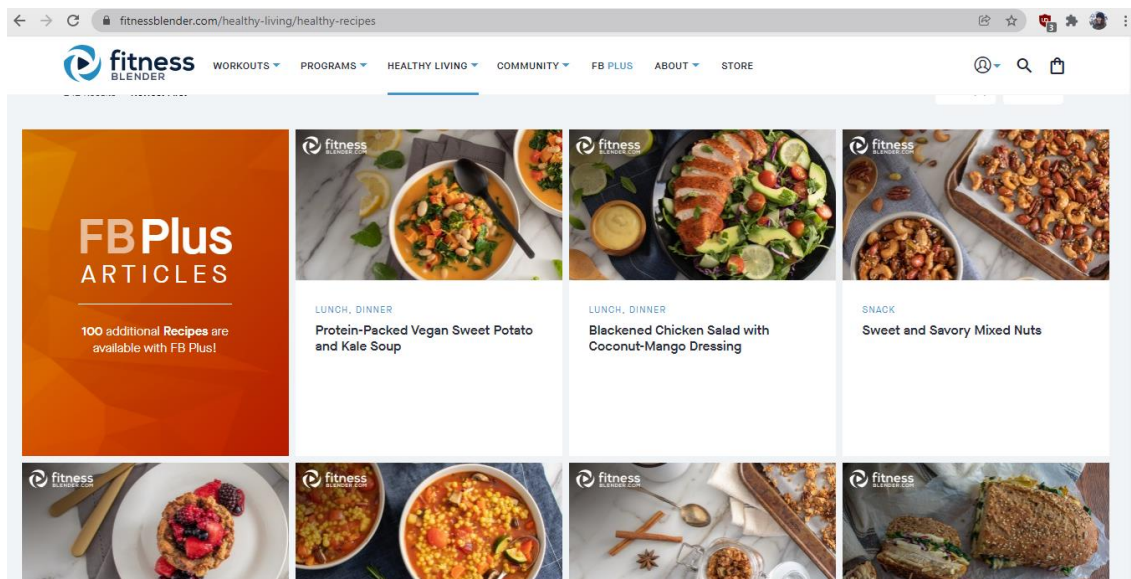
Εικόνα 1.5: Παράδειγμα δημιουργίας προφίλ με συμπλήρωση ερωτηματολογίου (Adidas Training)



Εικόνα 1.6: Παράδειγμα μέτρησης βημάτων, μιλίων και θερμίδων(Βηματομετρητής)



Εικόνα 1.7: Παράδειγμα καταγραφής ιστορικού (Βηματομετρητής & θερμιδομετρητής)



Εικόνα 1.8: Συνταγές για υγιεινή διατροφή – fitnessblender

## Κεφάλαιο 2<sup>ο</sup>: Android

### 2.1 Εισαγωγή

Το “Android” είναι ένα λειτουργικό σύστημα με λογισμικό ανοιχτού κώδικα και αναπτύχθηκε από τη “Google” και στη συνέχεια από την “Open Handset Alliance”. Είναι ένα λειτουργικό σύστημα που αρχικά κατασκευάστηκε για κινητές συσκευές με οθόνη αφής (π.χ. κινητά, τάμπλετ) και στη συνέχεια χρησιμοποιήθηκε και για άλλες συσκευές όπως τηλεοράσεις, έξυπνα τηλέφωνα, κλπ. Αποτελεί ένα ολοκληρωμένο λειτουργικό σύστημα που αποτελείται από τέσσερα επίπεδα που συνεργάζονται μεταξύ τους καθώς το ένα επηρεάζεται από το άλλο. Αυτά τα επίπεδα είναι ο πυρήνας, οι βιβλιοθήκες, το πλαίσιο εφαρμογών και οι εφαρμογές (θα αναφερθούμε σε αυτά στην επόμενη ενότητα) [1].

Παρουσιάστηκε για πρώτη φορά (από τη “Google”) τον Νοέμβριο του 2007 και από τότε έχουν υπάρξει στη αγορά πολλές διαφορετικές εκδόσεις του με πιο διαδεδομένες τη “Nougat” και τη “Marshmallow”. Όλες οι εφαρμογές του είναι γραμμένες σε “Java” και οι προγραμματιστές μπορούν να προγραμματίζουν δωρεάν τις εφαρμογές τους και να τις πουλούν στο “Play Store”.

Τα κινητά τηλέφωνα με λειτουργικό σύστημα Android εμφανίστηκαν για πρώτη φορά στην αγορά το 2008 και από τότε μέχρι το 2013 σύμφωνα με έκθεση της “Strategy Analytics”, υπερίσχυαν στην αγορά με ποσοστό αγοράς πάνω από 80%. Αν σκεφτεί κανείς ότι από το 2013 μέχρι σήμερα έχουν βγει πολλές ακόμη εκδόσεις και το “Android” είναι ένα συχνά εξελισσόμενο λειτουργικό σύστημα που πλέον δεν τοποθετείται μόνο σε κινητά τηλέφωνα αλλά και σε άλλες συσκευές, είναι προφανές ότι αποτελεί ακόμη μία δημοφιλή επιλογή.

Τέλος, τρέχει τον πυρήνα του λειτουργικού “Linux” και αποτελεί μία από τις πιο δημοφιλείς προτιμήσεις σε λειτουργικό σύστημα αφού οι πωλήσεις των συσκευών που το υποστηρίζουν ξεπερνούν το άθροισμα των πωλήσεων των συσκευών “Windows”, “iOS” και “Mac OS X”. Αξίζει να σημειωθεί ότι οι χρήστες συσκευών με λειτουργικό σύστημα “Android” έχει παρατηρηθεί ότι συνηθίζουν να προτιμούν τις εφαρμογές που κυκλοφορούν δωρεάν στο “play store” συγκριτικά με τους χρήστες του “iOS” οι οποίοι συνήθως πληρώνουν για τις εφαρμογές που τους ενδιαφέρουν.

Επίσης, οι προγραμματιστές των εφαρμογών για το λειτουργικό σύστημα “Android” δεν είναι απαραίτητο να έχουν στην κατοχή τους κάποια συσκευή με το παραπάνω λειτουργικό σύστημα, μπορούν να προγραμματίσουν δωρεάν μέσω ενός προγραμματιστικού περιβάλλοντος όπως το “Android” Studio, το οποίο χρησιμοποιήθηκε και για την υλοποίηση της εφαρμογής “Unikorn” και περιλαμβάνει και εξομοιωτή “Android” συσκευής. Αντίθετα, οι προγραμματιστές εφαρμογών για το λειτουργικό σύστημα “iOS” καλούνται να πληρώσουν συνδρομή χρόνου και για να προγραμματίσουν πρέπει να διαθέτουν κάποια “iOS” συσκευή.

Το “Android Studio” είναι αυτή τη χρονική στιγμή η πιο γνωστή και ευρέως χρησιμοποιούμενη πλατφόρμα για δημιουργία και ανάπτυξη εφαρμογών για το λειτουργικό σύστημα “Android” και χρησιμοποιήθηκε και για την ανάπτυξη της “Unikorn” και στο οποίο αναφέρεται το Κεφάλαιο 3 και πιο συγκεκριμένα η Ενότητα 3.1. Η χρήση του είναι δωρεάν, παρέχει στον προγραμματιστή πολλές βιβλιοθήκες και αποτελεί ένα καλά σχεδιασμένο και οργανωμένο περιβάλλον εργασίας. Επιπλέον, όπως αναφέρθηκε και παραπάνω, δίνει τη δυνατότητα στον χρήστη να φτιάξει έναν εξομοιωτή μιας “Android” συσκευής, με τα χαρακτηριστικά που ο ίδιος επιλέγει, και να τρέξει την εφαρμογή του εκεί, χωρίς να χρειαστεί να αγοράσει κάποια συσκευή με λειτουργικό σύστημα “Android”.



Εικόνα 2.1: Αρχική οθόνη του Android 12 [30]

## 2.2 Αρχιτεκτονική του Android

Στη βάση του “Android” υπάρχει ο πυρήνας “Linux”, ο οποίος προσφέρει εφαρμογές για υλικό, όπως ηχεία, πληκτρολόγιο, κλπ και διαχειρίζεται την κοινόχρηστη μνήμη. Εκτελεί βασικές για το σύστημα υπηρεσίες και είναι σημαντικός για τη σύνδεση μεταξύ του λογισμικού και του υλικού [1][7].

Το αμέσως επόμενο επίπεδο της αρχιτεκτονικής του “Android” είναι το “Hardware Abstraction Layer”, το οποίο προσφέρει βιβλιοθήκες με μοναδικά “interfaces” για συγκεκριμένα “abstractions” που υποστηρίζουν το “hardware”.

Πάνω από το “HAL” βρίσκονται το “Native C/C++ Libraries” που περιλαμβάνει βιβλιοθήκες γραμμένες σε C/C++ τις οποίες αξιοποιεί το “Java API Framework” για την ανάπτυξη εφαρμογών, και το “Android Runtime” το οποίο είναι ξεχωριστό για κάθε πρόγραμμα. Για την καλύτερη υποστήριξη των συσκευών “Android” χρησιμοποιείται μία βελτιωμένη έκδοση εικονική μηχανής “Java” η οποία χρησιμοποιεί πιο μικρά αρχεία ενδιάμεσου κώδικα από την “Java SE”.

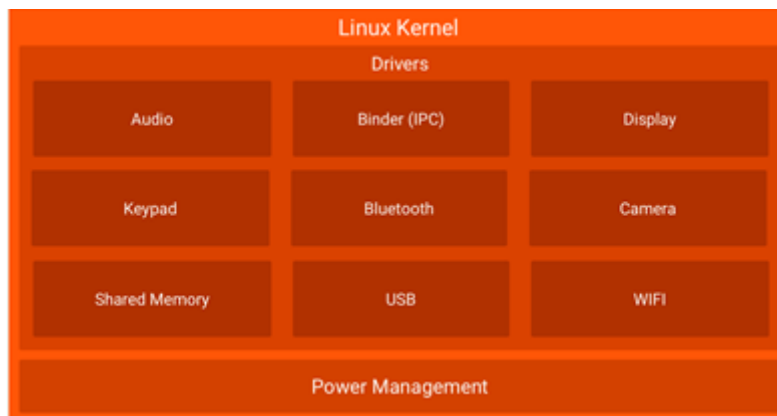
Αμέσως επόμενο επίπεδο είναι το “Applications Framework” το οποίο περιλαμβάνει διαχειριστή δραστηριότητας, πάροχο περιεχομένου, διαχειριστή πόρων, διαχειριστή τοποθεσίας και διαχειριστή ενημερώσεων. Με όλα τα παραπάνω παρέχετε στον προγραμματιστή η δυνατότητα να αναπτύξει τις εφαρμογές του αξιοποιώντας όλες τις δυνατότητες του “Android” και προσθέτοντας επιπλέον δυνατότητες με επιπλέον βιβλιοθήκες αν/όταν χρειάζεται.

Τέλος, στην κορυφή της αρχιτεκτονικής του “Android”, βρίσκονται τα “System Apps”, τα οποία αποτελούν τόσο τις εφαρμογές που είναι προεγκατεστημένες σε κάθε συσκευή “Android” όσο και τις εφαρμογές που εγκαθιστά ο χρήστης από το “Play Store”. Ένα κοινό όλων αυτών των εφαρμογών είναι ότι είναι γραμμένες σε “Java”.

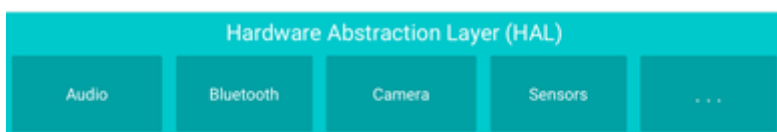
## Κεφάλαιο 2



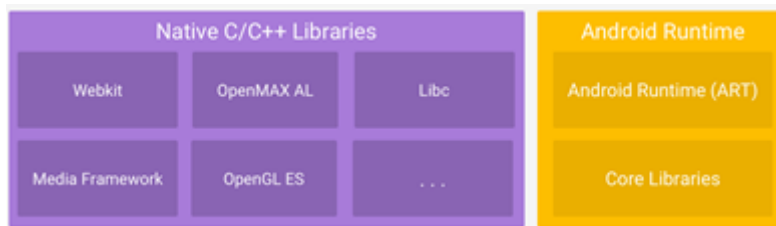
Εικόνα 2.2: Η αρχιτεκτονική του Android [7]



Εικόνα 2.3: Linux Kernel [7]



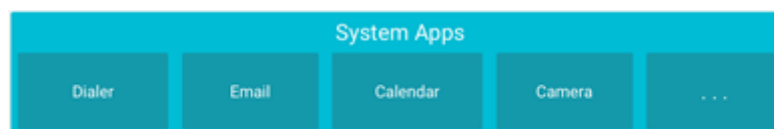
Εικόνα 2.4: Hardware Abstraction Layer (HAL) [7]



Εικόνα 2.5: Native C/C++ Libraries/Android Runtime [7]



Εικόνα 2.6: Java API Framework [7]



Εικόνα 2.7: System Apps [7]

### 2.3 Βασικά χαρακτηριστικά

Το λειτουργικό σύστημα “Android” έχει πάρα πολλές λειτουργίες, τις πιο σημαντικές εκ των οποίων θα αναφέρουμε στην παρούσα ενότητα. Αποτελεί μία τεχνολογία με πάρα πολλές δυνατότητες που παρουσιάζει διαρκή βελτίωση και εξέλιξη [1][2][15].

Πιο συγκεκριμένα, το λειτουργικό σύστημα Android είναι ένα λειτουργικό σύστημα ανοικτού πηγαίου κώδικα και παρέχεται δωρεάν. Αυτό σημαίνει ότι ο προγραμματιστής έχει τη δυνατότητα να τρέξει τον πηγαίο κώδικα και να παρατηρήσει τα χαρακτηριστικά του. Επίσης, μπορεί να δημοσιεύει τα σφάλματά του βοηθώντας το “Android” να εξελίσσεται, καθώς και να συμμετέχει σε δημόσιες πλατφόρμες επικοινωνίας και ανταλλαγής ιδεών και προβλημάτων.

Η πλατφόρμα “Android” προσαρμόζεται σε πολλές αναλύσεις οθόνης και υποστηρίζει την αποστολή και παραλαβή γραπτών μηνυμάτων και μηνυμάτων με εικόνες. Επίσης, υποστηρίζει πολλές γλώσσες, με αποτέλεσμα να μπορεί εύκολα να χρησιμοποιηθεί από άτομα διαφορετικών ηλικιών, από κάθε μέρος του πλανήτη και να καλύψει τις ανάγκες τους.

Επίσης, χρησιμοποιεί την “SQLite” για αποθήκευση δεδομένων και εξωτερική αποθήκευση σε “micro-SD” κάρτες. Η “SQLite” είναι μία βάση δεδομένων που προτιμάται για εφαρμογές που απαιτούν τοπική αποθήκευση, όπως για παράδειγμα ο φυλλομετρητής. Είναι πολύ δημοφιλής και αποτελεί τμήμα της παραδοτέας εφαρμογής και όχι μια απλή διεργασία της.

Επιπλέον, υποστηρίζει αρκετές τεχνολογίες για συνδεσιμότητα, όπως τις “GSM/EPGE/UMTS/HSPA/HSPA+/LTE”, “3G”, “4G”, “CDMA”, “EV-DO”, “Bluetooth”, “NFC”, “Wi-Fi” και πολυμέσα, όπως “MP3”, “JPEG”, “PNG”, κλπ.

Εάν ο χρήστης το επιθυμεί, μπορεί να εγκαταστήσει, από το “play store”, πληθώρα εφαρμογών είτε δωρεάν είτε επί πληρωμή, με στόχο τη βελτίωση της εμπειρίας του και την κάλυψη των αναγκών του. Επιπλέον, παρ’ όλο που κάποιες εφαρμογές όπως ο φυλλομετρητής, υπάρχουν στη συσκευή, προκειμένου ο χρήστης να μπορεί να συνδέεται στο διαδίκτυο, αυτό δεν αποτρέπει ούτε εμποδίζει τον χρήστη από το να εγκαταστήσει άλλες αντίστοιχες εφαρμογές με τις υπάρχουσες (σε αυτή την περίπτωση φυλλομετρητή), τις οποίες επίσης μπορεί να προμηθευτεί από το “play store”.

Αξίζει να αναφερθεί ότι το “Android” υποστηρίζει λογισμικό γραμμένο σε “Java”, πράγμα που σημαίνει ότι μπορεί να το μεταγλωττίσει και να το εκτελέσει. Οι περισσότερες εφαρμογές για “Android” είναι γραμμένες σε “Java”. Η “Java” είναι μία από τις πιο γνωστές γλώσσες προγραμματισμού και την χρησιμοποιούν πολλοί προγραμματιστές σε όλο τον κόσμο. Δεδομένου δε και ότι είναι μία αντικειμενοστρεφής, δυνατή, ανοιχτού κώδικα γλώσσα, που μπορεί ο καθένας να την χρησιμοποιήσει δωρεάν καθώς και να αξιοποιήσει το πλήθος βιβλιοθηκών που του παρέχει, ένας έμπειρος προγραμματιστής μπορεί μέσα σε μικρό χρονικό διάστημα να καταλάβει πως να προγραμματίσει σε Android και με την βοήθεια των έτοιμων εφαρμογών που υπάρχουν στο διαδίκτυο να φτιάξει σύνθετες και πρωτότυπες εφαρμογές.

Επίσης, το συγκεκριμένο λειτουργικό σύστημα επιτρέπει την ταυτόχρονη λειτουργία εφαρμογών. Οι οθόνες πολυεπαφής που περιλαμβάνουν οι συσκευές Android επιτρέπουν στον χρήστη να ελίσσεται εύκολα ανάμεσα στις εφαρμογές, να ανοίγει πληκτρολόγιο για να συνομιλήσει με κάποιον άλλο χρήστη, να ακούσει μουσική, να σερφάρει στο web και γενικά οι δυνατότητες που έχει ο χρήστης είναι αμέτρητες.



Εικόνα 2.8: Παράδειγμα οθονών στις οποίες προσαρμόζεται το Android [33]

## 2.4 Στοιχεία μίας Android εφαρμογής

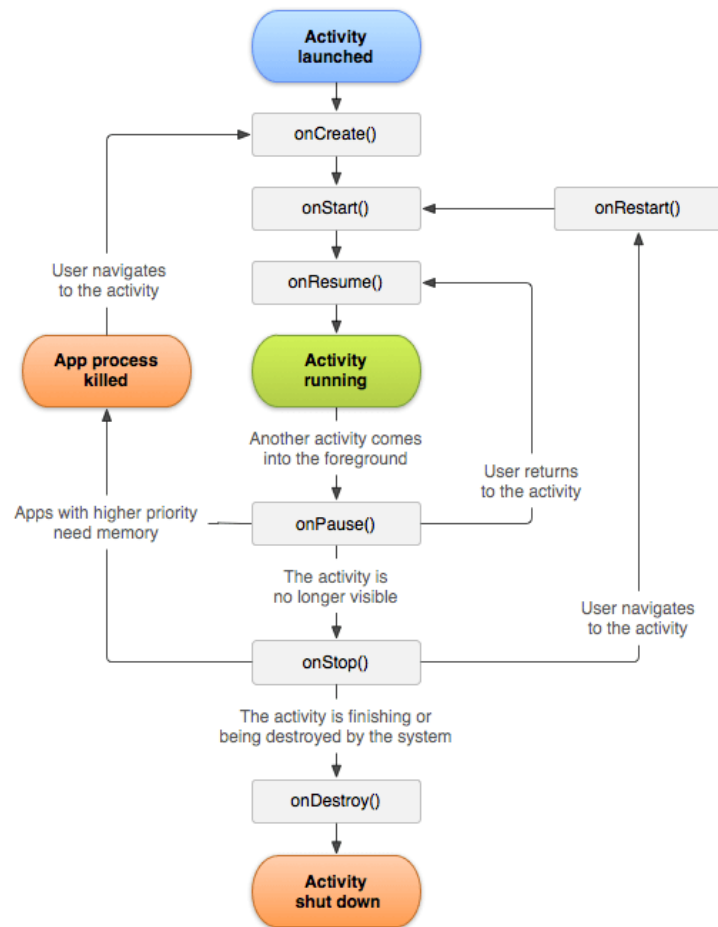
Κάθε εφαρμογή “Android” αποτελείται από κάποια επιμέρους συστατικά στοιχεία, κάποια από τα οποία αναφέρονται στην παρούσα ενότητα [1].

### 2.4.1 Activities

Όπως και στις ιστοσελίδες, έτσι και στις “Android” εφαρμογές υπάρχουν πολλές διαφορετικές οθόνες. Μπορούμε να τις σκεφτούμε σαν μία στοίβα με οθόνες από τις οποίες ο χρήστης μπορεί να δει μόνο αυτή που βρίσκεται στην κορυφή. Αυτές οι οθόνες στην περίπτωση μιας “Android” εφαρμογής, ονομάζονται “activities”. Οι δραστηριότητες, είναι ένα από τα βασικά στοιχεία μιας “Android” εφαρμογής και κάθε “activity” πρέπει να είναι δηλωμένο στο “AndroidManifest.xml”, στο οποίο θα αναφερθούμε στην υποενότητα 2.3.7.

Σε κάθε εφαρμογή Android δεν υπάρχει μία κλάση “main” αλλά υπάρχει μία αρχική οθόνη, η οποία συνηθίζεται να είναι η “Main Activity”. Όταν ο χρήστης παρατηρεί ένα “activity”, στην πραγματικότητα παρατηρεί ένα σύνολο από “Views”. Τα “views” μπορεί να είναι κουμπιά, μενού, μπάρες κύλισης, κλπ και βοηθούν στην αλληλεπίδραση του χρήστη με την εφαρμογή.

