

---

**Πανεπιστήμιο  
Πειραιά**



**Τμήμα  
Πληροφορικής**

---

## ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Τίτλος:

Εφαρμογή κινητού για καταγραφή και παρακολούθηση της υγείας και  
ανάπτυξης μωρών

Title:

Mobile application for recording and monitoring the health and  
development of babies

Ονοματεπώνυμο: Γκαρτζονίκα Αικατερίνη Π18026

Επιβλέπων καθηγητής: Ευθύμιος Αλέπης

## Πίνακας περιεχομένων

Εισαγωγή.....	5
---------------	---

### Α Μέρος – Λειτουργία της εφαρμογής

<b>1.Είσοδος.....</b>	<b>6</b>
1.1.Εγγραφή .....	6
1.2.Σύνδεση .....	6
1.3.Facebook .....	7
1.4.Google .....	7
<b>2.Προφίλ χρήστη .....</b>	<b>8</b>
2.1.Επιλογή τύπου χρήστη .....	8
2.2.Ιατρός .....	8
2.3.Κηδεμόνας.....	9
<b>3.Αρχική .....</b>	<b>10</b>
3.1.Ιατρός .....	10
3.1.1.Προβολή μωρών υπό επίβλεψη .....	10
3.1.2.Προβολή μωρών αποθηκευμένων στην βάση δεδομένων .....	10
3.1.3.Προβολή στοιχείων μωρού .....	11
3.1.4.Προβολή διαγραμμάτων ανάπτυξης .....	11
3.1.5.Διαγραφή μωρού .....	12
3.1.6.Προβολή στοιχείων κηδεμόνων.....	12
3.1.7.Καταγραφή υγείας και ανάπτυξης μωρού.....	13
3.1.8.Εμβολιασμός .....	14
3.1.9.Προβολή οικογενειακού ιστορικού .....	14
3.1.10.Προβολή υγείας και ανάπτυξης μωρού.....	15
3.2.Κηδεμόνας.....	15
3.2.1.Προβολή μωρών και στατιστικών.....	15
3.2.2.Εμβολιασμός .....	16
3.2.3.Προβολή διαγραμμάτων ανάπτυξης .....	16
3.2.4.Διαγραφή μωρού .....	17
3.2.5.Προβολή γιατρού .....	17
3.2.6.Προβολή υγείας και ανάπτυξης μωρού.....	18

3.2.7.Προσθήκη μωρού.....	18
<b>4.Ρυθμίσεις .....</b>	<b>19</b>
4.1.Ιατρός .....	19
4.1.1. Προβολή λογαριασμού χρήστη .....	19
4.1.2.Διαγραφή χρήστη.....	19
4.1.3.Αλλαγή διεύθυνσης ηλεκτρονικού ταχυδρομείου .....	20
4.1.4.Αλλαγή κωδικού πρόσβασης .....	20
4.1.5.Ενημέρωση στοιχείων χρήστη .....	21
4.2.Κηδεμόνας.....	21
4.2.1.Προβολή λογαριασμού χρήστη.....	21
4.2.2.Προβολή συν-κηδεμόνα .....	22
4.2.3.Διαγραφή χρήστη .....	22
4.2.4.Αλλαγή διεύθυνσης ηλεκτρονικού ταχυδρομείου .....	23
4.2.5.Αλλαγή κωδικού πρόσβασης .....	23
4.2.6.Ενημέρωση στοιχείων χρήστη .....	24

## Β μέρος- Σχεδιασμός και Υλοποίηση

<b>5.Κλάσεις .....</b>	<b>25</b>
5.1. AdapterChildSpinner Class .....	25
5.2.Baby Class.....	26
5.3.CustomAdapter Class .....	27
5.4.Development Class.....	28
5.5.DevelopmentalItems Class. ....	29
5.6.Doctor Class.....	30
5.7.ExaminationItems Class.....	31
5.8.FamilyHistoryIllnesses Class .....	32
5.9.Parent Class .....	33
5.10.RecyclerAdapter Class .....	34
5.11.SustenanceItems Class .....	36
5.12.Vaccination Class .....	37

<b>6.Activities</b>	<b>38</b>
6.1.AddBaby Activity	38
6.2.AddChildToDoctor Activity	38
6.3.CreateNewDoctor Activity	38
6.4.CreateNewParent Activity	38
6.5.DeleteChild Activity.	39
6.6.DoctorParentChoose	39
6.7.FamilyHistoryInput Activity	39
6.8.LoginRegister Activity	39
6.9.MainScreenDoctor Activity	39
6.10.MainScreenParents Activity	40
6.11.MonitoringDevelopment Activity	40
6.12.NextParent Activity	40
6.13.RegisterUsernamePassword Activity	40
6.14.ShowDevelopmentsList Activity	41
6.15.ShowHistoric Activity	41
6.16.UserAccount Activity	41
6.17.ViewCharts Activity	41
6.18.ViewParentInfo Activity	42
6.19.ViewVaccination Activity	42
<b>7.Βιβλιογραφικές πηγές</b>	<b>43</b>

## Εισαγωγή

Σκοπός της παρούσας πτυχιακής εργασίας είναι η δημιουργία μιας εφαρμογής για κινητά που καταγράφει την υγεία και την ανάπτυξη μωρών. Στόχος είναι οι χρήστες να μπορούν εύκολα και γρήγορα να παρακολουθούν την αναπτυξιακή εξέλιξη των παιδιών με το πάτημα ενός μόνο κουμπιού. Την εφαρμογή μπορούν να χρησιμοποιήσουν γονείς/κηδεμόνες και ιατροί. Ο κηδεμόνας έχει την δυνατότητα να παρακολουθήσει την ανάπτυξη του παιδιού, το ιστορικό των εμβολιασμών που έχουν γίνει καθώς και να ενημερωθεί για τα στοιχεία του ιατρού που έχει αναλάβει το κάθε μωρό. Από την άλλη, ο ιατρός μπορεί να παρακολουθήσει την μέχρι τώρα πορεία της ανάπτυξης, να έχει πρόσβαση στο οικογενειακό ιστορικό κάθε παιδιού, να καταγράψει νέες παρατηρήσεις και εμβολιασμούς και να ενημερώνεται για τα στοιχεία του/των κηδεμόνα/κηδεμόνων. Ο σχεδιασμός της εφαρμογής έγινε σύμφωνα με το *Βιβλιάριο Υγείας Παιδιού* που παρέχεται από το Υπουργείο Υγείας, ενώ για την δημιουργία των διαγραμμάτων χρησιμοποιήθηκαν τα *Πρότυπα Ανάπτυξης του Παιδιού* σύμφωνα με τον Παγκόσμιο Οργανισμό Υγείας. Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκε το περιβάλλον *Android Studio*, (version *Android Studio Dolphin | 2021.3.1 Patch*) και *Java 18* (version "18.0.1.1" 2022-04-22) ενώ για την αποθήκευση των δεδομένων αξιοποιήθηκε η *Realtime Database (Firebase)*.

## Ά Μέρος – Λειτουργία της εφαρμογής

### 1. Είσοδος

#### 1.1 Εγγραφή

Για την είσοδο στην εφαρμογή *BabyLand* απαιτείται εγγραφή του χρήστη. Μετά την εισαγωγή των στοιχείων, γίνεται έλεγχος εγκυρότητας αυτών και εμφανίζεται ενημερωτικό μήνυμα στο κάτω μέρος της οθόνης, σχετικά με το αποτέλεσμα της διαδικασίας. Σε περίπτωση έγκυρων στοιχείων αυτόματα γίνεται σύνδεση και είσοδος στην εφαρμογή. Σε διαφορετική περίπτωση ο χρήστης καλείται να προσπαθήσει πάλι.

Figure 1. Εγγραφή

#### 1.2 Σύνδεση

Για την είσοδο στην εφαρμογή ο χρήστης καλείται να εισάγει τα στοιχεία του (email και κωδικός) ή να συνδεθεί μέσω Facebook ή Google. Αν δεν έχει γίνει εγγραφή στην εφαρμογή του εμφανίζεται σχετικό προτροπτικό μήνυμα. Αν ο χρήστης έχει επιλέξει την εγγραφή με χρήση email και κωδικό, κατά την προσπάθεια της πρώτης εισόδου στην εφαρμογή θα του ζητηθεί να μεταβεί στο email για επιβεβαίωση της διεύθυνσης.

Figure 2. Σύνδεση

### 1.3 Facebook

Αν ο χρήστης επιλέξει ως μέσο πιστοποίησης το Facebook κατά την εγγραφή του στην εφαρμογή μεταφέρεται σε σελίδα του οργανισμού.

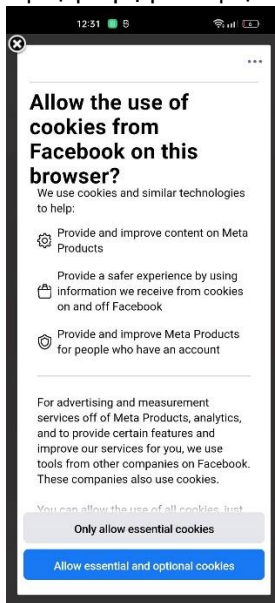


Figure 3. Facebook cookies

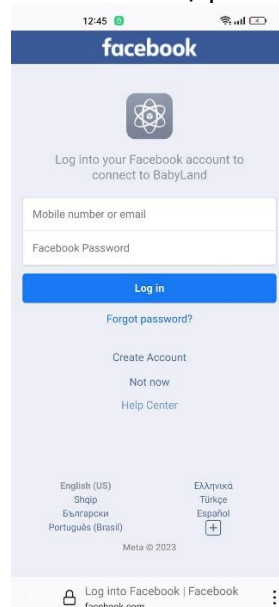


Figure 4. Facebook

### 1.4 Google

Αν ο χρήστης επιλέξει ως μέσο πιστοποίησης το Google κατά την εγγραφή του στην εφαρμογή μεταφέρεται σε σελίδα του οργανισμού.

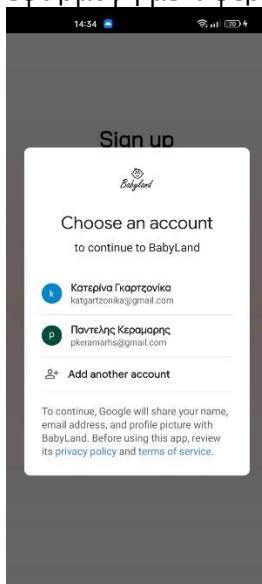


Figure 5. Google

## 2. Προφίλ χρήστη

### 2.1 Επιλογή τύπου χρήστη

Κατά την πρώτη είσοδο στην εφαρμογή ο χρήστης καλείται να επιλέξει τον τύπο προφίλ που θα έχει: γιατρός ή κηδεμόνας.



Figure 6. Επιλογή τύπου χρήστη

### 2.2 Ιατρός

Με την εκτέλεση της πρώτης εισόδου στην εφαρμογή ο χρήστης καλείται να εισάγει κάποιες πληροφορίες: Όνομα, Επίθετο, Τηλέφωνο επικοινωνίας και Αριθμό Ιατρικής Ταυτότητας και στην συνέχεια γίνεται έλεγχος εγκυρότητας όλων των στοιχείων καθώς και μοναδικότητας του αριθμού ιατρικής ταυτότητας.

