



Τεχνολογία Λογισμικού
Τομέας Ηλεκτρονικής και Υπολογιστών
Τμήμα ΗΜΜΥ
Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2019



Hap.io
Happy to remind your χάπι

Σχεδίαση Συστήματος

Del.3.1

Έκδοση 0.7
(draft)

Μηναδάκης Μιχαήλ minadakm@ece.auth.gr

Παπαγγέλης Άρης Ελευθέριος aris.papangelis@gmail.com

Παπαδάμ Στέφανος stefanospapadam@gmail.com

Πετρίδης Περικλής Σάββας periclespetrides@gmail.com

28/5/2019

Ιστορικό Αλλαγών

| Όνομα | Ημ/νία | Περιγραφή Αλλαγής | Εκδ. |
|---------------|------------|--|------|
| A. Συμεωνίδης | 29/05/2009 | Δημιουργία Εγγράφου Προσαρμογή του ESA software engineering standards guidelines (1991) και του εγγράφου SDD document, από τους Bruegge και Dutoit (2004). | 0.1 |
| hap.io | 20/5/2019 | Συγγραφή της ενότητας αρχιτεκτονικής συστήματος. | 0.2 |
| hap.io | 23/5/2019 | Συγγραφή μέρους του RESTful API. | 0.3 |
| hap.io | 24/5/2019 | Συγγραφή του υπόλοιπου RESTful API. | 0.4 |
| hap.io | 26/5/2019 | Συγγραφή της ανάλυσης SWOT, της κοστολόγησης και δημιουργία χρονοδιαγράμματος. | 0.5 |
| hap.io | 27/5/2019 | Δημιουργία διαγράμματος Gantt. | 0.6 |
| hap.io | 28/5/2019 | Διορθώσεις και τελική προσαρμογή εγγράφου. | 0.7 |

Μέλη Ομάδας Ανάπτυξης

| Όνομα | ΟΑ | Email |
|----------------------------|--------|--|
| A. Συμεωνίδης | * | asymeon@issel.ee.auth.gr |
| Μηναδάκης Μιχαήλ | Hap.io | minadakm@ece.auth.gr |
| Παπαγγέλης Άρης Ελευθέριος | Hap.io | aris.papagelis@gmail.com |
| Παπαδάμ Στέφανος | Hap.io | stefanospapadam@gmail.com |
| Πετρίδης Περικλής Σάββας | Hap.io | periclespetrides@gmail.com |

Πίνακας Περιεχομένων

| | |
|--|-----------|
| Πίνακας Περιεχομένων | 3 |
| Λίστα Σχημάτων | 5 |
| 1. Προτεινόμενη Αρχιτεκτονική Λογισμικού | 6 |
| 1.1 Αποδόμηση Συστήματος | 6 |
| 1.1.1 Υποσύστημα LoginHandler | 6 |
| 1.1.2 Υποσύστημα LoginGUI | 6 |
| 1.1.3 Υποσύστημα PatientHandler | 7 |
| 1.1.4 Υποσύστημα PatientPageGUI | 7 |
| 1.1.5 Υποσύστημα CaretakerHandler | 8 |
| 1.1.6 Υποσύστημα HomepageGUI | 9 |
| 1.1.7 Υποσύστημα Pill Handler | 9 |
| 1.1.8 Υποσύστημα BiomedicalDataHandler | 10 |
| 1.1.9 Υποσύστημα Notification GUI | 11 |
| 1.1.10 Υποσύστημα DatabaseHandler | 11 |
| 1.1.11 Διάγραμμα τμημάτων | 12 |
| 1.2 Απεικόνιση Υλικού/Λογισμικού | 13 |
| 1.2.1 Client Device | 13 |
| 1.2.2 Server System | 14 |
| 1.2.3 Συνολικό διάγραμμα ανάπτυξης | 14 |
| 1.3 Έλεγχος Πρόσβασης και Ασφάλεια | 16 |
| 2. Πίνακας ιχνηλασιμότητας εγγράφων Σχεδίασης και Απαιτήσεων Λογισμικού | 18 |
| 3. RESTful API | 19 |
| 3.1 Patient API | 19 |
| 3.1.1 GET Patient Info | 19 |
| 3.1.2 Update Patient Info | 20 |
| 3.1.3 Create New Patient | 21 |
| 3.1.4 GET Next Pill Time | 22 |
| 3.1.5 Pill Confirmation POST | 23 |
| 3.2 Pills API | 25 |
| 3.2.1 GET Pills Info | 25 |
| 3.2.2 Update Pills Info | 26 |
| 3.2.3 Add New Pill | 27 |
| 3.3 Biomedical Data API | 29 |
| 4. Ανάλυση SWOT | 30 |

| | |
|--|-----------|
| 5. Χρονοδιάγραμμα και Κοστολόγηση Έργου | 31 |
| 5.1 Έκταση έργου και κοστολόγηση | 31 |
| 5.2 Χρονοδιάγραμμα | 33 |
| 5.3 Διάγραμμα Gantt | 34 |
| Παράρτημα Ι – Ανοιχτά Θέματα | 35 |

Λίστα Σχημάτων

- Σχήμα 1. Υποσύστημα LoginHandler
- Σχήμα 2. Υποσύστημα LoginGUI
- Σχήμα 3. Υποσύστημα PatientHandler
- Σχήμα 4. Υποσύστημα PatientPageGUI
- Σχήμα 5. Υποσύστημα CaretakerHandler
- Σχήμα 6. Υποσύστημα HomepageGUI
- Σχήμα 7. Υποσύστημα PillHandler
- Σχήμα 8. Υποσύστημα BiomedicalDataHandler
- Σχήμα 9. Υποσύστημα NotificationGUI
- Σχήμα 10. Υποσύστημα DatabaseHandler
- Σχήμα 11. Διάγραμμα τμημάτων
- Σχήμα 12. Client Device
- Σχήμα 13. Server System
- Σχήμα 14. Συνολικό Διάγραμμα Ανάπτυξης
- Σχήμα 15. SWOT Ανάλυση
- Σχήμα 16. Χρονοδιάγραμμα
- Σχήμα 17. Διάγραμμα Gantt

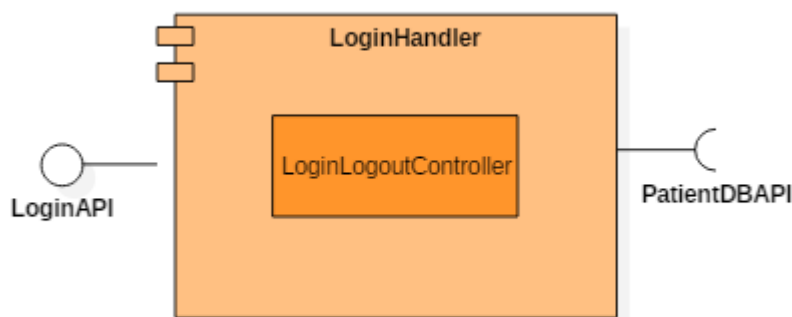
1. Προτεινόμενη Αρχιτεκτονική Λογισμικού

1.1 Αποδόμηση Συστήματος

Στην παράγραφο αυτή περιγράφεται η αποδόμηση του συστήματος, δηλαδή ο διαχωρισμός του συστήματος σε επιμέρους υποσυστήματα ανάλογα με τις λειτουργίες και τις αρμοδιότητες καθενός από αυτά. Κάθε υποσύστημα αποτελείται από ένα τμήμα (component), μέσα στο οποίο έχουν ομαδοποιηθεί κλάσεις του συστήματος που είχαν άμεση συσχέτιση, όπως αυτές αναπτύχθηκαν στο προηγούμενο παραδοτέο. Η αποδόμηση έγινε με κριτήριο της ύπαρξη υψηλής συνεκτικότητας και χαμηλής σύζευξης μεταξύ των κλάσεων κάθε τμήματος, έτσι ώστε αλλαγές σε κάποια κλάση κάποιου τμήματος να επηρεάζουν ελάχιστα ή καθόλου το υπόλοιπο σύστημα.