Στην παρακάτω εικόνα μπορούμε να παρατηρήσουμε τον κύκλο ζωής μιας δραστηριότητας. Αρχικά καλείται η μέθοδος “onCreate()”, η οποία ενεργοποιεί τη δημιουργία του “activity”. Έπειτα ενεργοποιείται η μέθοδος “onStart()” για να γίνει ορατή η εφαρμογή στον χρήστη, και αφού γίνει ορατή, καλείται η “onResume()”. Στη μέθοδο “onResume()” μένει μέχρι να χαθεί το “focus”, δηλαδή κάποιος να καλέσει τον χρήστη, να χτυπήσει κάποιο ξυπνητήρι, κλπ. Μετά, καλείται η μέθοδος “onPause()” για να παραμείνει η δραστηριότητα για ένα μικρό χρονικό διάστημα στο παρασκήνιο. Έπειτα καλείται η “onStop()”, που διατηρεί τα δεδομένα στη μνήμη και πλέον η δραστηριότητα δεν είναι ορατή. Τέλος, καλείται το “onDestroy()” σε περίπτωση που η εφαρμογή κληθεί να κλείσει οριστικά ή το “onRestart()” για να επανέλθει. Στην περίπτωση της κλήσης της μεθόδου “onDestroy()”, τα δεδομένα αποδεσμεύονται και η δραστηριότητα σταματά να εκτελείται.

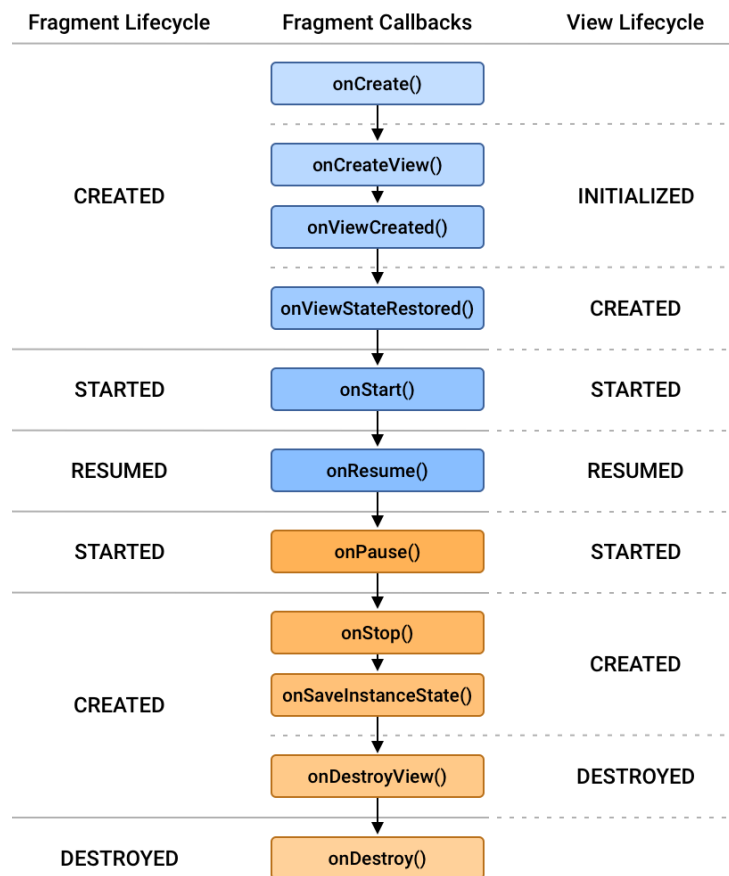


Εικόνα 2.9: Activity Lifecycle [31]

## 2.4.2 Fragments

Ένα fragment είναι ένα υποσύνολο μιας δραστηριότητας. Έχει δικό του κύκλο ζωής και διάδραση αλλά δεν μπορεί να υπάρξει χωρίς “activity”. Μία δραστηριότητα μπορεί να έχει πολλά “fragments” και ένα “fragment” μπορεί να καλείται από πολλές δραστηριότητες. Εκτός από προγραμματιστικά, μπορούμε να δημιουργήσουμε ένα “fragment” και μέσα στο αρχείο “xml” της δραστηριότητας που το χρησιμοποιεί.

Παρακάτω παρατηρούμε τον κύκλο ζωής ενός “fragment”. Οι μέθοδοι “onAttach(Activity)”, “onCreate(Bundle)”, “onCreateView(LayoutInflater”, “ViewGroup, Bundle)” καλούνται κατά τη δημιουργία του “fragment” για να το συσχετίσουν με τη δραστηριότητα, να το ενεργοποιήσουν και έπειτα να εμφανιστεί στην οθόνη με τη μέθοδο “onStart()”. Όταν ο χρήστης αλληλεπιδράσει θα εκτελεστεί η “onResume()” και όταν σταματήσει η αλληλεπίδραση, θα κληθεί η μέθοδος “onPause()”. Όταν πλέον δεν υπάρχει καμία αλληλεπίδραση και πλησιάζουμε στο κλείσιμο του “fragment”, θα κληθεί η “onStop()” που θα το κάνει άρατο, έπειτα η “onDestroyView()”, που θα εξαφανίσει τα σχετικά με τη διεπαφή αντικείμενα, η μέθοδος “onDestroy()”, που θα καταστρέψει το “fragment” και τέλος η μέθοδος “onDetach()” που θα ακυρώσει τη σχέση του “fragment” με την δραστηριότητα.



Εικόνα 2.10: Fragment Lifecycle [32]

### 2.4.3 Layouts

Τα “layouts” ορίζουν τη διάταξη μέσα σε μία δραστηριότητα ή/και “fragment”. Υπάρχουν διαφορετικά είδη “layouts”, ανάλογα με τις δυνατότητές τους. Για παράδειγμα μπορεί να είναι στατικά ή δυναμικά, να επιτρέπουν το “drag and drop” από το “Android Studio Layout Editor” ή να μην το επιτρέπουν, κλπ.

Παρακάτω αναφέρονται όλα τα διαφορετικά είδη “layout”:

- “Constraint Layout” (υποστηρίζει drag and drop από το Android Studio Layout και για να χρησιμοποιηθεί πρέπει να δοθούν οριζόντιος και κάθετος περιορισμός για κάθε στοιχείο)
- “Linear Layout (Μπορεί να υπάρξει εμφωλευμένο “Linear Layout” σε άλλο “Linear Layout”)
- “Relative Layout” (Τα στοιχεία τοποθετούνται το ένα σε σχέση με το άλλο)
- “Motion Layout”
- “RecyclerView” (Δυναμικό “Layout”)
- “Card Based Layout”
- “Adaptive Layout”
- “Adapter View”

## 2.5 Εκδόσεις Android

Η πρώτη έκδοση του “Android” (“Android Alpha”) κυκλοφόρησε στις 23 Σεπτεμβρίου του 2008. Αποτελεί τον προκάτοχο του “Android Beta” που κυκλοφόρησε στις 9 Φεβρουαρίου του 2009. Όσον αφορά τα ονόματα αυτών των δυο εκδόσεων, είναι “Alpha” και “Beta”, καθώς αποτελούν τις πρώτες 2 εκδόσεις του λειτουργικού συστήματος, οι οποίες είναι δοκιμαστικές, με την “Alpha” να χρησιμοποιείται μόνο από τους προγραμματιστές και τη Beta να είναι διαθέσιμη μόνο για ένα μικρό σύνολο χρηστών [2][8][9][10][11][12][13][14][23].

Η έκδοση Beta, περιλαμβάνει κάποιες βασικές λειτουργίες του λειτουργικού συστήματος “Android”, όπως είναι οι εφαρμογές τις “Google”, το “youtube”, το “calendar”, κλπ. Αξίζει να σημειωθεί ότι σε αυτές τις πρώτες εκδόσεις του λειτουργικού συστήματος “Android”, οι προγραμματιστές μπορούσαν να δημοσιεύουν τις εφαρμογές τους αλλά δεν μπορούσαν να τις πουλήσουν. Δηλαδή υπήρχε η ανάπτυξη εφαρμογών για android αλλά χωρίς δυνατότητα πληρωμής.

Από τον Απρίλιο του 2009 τα ονόματα των εκδόσεων του “Android” έχουν ονόματα γλυκών και κυκλοφορούν με αλφαβητική σειρά. Μετά από την έκδοση “Beta”, ακολουθεί η έκδοση “Cupcake”. Η συγκεκριμένη έκδοση έχει πάρα πολλές επιπλέον δυνατότητες από την προηγούμενή της, στις οποίες περιλαμβάνονται η δυνατότητα καταγραφής και παρακολούθησης βίντεο και το ανέβασμά τους στο “youtube”. Επίσης, με την έκδοση “Cupcake” πλέον ο χρήστης είναι σε θέση να χρησιμοποιήσει “Bluetooth” για ανταλλαγή δεδομένων και να ανεβάζει φωτογραφίες στο “Picasa”.

Στην έκδοση “Donut”, έχουν προστεθεί πλέον πολλές επιπλέον δυνατότητες σχετικά με την κάμερα και τις φωτογραφίες και υποστηρίζονται διάφορα μεγέθη οθονών. Επίσης, το “Android Market” έχει πλέον τη δυνατότητα να ανοίγει οθόνες από άλλες εφαρμογές, χαρίζοντας στον χρήστη καλύτερη διάδραση. Τέλος, ένα σημαντικό χαρακτηριστικό του συγκεκριμένου λειτουργικού συστήματος, για την εποχή του (Σεπτέμβριος 2009) αποτελεί το text-to-speech μέσω του οποίου ο χρήστης μπορεί πλέον να μιλάει και η φωνή του να μετατρέπεται σε κείμενο.

Η αμέσως επόμενη έκδοση του λειτουργικού συστήματος “Android”, είναι το “Android Éclair”. Με το “Google maps 3.1.2”, το “Android Éclair”, παρέχει στον χρήστη μεγαλύτερη ποικιλία και ποιότητα χαρτών και καλύτερη εμπειρία πλοήγησης. Επίσης υπάρχει μια εξέλιξη στο “Bluetooth” και στα κουμπιά πλοήγησης, τα οποία εμφανίζονται μόνο όποτε χρειάζονται. Με αυτή την έκδοση ο χρήστης μπορεί πλέον να χρησιμοποιεί κινούμενα φόντα στην επιφάνεια εργασίας του καθώς και να πραγματοποιεί ψηφιακό zoom.

Αμέσως μετά το “Éclair”, έρχεται το “Android Froyo”. Παρέχει στον χρήστη καλύτερη μνήμη και απόδοση και με την άμεση μεταγλώττιση επιτυγχάνει αύξηση της ταχύτητας του λειτουργικού συστήματος. Είναι η πρώτη έκδοση “Android” που επιτρέπει την μετατροπή της συσκευής σε “WiFi hotspot”, κάτι πολύ ιδιαίτερο και πρωτότυπο για το 2009 που κυκλοφόρησε.

Ακολουθεί το “Gingerbread” που μοιάζει περισσότερο, από τα προηγούμενά του, με το λειτουργικό σύστημα που έχουμε αυτή την χρονική στιγμή, παρέχοντας στον χρήστη τη δυνατότητα σύνδεσης στα κοινωνικά δίκτυα και τη δυνατότητα πραγματοποίησης βιντεοκλήσεων.

Το Honeycomb κυκλοφόρησε στις 22 Φεβρουαρίου του 2011 και ήταν διαθέσιμο μόνο για “tablets”. Σε αυτή την έκδοση παρατηρούμε την χρήση πολλαπλών πυρήνων. Το ακολουθεί το “Ice Cream, Sandwich”, το οποίο αποτελεί συρρίκνωσή του και επιτρέπει τη σύνδεση συσκευών ασύρματα μέσω του “WiFi Direct”.

Η επόμενη έκδοση του “Android” είναι το “Android Jelly Bean”, που παρουσιάζει μια βελτίωση στο πληκτρολόγιο, προβλέποντας την επόμενη λέξη από αυτή που πληκτρολογεί ο χρήστης και σημαντική βελτίωση στα “Bluetooth”, τα οποία πια χρησιμοποιούν λιγότερη ενέργεια. Και ακολουθεί το “KitKat” με μειωμένη χρήση μνήμης και ανανέωση εφαρμογών κλήσεων.

Το “Lollipop” κυκλοφόρησε το 2014. Παρουσιάζει σημαντική βελτίωση στην ταχύτητα και μία μέθοδο εξοικονόμησης μπαταρίας που κάνει τη συσκευή περισσότερο λειτουργική. Το περιβάλλον χρήστη πλέον είναι το “Materialized View” και με σύνδεση στο “Google Drive” αποκτά ο χρήστης τη δυνατότητα να έχει “back up” έως και 25MB ούτως ώστε να μην χάνει τα αρχεία, τις φωτογραφίες του, κλπ σε περίπτωση απώλειας της συσκευής του. Για πρώτη φορά δίνεται η δυνατότητα στον χρήστη να χρησιμοποιεί το “Google Pay” με τα δαχτυλικά του αποτυπώματα παρέχοντας του περισσότερη ασφάλεια καθώς και να ανοίγει μέχρι 4 παράθυρα ταυτόχρονα, με τη δυνατότητα ταυτόχρονης λειτουργίας εφαρμογών που παρέχεται στον χρήστη.

Το λειτουργικό σύστημα “Nougat” βελτιώνει την δυνατότητα του χρήστη να χρησιμοποιεί ταυτόχρονα πολλές εφαρμογές και βελτιώνει το σύστημα ειδοποιήσεων. Ένα χρόνο αργότερα, τον Αύγουστο του 2017, κυκλοφορεί το Android Oreo, το οποίο το οποίο κάνει ακόμα καλύτερη την εμπειρία του χρήστη σχετικά με τις εφαρμογές που μπορούν να λειτουργούν ταυτόχρονα στη συσκευή του, καθώς και πολλές άλλες εφαρμογές, όπως τις ρυθμίσεις και την αισθητική. Είναι η πρώτη φορά, που παρέχονται δυνατότητες ευφυΐας στον προγραμματιστή εφαρμογών του συγκεκριμένου λειτουργικού συστήματος.

Το “Android Pie” κυκλοφόρησε για πρώτη φορά 6 Αυγούστου του 2018. Παρείχε στον χρήστη τη δυνατότητα να περιστρέφει την οθόνη του χειροκίνητα καθώς και να περιηγείται στη συσκευή με μετακίνηση του χεριού του. Τον Σεπτέμβριο του 2019, εμφανίζεται στην αγορά η έκδοση 10, η οποία παρέχει στον χρήστη τη δυνατότητα να έχει τη συσκευή με φωτεινό θέμα. Επίσης ο χρήστης μπορεί πλέον όχι μόνο να κρατάει στιγμιότυπα αλλά και να βιντεοσκοπεί την οθόνη του. Τέλος, υπάρχει μία βελτίωση στην ασφάλεια, προσθέτοντας νέες άδειες σχετικά με τις εφαρμογές της συσκευής.

Αν και οι μικρές εικονίτσες που εμφανίζονται στην οθόνη για να μπορεί ο χρήστης να βλέπει για παράδειγμα ένα βίντεο στο “youtube” και ταυτόχρονα να έχει μία συνομιλία με έναν φίλο του στο “Messenger”, ήδη υπήρχαν από την έκδοση 10, στην έκδοση 11 είναι πλέον διαθέσιμες και για άλλες εφαρμογές και όχι μόνο για το “Messenger”. Η μπάρα των ειδοποιήσεων πλέον παρέχει περισσότερες λεπτομέρειες και είναι κατηγοριοποιημένες. Επιπλέον, όταν ο χρήστης δεν χρησιμοποιεί για ένα μεγάλο χρονικό διάστημα μία εφαρμογή, οι άδειες που της έχει παραχωρήσει διαγράφονται αυτόματα.

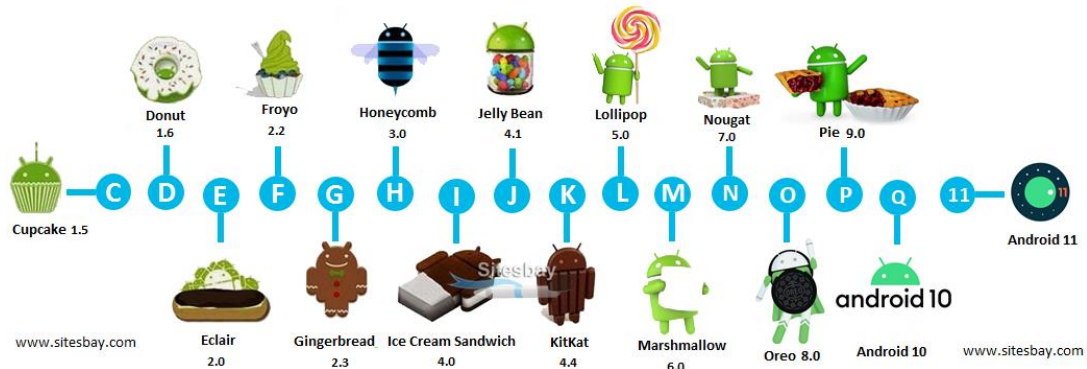
Η έκδοση που κυκλοφορεί την παρούσα χρονική στιγμή και κυκλοφόρησε για πρώτη φορά στις 4 Οκτωβρίου του 2021 είναι η “Android 12”. Με το “Material You” έχει βελτιωθεί η σχεδίαση και πλέον ο χρήστης έχει τη δυνατότητα, εκτός του να ρυθμίσει τη συσκευή και να αλλάξει φόντο η οθόνη μόνη της, να αλλάζει και το θέμα των χρωμάτων ανάλογα με το παρόν φόντο ταπετσαρίας. Αυτό σημαίνει ότι δεν έχουμε πλέον μόνο “dark mode” και το “light mode” αλλά και μία παλέτα χρωμάτων που αλλάζει ανάλογα με το επιλεγθέν φόντο. Επίσης ο χρήστης μπορεί ανά πάσα χρονική στιγμή να ελέγξει και να επεξεργαστεί τα δεδομένα που έχει παραχωρήσει στις εφαρμογές του και γενικά οι ρυθμίσεις έχουν βελτιωθεί αρκετά.

Η επόμενη έκδοση που - αν και δεν έχει ανακοινωθεί ακόμη – αναμένετε να κυκλοφορήσει το τελευταίο τρίμηνο του 2022, είναι το “Android 13”. Υποστηρίζεται ότι θα βασίζεται στο επίπεδο της μπαταρίας για να ορίσει τον αριθμό διεργασιών που μπορεί να χρησιμοποιήσει μια εφαρμογή και με το “Panlingual” ο χρήστης δεν θα είναι πια αναγκασμένος να έχει όλες τις εγκατεστημένες εφαρμογές του σε μία γλώσσα (π.χ. ελληνικά/αγγλικά) αλλά θα μπορεί να διαλέξει άλλη γλώσσα για κάθε εφαρμογή

## Κεφάλαιο 2

του. Τέλος, θα βελτιωθεί η υπηρεσία ειδοποιήσεων, καθώς θα γίνει όπως στις άδειες, δηλαδή, θα μπορούν να προστεθούν και να επεξεργαστούν οι κατηγορίες ειδοποιήσεων.

Όπως βλέπουμε στην Εικόνα 2.11 δεν υπάρχει καμία καθυστέρηση στην ανάπτυξη και κυκλοφορία νέας έκδοσης του λειτουργικού συστήματος “Android” και υπάρχει μία συνεχής εξέλιξη του λογισμικού. Με αυτή την εξέλιξη επιτυγχάνεται καλύτερη λειτουργικότητα στις εφαρμογές των συσκευών που έχουν σαν λειτουργικό σύστημα το “Android” καθώς και παροχή περισσότερων βιβλιοθηκών και δυνατοτήτων.



Εικόνα 2.11: Android versions [29]

Πίνακας 2.1: Εκδόσεις Android [11]

Όνομα	Νούμερο έκδοσης	Ημερομηνία αρχικής κυκλοφορίας	Επίπεδο API
Alpha	1.0	23 Σεπτεμβρίου 2008	1
Beta	1.1	9 Φεβρουαρίου 2009	2
Cupcake	1.5	27 Απριλίου 2009	3
Donut	1.6	15 Σεπτεμβρίου 2009	4
Éclair	2.0-2.1	26 Οκτωβρίου 2009	5-7
Froyo	2.2-2.2.3	20 Μαΐου 2010	8
Gingerbread	2.3-2.3.7	6 Δεκεμβρίου 2010	9-10
Honeycomb	3.0-3.2.6	22 Φεβρουαρίου 2011	11-13
Ice Cream Sandwich	4.0-4.0.4	18 Οκτωβρίου 2011	14-15
Jelly Bean	4.1-4.3.1	9 Ιουλίου 2012	16-18
KitKat	4.4-4.4.4	31 Οκτωβρίου 2013	19-20
Lollipop	5.0-5.1.1	12 Νοεμβρίου 2014	21-22

Marshmallow	6.0-6.0.1	5 Οκτωβρίου 2015	23
Nougat	7.0-7.1.2	22 Αυγούστου 2016	24-25
Oreo	8.0-8.1	21 Αυγούστου 2017	26-27
Pie	9.0	6 Αυγούστου 2018	28
10	10.0	3 Σεπτεμβρίου 2019	29
11	11.0	19 Φεβρουαρίου 2020	30
12	12.0	4 Οκτωβρίου 2021	31

## Κεφάλαιο 3<sup>ο</sup>: Προγράμματα & Εργαλεία που χρησιμοποιήθηκαν

### 3.1 Android Studio

Το “Android Studio” είναι ένα προγραμματιστικό περιβάλλον για “Android” εφαρμογές. Αποτελεί το βασικό εργαλείο που χρησιμοποιήθηκε για την υλοποίηση της παρούσας πτυχιακής εργασίας. Πρωτοεμφανίστηκε τον Μάιο του 2013 στο συνέδριο “Google I/O” από την “Google Product Manager”, “Katherine Chou” και η πρώτη σταθερή έκδοσή του ανακοινώθηκε τον Δεκέμβριο του 2014. Στηρίχτηκε στο λογισμικό της “JetBrains” IntelliJ IDEA” και είναι κατασκευασμένο σε “Java” [6][16][17].