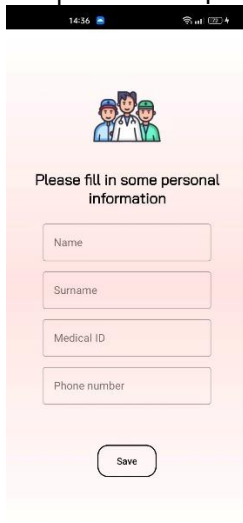


Figure 7. Εγγραφή ιατρού



## 2.3 Κηδεμόνας

Με την εκτέλεση της πρώτης εισόδου ο χρήστης καλείται να εισάγει κάποιες πληροφορίες για την ολοκλήρωση του λογαριασμού του στην εφαρμογή: Όνομα, Επίθετο, ΑΜΚΑ, Τηλέφωνο επικοινωνίας, Ημερομηνία γέννησης και Ομάδα αίματος. Γίνεται έλεγχος εγκυρότητας όλων των στοιχείων καθώς και έλεγχος μοναδικότητας του ΑΜΚΑ στην βάση δεδομένων. Αν δεν πληρείται η προϋπόθεση της μοναδικότητας αλλά ο εγγεγραμμένος χρήστης είναι κηδεμόνας και όχι χρήστης της εφαρμογής δίνεται η δυνατότητα αλλαγής αυτού. Στην συνέχεια καλείται να συμπληρώσει τα στοιχεία του συν-κηδεμόνα, αν αυτός υπάρχει, και εκτελείται και πάλι έλεγχος εγκυρότητας και μοναδικότητας στοιχείων. Αν ο χρήστης εισάγει τις απαραίτητες πληροφορίες αλλά ο ΑΜΚΑ δεν ανήκει σε χρήστη της εφαρμογής, οι πληροφορίες αποθηκεύονται στην βάση και ο συν-κηδεμόνας έχει μόνο την ιδιότητα του κηδεμόνα και όχι του χρήστη της εφαρμογής.

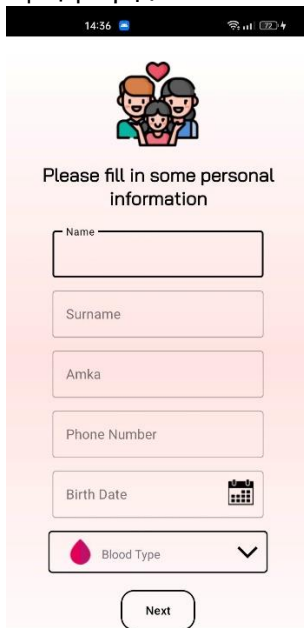


Figure 8. Εγγραφή κηδεμόνα

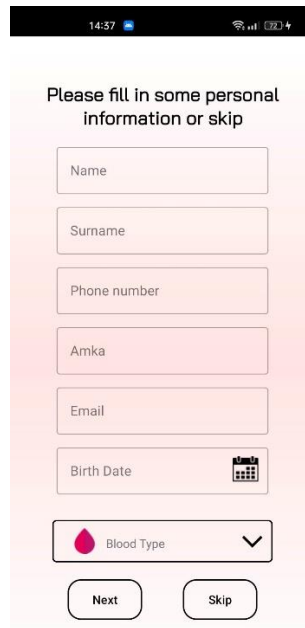


Figure 9. Εγγραφή συν-κηδεμόνα

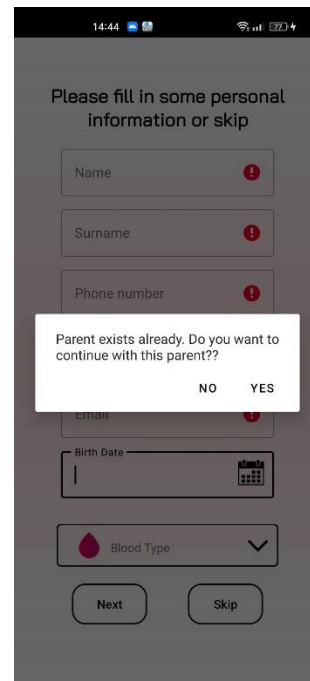


Figure 10. Ερώτηση για συν-κηδεμόνα

### 3. Αρχική σελίδα

#### 3.1 Ιατρός

##### 3.1.1 Προβολή μωρών υπό επίβλεψη

Στην αρχική αυτή σελίδα ο χρήστης (γιατρός) βλέπει τα μωρά που έχει αναλάβει υπό την επίβλεψή του. Επιλέγοντας κάποιο παιδί μεταβαίνει στην *Προβολή στοιχείων μωρού*.



Figure 11. Αρχική χωρίς μωρά

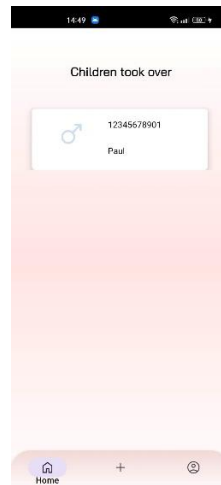


Figure 12. Αρχική με μωρά

##### 3.1.2 Προβολή μωρών αποθηκευμένων στην βάση δεδομένων

Ο χρήστης βλέπει μία λίστα με τα μωρά που υπάρχουν αποθηκευμένα στην βάση δεδομένων και μπορεί να κάνει αναζήτηση για κάποιο παιδί βάσει του ΑΜΚΑ του. Επιλέγοντας κάποιο παιδί του εμφανίζονται περισσότερες πληροφορίες και του ζητείται η εισαγωγή του ΑΜΚΑ ενός εκ των κηδεμόνων προκειμένου να το αναλάβει υπό την επίβλεψή του.

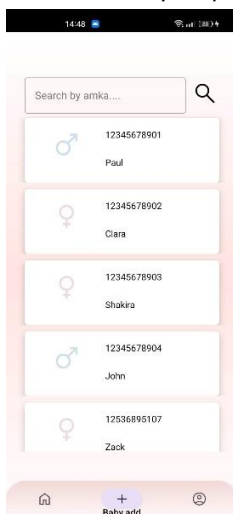


Figure 13. Προβολή πληροφοριών

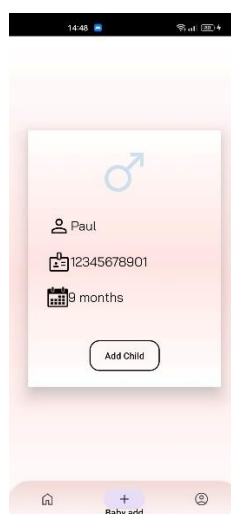


Figure 14. Μωρά στην βάση

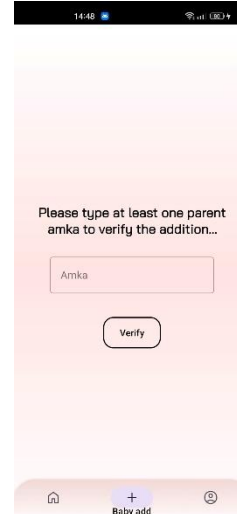


Figure 15. Βεβαίωση ΑΜΚΑ

### 3.1.3 Προβολή στοιχείων μωρού

Έχοντας επιλέξει κάποιο μωρό, ο χρήστης σε αυτή την σελίδα έχει την δυνατότητα να διαβάσει κάποιες πληροφορίες του παιδιού.

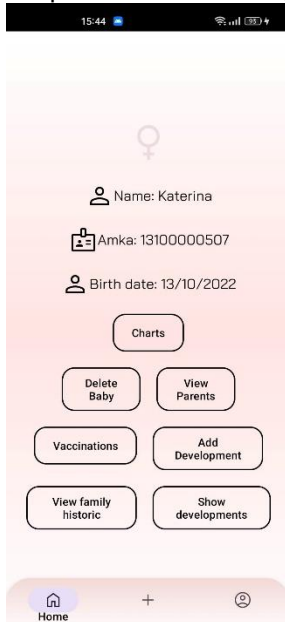


Figure 16. Προβολή μωρού

### 3.1.4 Προβολή διαγραμμάτων ανάπτυξης

Ο χρήστης (γιατρός) μπορεί να ενημερωθεί μέσω διαγραμμάτων ανάπτυξης για την εξέλιξη του βάρους και μήκους του μωρού σε σχέση με την ηλικία του.

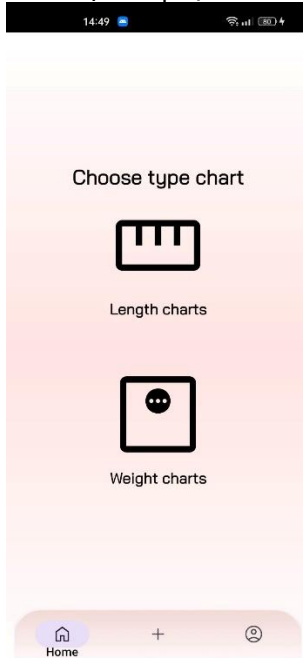


Figure 17. Επιλογή διαγράμματος

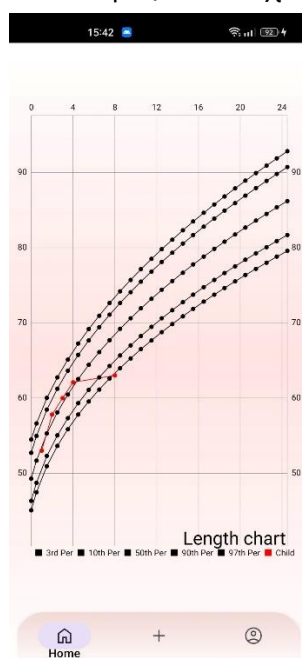


Figure 18. Διάγραμμα μήκους

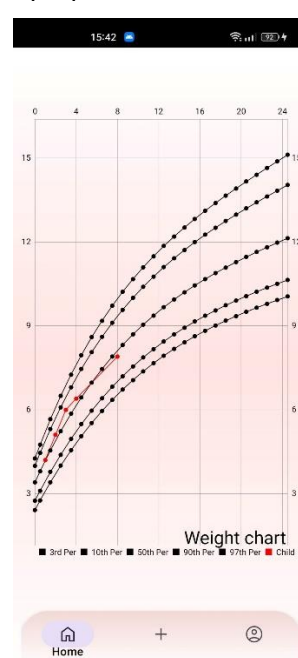


Figure 19. Διάγραμμα βάρους

### 3.1.5 Διαγραφή μωρού

Ο χρήστης έχει την δυνατότητα να διαγράψει από το προφίλ του κάποιο μωρό αν το επιθυμεί. Για την επιτυχή διαγραφή απαιτείται η συμπλήρωση του ΑΜΚΑ του παιδιού.

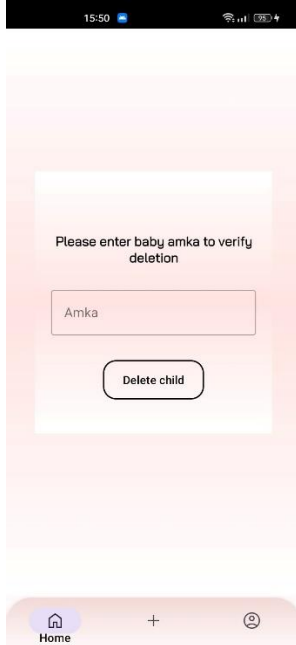


Figure 20. Διαγραφή μωρού

### 3.1.6 Προβολή στοιχείων κηδεμόνων

Στην σελίδα αυτή γίνεται προβολή των στοιχείων των κηδεμόνων του μωρού.