1.1.1 Υποσύστημα LoginHandler

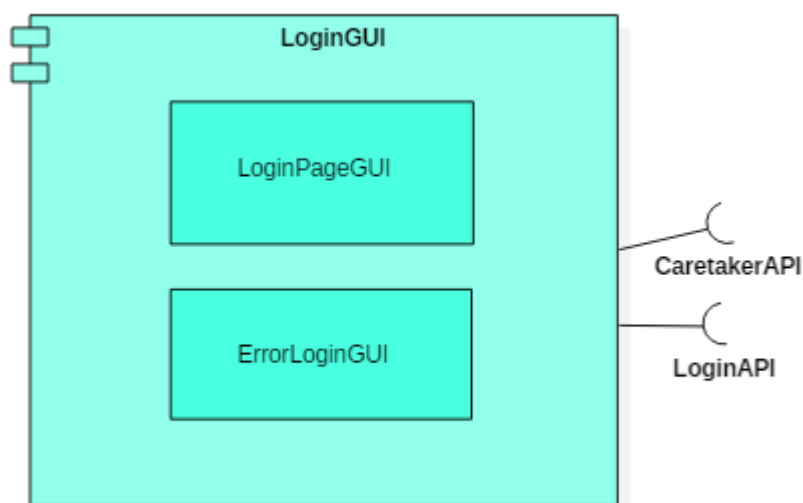
Το υποσύστημα αυτό περιλαμβάνει όλη τη λειτουργικότητα που απαιτείται προκειμένου να συνδεθεί ο χρήστης στο σύστημα ή να αποσυνδεθεί. Αποτελείται από τον ελεγκτή LoginLogoutController. Επικοινωνεί με το υποσύστημα DatabaseHandler, μέσω της διεπαφής PatientDBAPI, προκειμένου να επαληθεύσει τα στοιχεία του χρήστη και να τον συνδέσει στο σύστημα ή να τον αποσυνδέσει από αυτό. Επιπλέον, επικοινωνεί με το υποσυστήματα LoginGUI και HomepageGUI, μέσω της διεπαφής LoginAPI, ώστε να του παρέχει τα δεδομένα που πρέπει να απεικονίσει στην οθόνη.



Σχημα 1. Υποσύστημα LoginHandler

1.1.2 Υποσύστημα LoginGUI

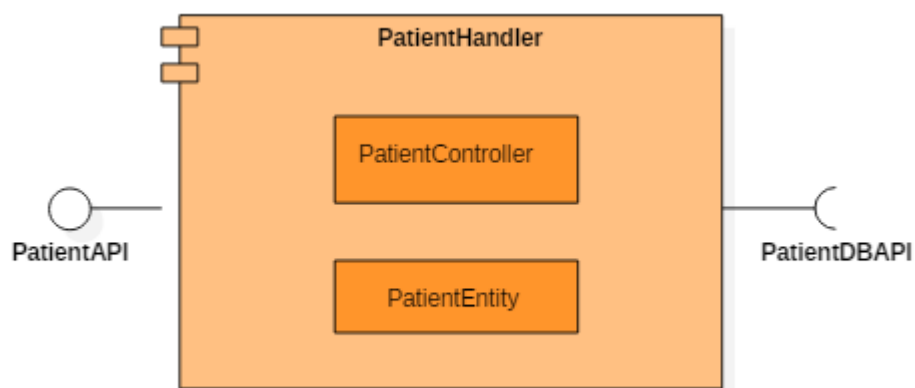
Το υποσύστημα αυτό περιλαμβάνει τις οριακές κλάσεις LoginPageGUI και ErrorLoginGUI, οι οποίες αποτελούν τις διεπαφές μέσω των οποίων ο χρήστης έχει τη δυνατότητα να εισάγει τα προσωπικά του στοιχεία και να συνδεθεί στο σύστημα καθώς και να επιστρέψει στην διαδικασία εισαγωγής σε περίπτωση εισαγωγής λανθασμένων στοιχείων. Επικοινωνεί με το υποσύστημα LoginHandler, μέσω του LoginAPI και με το CaretakerHandler μέσω του CaretakerAPI απαιτώντας από αυτό τις πληροφορίες τις οποίες θα εμφανίσει στην γραφική διεπαφή.



Σχήμα 2. Υποσύστημα LoginGUI

1.1.3 Υποσύστημα PatientHandler

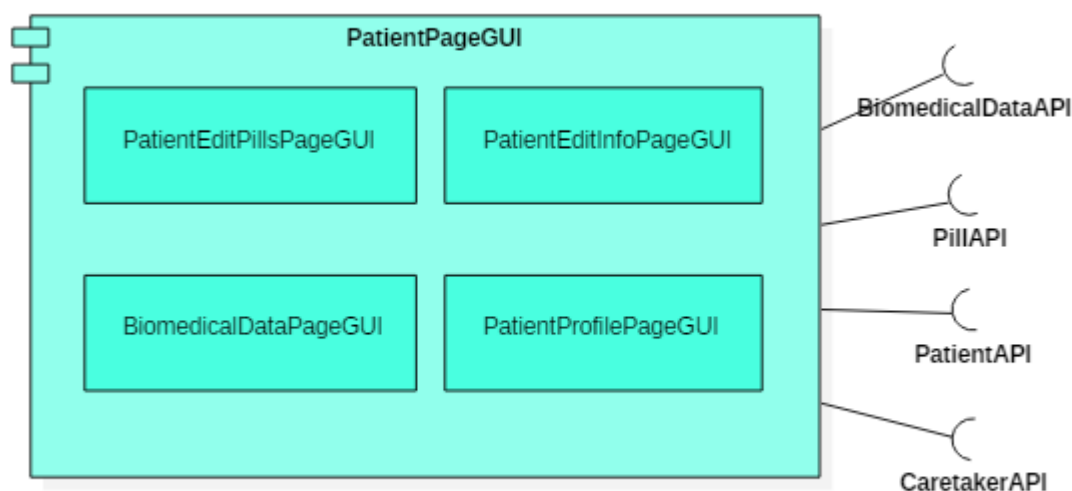
Το υποσύστημα αυτό περιλαμβάνει όλη τη λειτουργικότητα που απαιτείται για τον ασθενή. Αποτελείται από τον ελεγκτή **PatientController** και την οντότητα **PatientEntity**. Επικοινωνεί με το υποσύστημα **DatabaseHandler**, μέσω της διεπαφής **PatientDBAPI**, προκειμένου να έχει πρόσβαση στα στοιχεία του ασθενή. Επιπλέον, επικοινωνεί με τα υποσυστήματα **PatientPageGUI**, **NotificationGUI** και **HomepageGUI** μέσω της διεπαφής **PatientAPI**, ώστε να του παρέχει τα δεδομένα που πρέπει να απεικονίσει στην οθόνη.