#### 3.1.1 Γενικά χαρακτηριστικά

Κάποια από τα πιο σημαντικά χαρακτηριστικά του “Android Studio παρουσιάζονται παρακάτω:

- Χρησιμοποιεί πρότυπα κώδικα.
- Μπορεί να συνδεθεί με “Git”.
- Για τη συγγραφή κώδικα υποστηρίζει “Java”, “C”, “C++” και “Kotlin” και για ό,τι βλέπει ο χρήστης, “XML”.
- Περιέχει το “Android NDK” (εργαλείο που βοηθά στην υλοποίηση κώδικα χρησιμοποιώντας γλώσσες όπως “C” και “C++”) και το “Android SDK” (βιβλιοθήκες που βοηθούν στη συγγραφή και υλοποίηση “Android” εφαρμογών)
- Περιέχει προσομοιωτή, δίνοντας τη δυνατότητα στον προγραμματιστή να εκτελέσει τον κώδικα και να δει το αποτέλεσμα του στην οθόνη του.
- Περιέχει εντοπισμό σφαλμάτων και οδηγό για πιθανές λύσεις τους.
- Υποστηρίζει την ανάπτυξη εφαρμογών για όλες τις συσκευές που υποστηρίζουν αυτό το λειτουργικό σύστημα (για κινητά, τηλεοράσεις, κλπ)

#### 3.1.2 Το περιβάλλον του Android Studio

Στην παρακάτω εικόνα (Εικόνα 3.1) απεικονίζεται το περιβάλλον του “Android Studio” το οποίο αποτελείται από τη μπάρα εργαλείων (Εικόνα 3.2), τη μπάρα πλοήγησης (Εικόνα 3.3), τον χώρο συγγραφής κώδικα (Εικόνα 3.4), τα παράθυρα εργαλείων (Εικόνα 3.5) και τη γραμμή κατάστασης (Εικόνα 3.6).

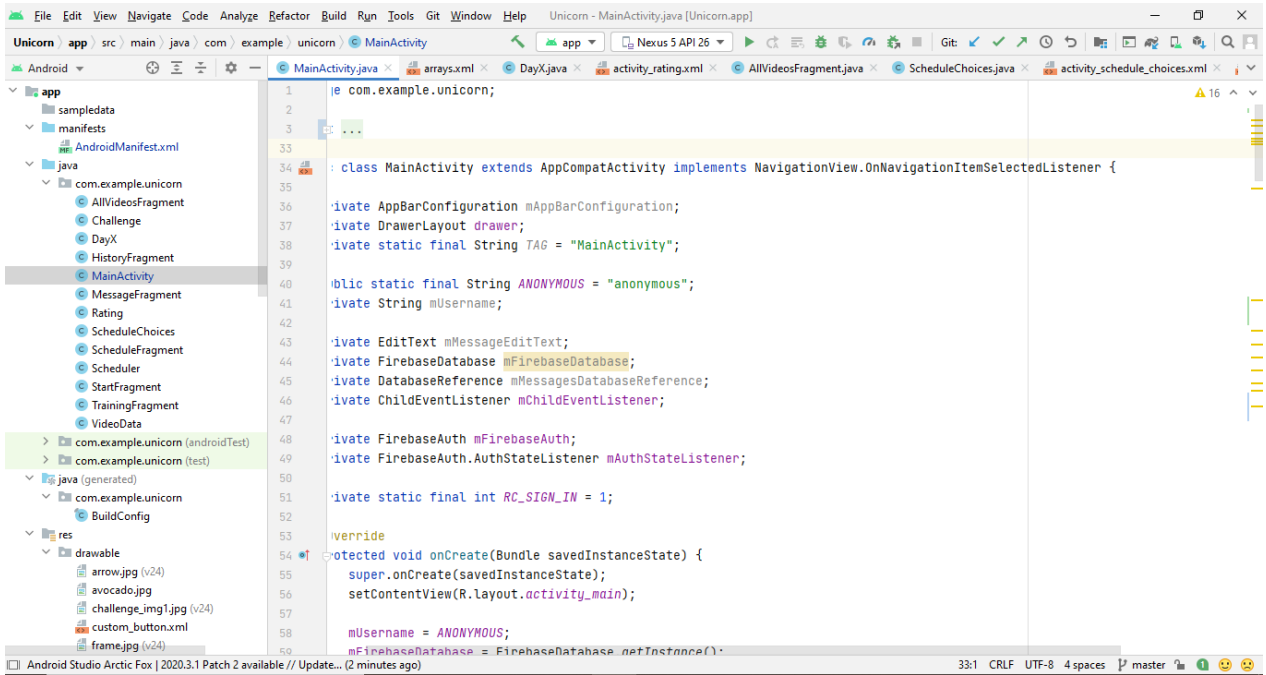
Η μπάρα εργαλείων δίνει μία σειρά από δυνατότητες στον προγραμματιστή που χρησιμοποιεί την εφαρμογή, όπως εκτέλεση του κώδικα, εντοπισμό σφαλμάτων και σύνδεση με το “git”. Διευκολύνει τον χρήστη να εκτελέσει βασικές λειτουργίες χωρίς να χρειάζεται να γνωρίζει πως θα μπορούσε να τις πραγματοποιήσει μέσω της γραμμής εντολών.

Η μπάρα πλοήγησης δίνει στο χρήστη τη δυνατότητα να γνωρίζει την ακριβή τοποθεσία της κλάσης στην οποία δουλεύει την παρούσα χρονική στιγμή, καθώς και να μπορεί να μεταφερθεί εύκολα και γρήγορα σε άλλο σημείο του έργου, από τη στιγμή που βρίσκεται στο “path” της κλάσης στην οποία εργάζεται.

Ο χώρος συγγραφής κώδικα περιέχει έτοιμα πρότυπα κώδικα και αυτόματη συμπλήρωση διευκολύνοντας τη συγγραφή του κώδικα. Επίσης υπάρχει εντοπισμός σφαλμάτων και πολλές φορές παρουσιάζονται προτάσεις σχετικά με τον τρόπο επίλυσής τους.

Τα παράθυρα εργαλείων βοηθούν στον προσανατολισμό του χρήστη στο έργο καθώς και στην εκτέλεση εντολών σε γραμμή εντολών. Προσφέρουν οργάνωση και προσανατολισμό του προγραμματιστή στο έργο. Επίσης, το παλιό παράθυρο βοηθάει τον προγραμματιστή να μεταφερθεί από τη μία κλάση του Project στην άλλη με μεγάλη ευκολία και ταχύτητα.

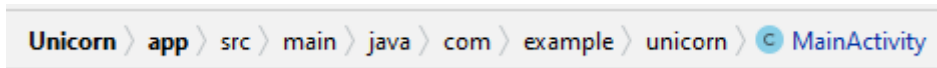
Η γραμμή κατάστασης δηλώνει την τρέχουσα κατάσταση της εφαρμογής (π.χ. αν κατεβαίνει κάποια βιβλιοθήκη, αν είναι συνδεδεμένη με “git”, κλπ).



Εικόνα 3.1: Το περιβάλλον του Android Studio

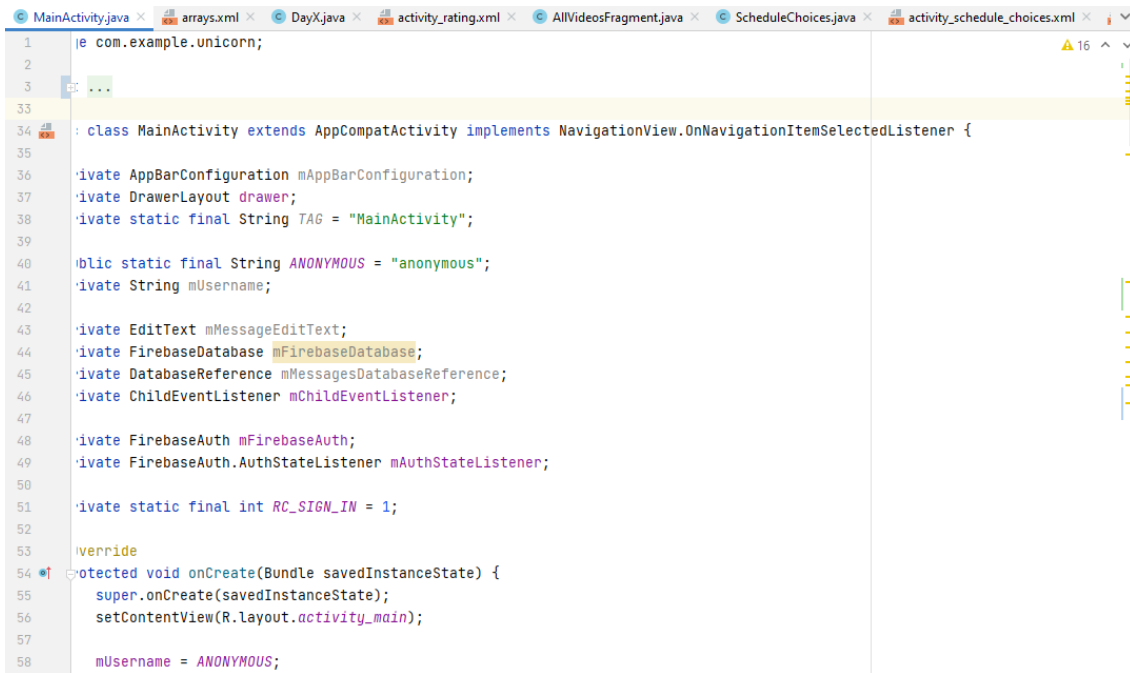


Εικόνα 3.2: Android Studio – Toolbar



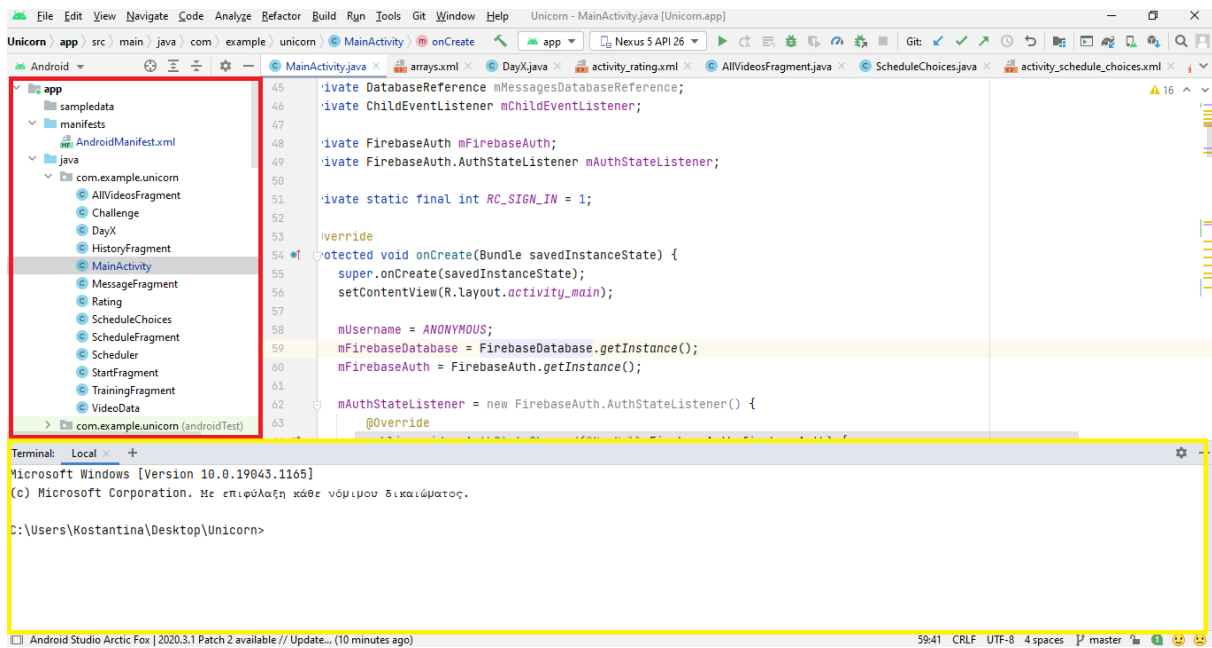
Εικόνα 3.3: Android Studio - Navigation Bar

## Κεφάλαιο 3

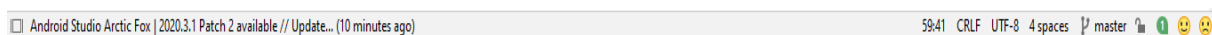


```
1 | je com.example.unicorn;
2 |
3 |
33 |
34 | : class MainActivity extends AppCompatActivity implements NavigationView.OnNavigationItemSelectedListener {
35 |
36 |     private AppBarConfiguration mAppBarConfiguration;
37 |     private DrawerLayout drawer;
38 |     private static final String TAG = "MainActivity";
39 |
40 |     public static final String ANONYMOUS = "anonymous";
41 |     private String mUsername;
42 |
43 |     private EditText mMessageEditText;
44 |     private FirebaseDatabase mFirebaseDatabase;
45 |     private DatabaseReference mMessagesDatabaseReference;
46 |     private ChildEventListener mChildEventListener;
47 |
48 |     private FirebaseAuth mFirebaseAuth;
49 |     private FirebaseAuth.AuthStateListener mAuthStateListener;
50 |
51 |     private static final int RC_SIGN_IN = 1;
52 |
53 |     override
54 |     protected void onCreate(Bundle savedInstanceState) {
55 |         super.onCreate(savedInstanceState);
56 |         setContentView(R.layout.activity_main);
57 |
58 |         mUsername = ANONYMOUS;
```

Εικόνα 3.4: Android Studio - Editor



Εικόνα 3.5: Android Studio – Tool Windows



Εικόνα 3.6: Android Studio - Status Bar

## 3.2 Apache NetBeans

Το “Apache NetBeans” είναι ένα προγραμματιστικό περιβάλλον για την ανάπτυξη “Java” εφαρμογών. Η πρώτη του εμφάνιση ήταν το 1996 με το όνομα “Xelfi” σαν μία εργασία φοιτητών του τμήματος Μαθηματικών και Φυσικής του “Charles University” (Πράγα). Το 1997 ο “Roman Stanek” δημιούργησε μία εταιρία για το έργο του και ξεκίνησε η εμπορική χρήση της πλατφόρμας. Το 1999 πουλήθηκε στην “Sun Microsystems” και το 2010 αγοράστηκε από την “Oracle” (το “NetBeans IDE”). Αφού το “NetBeans” συνέχισε να αναπτύσσεται και να εξελίσσεται και να αποκτά έδαφος στον χώρο του προγραμματισμού, δόθηκε τον Οκτώβριο του 2016 στην “Apache” και έκτοτε συνεχίζει να παρέχεται δωρεάν από αυτή. Στην παρούσα πτυχιακή εργασία χρησιμοποιήθηκε για τη δημιουργία ενός αρχείου κώδικα με “Java”, χρησιμοποιώντας την τεχνολογία “JSOUP” προκειμένου να αποσπάσει δεδομένα και πληροφορίες από βίντεο προπονήσεων που βρίσκονται δωρεάν στον παγκόσμιο ιστό και να τα παρέχει στους χρήστες της εφαρμογής [20].

### 3.2.1 Γενικά χαρακτηριστικά

Κάποια από τα πιο σημαντικά χαρακτηριστικά του Apache NetBeans παρουσιάζονται παρακάτω:

- Χρησιμοποιεί πρότυπα κώδικα.
- Μπορεί να συνδεθεί με “Git”.
- Υποστηρίζει τον προγραμματισμό εφαρμογών “Java”
- Έχει επεκτάσεις για “PHP”, “C”, “C++”, “HTML5” και “Javascript”
- Περιέχει εντοπισμό σφαλμάτων και προτάσεις πιθανών λύσεων.
- Προσφέρει επισήμανση σύνταξης και έξυπνη ολοκλήρωση κώδικα.

### 3.2.2 Το περιβάλλον του Apache NetBeans

Στην παρακάτω εικόνα (Εικόνα 3.7) απεικονίζεται το περιβάλλον του Apache NetBeans το οποίο αποτελείται από τη μπάρα εργαλείων (Εικόνα 3.8), τον χώρο συγγραφής κώδικα (Εικόνα 3.9), τα παράθυρα εργαλείων (Εικόνα 3.10) και τη γραμμή κατάστασης (Εικόνα 3.11).

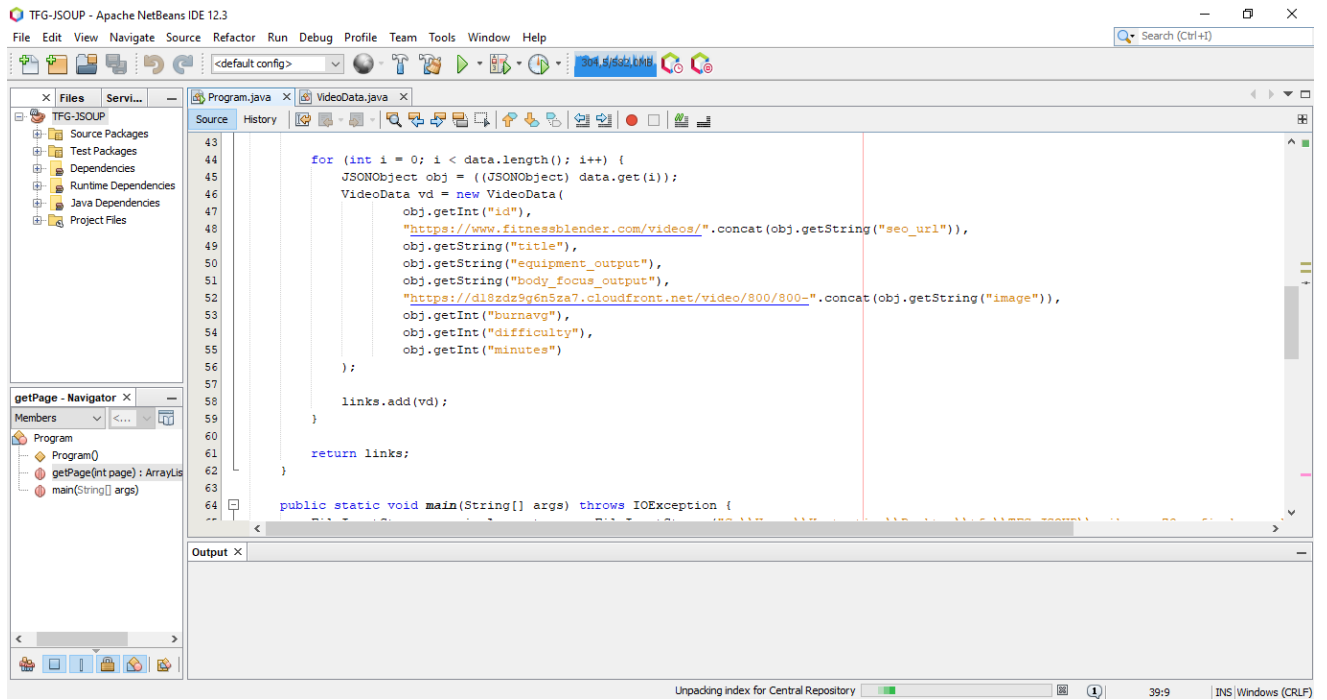
Όπως και στο “Android Studio”, η μπάρα εργαλείων διευκολύνει την ανάπτυξη της εφαρμογής από τον χρήστη, παρέχοντάς του μία σειρά από δυνατότητες όπως εκτέλεση κώδικα, δημιουργία νέου αρχείου, κλπ.

Ο χώρος συγγραφής κώδικα δίνει τη δυνατότητα στο χρήστη να χρησιμοποιήσει έτοιμα πρότυπα κώδικα καθώς και να εντοπίσει σφάλματα στον κώδικά του και πιθανές λύσεις τους. Επιπλέον του παρέχετε η δυνατότητα αυτόματης συμπλήρωσης.

Τα παράθυρα εργαλείων διευκολύνουν τον χρήστη στο να εντοπίζει γρήγορα μία κλάση στο έργο και να πηγαίνει σε αυτή. Επίσης δίνουν τη δυνατότητα στο χρήστη να χρησιμοποιήσει τη γραμμή εντολών.

Τέλος, η γραμμή κατάστασης παρουσιάζει την τρέχουσα κατάσταση της εφαρμογής καθώς και κάποια βασικά χαρακτηριστικά της.