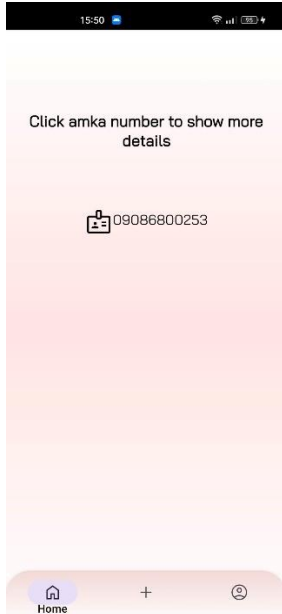


Figure 21. Επιλογή κηδεμόνα

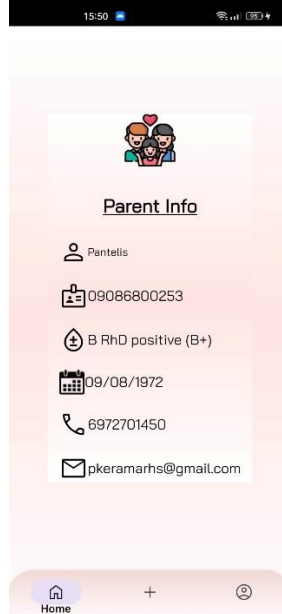


Figure 22. Στοιχεία κηδεμόνων

### 3.1.7 Καταγραφή υγείας και ανάπτυξης μωρού

Ο γιατρός μπορεί να καταγράψει την αναπτυξιακή εξέλιξη ενός μωρού. Καλείται να συμπληρώσει το βάρος, το μήκος, την περίμετρο κεφαλής, την διατροφή, την αναπτυξιακή παρακολούθηση του. Να κάνει κάποιες γενικές εξετάσεις καθώς και ανιχνευτικό έλεγχο ακοής, ενώ παράλληλα έχει την δυνατότητα καταγραφής παρατηρήσεων. Η ημερομηνία καταγραφής της κάθε παρακολούθησης θεωρείται η μέρα που αυτή καταγράφεται στον σύστημα, ωστόσο ο γιατρός έχει την δυνατότητα να το αλλάξει αν επιθυμεί. Τέλος, τα στοιχεία που βοηθούν στην καταγραφή της ανάπτυξης διαφέρουν από ηλικία σε ηλικία.

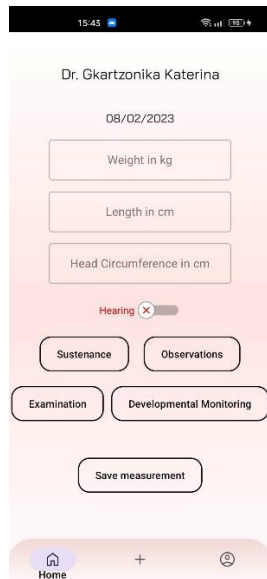


Figure 23. Καταγραφή ανάπτυξης

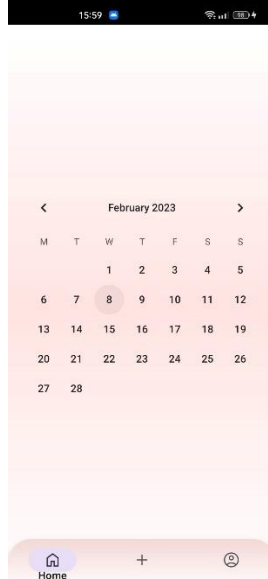


Figure 24. Επιλογή ημερομηνίας



Figure 25. Διατροφή

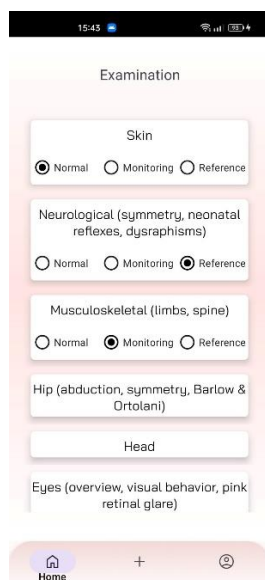


Figure 26. Εξέταση

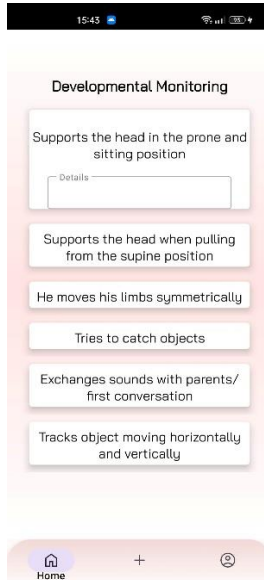


Figure 27. Αναπτυξιακή παρακολούθηση

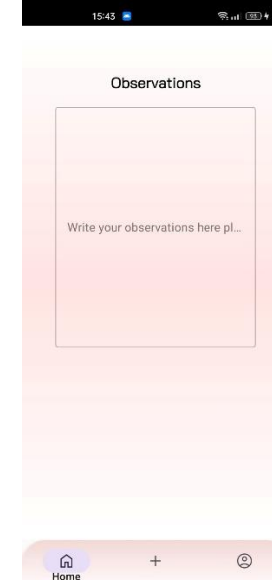


Figure 28. Παρατηρήσεις

### 3.1.8 Εμβολιασμός

Ο χρήστης έχει την δυνατότητα να δει τους εμβολιασμούς που αφορούν το κάθε παιδί, καθώς και λεπτομέρειες σχετικά με αυτούς (π.χ. ημερομηνία εμβολιασμού, γιατρός που ανέλαβε τον εμβολιασμό). Αν ο γιατρός εμβολιάσει κάποιο με παιδί με υπάρχον εμβόλιο πατάει το κουμπί *Add Vaccine*, διαφορετικά πατάει το κουμπί *Add another vaccine* και καλείται να εισάγει τα στοιχεία του εμβολίου που θα χορηγήσει στο παιδί.

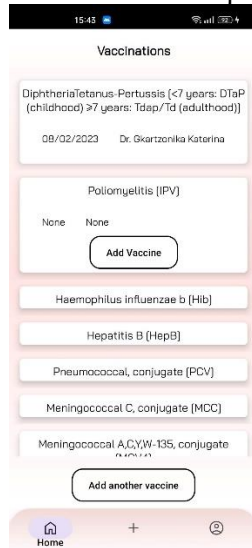


Figure 29. Εμβολιασμοί



Figure 30. Προσθήκη εμβολίου

### 3.1.9 Προβολή οικογενειακού ιστορικού

Ο χρήστης έχει την δυνατότητα προβολής του οικογενειακού ιστορικού κάθε μωρού. Αν στο ιστορικό της οικογένειας υπήρχε κάποια από τις αναφερόμενες παθήσεις εμφανίζεται με πράσινο χρώμα και ακριβώς από κάτω οι παρατηρήσεις που την αφορούν, σε διαφορετική περίπτωση εμφανίζεται με κόκκινο χρώμα.



Figure 31. Οικογενειακό ιστορικό

### 3.1.10 Προβολή υγείας και ανάπτυξης μωρού

Ο χρήστης έχει την δυνατότητα να δει όλες τις καταγραφές που έχουν γίνει σχετικά με την ανάπτυξη του μωρού καθώς και λεπτομέρειες σχετικά με την κάθε καταγραφή.

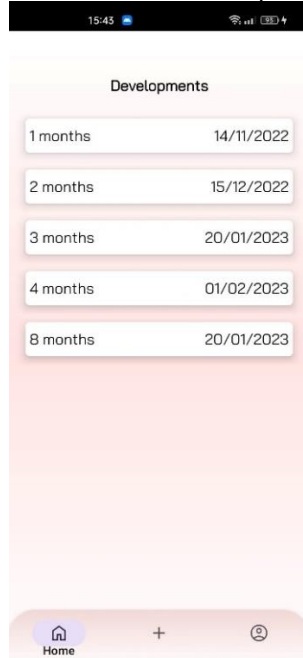


Figure 32. Καταγραφές

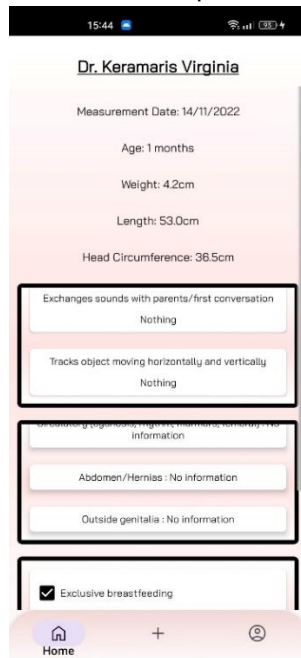


Figure 33. Αναλυτική καταγραφή α

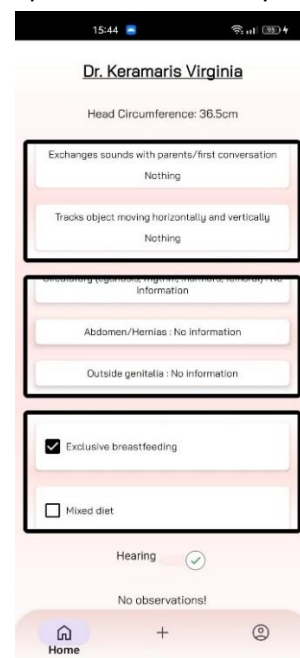


Figure 34. Αναλυτική καταγραφή β

## 3.2 Κηδεμόνας

### 3.2.1 Προβολή μωρών και στατιστικών

Στην αρχική σελίδα ο χρήστης έχει την δυνατότητα να δει κάποια στατιστικά στοιχεία για κάθε μωρό. Πατώντας στο drop down menu μπορεί να επιλέξει πιο παιδί θα προβληθεί μετέπειτα.



Figure 35. Αρχική χωρίς μωρά

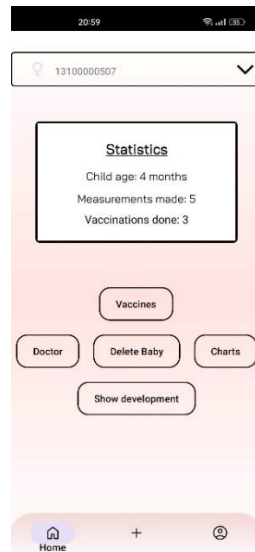


Figure 36. Αρχική με μωρά



Figure 37. Επιλογή μωρού

### 3.2.2 Εμβολιασμός

Ο χρήστης έχει την δυνατότητα να δει τους εμβολιασμούς που αφορούν το κάθε παιδί, καθώς και λεπτομέρειες σχετικά με αυτούς (π.χ. ημερομηνία εμβολιασμού, γιατρός που ανέλαβε τον εμβολιασμό). Αν δεν πραγματοποιηθεί ένας εμβολιασμός στα στοιχεία γράφει *None*.

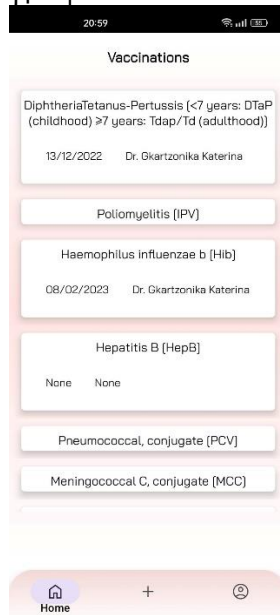


Figure 38. Εμβολιασμός

### 3.2.3 Προβολή διαγραμμάτων ανάπτυξης

Ο χρήστης (κηδεμόνας) μπορεί να ενημερωθεί μέσω διαγραμμάτων ανάπτυξης για την εξέλιξη του βάρους και μήκους του μωρού σε σχέση με την ηλικία του.