Σχήμα 3. Υποσύστημα PatientHandler

1.1.4 Υποσύστημα PatientPageGUI

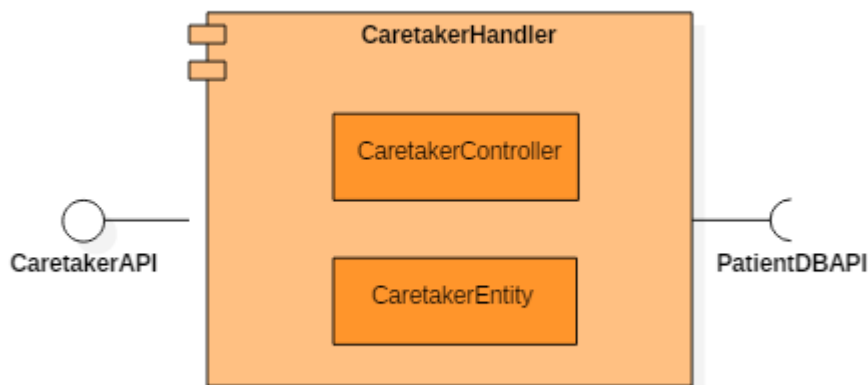
Το υποσύστημα αυτό περιλαμβάνει τις οριακές κλάσεις PatientEditPillsPageGUI, PatientEditInfoPageGUI, BiomedicalDataPageGUI και PatientProfilePageGUI οι οποίες αποτελούν τις διεπαφές μέσω των οποίων ο χρήστης έχει τη δυνατότητα να προβάλει και να επεξεργαστεί τα στοιχεία του ασθενή. Επικοινωνεί με το υποσύστημα BiomedicalDataHandler μέσω του BiomedicalDataAPI, με το PillHandler μέσω του PillAPI, με το PatientHandler μέσω του PatientAPI και με το CaretakerHandler μέσω του CaretakerAPI απαιτώντας από αυτό τις πληροφορίες τις οποίες θα εμφανίσει στην γραφική διεπαφή.



Σχημα 4. Υποσύστημα PatientPageGUI

1.1.5 Υποσύστημα CaretakerHandler

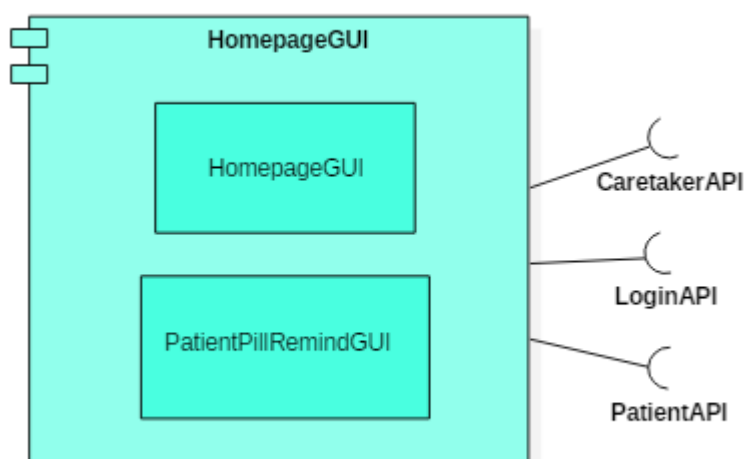
Το υποσύστημα αυτό περιλαμβάνει όλη τη λειτουργικότητα που απαιτείται για τον caretaker. Αποτελείται από τον ελεγκτή CaretakerController και την οντότητα CaretakerEntity. Επικοινωνεί με το υποσύστημα DatabaseHandler, μέσω της διεπαφής PatientDBAPI, προκειμένου να έχει πρόσβαση στα στοιχεία του ασθενή. Επιπλέον, επικοινωνεί με το υποσύστημα PatientPageGUI, NotificationGUI, LoginGUI και HomepageGUI μέσω της διεπαφής CaretakerAPI, ώστε να του παρέχει τα δεδομένα που πρέπει να απεικονίσει στην οθόνη.



Σχημα 5. Υποσύστημα CaretakerHandler

1.1.6 Υποσύστημα HomepageGUI

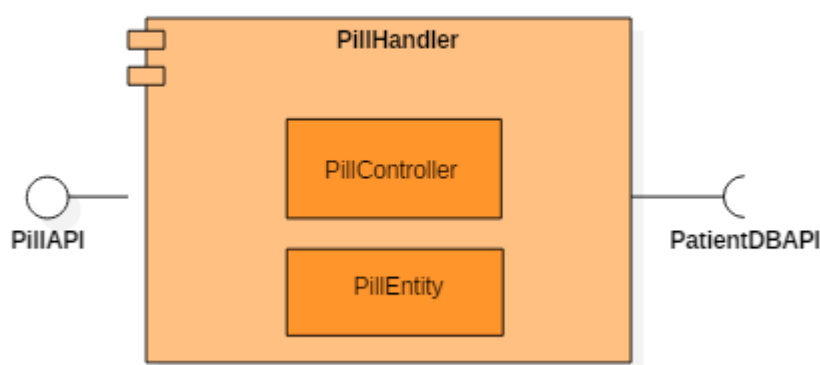
Το υποσύστημα αυτό περιλαμβάνει τις οριακές κλάσεις HomepageGUI και PatientPillRemindGUI, οι οποίες αποτελούν τις διεπαφές μέσω των οποίων ο ασθενής έχει την δυνατότητα να προβάλει την αρχική σελίδα του wearable και το wearable έχει την δυνατότητα να του υπενθυμίσει για το χάπι του. Επικοινωνεί με το υποσύστημα LoginHandler, μέσω του LoginAPI, με το CaretakerHandler μέσω του CaretakerAPI και με το PatientHandler με το PatientAPI απαιτώντας από αυτό τις πληροφορίες τις οποίες θα εμφανίσει στην γραφική διεπαφή.



Σχημα 6. Υποσύστημα HomepageGUI

1.1.7 Υποσύστημα Pill Handler

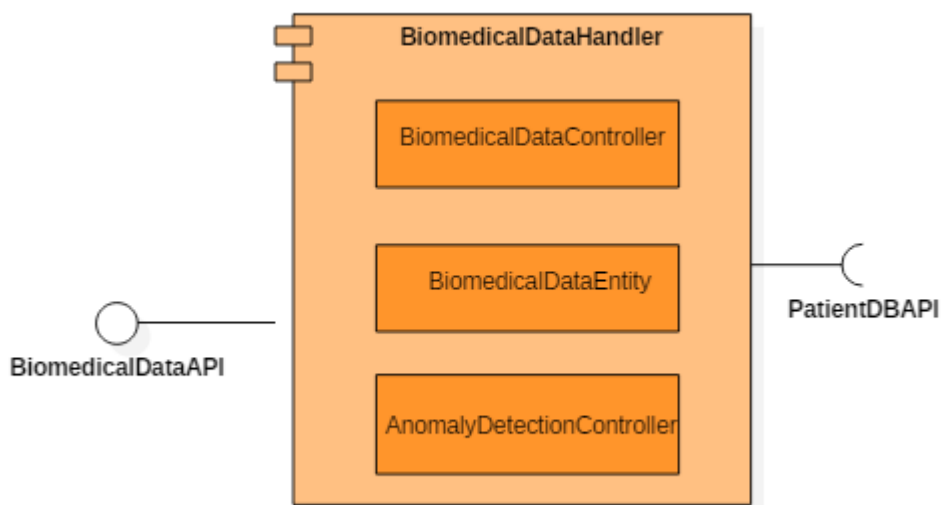
Το υποσύστημα αυτό περιλαμβάνει όλη τη λειτουργικότητα που απαιτείται για τον την οντότητα του χαπιού. Αποτελείται από τον ελεγκτή PillController και την οντότητα PillEntity. Επικοινωνεί με το υποσύστημα DatabaseHandler, μέσω της διεπαφής PatientDBAPI, προκειμένου να έχει πρόσβαση στα στοιχεία του ασθενή. Επιπλέον, επικοινωνεί με το υποσυστήματα PatientPageGUI και NotificationGUI μέσω της διεπαφής PillAPI, ώστε να του παρέχει τα δεδομένα που πρέπει να απεικονίσει στην οθόνη.