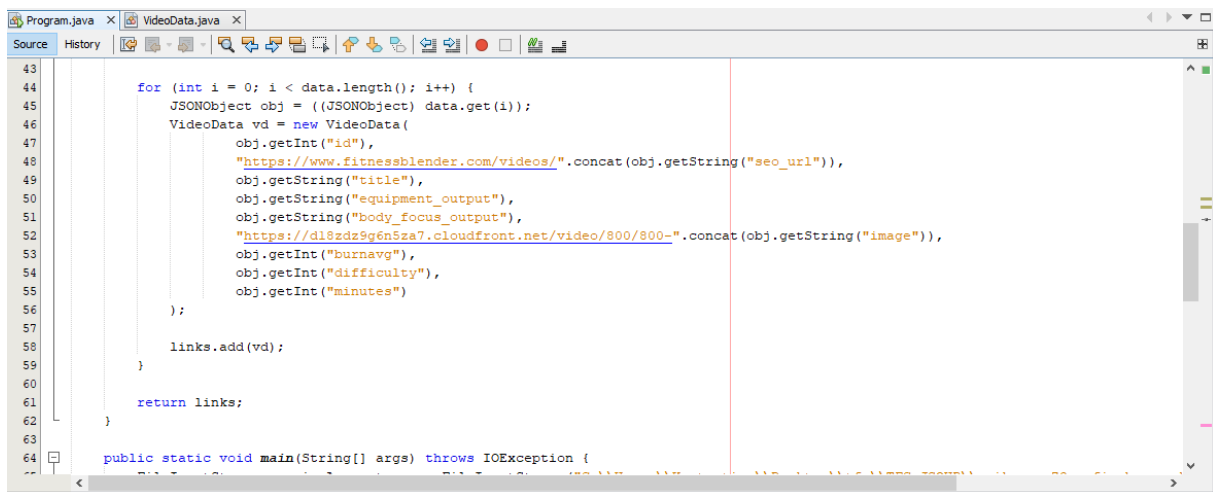
## Κεφάλαιο 3



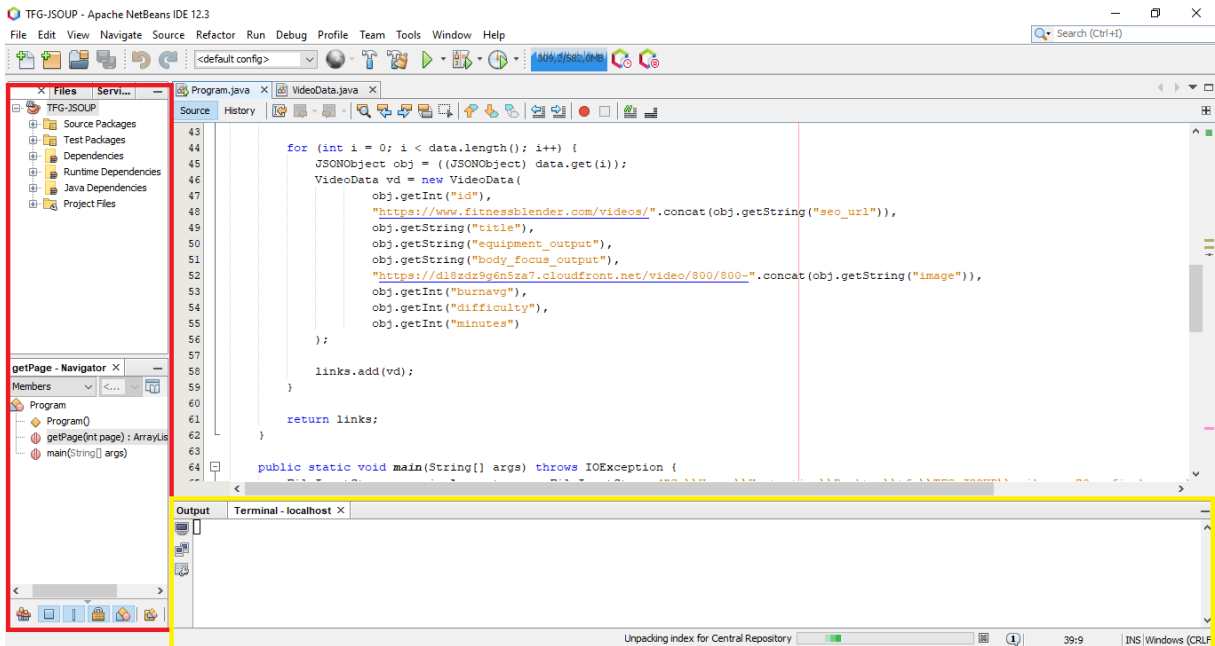
Εικόνα 3.7: Το περιβάλλον του Apache NetBeans



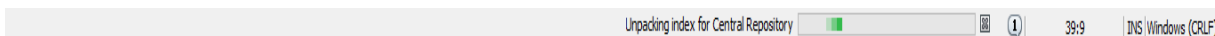
Εικόνα 3.8: Apache NetBeans- Toolbar



Εικόνα 3.9: Apache NetBeans – Editor



Εικόνα 3.10: Apache NetBeans – Tool Windows



Εικόνα 3.11: Apache NetBeans – Status Bar

### 3.3 Visual Studio Code

#### 3.3.1 Γενικά χαρακτηριστικά

Κάποια από τα πιο σημαντικά χαρακτηριστικά του “Visual Studio Code” παρουσιάζονται παρακάτω:

- Χρησιμοποιεί πρότυπα κώδικα.
- Μπορεί να συνδεθεί με “Git”.
- Υποστηρίζει τις πιο δημοφιλείς γλώσσες προγραμματισμού, όπως “Node.js”, “Python”, “C++”, “C#”, “PHP”, “Javascript”, “HTML”, “Java”, “CSS” και “TypeScript”.
- Περιέχει εντοπισμό σφαλμάτων.
- Προσφέρει επισήμανση σύνταξης και έξυπνη ολοκλήρωση κώδικα.

#### 3.3.2 Το περιβάλλον του Visual Studio Code

Στην παρακάτω εικόνα (Εικόνα 3.12) απεικονίζεται το περιβάλλον του “Visual Studio Code” το οποίο αποτελείται από τη μπάρα εργαλείων (Εικόνα 3.13), τη μπάρα πλοήγησης (Εικόνα 3.14), τον χώρο συγγραφής κώδικα (Εικόνα 3.15), τα παράθυρα εργαλείων (Εικόνα 3.16) και τη γραμμή κατάστασης (Εικόνα 3.17) [18][19].

## Κεφάλαιο 3

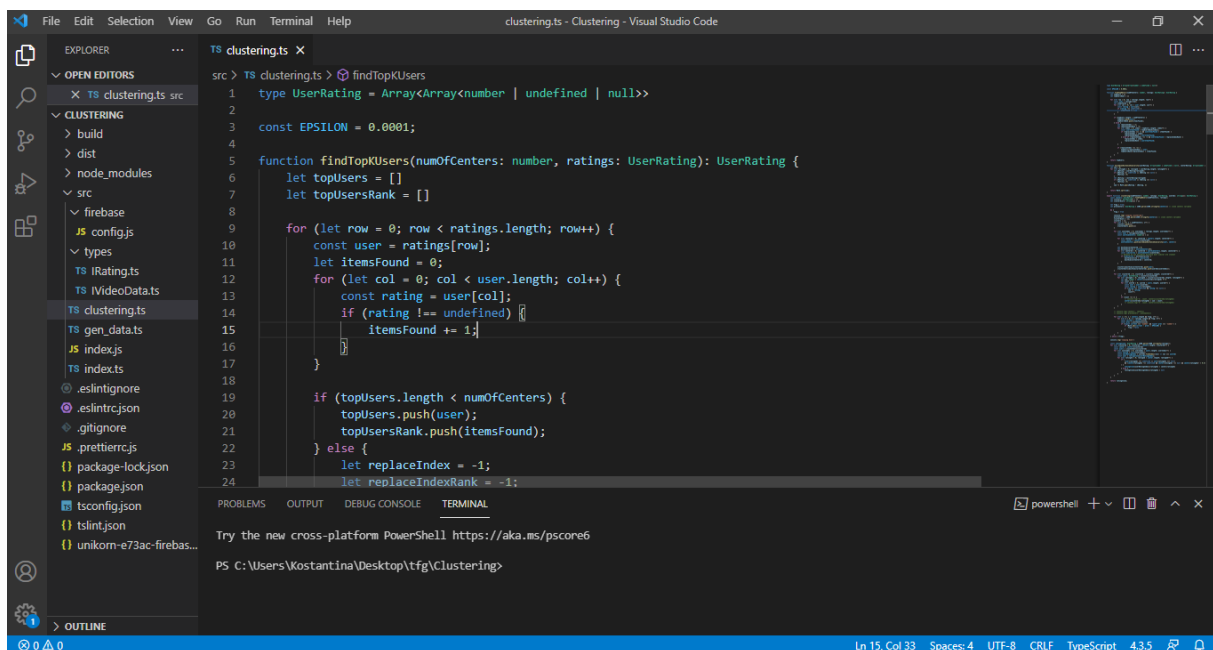
Η μπάρα εργαλείων παρέχει στον χρήστη μία σειρά από δυνατότητες, όπως αναζήτηση μέσα στο έργο, σύνδεση με “Git” και αναζήτηση, με σκοπό τη διευκόλυνση του χρήστη.

Με την μπάρα πλοήγησης ο χρήστης έχει τη δυνατότητα να γνωρίζει ανά πάσα χρονική στιγμή το “path” της κλάσης και της μεθόδου στην οποία βρίσκεται και να μετακινηθεί με άνεση σε έναν φάκελο / μία κλάση του “path” απλά πατώντας πάνω στο όνομά του.

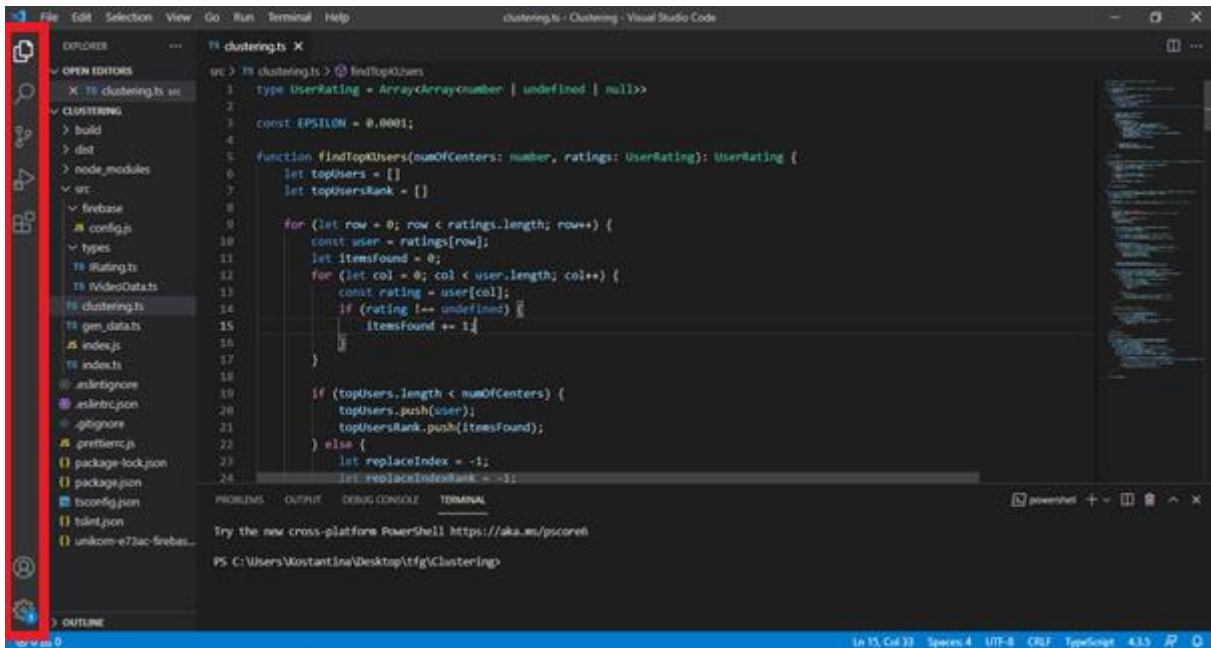
Ο χώρος συγγραφής κώδικα έχει αρίθμηση και τη δυνατότητα της αυτόματης συμπλήρωσης κώδικα και του εντοπισμού σφαλμάτων. Υποστηρίζει μία τεράστια γκάμα γλωσσών προγραμματισμού, όπως η “Javascript”, η “Typescript” και η “Python”.

Τα παράθυρα εργαλείων επιτρέπουν την εκτέλεση του προγράμματος (το κάτω παράθυρο) και διευκολύνουν την πλοήγηση του χρήστη μέσα στο έργο (το αριστερά παράθυρο).

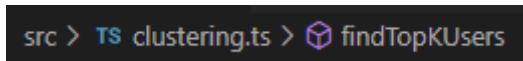
Η γραμμή κατάστασης δηλώνει κάποια στοιχεία της εφαρμογής, π.χ. τα προγραμματισμένα κενά στις εντολές και τη γλώσσα προγραμματισμού που χρησιμοποιείται και εκτελείται στο έργο.



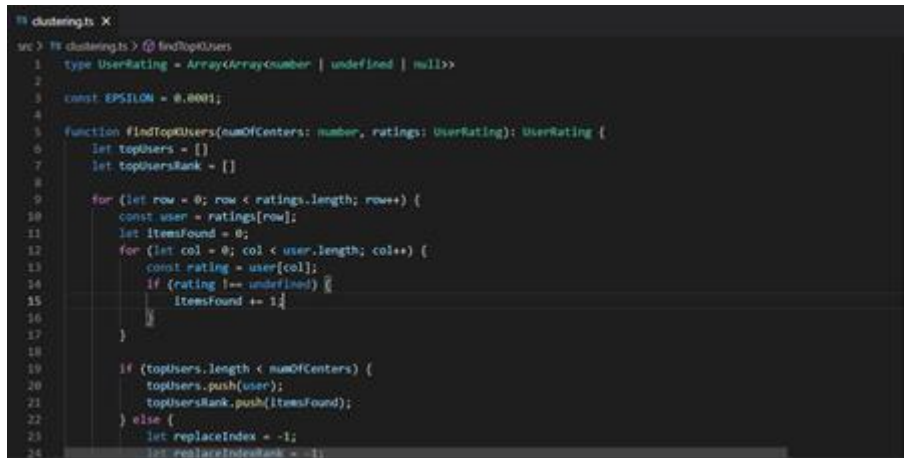
Εικόνα 3.12: Το περιβάλλον του Visual Studio Code



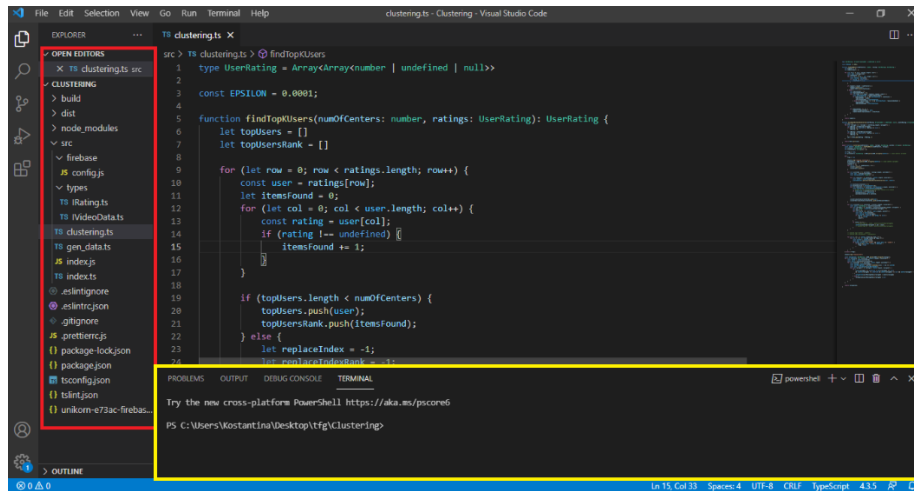
Εικόνα 3.13: Visual Studio Code – Toolbar



Εικόνα 3.14: Visual Studio Code – Navigation Bar



Εικόνα 3.15: Visual Studio Code – Editor



Εικόνα 3.16: Visual Studio Code – Tool Windows



Εικόνα 3.17: Visual Studio Code – Status Bar

### 3.4 Firebase

Η “Firebase” αρχικά ιδρύθηκε σαν μία ανεξάρτητη εταιρία (το 2011) και το 2014 την απέκτησε η “Google”. Υποστηρίζει ανάπτυξη λογισμικού για ιστοσελίδες και εφαρμογές κινητών και παρέχει πάρα πολλές δυνατότητες στον προγραμματιστή. Στη συγκεκριμένη εφαρμογή χρησιμοποιήθηκε σαν βάση δεδομένων, για τον έλεγχο των εισαγόμενων στοιχείων καθώς και για την αυθεντικοποίηση των χρηστών. Αξίζει να αναφερθεί ότι με τη “Firebase” όλα τα δεδομένα που αποθηκεύτηκαν στη βάση, αποθηκεύτηκαν στη “Realtime Database” με αποτέλεσμα να είναι διαρκώς ενημερωμένα [3][4][21][22].

#### 3.4.1 Γενικά χαρακτηριστικά

Κάποια από τα πιο σημαντικά χαρακτηριστικά της “Firebase” παρουσιάζονται παρακάτω:

- Παρέχει βάση δεδομένων
- Παρέχει δυνατότητα σύνδεσης εφαρμογών μεταξύ τους
- Υποστηρίζει καταγραφή αντιδράσεων χρήστη
- Παρέχει μεθόδους αυθεντικοποίησης χρηστών
- Ένα από τα πιο σημαντικά χαρακτηριστικά της είναι η ασφάλεια. Δίνει τη δυνατότητα στον προγραμματιστή να βάλει τους δικούς του κανόνες ασφαλείας και αν αδιαφορήσει και τα στοιχεία της βάσης βρεθούν σε κίνδυνο τον ενημερώνει και απενεργοποιεί κάθε κίνηση, προς το ανασφαλές σύνολο δεδομένων, μέχρι να αλλάξει τους κανόνες ασφαλείας ο προγραμματιστής.

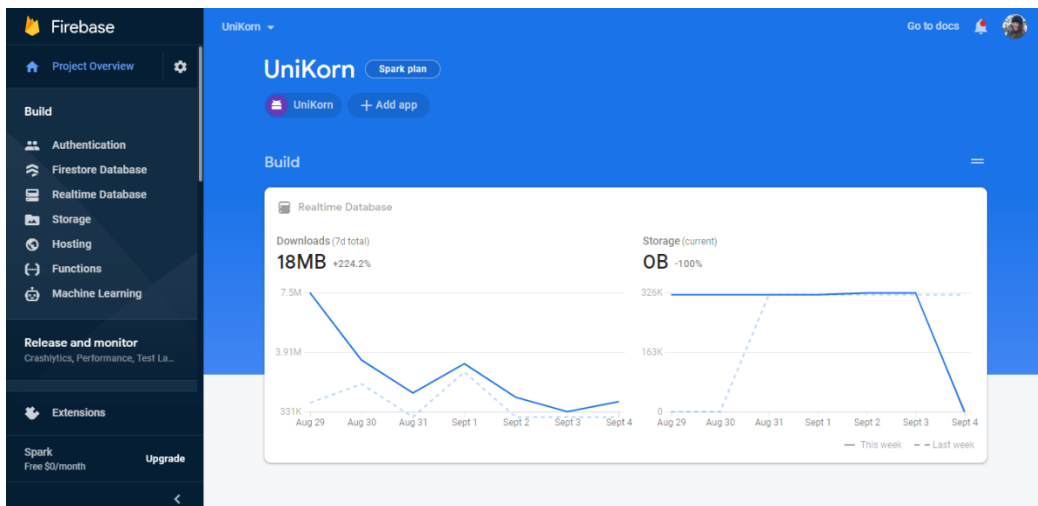
### 3.4.2 Η κονσόλα της Firebase (Firebase Console)

Η κονσόλα της “Firebase” (Εικόνα 3.18) δίνει τη δυνατότητα στο χρήστη να έχει πλήρη έλεγχο της εφαρμογής του, να δίνει δικαιώματα στους χρήστες πάνω στη βάση δεδομένων, κλπ και του παρέχει έτοιμες προτάσεις κώδικα για τη σωστή διαχείριση της εφαρμογής και της βάσης του. Από όλες τις δυνατότητες που παρέχει η “Firebase Console” για την παρούσα πτυχιακή χρησιμοποιήθηκαν η “Realtime Database” (Εικόνα 3.19) και το “Authentication” (Εικόνα 3.21).

Στην Εικόνα 3.18 βλέπουμε τη “Realtime Database” και πιο συγκεκριμένα βλέπουμε τη βάση δεδομένων της εφαρμογής. Στα “data” είναι τα στοιχεία των βίντεο, στο feedback οι αναφορές και τα σχόλια των χρηστών, στο “history” το ιστορικό του κάθε χρήστη, στα ratings οι βαθμολογίες των χρηστών στα βίντεο που έχουν παρακολουθήσει, στο “scheduler” το προσωπικό πρόγραμμα κάθε χρήστη και το “suggestions” οι προτάσεις βίντεο που δίνονται στον χρήστη με βάση το προφίλ του.

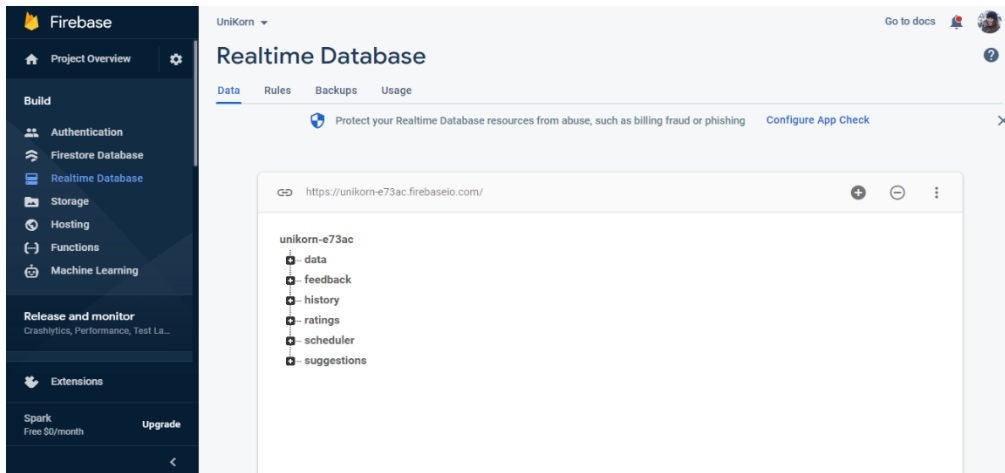
Μία εύλογη απορία που δημιουργείται όταν βλέπουμε όλα αυτά τα δεδομένα που αποθηκεύονται στη βάση είναι πόσο ασφαλές είναι αυτό για τους χρήστες. Για αυτό η “Firebase” παρέχει στον προγραμματιστή μία καρτέλα για τους κανόνες πάνω στη βάση δεδομένων (Εικόνα 3.20) και εάν ο προγραμματιστής δεν βάλει κανέναν κανόνα δεν επιτρέπει την λειτουργία της βάσης δεδομένων έως ότου το κάνει. Έτσι, δίνοντας για παράδειγμα το δικαίωμα να δει τις ψηφοφορίες που έχει δώσει μόνο για το προφίλ του ο αυθεντικοποιημένος χρήστης, αποτρέπει την διάρρευση δεδομένων χρηστών και παρέχει μεγαλύτερη ασφάλεια.