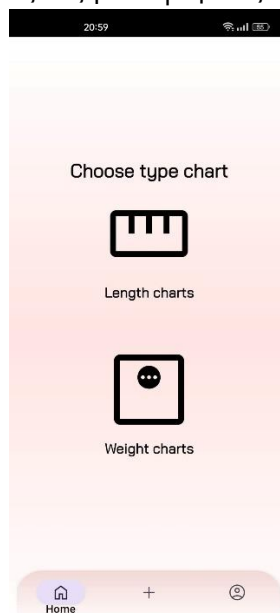


Figure 39. Επιλογή διαγράμματος

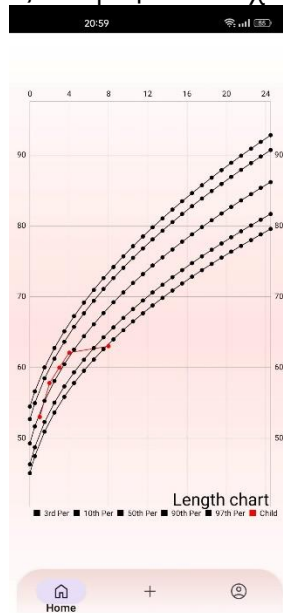


Figure 40. Διάγραμμα μήκους

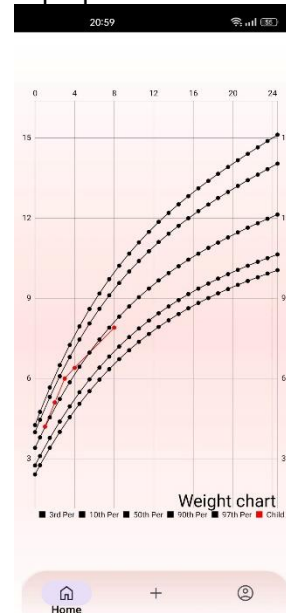


Figure 41. Διάγραμμα βάρους



### 3.2.4 Διαγραφή μωρού

Ο χρήστης έχει την δυνατότητα να διαγράψει κάποιο μωρό αν το επιθυμεί.

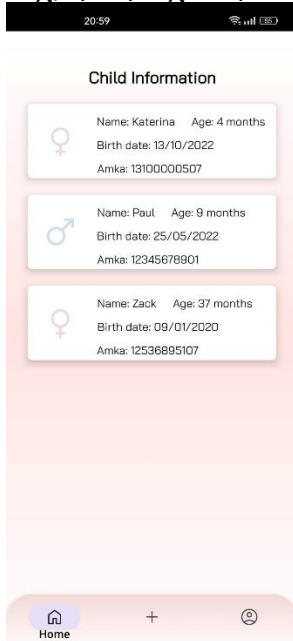


Figure 42. Επιλογή μωρού

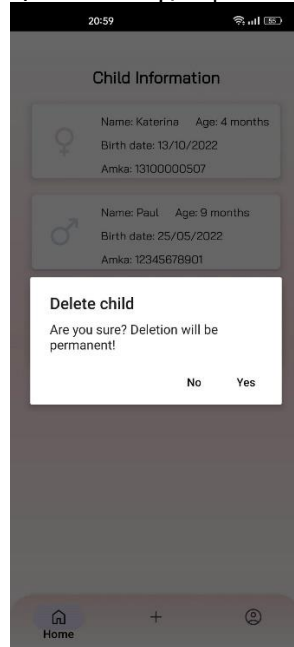


Figure 43. Διαγραφή μωρού

### 3.2.5 Προβολή γιατρού

Ο κηδεμόνας μπορεί να ενημερωθεί για τις πληροφορίες του γιατρού που έχει αναλάβει υπό την επίβλεψη του κάθε μωρό.



Figure 44. Στοιχεία γιατρού

### 3.2.6 Προβολή υγείας και ανάπτυξης μωρού

Ο χρήστης έχει την δυνατότητα να δει όλες τις καταγραφές που έχουν γίνει σχετικά με την ανάπτυξη του μωρού καθώς και λεπτομέρειες σχετικά με την κάθε καταγραφή.

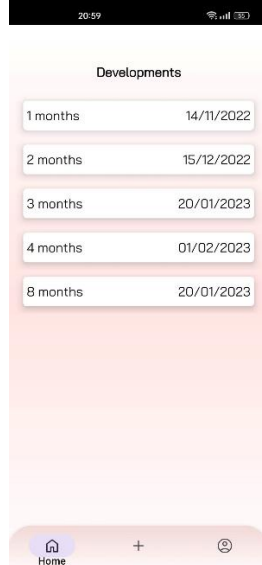


Figure 45. Επιλογή καταγραφής

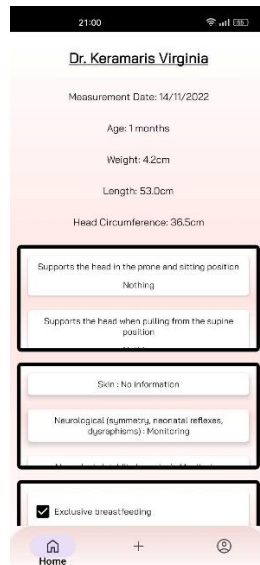


Figure 46. Αναλυτική καταγραφή α

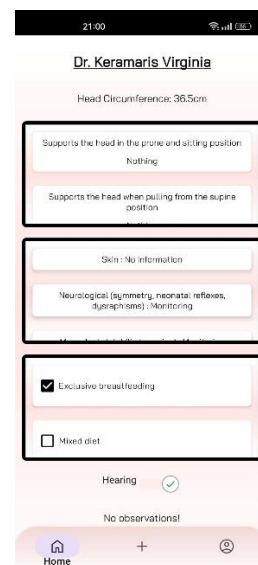


Figure 47. Αναλυτική καταγραφή β

### 3.2.7 Προσθήκη μωρού

Ο χρήστης έχει την δυνατότητα να εγγράψει κάποιο μωρό στην εφαρμογή και καλείται να εισάγει κάποιες πληροφορίες: Όνομα, ΑΜΚΑ, Ημερομηνία γέννησης, Τόπος γέννησης, Ομάδα αίματος καθώς και πληροφορίες για το οικογενειακό ιστορικό. Για όλα τα παραπάνω γίνεται έλεγχος εγκυρότητας καθώς και έλεγχος μοναδικότητας του ΑΜΚΑ. Σχετικά με την ημερομηνία γέννησης υπάρχει ο περιορισμός το έτος γέννησης να είναι μετά το 2019 και μέχρι και την σημερινή ημερομηνία.

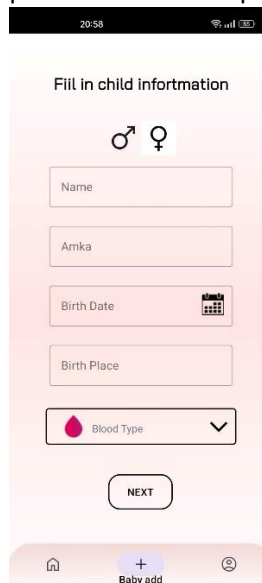


Figure 48. Εγγραφή μωρού

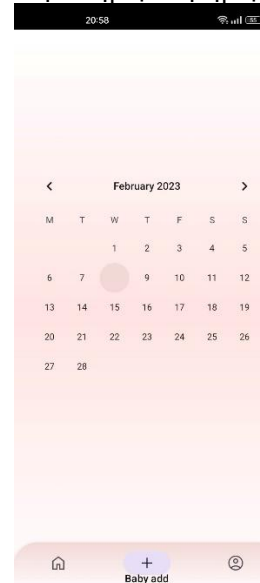


Figure 49. Επιλογή ημερομηνίας



Figure 50. Οικογενειακό ιστορικό

## 4. Ρυθμίσεις

### 4.1 Ιατρός

#### 4.1.1 Προβολή λογαριασμού χρήστη

Στην σελίδα αυτή προβάλλονται τα στοιχεία του χρήστη καθώς και οι δυνατότητες που έχει μέσω των ρυθμίσεων.



Figure 51. Ρυθμίσεις

#### 4.1.2 Διαγραφή χρήστη

Ο χρήστης έχει την δυνατότητα διαγραφής.

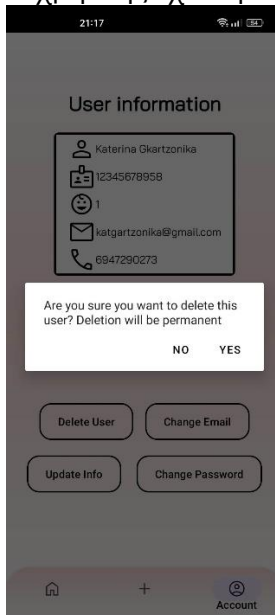


Figure 52. Διαγραφή χρήστη

### 4.1.3 Αλλαγή διεύθυνσης ηλεκτρονικού ταχυδρομείου

Ο χρήστης έχει την δυνατότητα αλλαγής της διεύθυνσης ηλεκτρονικού ταχυδρομείου, αν και μόνο αν έχει κάνει εγγραφή και είσοδο με την χρήση email και κωδικού. Σε περίπτωση που η πιστοποίηση του χρήστη γίνεται με Facebook και Google δεν έχει αυτή την επιλογή.

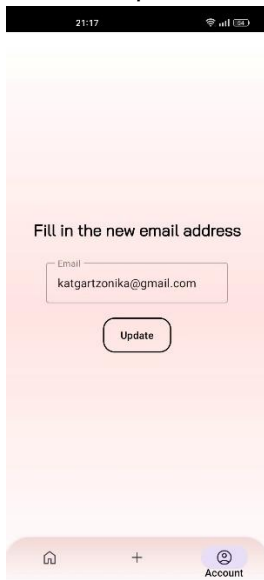


Figure 53. Αλλαγή email



Figure 54. Βεβαίωση αλλαγής

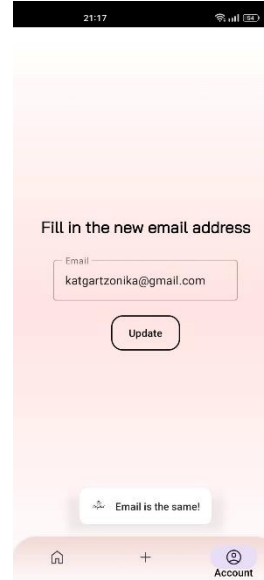


Figure 55. Ίδιο email

### 4.1.4 Αλλαγή κωδικού πρόσβασης

Ο χρήστης έχει την δυνατότητα αλλαγής του κωδικού πρόσβασης, αν και μόνο αν έχει κάνει εγγραφή και είσοδο με την χρήση email και κωδικού. Σε περίπτωση που η πιστοποίηση του χρήστη γίνεται με Facebook και Google δεν έχει αυτή την επιλογή.

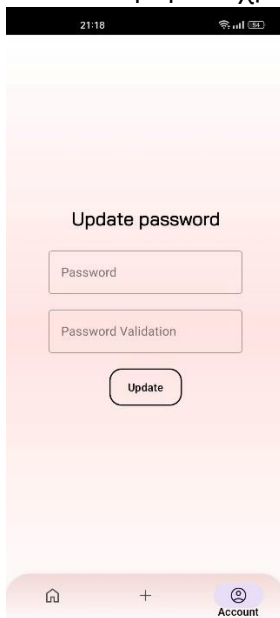


Figure 56. Αλλαγή κωδικού



Figure 57. Βεβαίωση αλλαγής

### 4.1.5 Ενημέρωση στοιχείων χρήστη

Ο χρήστης έχει την δυνατότητα αλλαγής των στοιχείων του.

Update information

Name  
Katerina

Surname  
Gkartzonika

Anika  
12345678958

Phone Number  
6947290273

Update Information

Account

Figure 58. Ενημέρωση στοιχείων

## 4.2 Κηδεμόνας

### 4.2.1 Προβολή λογαριασμού χρήστη

Στην σελίδα αυτή προβάλλονται τα στοιχεία του χρήστη καθώς και οι δυνατότητες που έχει μέσω των ρυθμίσεων.

User information

Ilias Papakonstantinou

10046403498

1

10/04/1983

O RhD positive (O+)

hgartzonikas@gmail.com

6947818974

No partner

CO-Parent

Delete User

Change Email

Update Info

Change Password

Account

Figure 59. Ρυθμίσεις χωρίς συν-κηδεμόνα

User information

Ilias Papakonstantinou

10046403498

1

10/04/1983

O RhD positive (O+)

hgartzonikas@gmail.com

6947818974

09086800253

CO-Parent

Delete User

Change Email

Update Info

Change Password

Account

Figure 60. Ρυθμίσεις με συν-κηδεμόνα

#### 4.2.2 Προβολή συν-κηδεμόνα

Ο χρήστης έχει την δυνατότητα να ενημερωθεί για τα στοιχεία του συν-κηδεμόνα, αν αυτός υπάρχει και να τον διαγράψει. Αν δεν υπάρχει μπορεί να προσθέσει συν-κηδεμόνα.