Σχημα 7. Υποσύστημα PillHandler

1.1.8 Υποσύστημα BiomedicalDataHandler

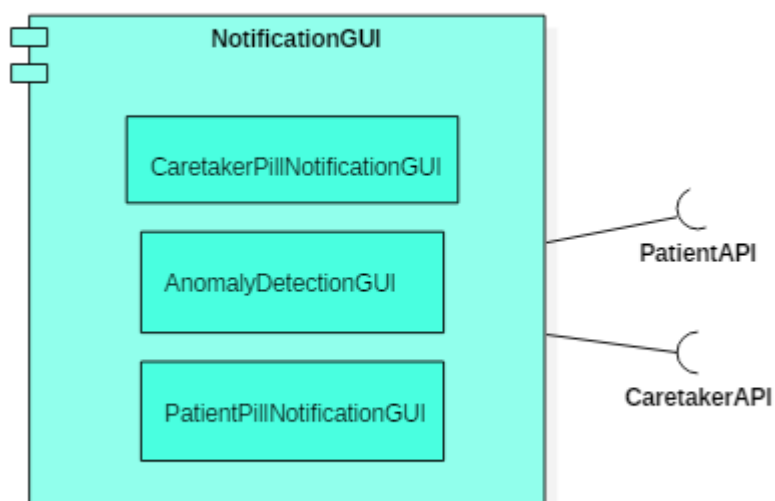
Το υποσύστημα αυτό περιλαμβάνει όλη τη λειτουργικότητα που απαιτείται για τον την οντότητα του χαπιού. Αποτελείται από τους ελεγκτές BiomedicalDataController και AnomalyDetectionController και την οντότητα BiomedicalDataEntity. Επιπλέον, επικοινωνεί με το υποσυστήματα PatientPageGUI και NotificationGUI μέσω της διεπαφής BiomedicalDataAPI, ώστε να του παρέχει τα δεδομένα που πρέπει να απεικονίσει στην οθόνη.



Σχημα 8. Υποσύστημα BiomedicalDataHandler

1.1.9 Υποσύστημα Notification GUI

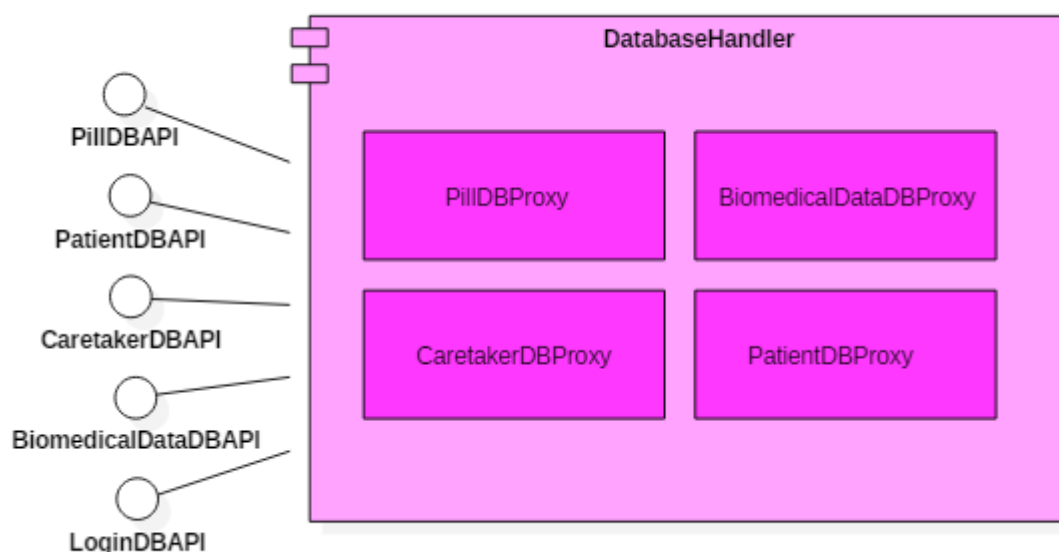
Το υποσύστημα αυτό περιλαμβάνει τις οριακές κλάσεις **CaretakerPillNotificationGUI** και **PatientPillNotificationGUI** οι οποίες αποτελούν τις διεπαφές μέσω των οποίων λαμβάνουν ειδοποιήσεις ο caretaker και ο ασθενής καθώς και την οριακή κλάση **AnomalyDetectionGUI** η οποία μας δείχνει της ανιχνευμένες ανωμαλίες. Επικοινωνεί με το υποσύστημα **LoginHandler**, μέσω του **LoginAPI**, με το **CaretakerHandler** μέσω του **CaretakerAPI** και με το **PatientHandler** με το **PatientAPI** απαιτώντας από αυτό τις πληροφορίες τις οποίες θα εμφανίσει στην γραφική διεπαφή.



Σχημα 9. Υποσύστημα NotificationGUI

1.1.10 Υποσύστημα DatabaseHandler

Όλα τα παραπάνω υποσυστήματα, τα οποία προσφέρουν τις διάφορες υπηρεσίες που απαιτεί το σύστημα μας, απαιτούν επικοινωνία με τη βάση δεδομένων. Πιο συγκεκριμένα, η βάση δεδομένων περιέχει δεδομένα τα οποία σχετίζονται με τον caretaker, τον ασθενή, τα biomedical data του ασθενή καθώς και με τα χάπια που λαμβάνει ο ασθενής. Η πρόσβαση σε όλες αυτές τις πληροφορίες επιτυγχάνεται μέσω των οριακών αντικειμένων PillDBProxy, BiomedicalDataDBProxy, CaretakerDBProxy και PatientDBProxy.