Στο “Authentication” (Εικόνα 3.21) ο προγραμματιστής μπορεί να δει τα στοιχεία των αυθεντικοποιημένων χρηστών. Αξίζει να αναφερθεί ότι και πάλι όλος ο κώδικας για την σύνδεση της εφαρμογής με τη Firebase παρέχεται από την πλατφόρμα όταν ο προγραμματιστής επιλέξει να έχει αυθεντικοποιημένους χρήστες στην εφαρμογή του. Φυσικά, για λόγους ασφαλείας και προστασίας προσωπικών δεδομένων, ο προγραμματιστής δεν έχει πρόσβαση στον κωδικό του χρήστη παρά μόνο σε βασικές πληροφορίες όπως το id που του δόθηκε κατά την αυθεντικοποίησή του και ο τρόπος σύνδεσής του (π.χ. “mail”).

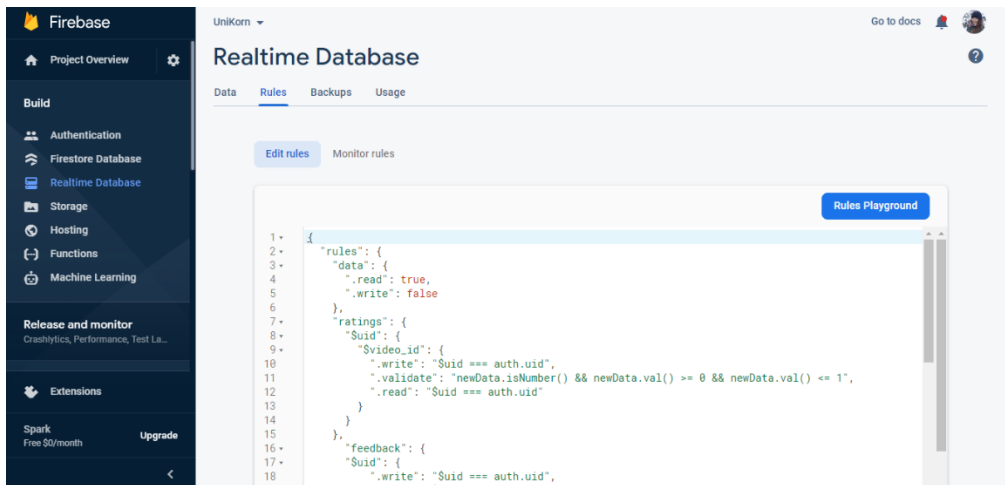


Εικόνα 3.18: Firebase Console

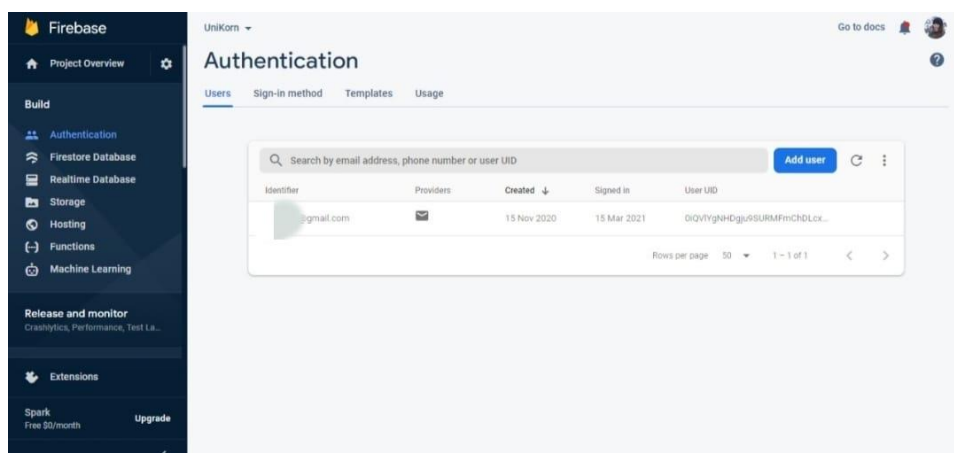
## Κεφάλαιο 4



Εικόνα 3.19: Firebase - Realtime Database



Εικόνα 3.20: Firebase - Realtime Database - Rules



Εικόνα 3.21: Firebase - Authentication

## Κεφάλαιο 4<sup>ο</sup>: Υλοποίηση εφαρμογής

### 4.1 Εισαγωγή στην εφαρμογή

Η εφαρμογή Unikorn είναι μία εφαρμογή που υποστηρίζει χρήστες που λόγω των αυξημένων απαιτήσεων της σημερινής κοινωνίας που οδηγεί σε έλλειψη χρόνου και, ίσως και, διάθεσης για γυμναστική και υγιεινό τρόπο ζωής χρειάζονται μία παρακίνηση. Η εφαρμογή εστιάζει στις ανάγκες τους και τους δίνει τη δυνατότητα δωρεάν παροχής προσωπικού προγράμματος γυμναστικής που μπορούν να πραγματοποιήσουν και στο σπίτι τους.

Είναι μία εφαρμογή για Android, που σημαίνει ότι ο χρήστης μπορεί να την έχει εγκατεστημένη στο κινητό του και να έχει πρόσβαση σε αυτήν ανά πάσα χρονική στιγμή. Με αυτόν τον τρόπο, ένας άνθρωπος που δεν έχει πολύ χρόνο διαθέσιμο για γυμναστική, θα μπορεί να εκμεταλλευτεί τον ελεύθερό του χρόνο, είτε βρίσκεται στο σπίτι του, είτε στο γήπεδο είτε στη δουλειά του. Υπάρχουν βίντεο μικρής χρονικής διάρκειας που δεν απαιτούν εξοπλισμό, με αποτέλεσμα οι ασκήσεις τους να μπορούν να εκτελεστούν οπουδήποτε.

Τα βίντεο με τις οδηγίες για την πραγματοποίηση των απαιτήσεων καθώς και οι εικόνες είναι προϊόντα που βρίσκονται ελεύθερα στο διαδίκτυο και βρίσκονται συγκεντρωμένα στην κατηγορία “All Videos” της εφαρμογής. Επίσης μέσω ενός προγράμματος τεχνητής νοημοσύνης, ο χρήστης, αφού έχει αξιολογήσει κάποια βίντεο που παρακολούθησε, έχει τη δυνατότητα να δει στην επιλογή “More Training” βίντεο που ταιριάζουν στο προφίλ του. Έτσι, ο χρήστης δεν χρειάζεται ούτε να έχει πολλή μνήμη στο κινητό του ούτε να ψάχνει με τις ώρες κάποιο βίντεο που να καλύπτει τις ανάγκες του (αυτό αυτόματα σημαίνει ότι κερδίζει χρόνο και διάθεση).

Πιο συγκεκριμένα, όποτε ο χρήστης αξιολογεί κάποιο βίντεο, η βαθμολογία που του βάζει αποθηκεύεται στη Firebase και, με ένα πρόγραμμα γραμμένο σε Typescript, ανά τακτά χρονικά διαστήματα ο διαχειριστής πραγματοποιεί κατηγοριοποίηση των χρηστών και παρέχει στον χρήστη τη δυνατότητα να παρακολουθήσει βίντεο που σύμφωνα με τις βαθμολογίες που έχει θέσει υπάρχουν μεγάλες πιθανότητες να του αρέσουν. Οπότε, το πρόγραμμα δεν στηρίζεται σε παροχή απαραίτητα παρόμοιων βίντεο, αλλά σε βίντεο που άρεσαν σε χρήστες με παρόμοια προφίλ.

Επιπλέον, κάθε φορά που ο χρήστης δηλώνει ότι έχει παρακολουθήσει ένα βίντεο, πέραν του ότι κάνει αξιολόγηση οπότε δέχεται προτάσεις για άλλα βίντεο που πιθανόν να του αρέσουν, αποθηκεύεται ο τίτλος του βίντεο στο ιστορικό του χρήστη προκειμένου να γνωρίζει και να οργανώνει τις προπονήσεις του. Για την καλύτερη οργάνωση των προπονήσεών του ο χρήστης έχει τη δυνατότητα να φτιάξει το δικό του προσωπικό πρόγραμμα γυμναστικής καθώς και να προσθέσει στο ιστορικό του δραστηριότητες που πραγματοποίησε εκτός εφαρμογής (π.χ. τρέξιμο, ποδηλασία, κλπ).

Επίσης, στην αρχική οθόνη του χρήστη καθώς και στην οθόνη του εβδομαδιαίου πλάνου του, εμφανίζεται στον χρήστη το κουμπί “Challenge”, το οποίο οδηγεί σε μία δραστηριότητα που αποτελεί πρόκληση για τον χρήστη προκειμένου να μην χάσει το ενδιαφέρον του για την εφαρμογή και γενικά για την δραστηριοποίησή του. Οι προκλήσεις θα ανανεώνονται ανά τακτά χρονικά διαστήματα από τον διαχειριστή.

## Κεφάλαιο 4

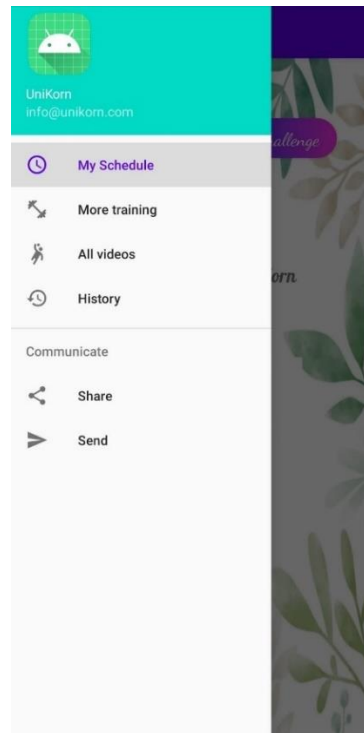
Τέλος, δίνεται στον χρήστη η δυνατότητα να αξιολογήσει την εφαρμογή, ανά πάσα χρονική στιγμή, προκειμένου η “Unicorn” να αποτελέσει μία συνεχώς εξελισσόμενη εφαρμογή που θα καλύπτει διαρκώς τις ανάγκες των χρηστών της.



Εικόνα 4.1: Αρχική οθόνη [25][26][27]

Στην Εικόνα 4.1 απεικονίζεται η αρχική οθόνη, έπειτα από τη σύνδεση του χρήστη.

### 4.2 Το κεντρικό μενού



Εικόνα 4.2: Κεντρικό μενού

Στην Εικόνα 4.2 απεικονίζεται το κεντρικό μενού. Αποτελείται από το “My Schedule”, το “More Training”, το “All Videos”, το “History”, το “Share” και το “Send”, τα οποία θα αναλύσουμε παρακάτω.



My Schedule: Μας πηγαίνει στο “fragment” με το προσωπικό πρόγραμμα του χρήστη

Εκεί, ο χρήστης θα μπορεί να δει το πρόγραμμά του και να παρακολουθήσει το βίντεο που έχει τοποθετήσει στη συγκεκριμένη μέρα της εβδομάδας.



More training: Μας πηγαίνει στο “fragment” με τα προτεινόμενα βίντεο για τον χρήστη, με βάση το προφίλ του (βγαίνει από έναν αλγόριθμο που παίρνει σαν είσοδο τις αξιολογήσεις που κάνει στα βίντεο ο χρήστης και χρησιμοποιώντας κατηγοριοποίηση και τον αλγόριθμο “k-means” κατηγοριοποιεί τους χρήστες και επιστρέφει τα προτεινόμενα βίντεο, θα το αναλύσουμε παρακάτω)



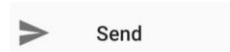
All videos: Μας πηγαίνει στο “fragment” που φορτώνει όλα τα βίντεο. Από εκεί, ακόμη και ένας χρήστης που δεν έχει ψηφίσει ακόμη κανένα βίντεο, μπορεί να ψάξει κάτι συγκεκριμένο π.χ. “Pilates” ή να πατήσει κάποιο βίντεο στην τύχη και να το βάλει στο πρόγραμμά του ή να το παρακολουθήσει εκείνη τη στιγμή και να το ψηφίσει, δημιουργώντας σιγά σιγά το προφίλ του.



History: Μας πηγαίνει στο “fragment” με το ιστορικό του χρήστη. Μία επιπλέον δυνατότητα που έχει ο χρήστης συγκριτικά με άλλες παρόμοιες εφαρμογές που κυκλοφορούν, είναι να μπορεί να προσθέσει στο πρόγραμμά του και δραστηριότητες που έχει πραγματοποιήσει εκτός εφαρμογής προκειμένου να οργανώνει και να χειρίζεται όπως ο ίδιος θέλει τις προπονήσεις του.



Share: Αυτή τη στιγμή δεν λειτουργεί αλλά στο μέλλον θα χρησιμοποιείται για να μπορεί ένας χρήστης να προωθήσει την εφαρμογή.



Send: Εμφανίζει στην οθόνη ένα “alert dialog” για να γράψει ο χρήστης μία αναφορά και να την στείλει απευθείας στον διαχειριστή της εφαρμογής.

Το παραπάνω μενού βρίσκεται μέσα στο αρχείο “drawer\_menu.xml”. Όπως βλέπουμε στην Εικόνα 4.3, το μενού αποτελείται από 2 κατηγορίες επιλογών, η μία εκ των οποίων περιέχει τις επιλογές που έχουν να κάνουν με την χρήση της εφαρμογής για την προπόνηση και τον προγραμματισμό προγράμματος του χρήστη και η άλλη περιέχει τις επιλογές που έχουν να κάνουν με την επικοινωνία με τον προγραμματιστή της εφαρμογής (για την αποστολή αναφοράς) και την προώθηση της εφαρμογής σε κάποιον “φίλο” (αυτή η επιλογή δεν έχει υλοποιηθεί ακόμη). Όπως παρατηρούμε, όλες οι επιλογές έχουν ένα “id” μέσω του οποίου τις εντοπίζουμε από την “main” κλάση (Εικόνα 4.4), μία εικόνα και έναν τίτλο, ο οποίος εμφανίζεται στον χρήστη.

```

<group android:checkableBehavior="single">
  <item
    android:id="@+id/nav_schedule"
    android:icon="@drawable/ic_schedule"
    android:title="My Schedule" />
  <item
    android:id="@+id/nav_training"
    android:icon="@drawable/ic_training"
    android:title="More training" />
  <item
    android:id="@+id/nav_allvideos"
    android:icon="@drawable/ic_allvideos"
    android:title="All videos" />
  <item
    android:id="@+id/nav_history"
    android:icon="@drawable/ic_history"
    android:title="History" />
</group>

<item android:title="Communicate">
  <menu>
    <item
      android:id="@+id/nav_share"
      android:icon="@drawable/ic_share"
      android:title="Share" />
  </menu>
</item>

```

Εικόνα 4.3: Απόσπασμα του “drawer\_menu.xml”

```

switch (item.getItemId()) {
    case R.id.nav_schedule:
        getSupportFragmentManager().beginTransaction().replace(R.id.fragment_container,
            new ScheduleFragment()).commit();
        this.setTitle("My schedule");
        break;
    case R.id.nav_training:
        getSupportFragmentManager().beginTransaction().replace(R.id.fragment_container,
            new TrainingFragment()).commit();
        this.setTitle("More training");
        break;
    case R.id.nav_allvideos:
        getSupportFragmentManager().beginTransaction().replace(R.id.fragment_container,
            new AllVideosFragment()).commit();
        this.setTitle("All videos");
        break;
    case R.id.nav_history:
        getSupportFragmentManager().beginTransaction().replace(R.id.fragment_container,
            new HistoryFragment()).commit();
        this.setTitle("History");
        break;
    case R.id.nav_share:
        Toast.makeText( context: this, text: "Share", Toast.LENGTH_SHORT).show();
        break;
    case R.id.nav_send:
        AlertDialog alertDialog = new AlertDialog.Builder( context: this).create();
        alertDialog.setTitle("Report");
        alertDialog.setMessage("Feel free to give us a feedback and/or tell us something you don't like! :)");
        alertDialog.setView(textLinearLayout);
}

```

Εικόνα 4.4: Απόσπασμα της “MainActivity.java” – Κεντρικό μενού

### 4.3 Οθόνη προγράμματος χρήστη (Schedule fragment)



Εικόνα 4.5: Η οθόνη του “My Schedule”

Μία από τις δυνατότητες που παρέχονται στον χρήστη της εφαρμογής είναι η δημιουργία και τροποποίηση ενός προσωπικού, εβδομαδιαίου πλάνου. Τα κουμπάκια “Monday”, “Tuesday”, “Wednesday”, “Thursday”, “Friday”, “Saturday” και “Sunday” περιέχουν το πρόγραμμα της ημέρας που αναγράφεται πάνω τους. Το κουμπάκι “Challenge” περιέχει την πρόκληση που έχει οριστεί από τον διαχειριστή με σκοπό την συνεχή παρότρυνση του χρήστη να συνεχίσει να χρησιμοποιεί την εφαρμογή και να μην χάσει τον ενθουσιασμό του για έναν πιο υγιεινό τρόπο ζωής.

Όπως βλέπουμε στην Εικόνα 4.6, για κάθε κουμπί ημέρας, στο “ScheduleFragment”, ελέγχεται αν υπάρχει κάποιο βίντεο αποθηκευμένο στην ημέρα που πατήθηκε (στο απόσπασμα την Πέμπτη) και αν υπάρχει, καλείται η κλάση “DayX” με τα στοιχεία του βίντεο που έχει προγραμματίσει ο χρήστης για εκείνη την ημέρα. Αν δεν υπάρχει αποθηκευμένο βίντεο, δεν γίνεται τίποτα.

Στην Εικόνα 4.7 βλέπουμε πως είναι αποθηκευμένα τα βίντεο στη βάση δεδομένων. Οι κωδικοί 717, 9, κλπ είναι τα “id” των βίντεο τα οποία μετά από το πάτημα του κουμπιού ημέρας πηγαίνουν σαν παράμετροι στην κλάση “DayX” προκειμένου να εμφανιστεί το βίντεο που έχει επιλέξει ο χρήστης για την συγκεκριμένη ημέρα που επιλέχτηκε.

Όταν ο χρήστης πατήσει το κουμπί μίας ημέρας, θα μεταφερθεί αυτόματα στο activity “DayX”, το οποίο απεικονίζεται στην Εικόνα 4.21 και περιλαμβάνει ένα βίντεο από το “youtube”, ένα κουμπί μέσω του οποίου ο χρήστης μπορεί να τοποθετήσει το βίντεο σε οποιαδήποτε μέρα του εβδομαδιαίου προγράμματός του και ένα κουμπί “Done” μέσω του οποίου το βίντεο καταγράφεται στο ιστορικό του χρήστη και ο χρήστης έχει τη δυνατότητα να αξιολογήσει το συγκεκριμένο βίντεο προκειμένου να λαμβάνει πιο εξειδικευμένες προτάσεις για τις επόμενες προπονήσεις του (στο fragment “More training”). Θα αναφερθούμε στη δραστηριότητα “DayX” με περισσότερες λεπτομέρειες στις επόμενες ενότητες.

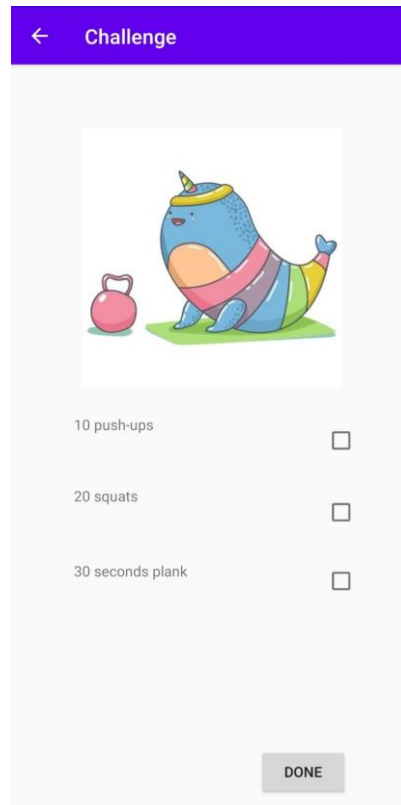
```
thursday.setOnClickListener(v -> {
    VideoData data = videoDataMap.get(schedule.get("thu"));
    if (data == null) {
        return;
    }
    Intent myIntent = new Intent(ScheduleFragment.this.getActivity(), DayX.class);
    myIntent.putExtra("videoId", data.getVideo());
    myIntent.putExtra("id", data.getId());
    ScheduleFragment.this.startActivity(myIntent);
});
```

Εικόνα 4.6: “ScheduleFragment” – Κουμπιά ημέρας

The screenshot shows the Firebase Scheduler interface. At the top, it displays the scheduler name "scheduler" and its instance ID "0iQVIYgNHDgju9SURMFmChDLcxP2". Below this, a list of days and their corresponding video IDs is shown:

Day	Video ID
fri	"717"
mon	"9"
sat	"155"
thu	"191"
tue	"360"
wed	"383"

Εικόνα 4.7: Firebase – Scheduler



Εικόνα 4.8: Ένα παράδειγμα Challenge

#### 4.4 Οθόνη προτεινόμενων προπονήσεων (Training fragment)

Όταν ο χρήστης χρησιμοποιήσει για πρώτη φορά την εφαρμογή η συγκεκριμένη οθόνη θα εμφανιστεί κενή. Αυτό συμβαίνει επειδή δεν έχει αξιολογήσει ακόμη κανένα βίντεο οπότε δεν υπάρχουν στοιχεία όσον αφορά το προφίλ του χρήστη.