Figure 61. Δεν υπάρχει συν-κηδεμόνας

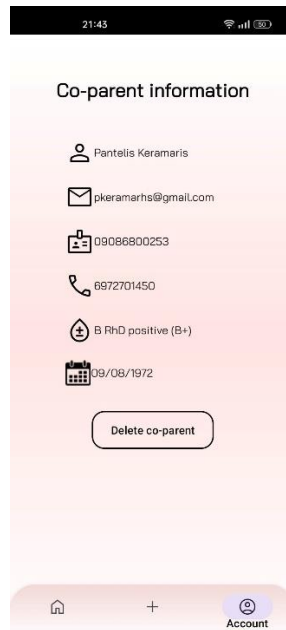


Figure 62. Στοιχεία συν-κηδεμόνα

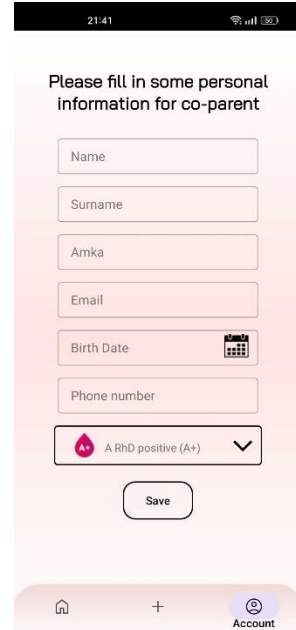


Figure 63. Εγγραφή συν-κηδεμόνα

#### 4.2.3 Διαγραφή χρήστη

Ο χρήστης έχει την δυνατότητα διαγραφής.



Figure 64. Διαγραφή χρήστη

#### 4.2.4 Αλλαγή διεύθυνσης ηλεκτρονικού ταχυδρομείου

Ο χρήστης έχει την δυνατότητα αλλαγής της διεύθυνσης ηλεκτρονικού ταχυδρομείου, αν και μόνο αν έχει κάνει εγγραφή και είσοδο με την χρήση email και κωδικού. Σε περίπτωση που η πιστοποίηση του χρήστη γίνεται με Facebook και Google δεν έχει αυτή την επιλογή.

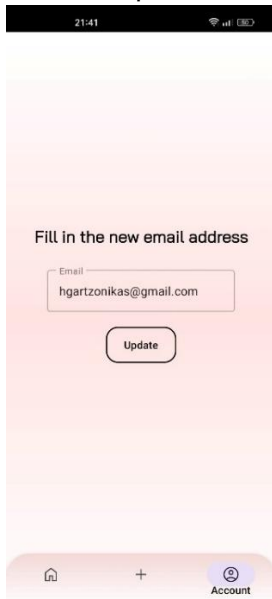


Figure 65. Αλλαγή email

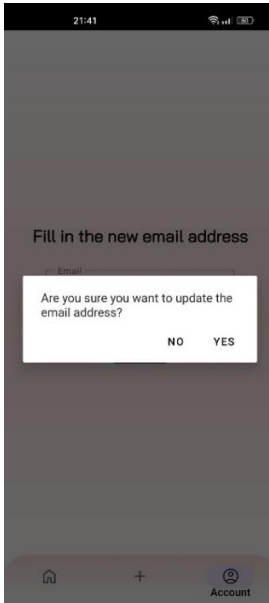


Figure 66.Βεβαίωση email

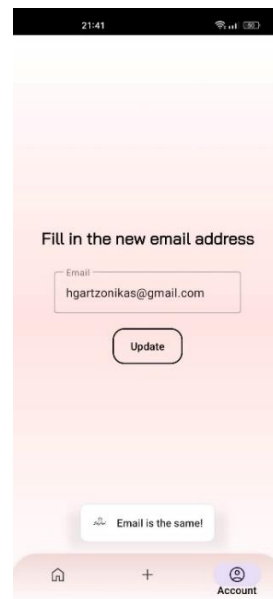


Figure 67. Ίδιο email

#### 4.2.5 Αλλαγή κωδικού πρόσβασης

Ο χρήστης έχει την δυνατότητα αλλαγής του κωδικού πρόσβασης, αν και μόνο αν έχει κάνει εγγραφή και είσοδο με την χρήση email και κωδικού. Σε περίπτωση που η πιστοποίηση του χρήστη γίνεται με Facebook και Google δεν έχει αυτή την επιλογή.

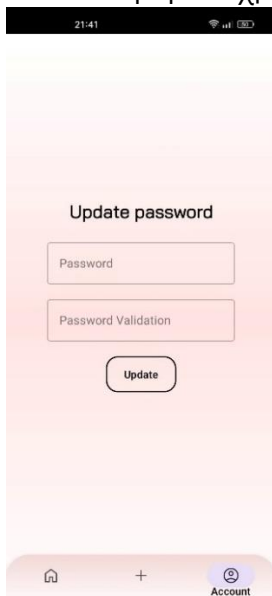


Figure 68.Αλλαγή κωδικού



Figure 69.Βεβαίωση αλλαγής

#### 4.2.6 Ενημέρωση στοιχείων χρήστη

Ο χρήστης έχει την δυνατότητα αλλαγής των στοιχείων του.

21:41

Update information

Name  
Ilias

Surname  
Papakonstantinou

Amka  
10046403498

Phone Number  
6947818974

Birth Date  
10/04/1983

Blood Type  
O RhD positive (O+)

Update Information

Home + Account

Figure 70.Ενημέρωση στοιχείων



## Β μέρος- Σχεδιασμός και Υλοποίηση

### 5. Κλάσεις

#### 5.1 AdapterChildSpinner Class

Η κλάση AdapterChildSpinner χρησιμοποιείται για την εμφάνιση του φύλου και του ΑΜΚΑ του παιδιού στον spinner. Ελέγχει το φύλο του μωρού και επιλέγει την αντίστοιχη φωτογραφία.

```
public class AdapterChildSpinner extends BaseAdapter {
    private Context context;
    private int[] images;
    private ArrayList<Baby> listKids;
    private LayoutInflater inflater;

    public AdapterChildSpinner(Context applicationContext, int[] images, ArrayList<Baby>listKids) {
        this.context = applicationContext;
        this.images = images;
        this.listKids = listKids;
        inflater = (LayoutInflater.from(applicationContext));
    }

    @Override
    public View getView(int i, View view, ViewGroup viewGroup) {
        view = inflater.inflate(R.layout.spinner_items, root: null);
        ImageView icon = (ImageView) view.findViewById(R.id.imageSex);
        TextView names = (TextView) view.findViewById(R.id.textAmka);
        if (listKids.get(i).getSex().equals("BOY")) {
            icon.setImageResource(images[0]);
        } else {
            icon.setImageResource(images[1]);
        }
        names.setText(listKids.get(i).getAmka());
        return view;
    }

    @Override
    public int getCount() { return listKids.size(); }

    @Override
    public Object getItem(int i) { return listKids.get(i).getAmka(); }

    @Override
    public long getItemId(int i) { return i; }
}
```

Figure 71. AdapterChildSpinner Class

## 5.2 Baby Class

Η κλάση Baby περιέχει τις πληροφορίες των μωρών. Αναλυτικότερα, περιέχει: το όνομα, τον αριθμό AMKA, το AMKA του γονέα 1, το AMKA του γονέα 2, τον τόπο γέννησης, τον τύπο αίματος, το φύλο, την ημερομηνία γέννησης, τη λίστα του οικογενειακού ιστορικού και την λίστα των εμβολιασμών. Τέλος, περιέχει τους constructors και τους αντίστοιχους getters και setters. Το AMKA κάθε μωρού είναι μοναδικό, το ίδιο και τον κηδεμόνων και αν δεν υπάρχει συν-κηδεμόνας αυτού το AMKA του γίνεται 0000000000.

```
12 public class Baby {
13     private String name, amka, parentOneAmka, parentTwoAmka, placeOfBirth, bloodType, sex;
14     private String dateOfBirth;
15     private ArrayList<FamilyHistoryIllnesses> iln;
16     private ArrayList<Vaccination> vaccinations;
17
18     @NotNull
19     @Override
20     public String toString() { return getAmka(); }
21
22
23     public Baby() {
24     }
25
26
27     public Baby(String name, String dateOfBirth, String amka, String placeOfBirth, String bloodType, String sex,
28                 String parentOneAmka, String parentTwoAmka, ArrayList<FamilyHistoryIllnesses> iln,
29                 ArrayList<Vaccination> vaccinations) {...}
30
31
32
33
34
35
36
37
38
39
40
41
42
43     public Baby(Baby baby) {...}
44
45
46
47
48
49
50
51
52
53
54
55     public ArrayList<Vaccination> getVaccinations() { return vaccinations; }
56
57
58
59     public void setVaccinations(ArrayList<Vaccination> vaccinations) {
60         this.vaccinations = vaccinations;
61     }
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79     public String getParentOneAmka() { return parentOneAmka; }
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Figure 72. Baby Class

```
79     public String getPlaceOfBirth() { return placeOfBirth; }
80
81
82
83     public void setPlaceOfBirth(String placeOfBirth) { this.placeOfBirth = placeOfBirth; }
84
85
86
87     public String getBloodType() { return bloodType; }
88
89
90
91     public void setBloodType(String bloodType) { this.bloodType = bloodType; }
92
93
94
95     public String getSex() { return sex; }
96
97
98
99     public void setSex(String sex) { this.sex = sex; }
100
101
102
103     public String getAmka() { return amka; }
104
105
106
107     public void setAmka(String amka) { this.amka = amka; }
108
109
110
111     public String getDateOfBirth() { return dateOfBirth; }
112
113
114
115     public void setDateOfBirth(String dateOfBirth) { this.dateOfBirth = dateOfBirth; }
116
117
118
119     public String getName() { return name; }
120
121
122
123     public void setName(String name) { this.name = name; }
124
125
126
127     public ArrayList<FamilyHistoryIllnesses> getIlIn() { return iln; }
128
129
130
131     public void setIlIn(ArrayList<FamilyHistoryIllnesses> iln) { this.iln = iln; }
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Figure 73. Baby Class

### 5.3 CustomAdapter Class

Η κλάση CustomAdapter χρησιμοποιείται για να εμφανίσει τους διαθέσιμους τύπους αίματος στον χρήστη.

```
public class CustomAdapter extends BaseAdapter {

    private Context context;
    private int images[];
    private String[] bloodType;
    private LayoutInflater inflater;

    public CustomAdapter(Context applicationContext, int[] images, String[] bloodType) {
        this.context = applicationContext;
        this.images = images;
        this.bloodType = bloodType;
        inflater = (LayoutInflater.from(applicationContext));
    }

    @Override
    public View getView(int i, View view, ViewGroup viewGroup) {
        view = inflater.inflate(R.layout.spinner_items, null);
        ImageView icon = (ImageView) view.findViewById(R.id.imageSex);
        TextView names = (TextView) view.findViewById(R.id.textAmko);
        icon.setImageResource(images[i]);
        names.setText(bloodType[i]);
        return view;
    }

    @Override
    public int getCount() { return bloodType.length; }

    @Override
    public Object getItem(int i) { return bloodType[i]; }

    @Override
    public long getItemId(int i) { return i; }
}
```