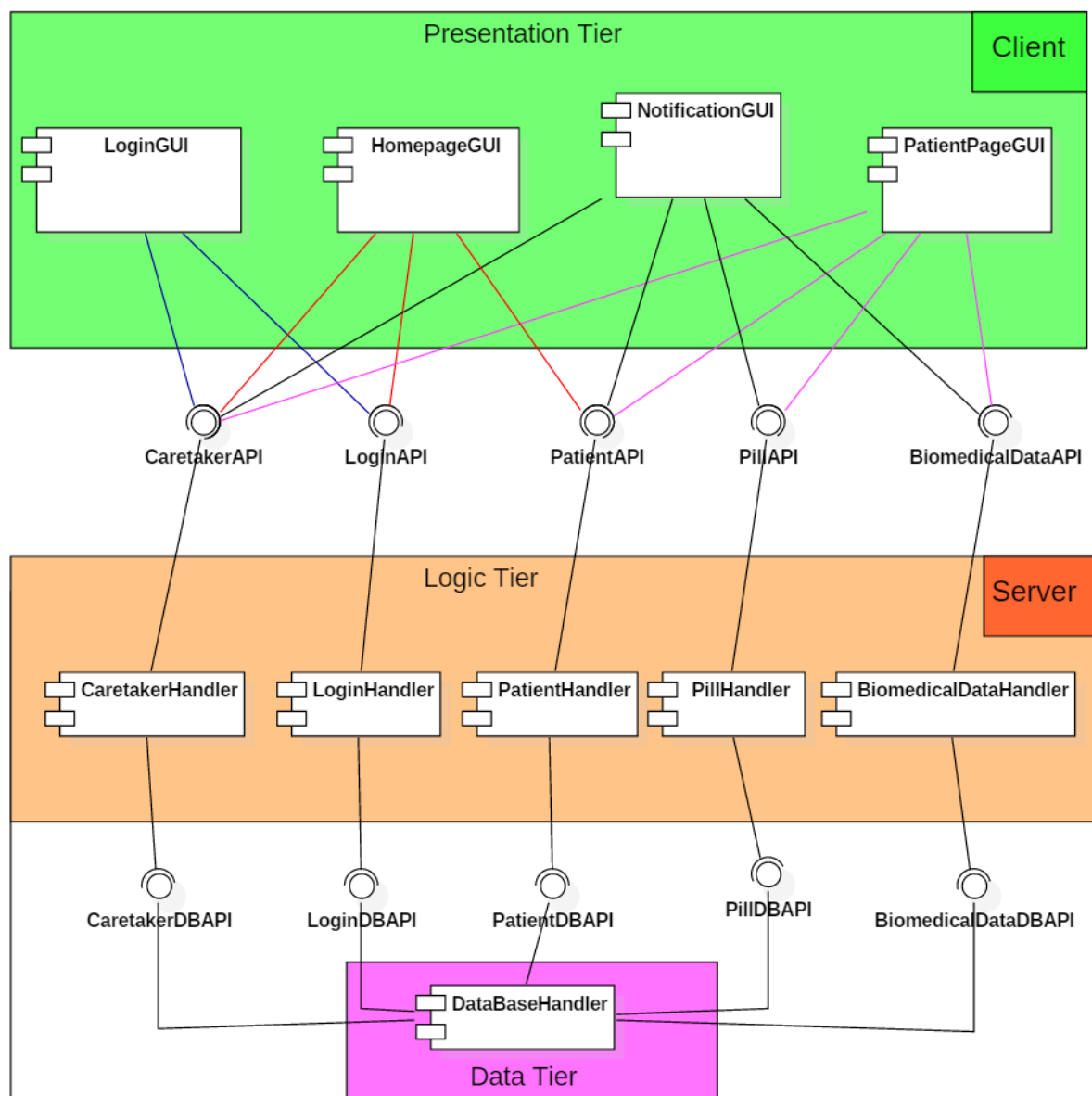


Σχημα 10. Υποσύστημα DatabaseHandler

1.1.11 Διάγραμμα τμημάτων

Το διάγραμμα τμημάτων παρουσιάζει τις εξαρτήσεις μεταξύ των επιμέρους υποσυστημάτων του συστήματος μας, το οποίο έχει υλοποιηθεί με βάση τον συνδυασμό των αρχιτεκτονικών Πελάτη - Διακομιστή (Client-Server) και Τριών Επιπέδων (3-Tier).

Τα διαγράμματα τμημάτων αξιοποιούνται για επιμερισμό τμημάτων του συστήματος σε ομάδες Ανάπτυξης και προς έλεγχο για το αν παρέχεται η απαιτούμενη λειτουργικότητα από το σύστημα.



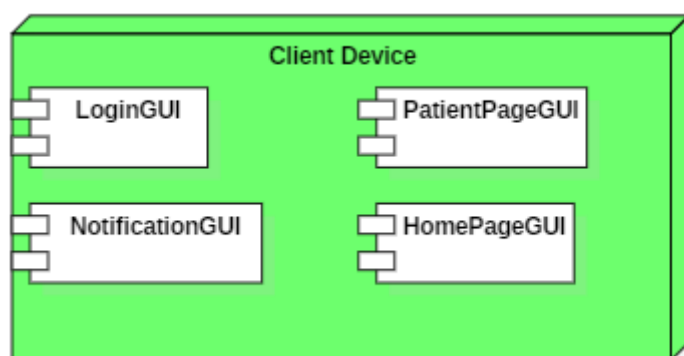
Σχημα 11. Διάγραμμα τμημάτων

1.2 Απεικόνιση Υλικού/Λογισμικού

Ακολουθεί η απεικόνιση υλικού λογισμικού που επιτυγχάνεται με χρήση των κατάλληλων διαγραμμάτων UML. Τα υποσυστήματα απεικονίζονται στο διάγραμμα ανάλογα με το αν αναφέρονται στην συσκευή του caretaker (client) ή στον server. Η συγκεκριμένη απεικόνιση σκοπεύει να κάνει κατανοητή τη σύνδεση των διαφόρων υποσυστημάτων και τη σχέση αυτών με το υλικό του συστήματος.

1.2.1 Client Device

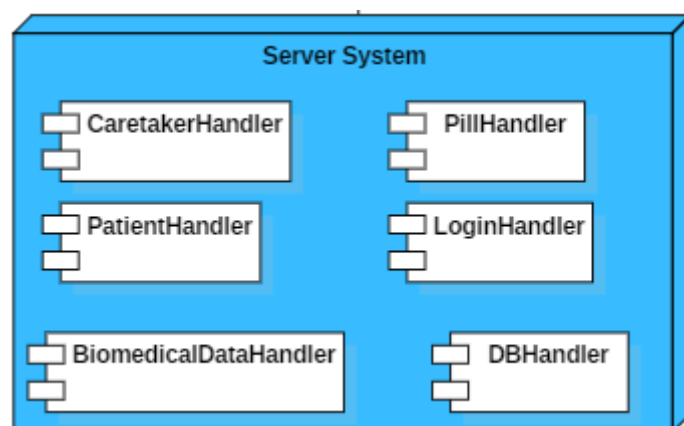
Το Client Device είναι ένας κόμβος του συστήματος και αποτελείται μόνο από τις γραφικές διεπαφές (GUI) που απαιτούνται προκειμένου να μπορεί ο caretaker να χρησιμοποιήσει την εφαρμογή. Το υλικό θα πρέπει να είναι κάποιο smartphone βασισμένο στο Android ή το iOS με σύνδεση στο διαδίκτυο.