Από τη στιγμή που ο χρήστης θα ξεκινήσει να χρησιμοποιεί την εφαρμογή και να ψηφίζει πια βίντεο του αρέσουν και πια όχι, ο διαχειριστής θα είναι σε θέση να τρέξει τον αλγόριθμο τεχνητής νοημοσύνης και αυτόματα να εμφανιστούν προτεινόμενα προγράμματα προπόνησης για τον χρήστη. Αυτό σημαίνει ότι ο χρήστης ακόμη και τις ημέρες που δεν έχει χρόνο για να μπει στη διαδικασία να ψάξει πρόγραμμα για γυμναστική, αν ανοίξει την εφαρμογή και πάει στη συγκεκριμένη δραστηριότητα, θα είναι σε θέση να βρει βίντεο που η πιθανότητα να του διεγείρουν το ενδιαφέρον είναι αυξημένη. Θα αναφερθούμε στον αλγόριθμο τεχνητής νοημοσύνης στις επόμενες ενότητες.

Στην Εικόνα 4.10 μπορούμε να δούμε ένα κομμάτι από τον κώδικα του “fragment”. Αφού γίνει έλεγχος σχετικά με το αν υπάρχουν ή όχι προτάσεις προπονήσεων για τον συγκεκριμένο χρήστη, θα εκτελεστεί το συγκεκριμένο κομμάτι του κώδικα προκειμένου με αυτή τη λούπα να εμφανιστούν εικονίτσες από όλα τα βίντεο και πατώντας τις εικονίτσες ο χρήστης να μπορεί να παρακολουθήσει κάποιο από αυτά.

Οι εικόνες που υπάρχουν στην οθόνη του “More training” είναι κουμπιά και οδηγούν στην δραστηριότητα “DayX” η οποία απεικονίζεται στην Εικόνα 4.21. Μέσω αυτής της δραστηριότητας, ο χρήστης έχει τη δυνατότητα να προσθέσει το βίντεο στο εβδομαδιαίο πλάνο του ή/και να ολοκληρώσει

την προπόνηση και να την αξιολογήσει. Θα αναφερθούμε με περισσότερες λεπτομέρειες στο παραπάνω Activity στις επόμενες ενότητες.



Εικόνα 4.9: Η οθόνη του “More training”

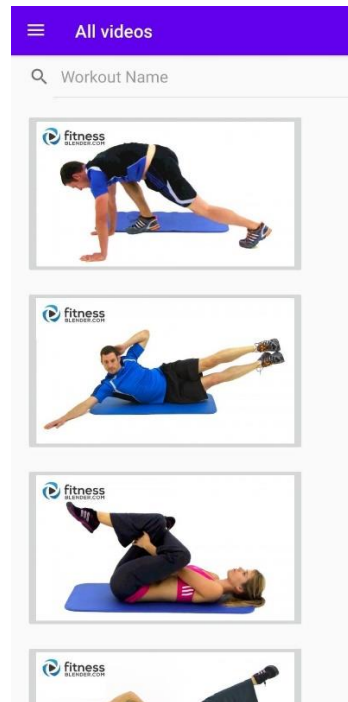
```
for (String videoId : userSuggestions.keySet()) {
    VideoData data = videoDataMap.get(videoId);

    ImageButton imageButton = new ImageButton(container.getContext());
    imageButton.setOnClickListener(v -> {
        Intent intent = new Intent (container.getContext(), DayX.class);
        intent.putExtra( name: "videoId", data.getVideo());
        intent.putExtra( name: "id", data.getId());
        intent.putExtra( name: "title", data.getBody_focus_output());
        startActivity(intent);
    });

    Thread th = new Thread(new Runnable() {
        @Override
        public void run() {
            try {
                URL url = new URL(data.getImage());
                Bitmap bpm = BitmapFactory.decodeStream(url.openConnection().getInputStream());
                getActivity().runOnUiThread(new Runnable() {
                    @Override
                    public void run() { imageButton.setImageBitmap(bpm); }
                });
            } catch (MalformedURLException e) {
                Log.getStackTraceString(e);
            } catch (IOException e) {
                Log.getStackTraceString(e);
            }
        }
    });
}
```

Εικόνα 4.10: Λούπα προτάσεων - More training

### 4.5 Οθόνη προπονήσεων (AllVideos fragment)



Εικόνα 4.11: Η οθόνη του “All videos”

Στην οθόνη προπονήσεων υπάρχουν κουμπιά με εικόνες προπονήσεων και ένα “search view”. Οι εικόνες οδηγούν στη δραστηριότητα “DayX” (Εικόνα 4.21) και στη θέση του βίντεο υπάρχει αυτό που έχει επιλεγεί από τον χρήστη. Το “search view” μπορεί να χρησιμοποιηθεί από τον χρήστη σε περίπτωση που θέλει να ψάξει κάποιο βίντεο μία συγκεκριμένης κατηγορίας, π.χ. “yoga”, “pilates”, κλπ. Οι επιλογές που έχει ο χρήστης στο συγκεκριμένο “fragment” δεν είναι οι προτεινόμενες από τις αξιολογήσεις των χρηστών, όπως συμβαίνει με το fragment “More training” αλλά είναι όλες οι επιλογές βίντεο που έχει ο χρήστης, τις οποίες λαμβάνουμε από την ιστοσελίδα “www.fitnessblender.com” με τη βοήθεια ενός αρχείου κώδικα που θα εξηγήσουμε στις επόμενες ενότητες. Έτσι ο χρήστης έχει πολλές επιλογές για προπονήσεις είτε για τη στιγμή που αναζητά προπόνηση είτε για να τις τοποθετήσει στο πρόγραμμά του για να τις πραγματοποιήσει μέσα κάποια άλλη στιγμή μέσα στην εβδομάδα του.

Τα στοιχεία των βίντεο παίρνονται από τη βάση δεδομένων και για την επεξεργασία τους χρησιμοποιείται η “HashMap” της Εικόνας 4.12. Με τη δημιουργία “HashMap” μπορούμε να διαχειριστούμε και να επεξεργαστούμε πιο εύκολα δεδομένα όπως αυτά των βίντεο.

Στην Εικόνα 4.13 βλέπουμε ένα κομμάτι του κώδικα του “AllVideosFragment” που περιλαμβάνει τη μέθοδο που χρησιμοποιεί το “search” προκειμένου ο χρήστης να έχει πρόσβαση στα βίντεο που αναζητεί (π.χ. αν πληκτρολογήσει “pilates” να του εμφανίζονται μόνο βίντεο προπονήσεων “pilates”). Ελέγχει αν αυτό που πληκτρολόγησε ο χρήστης ταιριάζει με τον τίτλο ή την κατηγορία του βίντεο και αν ναι, το επιστρέφει.

```

videoDataMap = new HashMap<>();

ArrayList<HashMap<String, Object>> list = (ArrayList<HashMap<String, Object>>) snapshot.getValue();

for (HashMap<String, Object> data: list) {
    VideoData vd = new VideoData(
        ((Long) data.get("id")).intValue(),
        (String) data.get("url"),
        (String) data.get("title"),
        (String) data.get("equipment_output"),
        (String) data.get("body_focus_output"),
        (String) data.get("image"),
        ((Long) data.get("burnavg")).intValue(),
        ((Long) data.get("difficulty")).intValue(),
        ((Long) data.get("minutes")).intValue()
    );
    vd.setVideo((String) data.get("video"));
    videoDataMap.put(String.valueOf(vd.getId()), vd);
}

try {
    setupImages(container);
} catch (IOException e) {
    e.printStackTrace();
}

```

Εικόνα 4.12: “AllVideosFragment” - HashMap

```

private void setSearch (ViewGroup container, String txt) throws IOException {
    if (videoDataMap == null) {
        return;
    }

    if (txt.equals("")) {
        buildButtons(container, videoDataMap);
        return;
    }

    Map<String, VideoData> searchedVideos = new HashMap<>();

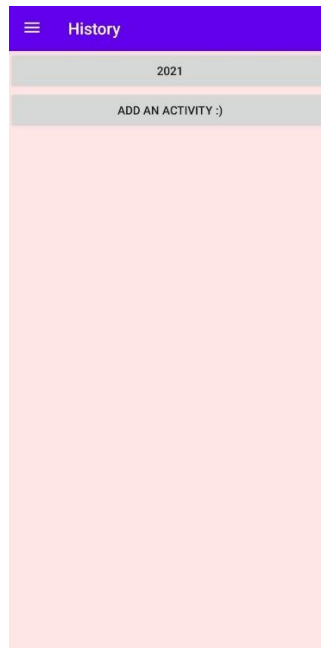
    for (VideoData vd : videoDataMap.values()) {
        if (
            vd.getTitle().toLowerCase().contains(txt.toLowerCase()) ||
            vd.getBody_focus_output().toLowerCase().contains(txt.toLowerCase())
        ) {
            searchedVideos.put(String.valueOf(vd.getId()), vd);
        }
    }

    buildButtons(container, searchedVideos);
}

```

Εικόνα 4.13: “AllVideosFragment” – Search option

## 4.6 Οθόνη ιστορικού (History fragment)



Εικόνα 4.14: Η οθόνη του “History”

Στο “fragment” καταγραφής ιστορικού ο χρήστης μπορεί να δει μία καταγραφή των προπονήσεών του πατώντας αρχικά το κουμπί με την χρονολογία για την οποία ενδιαφέρεται (Εικόνα 4.13), έπειτα πατώντας το κουμπί στο οποίο αναγράφεται ο μήνας στον οποίο πραγματοποιήθηκε την συγκεκριμένη προπόνηση (Εικόνα 4.14) και τέλος πατώντας το κουμπί στο οποίο αναγράφεται η ημέρα της προπόνησης με τη μορφή 05 → Για 5 του μήνα, 26 → Για 26 του μήνα, κλπ (Εικόνα 4.15).

Στη συγκεκριμένη οθόνη ο χρήστης μπορεί να διακρίνει και ένα κουμπί που γράφει “ADD AN ACTIVITY :)”. Πατώντας το συγκεκριμένο κουμπί, ο χρήστης μπορεί να προσθέσει στο ιστορικό του κάποια δραστηριότητα που πραγματοποίησε εκτός εφαρμογής, όπως τρέξιμο, κολύμβηση, κλπ προκειμένου να έχει μία πλήρη εικόνα των προπονήσεων που έχει πραγματοποιήσει και εντός και εκτός της εφαρμογής.

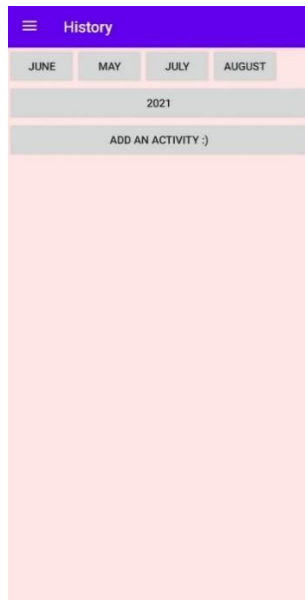
Στην Εικόνα 4.17 απεικονίζεται η οθόνη που εμφανίζεται στον χρήστη όταν πατήσει το κουμπί “ADD AN ACTIVITY :)” προκειμένου να προσθέσει στο ιστορικό του μία προπόνηση που πραγματοποίησε εκτός εφαρμογής. Με το κουμπί “+” προστίθεται η δραστηριότητα στο ιστορικό του χρήστη και εμφανίζεται μήνυμα επιβεβαίωσης ή μήνυμα λάθους εισαγωγής στοιχείων (Εικόνα 4.18).

Στην Εικόνα 4.19 παρατηρούμε ότι αφού γίνει mapping στις ημερομηνίες παίρνουμε τις δραστηριότητες ανάλογα με τις επιλογές που κάνει ο χρήστης. Έτσι, ο χρήστης μπορεί να επιλέξει να δει το πρόγραμμά του για μια συγκεκριμένη ημερομηνία. Στην Εικόνα 4.20 βλέπουμε ένα κομμάτι του κώδικα που εκτελείται με σκοπό να δίνει την επιλογή στον χρήστη να προσθέτει στο πρόγραμμά του δραστηριότητες που πραγματοποίησε εκτός της εφαρμογής.

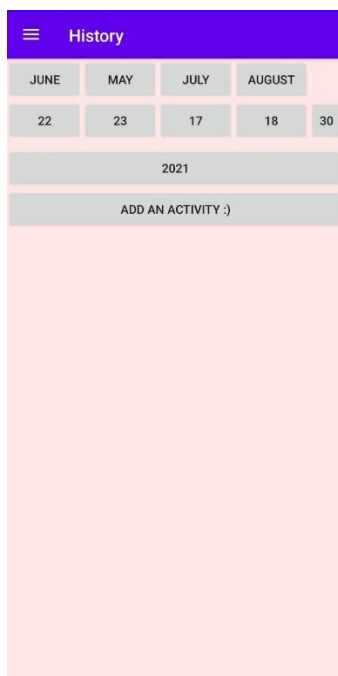
Αξίζει να σημειωθεί ότι η παραπάνω δυνατότητα που δίνεται στον χρήστη (η δυνατότητα να προσθέτει στο πρόγραμμά του δραστηριότητες που πραγματοποίησε εκτός εφαρμογής) μπορεί να χρησιμοποιηθεί και για δραστηριότητες άλλου τύπου και όχι μόνο για γυμναστική. Για παράδειγμα, μπορεί να χρησιμοποιηθεί για να κρατήσει ο χρήστης ημερολόγιο των ωρών που διάβασε και στον τίτλο δραστηριότητας να σημειώσει αυτά που τέλειωσε ή αυτά που θέλει να διαβάσει την επόμενη μέρα.

## Κεφάλαιο 4

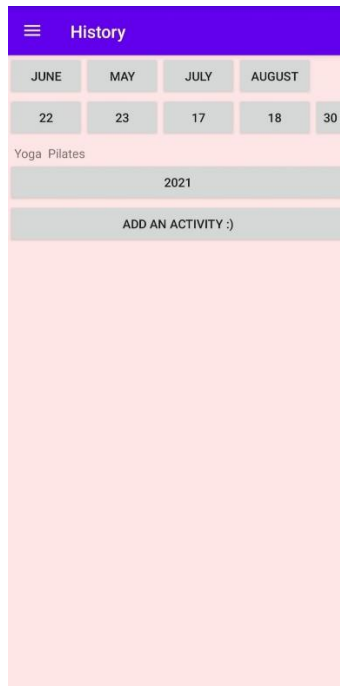
Γενικά, το “History fragment” δεν δημιουργήθηκε μόνο για να εξυπηρετεί στην δραστηριοποίηση του χρήστη αλλά και στη γενική οργάνωση της ζωής και τις ρουτίνες του.



Εικόνα 4.15: Η οθόνη του “History” αφού πατηθεί κουμπί χρονολογίας



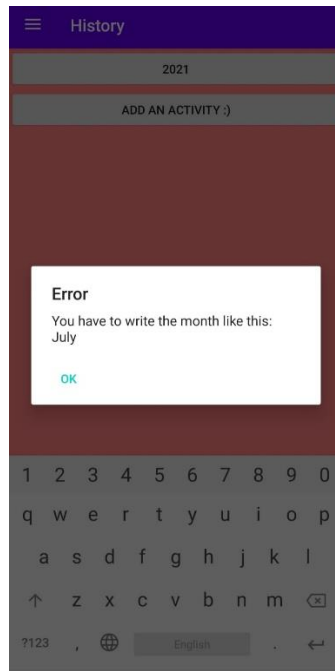
Εικόνα 4.16: Η οθόνη του “History” αφού πατηθεί κουμπί μήνα



Εικόνα 4.17: Η οθόνη του “History” αφού πατηθεί κουμπί ημέρας



Εικόνα 4.18: Η οθόνη του “History” αφού πατηθεί κουμπί εισαγωγής δραστηριότητας



Εικόνα 4.19: Μήνυμα λάθους εισαγωγής στοιχείων

```

for(String histDay : historyMonth.keySet()) {
    Button dayButton = setButtons(container, histDay, textLinearLayout3);

    dayButton.setOnClickListener(v -> {
        DatabaseReference historyActRef = database.getReference(path: "history").child(uid).child(histYear).child(histMonth).child(histDay);

        historyActRef.addListenerForSingleValueEvent(new ValueEventListener() {

            public void onDataChange(@NonNull @NotNull DataSnapshot snapshot) {
                historyDay = (Map<String, Double>) snapshot.getValue();
                textLinearLayout4.removeAllViews();
                for(String histAct : historyDay.keySet()) {
                    int len = historyDay.size();
                    setTextViews(container, histAct, textLinearLayout4);
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {

            }
        });
    });
}

```

Εικόνα 4.20: “HistoryFragment” – Εμφάνιση δραστηριότητων

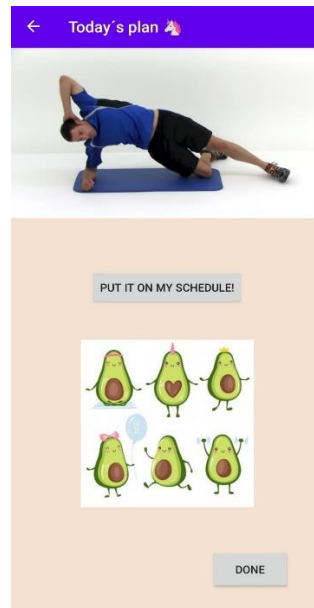
```

Button newActivity = setButtons(container, hist: "Add an activity :)", textLinearLayout);
newActivity.setOnClickListener(v -> {
    EditText persAct = new EditText(container.getContext());
    persAct.setHint("Activity");
    EditText monthAct = new EditText(container.getContext());
    monthAct.setHint("Month");
    EditText dayAct = new EditText(container.getContext());
    dayAct.setHint("Day");
    Button plus = new Button(container.getContext());
    plus.setText("+");
    toAddAnActivity.addView(persAct);
    toAddAnActivity.addView(monthAct);
    toAddAnActivity.addView(dayAct);
    toAddAnActivity.addView(plus);
    plus.setOnClickListener(w -> {
        String act = persAct.getText().toString();
        String monthStr = monthAct.getText().toString();
        String day = dayAct.getText().toString();
        Calendar calendar = Calendar.getInstance();
        int year = calendar.get(Calendar.YEAR);
        Boolean isValid = true;
        if (!monthStr.equals("January") && !monthStr.equals("February") && !monthStr.equals("March") && !monthStr.equals("April") && !monthStr.equals("May") && !monthStr.equals("June") && !monthStr.equals("July") && !monthStr.equals("August") && !monthStr.equals("September") && !monthStr.equals("October") && !monthStr.equals("November") && !monthStr.equals("December")) {
            isValid = false;
        }
        if (Integer.parseInt(day) > 31) {
            isValid = false;
        }
        if (isValid) {
            database.getReference(path: "history").child(uid).child(String.valueOf(year)).child(monthStr).child(day).child(act).setValue(true);
            Toast.makeText(container.getContext(), text: "Saved!! ^^", Toast.LENGTH_SHORT).show();
        }
    });
});
}
}

```

Εικόνα 4.21: “HistoryFragment” – Προσθήκη δραστηριότητας

## 4.7 DayX activity



Εικόνα 4.22: Η οθόνη του “DayX activity”

Στην οθόνη που εμφανίζεται όταν ο χρήστης επιλέξει την προβολή ενός βίντεο προπόνησης μπορούμε να παρατηρήσουμε μία οθόνη “youtube”, ένα κουμπί που γράφει “PUT IT ON MY SCHEDULE!” και ένα κουμπί που γράφει “DONE”. Στην οθόνη “youtube” ο χρήστης μπορεί να παρακολουθήσει το βίντεο που επέλεξε.

Πατώντας το κουμπί “DONE” ο χρήστης δηλώνει ότι πραγματοποίησε την προπόνηση και η κατηγορία της προπόνησής του καταγράφεται στο ιστορικό του, ενώ ο ίδιος μεταφέρεται σε μία οθόνη αξιολόγησης στην οποία θα αναφερθούμε παρακάτω. Με αυτό τον τρόπο ο χρήστης μπορεί να είναι διαρκώς ενημερωμένος σχετικά με τις προπονήσεις του και να είναι πιο οργανωμένος. Όπως θα δούμε παρακάτω, εκτός από καταγραφή των δραστηριοτήτων, στο ιστορικό μπορούν να καταγραφούν και δραστηριότητες που έχει πραγματοποιήσει ο χρήστης και εκτός εφαρμογής.

Εάν ο χρήστης δεν επιθυμεί να πραγματοποιήσει την προπόνηση εκείνη τη χρονική στιγμή, μπορεί να πατήσει το κουμπί “PUT IT ON MY SCHEDULE!” και κατευθείαν θα μεταφερθεί στην οθόνη του scheduler activity μέσω του οποίου μπορεί να αποθηκεύσει το βίντεο σε οποιαδήποτε ημέρα της εβδομάδας αυτός επιθυμεί, προκειμένου, όταν θα φτάσει η συγκεκριμένη μέρα, να πατήσει το κουμπί στο οποίο αναγράφεται το όνομά της στην οθόνη του προγράμματος (Εικόνα 4.21) και να πραγματοποιήσει την προπόνησή του. Αυτή η επιλογή υπάρχει για την καλύτερη οργάνωση των προπονήσεων του χρήστη. Επίσης, με αυτό τον τρόπο ο χρήστης δεν χρειάζεται να παρακολουθεί ένα σύνολο προπονήσεων σχεδιασμένων για έναν στόχο που δεν είναι απαραίτητα ο δικός του, αλλά να θέτει από μόνος του ένα πρόγραμμα προπονήσεων ειδικά προσαρμοσμένο στις ανάγκες του.