Figure 74. CustomAdapter Class

## 5.4 Development Class

Η κλάση Development περιέχει όλες τις πληροφορίες για την ανάπτυξη του μωρού, δηλαδή, το βάρος, το μήκος, την περίμετρο κεφαλής, την ηλικία, τον τύπο ηλικίας (βδομάδες ή μήνες), παρατηρήσεις, το όνομα του γιατρού που κάνει την καταγραφή, τον ΑΜΚΑ, την ημερομηνία καταγραφής, αν έγινε ανιχνευτικός έλεγχος ακοής, την λίστα διατροφής, την λίστα εξέτασης καθώς και την λίστα των παρακολουθήσεων ανάπτυξης. Τέλος περιέχουν οι constructors, οι setters και οι getters.

```
6 public class Development {
7     private String weight, length, headCircumference, age, ageType, observations, doctor, amka, measurementDate;
8     private ArrayList<SustenanceItems> sustenance;
9     private ArrayList<ExaminationItems> examination;
10    private ArrayList<DevelopmentalItems> developmentalMonitoring;
11    private Boolean hearing;
12
13    public Development(){}
14
15    public Development(String amka, String weight, String length, String headCircumference, String measurementDate,
16        String age, String ageType, ArrayList<SustenanceItems> sustenance,
17        ArrayList<ExaminationItems> examination, ArrayList<DevelopmentalItems> developmentalMonitoring,
18        Boolean hearing, String observations, String doctor) {...}
19
20    public String getAmka() { return amka; }
21
22    public void setAmka(String amka) { this.amka = amka; }
23
24    public String getWeight() { return weight; }
25
26    public void setWeight(String weight) { this.weight = weight; }
27
28    public String getLength() { return length; }
29
30    public void setLength(String length) { this.length = length; }
31
32    public String getHeadCircumference() { return headCircumference; }
33
34    public void setHeadCircumference(String headCircumference) {this.headCircumference = headCircumference;}
35
36    public String getMeasurementDate() { return measurementDate; }
37
38    public void setMeasurementDate(String measurementDate) {this.measurementDate = measurementDate;}
39
40    public String getAge() { return age; }
```

Figure 75.Development Class

```
73    public void setAge(String age) { this.age = age; }
74
75    public String getAgeType() { return ageType; }
76
77    public void setAgeType(String ageType) { this.ageType = ageType; }
78
79    public ArrayList<SustenanceItems> getSustenance() { return sustenance; }
80
81    public void setSustenance(ArrayList<SustenanceItems> sustenance) {this.sustenance = sustenance;}
82
83    public ArrayList<ExaminationItems> getExamination() { return examination; }
84
85    public void setExamination(ArrayList<ExaminationItems> examination) {this.examination = examination;}
86
87    public ArrayList<DevelopmentalItems> getDevelopmentalMonitoring() {return developmentalMonitoring;}
88
89    public void setDevelopmentalMonitoring(ArrayList<DevelopmentalItems> developmentalMonitoring) {
90        this.developmentalMonitoring = developmentalMonitoring;
91    }
92
93    public Boolean getHearing() { return hearing; }
94
95    public void setHearing(Boolean hearing) { this.hearing = hearing; }
96
97    public String getObservations() { return observations; }
98
99    public void setObservations(String observations) { this.observations = observations; }
100
101    public String getDoctor() { return doctor; }
102
103    public void setDoctor(String doctor) { this.doctor = doctor; }
104 }
```

Figure 76.Development Class

## 5.5 DevelopmentalItems Class

Η κλάση DevelopmentalItems περιλαμβάνει τα στοιχεία της παρακολούθησης της ανάπτυξης και συγκεκριμένα, το όνομα της ανάπτυξης, την ηλικία που αφορά και λεπτομέρειες που θα θέλει να συμπληρώσει ο γιατρός που κάνει την καταγραφή, τους constructors, τους getters και τους setters.

```
3 public class DevelopmentalItems {
4     private String name, ageGap, details;
5     |
6     public DevelopmentalItems(String name, String ageGap, String details) {
7         this.name = name;
8         this.ageGap = ageGap;
9         this.details = details;
10    }
11
12    public DevelopmentalItems(){ }
13
14    public String getName() { return name; }
15
16    public void setName(String name) { this.name = name; }
17
18    public String getAgeGap() {return ageGap;}
19
20    public void setAgeGap(String ageGap) { this.ageGap = ageGap; }
21
22    public String getDetails() { return details; }
23
24    public void setDetails(String details) { this.details = details; }
25
26 }
```

Figure 77.DevelopmentalItems Class

## 5.6 Doctor Class

Η κλάση Doctor περιέχει τα στοιχεία που είναι απαραίτητα για την ύπαρξη του ιατρού στην εφαρμογή. Πιο αναλυτικά, το όνομα, το επίθετο, τον αριθμό τηλεφώνου, το email, τον αριθμό ιατρικής ταυτότητας, την λίστα των παιδιών που παρακολουθεί, τους constructors, τους getters και τους setters. Ο αριθμός ιατρικής ταυτότητας είναι μοναδικός για κάθε γιατρό

```
6 public class Doctor {
7     private String name, medicalID, phoneNumber, email, surname;
8     private ArrayList<Baby> kids;
9
10    public Doctor(String name, String medicalID, String phoneNumber, String email,
11        ArrayList<Baby> kids, String surname) {...}
12
13
14
15
16
17
18
19
20    public Doctor() {}
21
22    public String getSurname() { return surname; }
23
24
25
26    public void setSurname(String surname) { this.surname = surname; }
27
28
29
30    public String getName() { return name; }
31
32
33
34    public void setName(String name) { this.name = name; }
35
36
37
38    public String getMedicalID() { return medicalID; }
39
40
41
42    public void setMedicalID(String medicalID) { this.medicalID = medicalID; }
43
44
45
46    public String getPhoneNumber() { return phoneNumber; }
47
48
49
50    public void setPhoneNumber(String phoneNumber) { this.phoneNumber = phoneNumber; }
51
52
53
54    public String getEmail() { return email; }
55
56
57
58    public void setEmail(String email) { this.email = email; }
59
60
61
62    public ArrayList<Baby> getKids() { return kids; }
63
64
65
66    public void setKids(ArrayList<Baby> kids) { this.kids = kids; }
67
68
69 }
70
```

Figure 78. Doctor Class

## 5.7 ExaminationItems Class

Η κλάση ExaminationItems εμπεριέχει όλα τα στοιχεία που θα χρειαστούν κατά την εξέταση του μωρού από τον γιατρό, δηλαδή, το όνομα της εξέτασης, οι ηλικίες που αφορά καθώς και λεπτομέρειες που μπορεί ο ίδιος να συμπληρώσει. Τέλος οι constructors, οι getters και οι setters.

```
2
3  public class ExaminationItems {
4      private String name, ageGap;
5      private int details;
6
7      public ExaminationItems() {
8      }
9
10     public ExaminationItems(String name, String ageGap, Integer details) {
11         this.name = name;
12         this.ageGap = ageGap;
13         this.details = details;
14     }
15
16     public String getName() { return name; }
17
18
19     public void setName(String name) { this.name = name; }
20
21
22
23     public String getAgeGap() { return ageGap; }
24
25
26
27     public void setAgeGap(String ageGap) { this.ageGap = ageGap; }
28
29
30
31     public int getDetails() { return details; }
32
33
34
35     public void setDetails(int details) { this.details = details; }
36
37
38
39 }
```

Figure 79. ExaminationItems Class

## 5.8 FamilyHistoryIllnesses Class

Η κλάση `FamilyHistoryIllnesses` περιέχει τις λεπτομέρειες που αφορούν το οικογενειακό ιστορικό. Για τον λόγο αυτό, για κάθε ασθένεια αποθηκεύεται το όνομα, οι λεπτομέρειες και μία μεταβλητή `switches` που εξηγεί αν το παιδί έχει την συγκεκριμένη ασθένεια. Τέλος υπάρχουν οι constructors και οι αντίστοιχοι getters και setters.

```
2
3  public class FamilyHistoryIllnesses {
4      private String illness, details;
5      private Boolean switches;
6
7      public FamilyHistoryIllnesses(String illness, Boolean switches, String details) {
8          this.illness = illness;
9          this.switches = switches;
10         this.details = details;
11     }
12
13     public FamilyHistoryIllnesses() {}
14
15     public String getIllness() { return illness; }
16
17
18     public void setIllness(String illness) { this.illness = illness; }
19
20
21
22     public Boolean getSwitches() { return switches; }
23
24
25
26     public void setSwitches(Boolean switches) { this.switches = switches; }
27
28
29
30     public String getDetails() { return details; }
31
32
33
34     public void setDetails(String details) { this.details = details; }
35
36
37 }
38
39
```

Figure 80. `FamilyHistoryIllnesses` Class



## 5.9 Parent Class

Η κλάση Parent περιέχει τις απαραίτητες πληροφορίες για τους γονείς/κηδεμόνες. Δηλαδή, το όνομα, το επίθετο, τον ΑΜΚΑ, το τηλέφωνο επικοινωνίας, το email, την ημερομηνία γέννησης, τον τύπο αίματος, την λίστα των παιδιών που έχει εγγράψει στην εφαρμογή, αν υπάρχει συν-κηδεμόνας και αν αυτός υπάρχει το ΑΜΚΑ του, οι constructors, οι getters και οι setters. Αν ο κηδεμόνας δεν υπάρχει το partner είναι false και το partnersAmka παίρνει την τιμή 00000000000. Επίσης κάθε αριθμός ΑΜΚΑ είναι μοναδικός.

```
6 public class Parent implements Serializable {
7     private String name, surname, amka, phoneNumber, email, dateOfBirth, bloodType, partnersAmka;
8     private Boolean partner;
9     private ArrayList<Baby> kids;
10
11     public Parent(String name, String surname, String amka, String phoneNumber, String email,
12         String dateOfBirth, String bloodType, String partnersAmka, Boolean partner,
13         ArrayList<Baby> kids){...}
14
15
16
17
18
19
20
21
22
23
24
25
26 @ public Parent(Parent u){...}
27
28
29 public Parent() {}
30
31
32
33
34
35
36
37
38
39
40
41 public ArrayList<Baby> getKids() { return kids; }
42
43
44
45 public void setKids(ArrayList<Baby> kids) { this.kids = kids; }
46
47
48
49 public String getName() { return name; }
50
51
52 public void setName(String name) { this.name = name; }
53
54
55
56 public String getSurname() { return surname; }
57
58
59 public void setSurname(String surname) { this.surname = surname; }
60
61
62
63
64 public String getAmka() { return amka; }
65
66
67 public void setAmka(String amka) { this.amka = amka; }
68
69
70
71 public String getPhoneNumber() { return phoneNumber; }
72
73
74 public void setPhoneNumber(String phoneNumber) { this.phoneNumber = phoneNumber; }
75
76
77
78
79 public String getEmail() { return email; }
80
81
82 public void setEmail(String email) { this.email = email; }
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2
```