Σχημα 12. Client Device

1.2.2 Server System

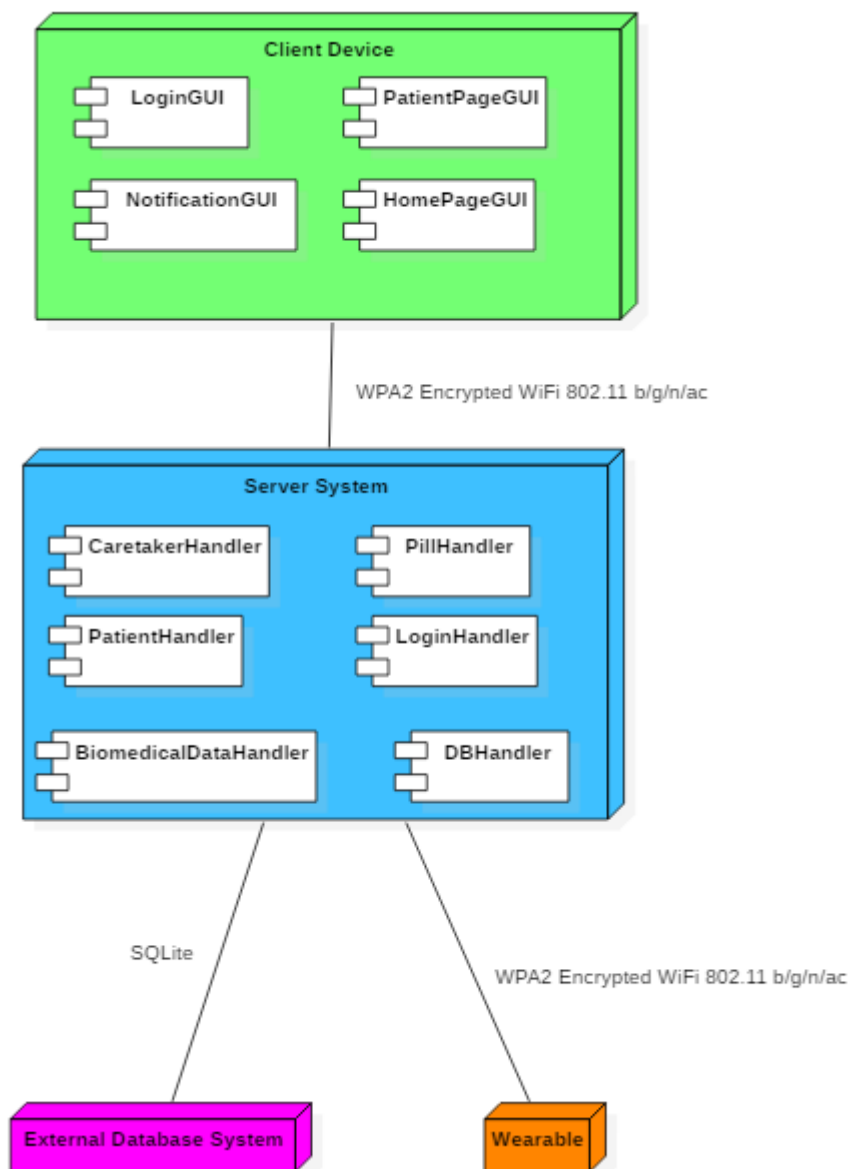
Το Server System είναι ο κυριότερος κόμβος του συστήματος. Είναι υπεύθυνος για την επικοινωνία του συστήματος με τον caretaker μέσω του κόμβου Client Device και με το wearable που φορά ο ασθενής. Επικοινωνεί επίσης με το σύστημα External Database για την αποθήκευση και ανάκτηση πληροφοριών.



Σχημα 13. Server System

1.2.3 Συνολικό διάγραμμα ανάπτυξης

Στο συνολικό διάγραμμα ανάπτυξης παρουσιάζεται ο τρόπος σύνδεσης των κόμβων του συστήματος. Αρχικά, έχουμε τον Client Device που επικοινωνεί με το Server System μέσω ενός ασύρματου τοπικού δικτύου LAN το οποίο χρησιμοποιεί το πρωτόκολλο WiFi 802.11 b/g/n/ac και κρυπτογράφηση WPA2. Η επικοινωνία με την βάση δεδομένων γίνεται με βάση το RDBMS SQLite, ενώ αυτή επικοινωνεί με το wearable του ασθενή μέσω του ασύρματου τοπικού δικτύου LAN που χρησιμοποιείται και για την επικοινωνία Client-Server.



Σχημα 14. Συνολικό Διάγραμμα Ανάπτυξης

Παρακάτω παρουσιάζονται οι ελάχιστες και οι προτεινόμενες απαιτήσεις σε επίπεδο χρήστη και server για το σύστημα μας.

| | | Minimum Requirements | Recommended Requirements |
|---------------|-----------------|-------------------------------|--------------------------|
| Client | CPU | MediaTek MT6570 | Qualcomm Snapdragon 801 |
| | RAM | 1GB | 2GB |
| | Internal Memory | 8GB | 16GB |
| | Android Version | 6.0 | 6.0 or higher |
| | iOS Version | 9 | 9 or higher |
| Server | CPU | Intel Xeon X3210 2.13Ghz | Intel Xeon X5570 2.93 |
| | RAM | 4GB | 8GB |
| | HDD | 750GB | 1TB |
| | OS | CentOS 7, Ubuntu Server 16.04 | Red Hat enterprise Linux |

1.3 Έλεγχος Πρόσβασης και Ασφάλεια

Στην ενότητα αυτή παρουσιάζεται ο πίνακας πρόσβασης, ο οποίος περιγράφει τους διάφορους χρήστες που έχουν πρόσβαση στην πλατφόρμα του Hapio και τις λειτουργίες που αυτοί μπορούν να επιτελέσουν.

Στην πλατφόρμα που αναπτύξαμε, πρόσβαση έχει η κατηγορία χρηστών με την ιδιότητα του caretaker ή του ασθενή. Ο έλεγχος πρόσβασης ξεκινά με την επιβεβαίωση των στοιχείων του λογαριασμού του caretaker. Ο λογαριασμός αυτός θεωρούμε πως του έχει δοθεί από τους δημιουργούς της εφαρμογής, επομένως δεν μας ενδιαφέρει η διαδικασία δημιουργίας λογαριασμού. Τα στοιχεία αυτά, κρατούνται κρυπτογραφημένα στη βάση δεδομένων.

Μετά τη σύνδεση, ο caretaker μπορεί μέσω των διεπαφών της εφαρμογής μας, να αλληλεπιδρά με το σύστημα και να εκτελεί συγκεκριμένες ενέργειες. Οι ενέργειες που μπορεί να επιτελέσει ο caretaker ή ο ασθενής αναφέρονται στον παρακάτω πίνακα με τη μορφή συναρτήσεων που έχουν περιγραφεί ήδη στο προηγούμενο παραδοτέο των απαιτήσεων λογισμικού.

| Actor | Class | Class | Class |
|-----------|------------------------------|-------------------------|------------------------|
| Caretaker | LoginPageGUI | ErrorLoginGUI | HomePageGUI |
| | loginClick() | backClick() | patientManageClick() |
| | | | patientAddClick() |
| | | | logoutClick() |
| Caretaker | PatientProfilePageGUI | PatientEditPillsPageGUI | PatientEditInfoPageGUI |
| | patientEditClick() | backClick() | editCommentClick() |
| | patientDeleteClick() | editPillClick() | editPillsClick() |
| | patientBiomedicalDataClick() | selectPillClick() | backClick() |
| | homepageClick() | | editNameClick() |
| | | | editSurnameClick() |
| | | | editAge() |

| Caretaker | AnomalyDetectionGUI | CaretakerPillNotificationGUI | BiomedicalDataPageGUI |
|-----------|---------------------|------------------------------|-----------------------|
| | saveAnomalyClick() | clickConfirm() | backClick() |
| | clickHomepage() | clickReject() | |
| | | clickHomepage() | |

| Actor | Class |
|---------|----------------------------|
| Ασθενής | PatientPillNotificationGUI |
| | clickConfirm() |
| | clickReject() |

2. Πίνακας ιχνηλασιμότητας εγγράφων Σχεδίασης και Απαιτήσεων Λογισμικού

Η μετάβαση από το έγγραφο απαιτήσεων λογισμικού στο έγγραφο σχεδίασης συστήματος δεν εμπεριέχει μεταβολές.