Στην παρακάτω εικόνα βλέπουμε ένα κομμάτι κώδικα σχετικό με τα κουμπιά του “activity”. Όπως βλέπουμε στο στιγμιότυπο, το κουμπί “done” προσθέτει το βίντεο στο ιστορικό του χρήστη και το κουμπί “scheduler” (με text: “put it on my schedule!”) οδηγεί τον χρήστη στην κλάση “Scheduler” περνώντας σαν παράμετρο το id του βίντεο. Στη συνέχεια, ο χρήστης θα κληθεί να διαλέξει μία ημέρα της εβδομάδας για να τοποθετηθεί το βίντεο σε αυτή (δηλαδή στο εβδομαδιαίο πρόγραμμά του) και όταν φτάσει αυτή η μέρα της εβδομάδας ή και οποιαδήποτε στιγμή, επιλέγοντας εκείνη την ημέρα, θα μπορεί να το παρακολουθήσει. Χωρίς απαραίτητα να κατεβάσει το βίντεο στο κινητό του και να

εξαντλήσει τη μνήμη της συσκευής του αλλά ούτε και να χάσει το βίντεο λόγω του ότι την παρούσα χρονική στιγμή δεν μπορεί να πραγματοποιήσει την προπόνηση.

```

Button Button = findViewById(R.id.btn_done);

Button.setOnClickListener(v -> {
    final FirebaseDatabase database = FirebaseDatabase.getInstance();
    database.getReference( path: "history").child(vid).child(String.valueOf(year)).child(monthStr).child(String.valueOf(day)).
    Intent intent = new Intent ( packageContext: this, Rating.class);
    intent.putExtra( name: "videoId", id);
    startActivity(intent);
});

Button scheduler = findViewById(R.id.btn_scheduler);

scheduler.setOnClickListener(v -> {
    Intent intent = new Intent ( packageContext: this, Scheduler.class);
    intent.putExtra( name: "videoId", id);
    startActivity(intent);
});

```

Εικόνα 4.23: DayX activity – buttons

## 4.8 Send Dialog

### Report

Feel free to give us a feedback and/or tell us something you dont like! :)

Feedback

OK

Εικόνα 4.24: Send Dialog

Εάν ο χρήστης επιλέξει από το κεντρικό μενού την επιλογή “Send”, του εμφανίζεται ένα “Dialog”, στο οποίο μπορεί αν σημειώσει ένα θετικό ή αρνητικό σχόλιο και πατώντας το OK το “feedback” του να αποθηκευτεί στη firebase και να μπορεί να το δει ο διαχειριστής του συστήματος. Είναι μία σημαντική πινελιά της εφαρμογής καθώς έχει σαν κέντρο το χρήστη και η γνώμη του είναι σημαντική.

Στην Εικόνα 4.24 μπορούμε να δούμε τον κώδικα του “send dialog”, ο οποίος βρίσκεται μέσα στη “MainActivity”. Όπως βλέπουμε στο απόσπασμα, ενώ όταν ο χρήστης επιλέγει κάτι από το κεντρικό μενού, ο κώδικας τον στέλνει σε μία άλλη κλάση ή “fragment”, όταν ο χρήστης επιλέγει να στείλει ένα μήνυμα (μια αναφορά) στον προγραμματιστή, ο κώδικας του ανοίγει ένα “dialog” (χωρίς να τον συνδέσει σε άλλη κλάση/fragment).

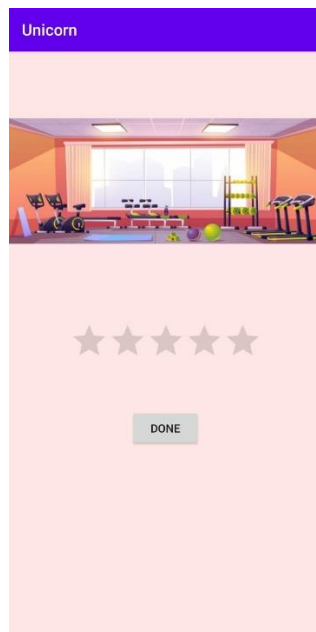
```

        new TrainingFragment()).commit();
        this.setTitle("More training");
        break;
    case R.id.nav_allvideos:
        getSupportFragmentManager().beginTransaction().replace(R.id.fragment_container,
            new ALLVideosFragment()).commit();
        this.setTitle("All videos");
        break;
    case R.id.nav_history:
        getSupportFragmentManager().beginTransaction().replace(R.id.fragment_container,
            new HistoryFragment()).commit();
        this.setTitle("History");
        break;
    case R.id.nav_share:
        Toast.makeText(context, this, text: "Share", Toast.LENGTH_SHORT).show();
        break;
    case R.id.nav_send:
        AlertDialog alertDialog = new AlertDialog.Builder(context, this).create();
        alertDialog.setTitle("Report");
        alertDialog.setMessage("Feel free to give us a feedback and/or tell us something you don't like! :)");
        alertDialog.setView(textLinearLayout);
        alertDialog.setButton(AlertDialog.BUTTON_NEUTRAL, text: "OK",
            new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int which) {
                    database.getReference(path: "feedback").child(uid).child(String.valueOf(feedback.getText())).setValue(
                        dialog.dismiss());
                }
            });
        alertDialog.show();
        break;

```

Εικόνα 4.25: Send Dialog Code

## 4.9 Rating activity



Εικόνα 4.26: Οθόνη αξιολόγησης βίντεο [5][28]

Η οθόνη αξιολόγησης που εμφανίζεται στον χρήστη έπειτα από την ολοκλήρωση κάποιας προπόνησής του, περιλαμβάνει ένα “Rating bar” και ένα κουμπί με το κείμενο “DONE”. Αξιοποιώντας τα αστεράκια του “Rating Bar” ο χρήστης είναι σε θέση να αξιολογήσει την προπόνηση που μόλις πραγματοποίησε και πατώντας το κουμπί “DONE” η ψηφοφορία του να αποθηκευτεί στη βάση.

Μέσω ενός προγράμματος τεχνητής νοημοσύνης γραμμένο σε “Typescript” και χρησιμοποιώντας την ευκλείδια απόσταση σαν μέτρο για την κατηγοριοποίηση, ο διαχειριστής είναι σε θέση να αξιοποιήσει τις πληροφορίες που είναι αποθηκευμένες στη βάση σχετικά με τις αξιολογήσεις που έχει κάνει ο χρήστης και να εμφανιστούν στο fragment “More training” καινούργια προτεινόμενα βίντεο σύμφωνα με το προφίλ του χρήστη. Θα αναφερθούμε με περισσότερες λεπτομέρειες σε αυτό το πρόγραμμα στις επόμενες ενότητες.

Το rating activity, όπως βλέπουμε στην Εικόνα 4.26 περιέχει μόνο ένα “rating bar” και ένα “button”. Όταν ο χρήστης πατάει το κουμπί, η βαθμολογία του αποθηκεύεται στα “ratings” (Εικόνα 4.27) και αξιοποιείται για την κατηγοριοποίηση του χρήστη, με σκοπό την παροχή προτάσεων προπονήσεων με βάση τα ενδιαφέροντά του.

```
public class Rating extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_rating);
        int id = getIntent().getIntExtra( name: "videoId", defaultValue: -1);

        FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
        String uid = user.getId();

        Button Button = findViewById(R.id.btn_rating);

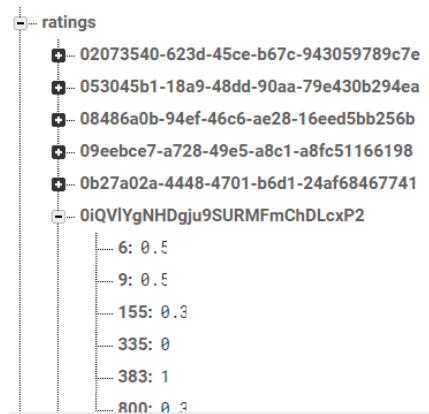
        Button.setOnClickListener(v -> {
            RatingBar ratingBar = (RatingBar) findViewById(R.id.ratingBar);
            double rate = ratingBar.getRating();

            final FirebaseDatabase database = FirebaseDatabase.getInstance();

            database.getReference( path: "ratings").child(uid).child(String.valueOf(id)).setValue(rate / 5.0);

            Intent intent = new Intent ( packageContext: this, MainActivity.class);
            startActivity(intent);
        });
    }
}
```

Εικόνα 4.27: Rating activity



Εικόνα 4.28: Firebase – ratings

#### 4.10 Scheduler activity



Εικόνα 4.29: Οθόνη κατανομής βίντεο σε ημέρες

Στη οθόνη του “scheduler” ο χρήστης μπορεί να δει 7 κουμπάκια, ένα για κάθε ημέρα της εβδομάδας. Όταν ο χρήστης πατήσει ένα από αυτά τα κουμπάκια, το βίντεο αυτόματα αποθηκεύεται στο εβδομαδιαίο πρόγραμμα του χρήστη και μέσω του “Fragment Schedule” ο χρήστης είναι σε θέση να παρακολουθήσει αυτή την προπόνηση πατώντας το αντίστοιχο κουμπί στην οθόνη του προσωπικού του προγράμματος (Εικόνα 4.2). Με αυτό τον τρόπο ο χρήστης οργανώνει τις προπονήσεις του και είναι σε θέση να διαμορφώσει μόνος του το εβδομαδιαίο πλάνο προπονήσεών του.

Όπως φαίνεται στο τμήμα του κώδικα της “Scheduler Activity”, στην Εικόνα 4.29, υπάρχει ένα κουμπί για κάθε μέρα. Στο παράδειγμα φαίνεται ο προγραμματισμός πίσω από το κουμπί της Τετάρτης. Κάθε φορά που επιλέγεται ένα από τα κουμπιά από το “scheduler”, το βίντεο το οποίο έχει επιλεγεί για εισαγωγή στο εβδομαδιαίο πρόγραμμα του χρήστη περνάει στη βάση προκειμένου να έχει πρόσβαση σε αυτό την ημέρα που το επιθυμεί. Αξίζει να σημειωθεί ότι για την αναγνώριση του βίντεο και την αναπαραγωγή του από την εφαρμογή, εκτός από το id χρησιμοποιείται και το αναγνωριστικό του βίντεο που ανακτάται απ’ το “youtube”. Για παράδειγμα, για το βίντεο “https://www.youtube.com/watch?v=FFHwoIQmFEA”, το προαναφερθέν αναγνωριστικό, είναι “FFHwoIQmFEA”. Ο τρόπος με τον οποίο πραγματοποιείται η αποκοπή αυτού του αποσπάσματος του συνδέσμου περιγράφεται στην επόμενη ενότητα (“JSOUP”).

```
wednesday.setOnClickListener(v -> {
    database.getReference( path: "scheduler").child(uid).child("wed").setValue(String.valueOf(id));
    Intent intent = new Intent ( packageContext: this, MainActivity.class);
    startActivity(intent);
});
```

Εικόνα 4.30: Scheduler Activity

## 4.11 JSOUP

Χρησιμοποιώντας την τεχνολογία “JSOUP” καταγράφηκε ένα κομμάτι κώδικα σε “Java” με σκοπό την αποθήκευση των βίντεο και των χαρακτηριστικών τους καθώς και των εικόνων, από την ιστοσελίδα “www.fitnessblender.com”. Στην Εικόνα 4.31 φαίνεται ένα μέρος του κώδικα.

```
Document doc = Jsoup.connect("https://www.fitnessblender.com/videos?page=" + concat(String.valueOf(page))).get()

Elements script = doc.select("body > script:nth-child(7)");

String[] lines = script.html().split("\n");

String jsonString = lines[0].substring(28, lines[0].length() - 1);

JSONObject json = new JSONObject(jsonString);

JSONArray data = json.getJSONArray("data");

ArrayList<VideoData> links = new ArrayList<>();

for (int i = 0; i < data.length(); i++) {
    JSONObject obj = ((JSONObject) data.get(i));
    VideoData vd = new VideoData(
        obj.getInt("id"),
        "https://www.fitnessblender.com/videos/" + concat(obj.getString("seo_url")),
        obj.getString("title"),
```

Εικόνα 4.31: Στιγμιότυπο από τον κώδικα που φέρνει τα δεδομένα από τη ιστοσελίδα

Μαζί με τα βίντεο, φροντίζουμε να πάρουμε και όλες τις πληροφορίες που σχετίζονται με αυτά, όπως ο χρόνος διάρκειας του βίντεο, ο βαθμός δυσκολίας, αν χρειάζεται ή όχι εξοπλισμό κλπ. Με αυτόν τον

τρόπο ακόμη και τα στοιχεία που αυτή τη στιγμή δεν χρειάζεται η εφαρμογή για να λειτουργήσει, θα μπορούν να χρησιμοποιηθούν για μελλοντική χρήση με σκοπό την παροχή επιπλέον δυνατοτήτων, όπως προτάσεις προπονήσεων με βάση τον χρόνο που διαθέτει ο χρήστης και ένα καλύτερο πρόγραμμα κατηγοριοποίησης που εκτός από τις ψηφοφορίες του χρήστη, θα παίρνει σαν δεδομένα και κάποια χαρακτηριστικά του βίντεο προκειμένου να δίνει καλύτερες προτάσεις και γενικά να είναι πιο αξιόπιστο.

Στην Εικόνα 4.32 μπορεί κανείς να παρατηρήσει ένα κομμάτι της κλάσης “VideoData”, του κώδικα, η οποία χρησιμοποιείται για τον παραπάνω σκοπό. Στην Εικόνα 4.33 μπορεί κανείς να δει πως αποθηκεύονται τελικά όλα αυτά τα στοιχεία στη βάση για να χρησιμοποιηθούν από το πρόγραμμα γραμμένο σε “Java for Android”, μέσω της “Firebase”.

```
public VideoData(int id, String url, String title, String equipment_output,
String body_focus_output, String image, int burnavg, int difficulty, int minutes) {
    this.id = id;
    this.url = url;
    this.title = title;
    this.equipment_output = equipment_output;
    this.body_focus_output = body_focus_output;
    this.image = image;
    this.burnavg = burnavg;
    this.difficulty = difficulty;
    this.minutes = minutes;
    this.video = "";
}
```

Εικόνα 4.32: Στιγμιότυπο της κλάσης VideoData

```
FirebaseApp.initializeApp(options);

ArrayList<VideoData> data = new ArrayList<>();

for (int i = 1; i <= 33; i++) {
    data.addAll(getPage(i));
}

for (VideoData vd : data) {
    Document doc1 = Jsoup.connect(vd.getUrl()).get();

    Element el = doc1.select("iframe").first();

    if (el == null) {
        data.remove(vd);
    } else {
        vd.setVideo(el.attr("src").substring(30, 41));
    }
}

final FirebaseDatabase database = FirebaseDatabase.getInstance();
DatabaseReference ref = database.getReference("data");

ref.setValueAsync(data);
```

Εικόνα 4.33: Αποθήκευση δεδομένων στη βάση (Program.java)

## 4.12 Clustering

Αξιοποιώντας την μέθοδο “k-means” και την ευκλείδια απόσταση πραγματοποιήθηκε η κατηγοριοποίηση των χρηστών. Είναι ένας κώδικας γραμμένος σε “Typescript” μέσω του οποίου κάθε φορά που ο χρήστης τον εφαρμόζει, οι χρήστες (επανα)κατηγοριοποιούνται, δηλαδή χωρίζονται σε ομάδες, με σκοπό να προσφέρονται στον κάθε χρήστη (στην οθόνη του “More training”) προτάσεις για βίντεο που υπάρχουν αυξημένες πιθανότητες να του αρέσουν λαμβάνοντας υπ’ όψιν τις αξιολογήσεις που έχει κάνει σε άλλα βίντεο προπονήσεων (η κατηγοριοποίηση γίνεται με βάση τις αξιολογήσεις των χρηστών). Στην Εικόνα 4.34 φαίνεται ένα μέρος του κώδικα “clustering” [34].

Στην Εικόνα 4.35 παρατηρούμε πως δημιουργείται ένα αντίγραφο του πίνακα αξιολογήσεων των χρηστών και στη συνέχεια, εάν πληρούνται κάποιες συνθήκες σχετικά με τη αξιολόγηση του χρήστη και του κέντρου (του “cluster” στο οποίο ανήκει ο χρήστης), το βίντεο αποθηκεύεται στις προτάσεις προπονήσεων που θα εμφανιστούν στον χρήστη.

```

flag = true

console.log('looping clustering')
prevCenters = JSON.parse(JSON.stringify(centers)) // clone centers variable
clusterUsers = [];
clusters = [];
for (let i = 0; i < numOfCenters; i++) {
  clusters.push([])
  clusterUsers.push([])
}

for (let userIndex = 0; userIndex < ratings.length; userIndex++) {
  const user = ratings[userIndex];
  const userSimCenters: number[] = []

  for (let centerId = 0; centerId < centers.length; centerId++) {
    const center = centers[centerId];
    userSimCenters.push(euclidianDistanceSimilarity(user, center))
  }

  let mostSimilarCenterId = 0;
  let mostSimilarity = userSimCenters[0];
  for (let centerId = 0; centerId < userSimCenters.length; centerId++) {
    const similarity = userSimCenters[centerId];
    // because we use euclidian distance most similar are closest
    if (similarity < mostSimilarity) {
      mostSimilarity = similarity;
      mostSimilarCenterId = centerId;
    }
  }
}

```

Εικόνα 4.34: Στιγμιότυπο από τον κώδικα κατηγοριοποίησης

```

const ratingClone: UserRating = JSON.parse(JSON.stringify(ratings));
for (let clusterId = 0; clusterId < centers.length; clusterId++) {
  const center = centers[clusterId]
  const users = clusterUsers[clusterId]
  for (let userIndex = 0; userIndex < users.length; userIndex++) {
    const userId = users[userIndex];
    const userRatingIndex = userIds.findIndex((val) => val === userId)
    const user = ratingClone[userRatingIndex];
    for (let ratingId = 0; ratingId < center.length; ratingId++) {
      if (
        (user[ratingId] === undefined || user[ratingId] === null)
        && (center[ratingId] !== undefined && center[ratingId] !== null && center[ratingId]! > 0.5)
      ) {
        ratingClone[userRatingIndex][ratingId] = center[ratingId]
      } else {
        ratingClone[userRatingIndex][ratingId] = null
      }
    }
  }
}

```

Εικόνα 4.35: Αντίγραφο πίνακα αξιολογήσεων χρηστών

### 4.13 Επίλογος

Μέσα από ένα κεντρικό μενού, δίνονται στον χρήστη της εφαρμογής πολλές δυνατότητες με στόχο την οργάνωση του προγράμματος προπονήσεων του αλλά και την παρότρυνσή του για έναν πιο υγιεινό τρόπο ζωής. Μέσα σε αυτές τις επιλογές είναι η παρακολούθηση κάποιου βίντεο προπόνησης και η αξιολόγησή του καθώς και η παροχή προτεινόμενων, από την εφαρμογή, βίντεο, με βάση το προφίλ του χρήστη.

Δεν είναι λίγες οι φορές που ένας χρήστης διακόπτει τη χρήση μιας εφαρμογής ευεξίας καθώς νιώθει χαμένος μέσα στην πολλή πληροφορία. Για αυτό το λόγο η εφαρμογή “Unikorn” είναι προσανατολισμένη προς τον χρήστη και τις ανάγκες του. Δίνει προτάσεις προπονήσεων στον χρήστη αξιοποιώντας τις αξιολογήσεις του προκειμένου να φτιάξει το προφίλ του και παρέχει τη δυνατότητα αποστολής της άποψης και των παραπόνων του, απ’ ευθείας στον διαχειριστή.

Τέλος, με τη δυνατότητα της απογραφής των προπονήσεων καθώς και της προσθήκης δραστηριοτήτων που πραγματοποίησε ο χρήστης εκτός εφαρμογής, η εφαρμογή βοηθάει και παροτρύνει χρήστες που δυσκολεύονται να οργανώσουν το πρόγραμμά τους. Αυτό το πετυχαίνει και με την παροχή ενός “fragment” για τη δημιουργία προσωπικού προγράμματος από τον ίδιο τον χρήστη προκειμένου να βρίσκονται στο πρόγραμμά του μόνο ασκήσεις/προγράμματα που του αρέσουν και να μην παραβλέπει κάποιο πρόγραμμά του λόγω έλλειψης χρόνου εντόπισης στην εφαρμογή.

## Κεφάλαιο 5ο: Επίλογος

### 5.1 Η Unikorn σήμερα

Η εφαρμογή “Unikorn” με την τωρινή της μορφή είναι σε θέση να βοηθήσει τόσο τα άτομα που δεν έχουν χρόνο ή/και διάθεση για γυμναστική λόγω του σύγχρονου τρόπου ζωής όσο και τα άτομα που επιθυμούν και έχουν ήδη βάλει τη γυμναστική στη ζωή τους και χρειάζονται μία διευκόλυνση για την οργάνωση της ρουτίνας τους με σκοπό την επίτευξη των στόχων τους.