## 5.10 RecyclerView Adapter Class

Η κλάση RecyclerViewAdapter χρησιμοποιείται για την φόρτωση των δεδομένων στους recycler viewers. Παρακάτω αναλύεται ένα τμήμα του κώδικα καθώς και το πως επιτυγχάνεται η σωστή λειτουργία αυτού. Αρχικά δηλώνουμε τα attributes και τα interface (Figure 83). Στην συνέχεια στην συνάρτηση onCreateViewHolder διευκρινίζουμε πιο αρχείο xml θα χρησιμοποιηθεί κάθε φορά (Figure 84). Ακολουθεί η συνάρτηση onBindViewHolder η οποία τοποθετεί όλα τα δεδομένα στα αντίστοιχα views (Figure 85), η getItemCount που επιστρέφει το μέγεθος της λίστας (Figure 86) και ο constructor της κλάσης (Figure 86). Υποκλάση της RecyclerViewAdapter είναι η MyViewHolder στην οποία αρχικά δηλώνουμε τα views που θα χρησιμοποιηθούν (Figure 87). Ο constructor της υποκλάσης βρίσκει τα views στο xml αρχείο για να μπορέσουν να φορτωθούν σωστά τα δεδομένα αλλά και να υπάρχει αλληλεπίδραση με τον χρήστη (Figure 88).

```
31 public class RecyclerViewAdapter extends RecyclerView.Adapter<RecyclerViewAdapter.MyViewHolder> {
32     private ArrayList<list>;
33     private String id, userType;
34     private RecyclerView.OnClickListeners listener;
35     private radioButtonChange radioButtonChange;
36     private textChange textChange;
37     private sustenanceCheck sustenanceCheck;
38     private addVaccination addVaccination;
39
40     public interface addVaccination{ void addVaccine(int position);}
41
42     public void addVaccination(addVaccination addVaccination){this.addVaccination = addVaccination;}
43
44     public interface sustenanceCheck{void sustenanceChecked(int position, Boolean value);
45     }
46
47     public void sustenanceCheck(sustenanceCheck sustenanceCheck){this.sustenanceCheck = sustenanceCheck;}
48
49     public interface textChange {void textChanged(int position, String text);
50     }
51
52     public void textChange(textChange textChange) { this.textChange =textChange; }
53
54     public interface radioButtonChange{void rChange(int id, int position);}
55
56     public void radioButtonChange(radioButtonChange radioButtonChange){
57         this.radioButtonChange = radioButtonChange;
58     }
59
60 }
61
```

Figure 83.RecyclerView Class

```
64 @NonNull
65 @Override
66 public RecyclerViewAdapter.MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
67     View itemView = LayoutInflater.from(parent.getContext()).inflate(R.layout.family_history_recycler, parent, attachToRoot: false);
68     if (id.equals("illnessInput")) {
69         itemView = LayoutInflater.from(parent.getContext()).inflate(R.layout.family_history_recycler, parent, attachToRoot: false);
70         RecyclerViewAdapter.MyViewHolder view = new RecyclerViewAdapter.MyViewHolder(itemView, radioButtonChange, textChange, sustenanceCheck);
71     }else if(id.equals("examination")){...}else if(id.equals("developmental")){...}else if(id.equals("sustenance")){...} else if(id.equals("
94     return new MyViewHolder(itemView, radioButtonChange, textChange, sustenanceCheck);
95 }

```

Figure 84.RecyclerView Class

```

97      @Override
98      public void onBindViewHolder(@NonNull RecyclerView.ViewHolder holder, int position) {
99          if (id.equals("illnessInput")) {
100              List<FamilyHistoryIllnesses> lists = (List<FamilyHistoryIllnesses>) (List<?>) list;
101              String name = lists.get(position).getIllness();
102              holder.illnessName.setText(name);
103              String details = lists.get(position).getDetails();
104              holder.illnessDetails.setText(details);
105              Boolean sw = lists.get(position).getSwitches();
106              if(sw){
107                  holder.switches.setChecked(true);
108                  holder.illnessDetailsLayout.setVisibility(View.VISIBLE);
109              }else{
110                  holder.switches.setChecked(false);
111                  holder.illnessDetailsLayout.setVisibility(View.GONE);
112              }
113          }
114      }
115  }
116

```

Figure 85. RecyclerView Class

```

280      @Override
281      public int getItemCount() {return list.size();}
282
283      public RecyclerViewAdapter(recyclerViewOnClickListener listener, ArrayList list, String id, String userType) {
284          this.list = list;
285          this.id=id;
286          this.listener = listener;
287          this.userType = userType;
288      }
289

```

Figure 86. RecyclerView Class

```

289      public class MyViewHolder extends RecyclerView.ViewHolder implements View.OnClickListener{
290          private TextView illnessName, examinationListitemText, developmentalListitemText, ageTextView, dateTextView,
291              nameDevelopmentalMonitoringTextView, detailsDevelopmentalMonitoringTextView, nameExaminationTextView,
292              nameTextViewDelete, ageTextViewDelete, birthDateTextViewDelete, ageTextViewDelete, ageAvailableChildrenText,
293              nameAvailableChildrenText, illnessNameTextView, illnessDetailsTextView, vaccinationNameTextView,
294              vaccinationDateTextView, vaccinationDoctorTextView;
295          private TextInputEditText illnessDetails, developmentalEditText;
296          private TextInputLayout illnessDetailsLayout, developmentalLayout;
297          private Switch switches;
298          private RadioButton radioButton1, radioButton2, radioButton3;
299          private RadioGroup radioGroup;
300          private CardView card, cardViewAdd, cardViewExamination, cardViewVaccinations, cardViewDevelopments, cardViewDeleteChild;
301          private boolean is=true;
302          private Button addVaccineButton;
303          private ImageView sexImageViewDelete, sexAvailableChildren;
304          private CheckBox checkedSustenanceMonitoring, checkShowDevelopments;
305      }
306

```

Figure 87. RecyclerView Class

```

307      public MyViewHolder(Final View itemView, radioButtonChange radioButtonChange, textChange textChange, sustenanceCheck sustenanceCheck) {
308          super(itemView);
309          if (id.equals("illnessInput")) {
310              List<FamilyHistoryIllnesses> lists = (List<FamilyHistoryIllnesses>) (List<?>) list;
311              illnessName = itemView.findViewById(R.id.illnessName);
312              switches = itemView.findViewById(R.id.switches);
313              illnessDetails = itemView.findViewById(R.id.illnessDetails);
314              illnessDetailsLayout = itemView.findViewById(R.id.illnessDetailsTextInputLayout);
315              switches.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
316                  @Override
317                  public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
318                      int i = getAdapterPosition();
319                      if(switches.isChecked()){
320                          illnessDetailsLayout.setVisibility(View.VISIBLE);
321                      }else{
322                          illnessDetailsLayout.setVisibility(View.GONE);
323                      }
324                      FamilyHistoryIllnesses f1 = new FamilyHistoryIllnesses(lists.get(i).getIllness(), switches.isChecked(), lists.get(i).getDetails());
325                      lists.set(i, f1);
326                  }
327              });
328          }
329      }
330
331      @Override
332      public void onClick(View v) {...}
333
334      public interface recyclerViewOnClickListener {
335          void onClick (View view,int position);
336      }
337

```

Figure 88. RecyclerView Class

## 5.11 Sustenanceltems Class

Η κλάση Sustenanceltems περιέχει τις λεπτομέρειες που αφορούν την διατροφή του παιδιού και πιο συγκεκριμένα, το όνομα του διατροφικού προϊόντος, την ηλικιακή ομάδα που αφορά, και μία μεταβλητή σχετικά με τον αν το μωρό καταναλώνει αυτό τον τύπου φαγητού. Τέλος περιέχει τους constructors, τους getters και τους setters.

```
3 public class SustenanceItems {
4     private String name, ageGap;
5     private Boolean checked;
6
7     public SustenanceItems(String name, String ageGap, Boolean checked) {
8         this.name = name;
9         this.ageGap = ageGap;
10        this.checked = checked;
11    }
12
13    public SustenanceItems(){ }
14
15    public String getName() { return name; }
16
17    public void setName(String name) { this.name = name; }
18
19    public String getAgeGap() { return ageGap; }
20
21    public void setAgeGap(String ageGap) { this.ageGap = ageGap; }
22
23    public Boolean getChecked() { return checked; }
24
25    public void setChecked(Boolean checked) { this.checked = checked; }
26 }
27
28
29
```

Figure 89.Sustenance Class

## 5.12 Vaccination Class

Η κλάση Vaccination περιέχει τις απαραίτητες πληροφορίες για κάθε εμβόλιο. Δηλαδή, το όνομα του εμβολίου, έναν μοναδικό κωδικό, το όνομα του γιατρού που χορήγησε το εμβόλιο και την ημερομηνία εμβολιασμού. Αν δεν έχει εκτελεστεί κάποιο εμβόλιο ακόμα, τα δύο τελευταία αποθηκεύονται στην βάση ως null.

```
3 public class Vaccination {
4     private String name, date, doctorName;
5     private int uniqueID;
6
7     public Vaccination() {}
8
9     public Vaccination(String name, String date, String doctorName, int uniqueID) {
10         this.name = name;
11         this.date = date;
12         this.doctorName = doctorName;
13         this.uniqueID = uniqueID;
14     }
15
16     public String getName() { return name; }
17
18     public void setName(String name) { this.name = name; }
19
20     public String getDate() { return date; }
21
22     public void setDate(String date) { this.date = date; }
23
24     public String getDoctorName() { return doctorName; }
25
26     public void setDoctorName(String doctorName) { this.doctorName = doctorName; }
27
28     public int getUniqueID() { return uniqueID; }
29
30     public void setUniqueID(int uniqueID) { this.uniqueID = uniqueID; }
31 }
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
```

Figure 90.Vaccination Class

## 6. Activities

### 6.1 AddBaby Activity

Το `addBaby Activity` είναι υπεύθυνο για την εγγραφή μωρού στην εφαρμογή. Συγκεκριμένα, κατά την συμπλήρωση των στοιχείων του παιδιού γίνεται έλεγχος ορθότητας των στοιχείων. Με το πάτημα του κουμπιού *Next* εκτελείται έλεγχος μοναδικότητας του AMKA του μωρού και ο έλεγχος της εφαρμογής μεταφέρεται στο `FamilyHistoryInput Activity`.

### 6.2 AddChildToDoctor Activity

Το `addChildToDoctor Activity` παρέχει την δυνατότητα στο χρήστη (γιατρό) να ενημερωθεί για τα μωρά που υπάρχουν στην βάση δεδομένων αλλά και να ψάξει κάποιο βάσει του AMKA του. Επιλέγοντας κάποιο παιδί ο χρήστης βλέπει περισσότερα στοιχεία για αυτό και αν πατήσει το κουμπί *Add Child* του ζητείται να εισάγει το AMKA ενός εκ των κηδεμόνων του για να γίνει η βεβαίωση προσθήκης. Σε περίπτωση που ο γιατρός έχει ήδη αναλάβει το παιδί του εμφανίζεται μήνυμα, διαφορετικά (και έχοντας εισάγει σωστά τον AMKA) ενημερώνεται η βάση δεδομένων, αλλά και ο ίδιος με μήνυμα.

### 6.3 CreateNewDoctor Activity

Το Activity `createNewDoctor` χρησιμοποιείται για την εγγραφή του χρήστη (γιατρός). Κατά την συμπλήρωση των πεδίων γίνεται έλεγχος ορθότητας των στοιχείων που εισάγει ο χρήστης και με το πάτημα του κουμπιού *Save* εκτελείται έλεγχος μοναδικότητας του αριθμού ιατρικής ταυτότητας.

### 6.4 CreateNewParent Activity

Στο `createNewParent Activity` εκτελείται η εγγραφή του χρήστη (κηδεμόνας). Κατά την συμπλήρωση των στοιχείων γίνεται έλεγχος ορθότητας των πληροφοριών καθώς και έλεγχος μοναδικότητας του AMKA. Αν ο AMKA είναι μοναδικός ο έλεγχος της εφαρμογής μεταβαίνει στο `nextParent Activity`, σε διαφορετική περίπτωση γίνεται έλεγχος αν ο κηδεμόνας έχει την ιδιότητα του χρήστη ή είναι απλώς κηδεμόνας (π.χ. συν-κηδεμόνας άλλου χρήστη). Αν ισχύει η ιδιότητα του χρήστη εμφανίζεται μήνυμα ανάγκης συμπλήρωσης διαφορετικού AMKA αλλιώς ο χρήστης ρωτάτε αν θέλει να χρησιμοποιήσει τα στοιχεία του κηδεμόνα που υπάρχει ήδη στην βάση.