Επομένως, δεν ορίζουμε κάποιον πίνακα ιχνηλασιμότητας.

3. RESTful API

3.1 Patient API

Το Patient API αφορά την διασύνδεση του συστήματος με τα στοιχεία του Patient και τα resources του. Συγκεκριμένα παίρνουμε και ανανεώνουμε τις πληροφορίες του ασθενή, δημιουργούμε καινούργιο ασθενή, παίρνουμε την ώρα λήψης του χαπιού και επιβεβαιώνουμε την λήψη του χαπιού.

3.1.1 GET Patient Info

| Method | URL | | |
|---|---|-------------|------|
| GET | patient/{patientID} | | |
| Description | | | |
| Gets a patient’s information for a specific patient ID. | | | |
| Parameters | | | |
| Path Parameters | | | |
| Path Parameter | Description | | |
| {patientID} | Refers to the ID of the patient in question. | | |
| Query string parameters | | | |
| Query Parameters | Required/Optional | Description | Type |
| None | | | |
| Sample Request Body | | | |
| Not needed | | | |
| Sample Responses | | | |
| Status | Response | | |
| 200 | { "patient":{ "patientID":"1234", "name": "David Bowie", "comment": "Anisocoria, Liver Issues", } } | | |
| 400 | { "error": "Please specify database version." } | | |
| 404 | { "error": "patientID not found" } | | |
| 500 | { "error": "Something went wrong. Please try again later." } | | |

3.1.2 Update Patient Info

| Method | URL | | |
|--|---|-------------|------|
| PUT | pills/{patientID} | | |
| Description | | | |
| Updates a pill list's information for a specific patient ID. | | | |
| Parameters | | | |
| Path Parameters | | | |
| Path Parameter | Description | | |
| {patientID} | Refers to the ID of the patient in question. | | |
| Query string parameters | | | |
| Query Parameters | Required/Optional | Description | Type |
| None | | | |
| Sample Request Body | | | |
| <pre>{ "patientID": "1234", "pill": { "pillID": "123", "stock": { "current": 2, "total": 10 }, "schedule": [{ "frequency": "Daily", "firstTime": "10:00 AM", "secondTime": "22:00 PM" }] } }</pre> | | | |
| Sample Responses | | | |
| Status | Response | | |
| 200 | Not needed | | |
| 400 | { "error" : "Please specify database version."} | | |
| 404 | { "error" : "patientID not found"} | | |
| 500 | { "error" : "Something went wrong. Please try again later."} | | |

3.1.3 Create New Patient

| Method | URL | | |
|---|--|-------------|------|
| POST | patient/ | | |
| Description | | | |
| Creates a new patient. | | | |
| Parameters | | | |
| Path Parameters | | | |
| Path Parameter | Description | | |
| {patientID} | Refers to the ID of the patient in question. | | |
| Query string parameters | | | |
| Query Parameters | Required/Optional | Description | Type |
| None | | | |
| Sample Request Body | | | |
| <pre>{ "patient":{ "wearableID":"1a2b3c4d", "name": "David Bowie", "comment": "Anisocoria, Liver Issues", } }</pre> | | | |
| Sample Responses | | | |
| Status | Response | | |
| 201 | <pre>{ "patient":{ "patientID":"1234", "name": "David Bowie", "comment": "Anisocoria, Liver Issues", } }</pre> | | |
| 400 | <pre>{"error":"Please specify database version."}</pre> | | |
| 404 | <pre>{"error":"patientID not found"}</pre> | | |
| 500 | <pre>{"error":"Something went wrong. Please try again later."}</pre> | | |

3.1.4 GET Next Pill Time

| Method | URL | | |
|--|--|-------------|------|
| GET | patients/{patientID}/pillCountdown | | |
| Description | | | |
| Gets a patient’s remaining time for next pill. | | | |
| Parameters | | | |
| Path Parameters | | | |
| Path Parameter | Description | | |
| {patientID} | Refers to the ID of the patient in question. | | |
| Query string parameters | | | |
| Query Parameters | Required/Optional | Description | Type |
| None | | | |
| Sample Request Body | | | |
| Not needed | | | |
| Sample Responses | | | |
| Status | Response | | |
| 200 | <pre>{ "countdown":{ "nextPill":{ "pillID":"123", "pillName":"Sorafenib", "pillImage":"pill123Img.jpg", "schedule":[{ "frequency":"Daily", "firstTime":"10:00 AM", "secondTime":"22:00 PM" }], "stock":{ "current":3, "total":10 } }, "currentTime":"09:46:36 AM", "timeLeft":"00:13:24" } }</pre> | | |

| | |
|-----|--|
| | } |
| 400 | {"error": "Please specify database version."} |
| 404 | {"error": "patientID not found"} |
| 500 | {"error": "Something went wrong. Please try again later."} |

3.1.5 Pill Confirmation POST

| Method | URL | | |
|--|----------------------------------|-------------|------|
| POST | patients/{patientID}/pillConfirm | | |
| Description | | | |
| Confirms that a patient took their pill. | | | |
| Parameters | | | |
| Path Parameters | | | |
| Path Parameter | Description | | |
| None | | | |
| Query string parameters | | | |
| Query Parameters | Required/Optional | Description | Type |
| None | | | |
| Sample Request Body | | | |
| <pre>{ "patientID": "1234", "confirmationReport": { "confirmedFrom": "caretaker", "confirmationStatus": "True", "pillConfirmed": { "pillID": "123", "stock": { "current": 3, "total": 10 } } } }</pre> | | | |
| Sample Responses | | | |
| Status | Response | | |
| 201 | { | | |

| | |
|-----|---|
| | <pre>"patientID":"1234", "pill":{ "pillID":"123", "stock":{ "current":2, "total":10 } }</pre> |
| 400 | {"error":"Please specify database version."} |
| 404 | {"error":"patientID not found"} |
| 500 | {"error":"Something went wrong. Please try again later."} |

3.2 Pills API

Το Pills API αφορά την διασύνδεση του συστήματος με τα στοιχεία των Pills και τα resources τους. Συγκεκριμένα παίρνουμε και ανανεώνουμε τις πληροφορίες των χαπιών του ασθενή και προσθέτουμε καινούργιο χάπι για τον ασθενή.

3.2.1 GET Pills Info

| Method | URL | | |
|---|--|-------------|------|
| GET | patients/{patientID}/pills | | |
| Description | | | |
| Gets the pill list for a specific patient ID. | | | |
| Parameters | | | |
| Path Parameters | | | |
| Path Parameter | Description | | |
| {patientID} | Refers to the ID of the patient in question. | | |
| Query string parameters | | | |
| Query Parameters | Required/Optional | Description | Type |
| None | | | |
| Sample Request Body | | | |
| Not needed | | | |
| Sample Responses | | | |
| Status | Response | | |
| 200 | <pre>{ "patientID":"1234", "pills":[{ "pill_1":{ "pillID":"123", "pillName":"Sorafenib", "pillImage":"pill123Img.jpg", "schedule":[{ "frequency":"Daily", "firstTime":"10:00 AM", "secondTime":"22:00 PM" }], }, "stock":{ "current":3,</pre> | | |

| | |
|-----|---|
| | <pre> "total":10 } }, "pill_2":{ "pillID":"124", "pillName":"Legolas", "pillImage":"pill124Img.jpg", "schedule":[{ "frequency":"Every Other Day", "firstTime":"12:00 AM", },], "stock":{ "current":5, "total":15 } }] } </pre> |
| 400 | {"error":"Please specify database version."} |
| 404 | {"error":"patientID not found"} |
| 500 | {"error":"Something went wrong. Please try again later."} |