Ο χρήστης της εφαρμογής έχει τη δυνατότητα να επιλέξει και να παρακολουθήσει κάποιο βίντεο μέσα από μία τεράστια ποικιλία από διαφορετικά είδη προπονήσεων. Χρησιμοποιώντας δε τον χώρο αναζήτησης μπορεί να ψάξει και να βρει κάτι συγκεκριμένο ανάλογα με την τότε διάθεσή του, χωρίς να χάσει χρόνο ψάχνοντας π.χ. κάποιο βίντεο “pilates” στο διαδίκτυο. Απ’ ευθείας μεταφέρεται σε μια ιστοσελίδα που του δείχνει το βίντεο από το “youtube” και με το πάτημα ενός κουμπιού αποθηκεύει αυτή τη δραστηριότητά του στο ιστορικό του και την αξιολογεί προκειμένου να έχει προτάσεις βίντεο σύμφωνα τον αλγόριθμο κατηγοριοποίησης που τον ομαδοποιεί με χρήστες με τα ίδια ενδιαφέροντα.

Επίσης παρέχεται στον χρήστη η δυνατότητα να αποθηκεύει τα βίντεο για να τα παρακολουθήσει κάποια άλλη χρονική στιγμή, τοποθετώντας τα σε ένα εβδομαδιαίο πλάνο προπονήσεων. Με αυτό τον τρόπο οργανώνεται και δεν χάνει χρόνο τις επόμενες μέρες που είναι πολύ πιθανό, αν δεν χρησιμοποιούσε την εφαρμογή, να παρέλειπε την προπόνησή του λόγω έλλειψης χρόνου για να ψάξει και να βρει το βίντεο προπόνησης που καλύπτει τις ανάγκες του.

Κάτι ακόμα που ξεχωρίζει τη “Unikorn” από άλλες αντίστοιχες εφαρμογές είναι η δυνατότητα που παρέχει στο χρήστη να αποθηκεύει στο ιστορικό του ακόμη και δραστηριότητες που παρακολούθησε εκτός εφαρμογής προκειμένου, όπως αναφέρθηκε και παραπάνω, να ενθαρρύνει και να βοηθάει τους χρήστες να οργανώσουν και να διαχειριστούν σωστά το πλάνο προπονήσεών τους.

Τέλος, ο χρήστης μπορεί οποιαδήποτε στιγμή το επιθυμήσει, να στείλει σχόλια στον διαχειριστή, κάτι που δείχνει το πόσο προσανατολισμένη προς τον χρήστη είναι η Unikorn καθώς δεν επαναπαύεται στις δυνατότητες που παρέχει σήμερα αλλά ψάχνει συνεχώς τρόπους βελτίωσης προκειμένου να είναι πάντα κοντά στον χρήστη και να καλύπτει τις ανάγκες του.

### 5.2 Η Unikorn στο μέλλον

Μετά το πέρας της παρούσας πτυχιακής εργασίας η εφαρμογή Unikorn θα συνεχίσει να εξελίσσεται και υπάρχουν κάποια άμεσα πλάνα για να προσεγγίσει περισσότερο τους χρήστες που αγαπούν τον υγιεινό τρόπο ζωής. Μία από τις δυνατότητες που θα προστεθούν είναι ο θερμοδομετρητής με παροχή πληροφοριών για τα τρόφιμα, όπως πρωτεΐνες, υδατάνθρακες, κλπ. Ένα τέτοιο εργαλείο θα μπορέσει να εξυπηρετήσει και τον αθλητή που πρέπει να μετράει τις θερμίδες και τις πρωτεΐνες του για να έχει καλύτερη απόδοση αλλά και τον απλό ασκούμενο που επιθυμεί μία αλλαγή στην διατροφική συμπεριφορά του.

Επίσης, θα προστεθεί η δυνατότητα απόκτησης φίλων. Αυτό θα επιτευχθεί είτε μέσω του “mail” είτε μέσω του κινητού τηλεφώνου του φίλου τους. Με αυτόν τον τρόπο και κρατώντας ιστορικό από τη δραστηριότητα όλων των χρηστών, θα επιτευχθεί οι χρήστες να χρησιμοποιούν περισσότερο την εφαρμογή και να έχουν διαρκώς ένα κίνητρο. Έχει παρατηρηθεί ότι όταν κάνεις γυμναστική με τους φίλους σου τα παρατάς πιο δύσκολα. Με βάση αυτή την ιδέα, θα αναπτυχθεί ένα σύστημα πόντων και

επιβράβευσης προκειμένου οι χρήστες να παραμένουν ενεργοί και να έχουν συνεχώς κίνητρα να χρησιμοποιήσουν την εφαρμογή.

Με τη δυνατότητα των φίλων στο μέλλον θα δημιουργηθεί ένα καινούργιο προφίλ, αυτό του προπονητή. Με αυτόν τον τρόπο, ένας προπονητής θα έχει τη δυνατότητα να παρακολουθεί το προσωπικό πρόγραμμα των αθλητών του και να τον ελέγχει χωρίς να χρειάζεται να έρχεται διαρκώς σε επικοινωνία μαζί του. Επίσης, θα μπορούν να επικοινωνήσουν μεταξύ τους μέσω της εφαρμογής καθώς και να μοιραστούν αρχεία προκειμένου να μην χρειάζεται να χρησιμοποιούν άλλες εφαρμογές παρά μόνο τη Unikorn.

Κάτι που θα ήταν καλό να επιτευχθεί στο μέλλον είναι μία συνεργασία με γυμναστήρια και διατροφολόγους. Από τη μία με αυτό τον τρόπο θα τους διαφημίζουμε και από την άλλη θα μπορούμε και να παρέχουμε στους χρήστες δυνατότητα να δουν που βρίσκεται κοντά τους γυμναστήριο ή και διατροφολόγος καθώς και να λαμβάνουν εκπτώσεις ή/και δώρα σύμφωνα με τους πόντους που μαζεύουν από τη γυμναστική τους. Με αυτόν τον τρόπο οι χρήστες θα έχουν μόνιμα έναν λόγο να χρησιμοποιούν την εφαρμογή και τα γυμναστήρια και οι διατροφολόγοι θα διαφημίζονται αρκετά καθώς τα άτομα που χρησιμοποιούν την εφαρμογή είναι κατά βάση άτομα που ή ασχολούνται με τον υγιεινό τρόπο ζωής ή θέλουν να ασχοληθούν, οπότε είναι πολύ πιθανό να αναζητήσουν υποστήριξη από ειδικούς.

Με τις συνεργασίες θα επιτύχουμε να δημιουργήσουμε και δύο φόρουμ. Το ένα μπορεί να είναι σχετικό με τα αθλήματα και το άλλο με τη διατροφή. Το φόρουμ των αθλημάτων θα έχει φίλτρα για κάποια συγκεκριμένα αθλήματα και σε περίπτωση που ο χρήστης πάει να χρησιμοποιήσει φίλτρο που δεν υπάρχει με μια εφαρμογή ευφυΐας θα ελέγχεται αν όντως είναι άθλημα και θα προστίθεται στα φίλτρα. Με αυτόν τον τρόπο για παράδειγμα θα μπορεί η Μαρία να μιλήσει μόνο με άτομα που ασχολούνται με την κολύμβηση και όποτε επιθυμούν οι προπονητές των γυμναστηρίων θα μπορούν να δημοσιεύουν άρθρα ή/και να ανοίγουν συζητήσεις με τους αθλητές επάνω σε θέματα που σύμφωνα με τα φίλτρα θα καταλαβαίνουν ότι τους ενδιαφέρουν (και θα έχουν γνώσεις).

Το φόρουμ που θα είναι σχετικό με τη διατροφή θα υπάρχει με στόχο την ανταλλαγή συνταγών μεταξύ των χρηστών και την προώθηση συνταγών από διατροφολόγους. Οι διατροφολόγοι συνηθίζουν να δημοσιεύουν συνταγές τους με σκοπό να διαφημιστούν και οι αθλητές συνηθίζουν να ανταλλάσσουν μεταξύ τους συνταγές με παραπάνω πρωτεΐνες ή/και ενέργεια καθώς και διατροφικές συμβουλές. Ακόμη και ο χρήστης που δεν θα είναι αθλητής αλλά θα προσπαθεί να μνηθεί στην κουλτούρα ενός άλλου τρόπου ζωής θα έχει ενδιαφέρον και περιέργεια για να δοκιμάσει ή να μοιραστεί μία συνταγή/συμβουλή.

Φυσικά, για όλες αυτές τις λειτουργίες θα χρειαστεί πάρα πολύς χρόνος και μπορεί να προστεθούν και άλλες δυνατότητες, ανάλογα με τα feedback που θα δίνουν οι χρήστες. Το Unikorn είναι ένα ξεκίνημα, μια ιδέα και ένας τρόπος να μνηθούν όσο περισσότεροι γίνεται σε έναν πιο υγιεινό τρόπο ζωής. Είναι ένας τρόπος να ενωθούν οι κόσμοι αυτών που μόνιμα προσπαθούν και στο τέλος τα παρατάνε λόγω έλλειψης κινήτρου αλλά με αυτών που η άθληση είναι τρόπος ζωής.



# ΒΙΒΛΙΟΓΡΑΦΙΑ

## Βιβλία

[1] Δαμιανός Γαβαλάς, Βλάσης Κασαπάκης, Θωμάς Χατζηδημήτρης, *Κινητές Τεχνολογίες*. Αθήνα, NewTech Pub, 2015.

[2] Paul Deitel, Harvey Deitel, Abbey Deitel, *Android – How to Program – Second Edition*. New Jersey: Pearson Education Inc, 2015.

## Internet Sites

[3] Udacity, “Firebase in a weekend” *Udacity*. [Online]. Available: <https://classroom.udacity.com/courses/ud0352>

[4] Educatree, “How to retrieve data from Firebase in Android || Retrieve data from firebase || Android Firebase #4,” Youtube, 2019. [Online]. Available: <https://www.youtube.com/watch?app=desktop&v=LpWhAz3e1sI>

[5] AbhiAndroid, “Rating Bar Tutorial With Example In Android Studio,” Abhi Android [Online]. Available: <https://abhiandroid.com/ui/ratingbar>

[6] Developers Android, “Android Studio”, Developers Android. [Online]. Available: <https://developer.android.com/studio>

[7] Developers Android, “Platform Architecture”, Developers Android. [Online]. Available: <https://developer.android.com/guide/platform>

[8] inkstory, “Android 12: Τι είναι και όλα τα νέα χαρακτηριστικά”, inkstory/tech. [Online]. Available: <https://inkstory.gr/tech/android-12-ti-einai-charaktiristika-fernei/>

[9] Xiaomi Miui Hellas, “Android 13 : Αποκαλύφθηκαν τα νέα χαρακτηριστικά που θα μας φέρει”, Xiaomi Miui Hellas. [Online]. Available: <https://news.xiaomi-miui.gr/android-13-features-revealed-74567-2/>

[10] technoglitz, “Όλα όσα γνωρίζουμε για το Android 13 “Tiramisu” India News, The Indian Express”, Developers technoglitz. [Online]. Available: <https://technoglitz.com/greece/%CF%8C%CE%BB%CE%B1-%CF%8C%CF%83%CE%B1-%CE%B3%CE%BD%CF%89%CF%81%CE%AF%CE%B6%CE%BF%CF%85%CE%BC%CE%B5-%CE%B3%CE%B9%CE%B1-%CF%84%CE%BF-android-13-tiramisu-india-news-the-indian-express/>

[11] Wikipedia, “Android”, Wikipedia. [Online]. Available: <https://el.wikipedia.org/wiki/Android>

[12] Wikipedia, “Android Alpha”, Wikipedia. [Online]. Available: [https://el.wikipedia.org/wiki/Android\\_Alpha](https://el.wikipedia.org/wiki/Android_Alpha)

[13] Public, “To Android 11 έφτασε – Τι νέο φέρνει;”, Public blog. [Online]. Available: <https://blog.public.gr/smartphone/android-11-eftase-ti-neo-fernei>

[14] xda-developers, “Exclusive: This is our first look at Android 13 “Tiramisu” and some of its upcoming features”, xda-developers. [Online]. Available: <https://www.xda-developers.com/android-13-tiramisu-exclusive-first-look/>

[15] Wikipedia, “SQLite,” Wikipedia. [Online]. Available: <https://el.wikipedia.org/wiki/SQLite>

[16] Developers Android, “Meet Android Studio,” Developers Android. [Online]. Available: <https://developer.android.com/studio/intro>

[17] Wikipedia, “Android Studio,” Wikipedia. [Online]. Available: [https://el.wikipedia.org/wiki/Android\\_Studio](https://el.wikipedia.org/wiki/Android_Studio)

[18] Wikipedia, “Visual Studio Code,” Wikipedia. [Online]. Available: [https://en.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://en.wikipedia.org/wiki/Visual_Studio_Code)

[19] Visual Studio, “Visual Studio Code FAQ,” Visual Studio. [Online]. Available: <https://code.visualstudio.com/docs/supporting/faq>

[20] Wikipedia, “NetBeans,” Wikipedia. [Online]. Available: <https://en.wikipedia.org/wiki/NetBeans>

[21] Wikipedia, “Firebase,” Wikipedia. [Online]. Available: <https://en.wikipedia.org/wiki/Firebase>

[22] Firebase, “Firebase,” Firebase. [Online]. Available: <https://firebase.google.com/>

[23] Wikipedia, “Η ιστορία των εκδόσεων του Android,” Wikipedia. [Online]. Available: [https://el.wikipedia.org/wiki/%CE%99%CF%83%CF%84%CE%BF%CF%81%CE%AF%CE%B1\\_%CE%B5%CE%BA%CE%B4%CF%8C%CF%83%CE%B5%CF%89%CE%BD\\_%CF%84%CE%BF%CF%85\\_Android](https://el.wikipedia.org/wiki/%CE%99%CF%83%CF%84%CE%BF%CF%81%CE%AF%CE%B1_%CE%B5%CE%BA%CE%B4%CF%8C%CF%83%CE%B5%CF%89%CE%BD_%CF%84%CE%BF%CF%85_Android)

[24] Virtual Yoga Class by [Kari Shea](#) on [Unsplash](#), <https://unsplash.com/photos/qa1wvrlWCio>

[25] rocket with sparklers, <https://arrow.tudublin.ie/assets/md5images/419e1862bf08bab457a264787bf5ead2.png>

[26] Cute unicorn cat with rainbow hula hoop doing fitness exercise cartoon animal character isolated on a white background, [https://www.freepik.com/premium-vector/cute-unicorn-cat-with-rainbow-hula-hoop-doing-fitness-exerssise-cartoon-animal-character-isolated-white-background\\_7781036.htm](https://www.freepik.com/premium-vector/cute-unicorn-cat-with-rainbow-hula-hoop-doing-fitness-exerssise-cartoon-animal-character-isolated-white-background_7781036.htm)

[27] Watercolor leaves with copy space design, [https://www.freepik.com/free-vector/watercolor-leaves-with-copy-space-design\\_3463072.htm](https://www.freepik.com/free-vector/watercolor-leaves-with-copy-space-design_3463072.htm)

[28] Sport and fitness equipment in gym, [https://www.freepik.com/free-vector/sport-fitness-equipment-gym\\_7743070.htm](https://www.freepik.com/free-vector/sport-fitness-equipment-gym_7743070.htm)

[29] History of Android, <https://www.sitesbay.com/android/android-history-and-versions-of-android?fbclid=IwAR2jdR08CVt4k8z6Mk69OlxBmgCwhRrZXbfXfP2yQQObKah1r0sFUVi8F8I>

[30] Android 12, <https://fuentitech.com/7-important-android-12-features-to-tap-on-first-day/308323/>

[31] Android Activity Lifecycle, <https://www.javatpoint.com/android-life-cycle-of-activity>

[32] Android Fragment Lifecycle, <https://developer.android.com/guide/fragments/lifecycle>

[33] Generic System Image, <https://developer.android.com/topic/generic-system-image>

## **Journal Articles**

[34] SonJie Gond, “A Collaborative Filtering Recommendation Algorithm Based on User Clustering and Item Clustering,” *Journal of Software*, vol. 5, no. 7, pp. 745-751, July 2010.

## ΠΑΡΑΡΤΗΜΑ Α: ΚΩΔΙΚΑΣ ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗΣ

```
TS clustering.ts X
src > TS clustering.ts > findTopKUsers
1  type UserRating = Array<Array<number | undefined | null>>
2
3  const EPSILON = 0.0001;
4
5  function findTopKUsers(numOfCenters: number, ratings: UserRating): UserRating {
6      let topUsers = []
7      let topUsersRank = []
8
9      for (let row = 0; row < ratings.length; row++) {
10         const user = ratings[row];
11         let itemsFound = 0;
12         for (let col = 0; col < user.length; col++) {
13             const rating = user[col];
14             if (rating !== undefined) {
15                 itemsFound += 1;
16             }
17         }
18
19         if (topUsers.length < numOfCenters) {
20             topUsers.push(user);
21             topUsersRank.push(itemsFound);
22         } else {
23             let replaceIndex = -1;
24             let replaceIndexRank = -1;
25
26             for (let index = 0; index < topUsers.length; index++) {
27                 const userItemsFound = topUsersRank[index];
28                 if (replaceIndex === -1 && userItemsFound < itemsFound) {
29                     replaceIndex = index;
30                     replaceIndexRank = userItemsFound;
31                 } else if (replaceIndex !== -1 && userItemsFound < replaceIndexRank) {
32                     replaceIndex = index;
33                     replaceIndexRank = userItemsFound;
34                 }
35             }
36
37             if (replaceIndex !== -1) {
38                 topUsers[replaceIndex] = user;
39                 topUsersRank[replaceIndex] = itemsFound;
40             }
41         }
42     }
43     return topUsers;
44 }
45
46 function euclidianDistanceSimilarity(userRating: Array<number | undefined | null>, centerRating: Array<number | undefined | null>): number {
47     let sum = 0;
48     for (let i = 0; i < userRating.length; i++) {
49         if (userRating[i] !== undefined && centerRating[i] !== undefined) {
50             sum += Math.abs(userRating[i] - centerRating[i]);
51         }
52     }
53     return sum;
54 }
```

```

48   for (let ratingId = 0; ratingId < userRating.length; ratingId++) {
49     let uRating = userRating[ratingId];
50     if (uRating === undefined || uRating === null) {
51       uRating = 0;
52     }
53     let cRating = centerRating[ratingId]
54     if (cRating === undefined || cRating === null) {
55       cRating = 0;
56     }
57     sum += Math.pow(uRating - cRating, 2)
58   }
59
60   return Math.sqrt(sum);
61 }
62
63 export function clustering(numOfCenters: number, ratings: UserRating, userIds: string[]): UserRating {
64   const centers: UserRating = findTopKUsers(numOfCenters, ratings);
65   let clusters: UserRating[] = []
66   let clusterUsers: string[][] = []
67
68   let flag = true;
69   let prevCenters: UserRating = JSON.parse(JSON.stringify(centers)) // clone centers variable
70

```

```

71   do {
72     flag = true
73
74     console.log('looping clustering')
75     prevCenters = JSON.parse(JSON.stringify(centers)) // clone centers variable
76     clusterUsers = [];
77     clusters = [];
78     for (let i = 0; i < numOfCenters; i++) {
79       clusters.push([])
80       clusterUsers.push([])
81     }
82
83     for (let userIndex = 0; userIndex < ratings.length; userIndex++) {
84       const user = ratings[userIndex];
85       const userSimCenters: number[] = []
86
87       for (let centerId = 0; centerId < centers.length; centerId++) {
88         const center = centers[centerId];
89         userSimCenters.push(euclidianDistanceSimilarity(user, center))
90       }
91
92       let mostSimilarCenterId = 0;
93       let mostSimilarity = userSimCenters[0];
94       for (let centerId = 0; centerId < userSimCenters.length; centerId++) {

```

```

95     const similarity = userSimCenters[centerId];
96     // because we use euclidian distance most similar are closest
97     if (similarity < mostSimilarity) {
98         mostSimilarity = similarity;
99         mostSimilarCenterId = centerId;
100     }
101 }
102
103 clusters[mostSimilarCenterId].push(user);
104 clusterUsers[mostSimilarCenterId].push(userIds[userIndex]);
105 }
106
107 for (let clusterId = 0; clusterId < clusters.length; clusterId++) {
108     const users: UserRating = clusters[clusterId]
109     for (let ratingId = 0; ratingId < clusters[clusterId].length; ratingId++) {
110         let sum: number = centers[clusterId][ratingId] ?? 0
111         let count = 1;
112         for (let userId = 0; userId < users.length; userId++) {
113             const user = users[userId];
114             const rating = user[ratingId];
115             if (rating !== undefined && rating !== null) {
116                 sum += rating;
117                 count++;
118             }
119         }
120         if (count !== 1) {
121             centers[clusterId][ratingId] = sum / count;
122         }
123     }
124 }
125
126 for (let i = 0; i < centers.length && flag; i++) {
127     for (let k = 0; k < ratings.length && flag; k++) {
128         const current = centers[i][k]
129         const prev = prevCenters[i][k]
130         if (typeof current === 'number' && typeof prev === 'number') {
131             if (Math.abs(current - prev) > EPSILON) {
132                 flag = false
133             }
134         }
135     }
136 }
137 } while (!flag);
138
139 console.log('looping done')
140

```

```

141     const ratingClone: UserRating = JSON.parse(JSON.stringify(ratings));
142     for (let clusterId = 0; clusterId < centers.length; clusterId++) {
143         const center = centers[clusterId]
144         const users = clusterUsers[clusterId]
145         for (let userIndex = 0; userIndex < users.length; userIndex++) {
146             const userId = users[userIndex];
147             const userRatingIndex = userIds.findIndex((val) => val === userId)
148             const user = ratingClone[userRatingIndex];
149             for (let ratingId = 0; ratingId < center.length; ratingId++) {
150                 if (
151                     (user[ratingId] === undefined || user[ratingId] === null)
152                     && (center[ratingId] !== undefined && center[ratingId] !== null && center[ratingId]! > 0.5)
153                 ) {
154                     ratingClone[userRatingIndex][ratingId] = center[ratingId]
155                 } else {}
156                 ratingClone[userRatingIndex][ratingId] = null
157             }
158         }
159     }
160 }
161
162     return ratingClone;
163 }

```