## 6.5 DeleteChild Activity

Στο deleteChild Activity εκτελείται διαγραφή κάποιου μωρού από την βάση. Συγκεκριμένα με την είσοδο του χρήστη σε αυτή την σελίδα εμφανίζονται τα μωρά που έχει εγγράψει στην εφαρμογή, στην συνέχεια αφού επιλέξει κάποιο μωρό και επιβεβαιώσει την διαγραφή του ενημερώνεται ο πίνακας του χρήστη, ο πίνακας των γιατρών καθώς και ο πίνακας με τις αναπτυξιακές παρακολουθήσεις.

## 6.6 DoctorParentChoose Activity

Το doctorParentChoose Activity είναι υπεύθυνο για τον έλεγχο του τύπου χρήστη που χρησιμοποιεί την εφαρμογή. Ειδικότερα μετά την είσοδο του, το πρόγραμμα ελέγχει αν ο χρήστης είναι κηδεμόνας ή γιατρός και μεταβαίνει στην αντίστοιχη σελίδα. Αν δεν υπάρχουν πληροφορίες για τον τύπο χρήστη, αυτό σημαίνει ότι δεν έχει ολοκληρωθεί η εγγραφή του στην εφαρμογή και καλείται ο ίδιος να επιλέξει τον τύπο που θέλει και ο έλεγχος μεταβαίνει στην αντίστοιχη σελίδα ολοκλήρωσης εγγραφής.

## 6.7 FamilyHistoryInput Activity

Μέσω του familyHistoryInput Activity ο χρήστης ολοκληρώνει την εγγραφή του μωρού με την συμπλήρωση του οικογενειακού ιστορικού. Έτσι με το πάτημα του κουμπιού Save ενημερώνεται ο πίνακας των παιδιών και του κηδεμόνα.

## 6.8 LoginRegister Activity

Μέσω του loginRegister Activity επιτυγχάνεται η είσοδος του χρήστη στην εφαρμογή είτε με κωδικό και email είτε με χρήση λογαριασμού Facebook ή Google. Σε κάθε περίπτωση πριν την είσοδο γίνεται έλεγχος αν έχει πραγματοποιηθεί πρώτα εγγραφή και εμφανίζεται αντίστοιχο μήνυμα. Για την χρήση Facebook και Google ο χρήστης μεταβαίνει στον αντίστοιχο πάροχο για επιβεβαίωση των στοιχείων.

## 6.9 MainScreenDoctor Activity

Το mainScreenDoctor Activity αποτελεί την αρχική οθόνη του χρήστη (γιατρός). Αρχικά γίνεται έλεγχος σχετικά με το αν ο χρήστης έχει αναλάβει υπό την παρακολούθηση του παιδιά. Αν ναι του εμφανίζεται μία λίστα με αυτά, αλλιώς του εμφανίζεται ανάλογο μήνυμα. Επιλέγοντας κάποιο παιδί του εμφανίζονται περισσότερες πληροφορίες για αυτό αλλά και οι διαθέσιμες ενέργειες που έχει. Αν επιλέξει να διαγράψει κάποιο μωρό του ζητείται πρώτα η εισαγωγή AMKA για την επιβεβαίωση της διαγραφής.

## 6.10 mainScreenParents Activity

Το mainScreenParents Activity αποτελεί την αρχική σελίδα του χρήστη (κηδεμόνας). Με την είσοδο στην οθόνη αυτή γίνεται έλεγχος ύπαρξης εγγεγραμμένων μωρών από το χρήστη και του εμφανίζεται είτε ένα μήνυμα είτε κάποια στατιστικά στοιχεία αναφορικά με την ανάπτυξη του μωρού.

## 6.11 MonitoringDevelopment Activity

Στο monitorinDevelopment Activity επιτυγχάνεται η καταγραφή της αναπτυξιακής εξέλιξης για κάποιο μωρό από τον χρήστη (γιατρό). Το πρόγραμμα παίρνει από την βάση δεδομένων τα στοιχεία που αφορούν την διατροφή, την αναπτυξιακή παρακολούθηση και την εξέταση του παιδιού ανάλογα με την ηλικία του. Με το πάτημα του κουμπιού *Save Development* εκτελείται έλεγχος ορθότητας όλων των στοιχείων και ενημερώνεται η βάση δεδομένων.

## 6.12 NextParent Activity

Το nextParent Activity είναι υπεύθυνο για την εγγραφή του συν-κηδεμόνα. Αν ο χρήστης επιλέξει το κουμπί *Skip* ολοκληρώνεται η εγγραφή του χρήστη χωρίς συν-κηδεμόνα. Αν συμπληρώσει τα πεδία και πατήσει το κουμπί *Next* γίνεται έλεγχος ορθότητας των στοιχείων καθώς και έλεγχος μοναδικότητας του ΑΜΚΑ. Αν δεν πληρείται η προϋπόθεση της μοναδικότητας ο χρήστης ερωτάται αν θέλει να συνεχίσει με συν-κηδεμόνα τον ήδη εγγεγραμμένο χρήστη ή αν θέλει να αλλάξει τον ΑΜΚΑ που εισήγαγε. Σε κάθε περίπτωση ολοκληρώνεται η εγγραφή του χρήστη και ενημερώνεται η βάση.

## 6.13 RegisterUsernamePassword Activity

Το registerUsername Activity χρησιμοποιείται για την εγγραφή του χρήστη στην εφαρμογή. Ο χρήστης έχει την δυνατότητα να χρησιμοποιήσει email και κωδικό ή λογαριασμό Facebook ή λογαριασμό Google. Για την πρώτη επιλογή απαιτείται η συμπλήρωση του κωδικού ξανά για επιβεβαίωση. Για την χρήση Facebook και Google γίνεται μετάβαση στον αντίστοιχο πάροχο όπου ζητούνται τα στοιχεία σύνδεσης για την ολοκλήρωση πιστοποίησης του χρήστη. Σε κάθε περίπτωση γίνεται έλεγχος ύπαρξης του email στην βάση δεδομένων και εμφανίζεται μήνυμα λάθους στην περίπτωση που υπάρχει.



### 6.14 ShowDevelopmentsList Activity

Το showDevelopmentsList Activity χρησιμοποιείται για την παρακολούθηση της ανάπτυξης του μωρού και συγκεκριμένα για την εμφάνιση των μέχρι τώρα καταγραφών. Το πρόγραμμα εμφανίζει μία λίστα με τις καταγραφές αναφορικά με το κάθε μωρό και επιλέγοντας ο χρήστης κάποια του εμφανίζονται όλες οι πληροφορίες σχετικά με αυτή.

### 6.15 ShowHistoric Activity

Το showHistoric Activity χρησιμοποιείται για την εμφάνιση του οικογενειακού ιστορικού κάποιου μωρού στον χρήστη (γιατρό). Το πρόγραμμα παίρνει από την βάση δεδομένων το ιστορικό και το εμφανίζει σε μία λίστα, στην οποία με κόκκινο χρώμα είναι οι ασθένειες που δεν σχετίζονται με την οικογένεια, ενώ με πράσινο αυτές που υπάρχουν στο οικογενειακό περιβάλλον.

### 6.16 UserAccount Activity

Στο userAccount Activity περιέχονται οι ρυθμίσεις της εφαρμογής. Αναλυτικότερα εμφανίζονται τα στοιχεία του χρήστη και οι διαθέσιμες επιλογές του ανάλογα με τον τύπο χρήστη. Σε οποιαδήποτε περίπτωση αν θελήσει να κάνει αλλαγή email ή και κωδικού πρόσβασης αλλά έχει συνδεθεί με Facebook ή Google θα του εμφανιστεί ανάλογο μήνυμα, διαφορετικά αφού επιβεβαιώσει την επιλογή του ενημερώνεται η βάση δεδομένων. Ομοίως, σε οποιαδήποτε περίπτωση ενημέρωσης των στοιχείων ενημερώνεται η βάση δεδομένων. Ωστόσο σε περίπτωση διαγραφής χρήστη (κηδεμόνα) διαγράφονται τα εγγεγραμμένα από αυτών μωρά αν δεν υπάρχει συν-κηδεμόνας, οι καταγραφές αναπτυξιακής παρακολούθησης που έχουν γίνει και ενημερώνεται ο πίνακας των γιατρών.

### 6.17 ViewCharts Activity

Το viewCharts Activity χρησιμοποιείται για την εμφάνιση των διαγραμμάτων βάρους και μήκους. Πιο συγκεκριμένα, με την είσοδο στην σελίδα, το πρόγραμμα παίρνει τα στοιχεία από την βάση δεδομένων ανάλογα με το φύλο του παιδιού και τον τύπο διαγράμματος που επιλέχθηκε. Με την χρήση του MPAndroidChart δημιουργείται το διάγραμμα, στο οποίο με μαύρα χρώμα εμφανίζονται οι καμπύλες που προτείνει ο ΠΟΥ, ενώ με κόκκινο εμφανίζεται η καμπύλη που δημιουργείται από τα στοιχεία των μέχρι τώρα καταγραφών του μωρού.

### 6.18 ViewParentInfo Activity

Το viewParentInfo Activity είναι υπεύθυνο για την εμφάνιση των στοιχείων των κηδεμόνων του μωρού. Αναλυτικά, το πρόγραμμα εμφανίζει αρχικά το ΑΜΚΑ των κηδεμόνων και αφού επιλέξει ΑΜΚΑ ο χρήστης (γιατρός) του εμφανίζονται και όλες οι υπόλοιπες πληροφορίες.

### 6.19 ViewVaccination Activity

Τέλος, το viewVaccination Activity είναι υπεύθυνο για τον εμβολιασμό των μωρών. Ειδικότερα με την είσοδο στην οθόνη εμφανίζεται μία λίστα με τα διαθέσιμα εμβόλια σύμφωνα με το *Βιβλιάριο Υγείας του Παιδιού* που παρέχει το Υπουργείο Παιδείας. Αν έχει γίνει κάποιος εμβολιασμός εμφανίζεται το όνομα του γιατρού και η ημερομηνία που αυτός καταχωρήθηκε αλλιώς εμφανίζεται η λέξη *None*. Αν ο χρήστης έχει την ιδιότητα του γιατρού του εμφανίζονται 2 επιπλέον κουμπιά. Το *Add Vaccine* με το οποίο καταγράφει ένα εμβόλιο που υπάρχει ήδη στην λίστα και το *Add Other Vaccine* με το οποίο καταγράφει έναν νέο εμβολισμό και δηλώνει το όνομα του εμβολίου. Σε κάθε περίπτωση ενημερώνεται αυτόματα η βάση δεδομένων.

## 7. Βιβλιογραφικές πηγές

Για την υλοποίηση της εργασίας χρησιμοποιήθηκαν πληροφορίες από τις παρακάτω ιστοσελίδες:

- <https://www.moh.gov.gr/articles/health/dieythynsh-prwtobathmias-frontidas-ygeias/drasedis-kai-programmata-agwghs-ygeias/oikogeneiakos-programmatismos/4392-bibliario-ygeias-paidiou> ,για την λήψη του Βιβλιαρίου Υγείας Παιδιού
- <https://www.who.int/tools/child-growth-standards> , για πληροφορίες σχετικά με την ανάπτυξη των μωρών
- <https://firebase.google.com/docs> ,για την βάση δεδομένων
- <https://stackoverflow.com/> ,για τον κώδικα
- [https://www.cdc.gov/growthcharts/html\\_charts/wtageinf.htm#females](https://www.cdc.gov/growthcharts/html_charts/wtageinf.htm#females) ,για την δημιουργία των διαγραμμάτων βάρους
- [https://www.cdc.gov/growthcharts/html\\_charts/lenageinf.htm#females](https://www.cdc.gov/growthcharts/html_charts/lenageinf.htm#females) ,για την δημιουργία των διαγραμμάτων μήκους
- <https://developers.facebook.com/> ,για την σύνδεση με το Facebook
- <https://firebase.google.com/docs/auth/android/google-signin> ,για την σύνδεση με το Google.