3.2.2 Update Pills Info

| Method | URL | | |
|---|--|-------------|------|
| PUT | patients/{patientID}/pills/{pillID} | | |
| Description | | | |
| Updates a specific patient's pill using the patient's and the pill's IDs. | | | |
| Parameters | | | |
| Path Parameters | | | |
| Path Parameter | Description | | |
| {patientID} | Refers to the ID of the patient in question. | | |
| Query string parameters | | | |
| Query Parameters | Required/Optional | Description | Type |
| None | | | |
| Sample Request Body | | | |

```
{
  "patientID":"1234",
  "pill":{
    "pillID":"123",
    "stock":{
      "current":2,
      "total":10
    },
    "schedule":[
      {
        "frequency":"Daily",
        "firstTime":"10:00 AM",
        "secondTime":"22:00 PM"
      }
    ],
  }
}
```

Sample Responses

| Status | Response |
|--------|---|
| 200 | OK |
| 400 | {"error":"Please specify database version."} |
| 404 | {"error":"patientID not found"} |
| 500 | {"error":"Something went wrong. Please try again later."} |

3.2.3 Add New Pill

| Method | URL | | |
|----------------------------------|--|-------------|------|
| POST | patient/{patientID}/pills | | |
| Description | | | |
| Adds a new pill for the patient. | | | |
| Parameters | | | |
| Path Parameters | | | |
| Path Parameter | Description | | |
| {patientID} | Refers to the ID of the patient in question. | | |
| Query string parameters | | | |
| Query Parameters | Required/Optional | Description | Type |
| None | | | |

| Sample Request Body | |
|---|---|
| <pre>{ "patientID":"1234", "pillID":"123" }</pre> | |
| Sample Responses | |
| Status | Response |
| 201 | <pre>{ "patientID":"1234", "pill_1":{ "pillID":"123", "pillName":"Sorafenib", "pillImage":"pill123Img.jpg", "stock":{ "current":10, "total":10 } }, }</pre> |
| 400 | {"error":"Please specify database version."} |
| 404 | {"error":"patientID not found"} |
| 500 | {"error":"Something went wrong. Please try again later."} |

3.3 Biomedical Data API

Το Biomedical Data API αφορά την διασύνδεση του συστήματος με τα στοιχεία των Biomedical Data. Συγκεκριμένα παίρνουμε τα Biomedical Data ενός ασθενή.

3.3.1 GET Biomedical Data

| Method | URL | | |
|---|---|-------------|------|
| GET | biomedicalData/{patientID} | | |
| Description | | | |
| Gets the biomedical data for a specific patient ID. | | | |
| Parameters | | | |
| Path Parameters | | | |
| Path Parameter | Description | | |
| {patientID} | Refers to the ID of the patient in question. | | |
| Query string parameters | | | |
| Query Parameters | Required/Optional | Description | Type |
| None | | | |
| Sample Request Body | | | |
| Not needed | | | |
| Sample Responses | | | |
| Status | Response | | |
| 200 | { "biomedicalData":{ "bloodPressure":13.6, "cardioGram":[70,80,75,80,78], "bloodGlucose":23.5, "bloodOxygen":56, "skinConductivity":89, } } | | |
| 400 | { "error": "Please specify database version." } | | |
| 404 | { "error": "patientID not found" } | | |
| 500 | { "error": "Something went wrong. Please try again later." } | | |

4. Ανάλυση SWOT

Η ανάλυση SWOT είναι ένα εργαλείο στρατηγικού σχεδιασμού το οποίο χρησιμοποιείται για την ανάλυση του εσωτερικού και εξωτερικού περιβάλλοντος μίας επιχείρησης, όταν η επιχείρηση πρέπει να λάβει μία απόφαση σε σχέση με τους στόχους που έχει θέσει ή με σκοπό την επίτευξή τους.

Κατά την ανάλυση SWOT μελετώνται τα δυνατά (Strengths) και αδύνατα (Weaknesses) σημεία μίας επιχείρησης, οργανισμού ή και περιοχής, καθώς και οι ευκαιρίες (Opportunities) και οι απειλές (Threats) που υπάρχουν.

| | |
|--|--|
| STRENGTHS <ul style="list-style-type: none">✓ Είμαστε GDPR compliant✓ Εφαρμογή πολύ φιλική για τους χρήστες✓ Η υλοποίηση μας επιτρέπει επέκταση με άλλες υπηρεσίες (π.χ. φαρμακοποιεί)✓ Η λύση μας είναι καινοτόμα και αντιμετωπίζει μεγάλο υπαρκτό πρόβλημα✓ Wearable σχεδιασμένο να μην χρειάζεται σχεδόν καθόλου user input από patient | WEAKNESSES <ul style="list-style-type: none">✗ Περιορισμένες εξατομικευμένες υπηρεσίες προς το παρών✗ Υψηλό αρχικό κόστος για τους πελάτες καθώς χρειάζονται wearable |
| OPPORTUNITIES <ul style="list-style-type: none">✓ Είμαστε σε τομέα που έχει πολύ άνθηση τα τελευταία χρόνια και εξελίσσονται διαρκώς αλγόριθμοι/τεχνολογίες που μας βοηθάνε✓ Υπάρχει ανταγωνισμός αλλά περιορισμένων δυνατοτήτων✓ Οι νέες γενιές ηλικιωμένων είναι όλο και περισσότερο τεχνολογικά καταρτισμένοι✓ Το target audience μας είναι πολύ μεγάλο και ανανεώνεται διαρκώς | THREATS <ul style="list-style-type: none">✗ Ο τομέας της βιοϊατρικής μηχανικής είναι πολύ δύσκολος στην εισαγωγή καθώς υπάρχουν πολύ αυστηροί νόμοι και άρα νομοθετικά εμπόδια✗ Στην αρχή χρειάζεται μεγάλη αρχική επένδυση για την κατασκευή wearable✗ Τεχνοφοβία παλαιών γενεών ηλικιωμένων✗ Πρέπει να κερδίσουμε την εμπιστοσύνη του κόσμου για να χρησιμοποιεί το προϊόν μας |

Σχημα 15. SWOT Ανάλυση

5. Χρονοδιάγραμμα και Κοστολόγηση Έργου

5.1 Έκταση έργου και κοστολόγηση

| Τύπος κλάσης | <u>UI</u> | <u>Controllers</u> | <u>Entities</u> | <u>Proxies / APIs</u> |
|--|-----------|--------------------|-----------------|-----------------------|
| Αριθμός κλάσεων | 11 | 7 | 5 | 6 |
| Εκτίμηση γραμμών κώδικα για κάθε κλάση | 300 | 600 | 400 | 600 |
| Συνολικές γραμμές κώδικα για κάθε τύπο | 3.300 | 4.200 | 2.000 | 3.600 |

Συνολικό αποτέλεσμα γραμμών κώδικα: 13.100 γραμμές κώδικα → 13,1 ανθρωπομήνες → 26.200 Ευρώ

Το παραπάνω κόστος αφορά μόνο τη συγγραφή του κώδικα από τους μηχανικούς λογισμικού. Πρέπει να ληφθεί υπόψιν και το κόστος ολοκλήρωσης.

Το σύστημά μας διαθέτει 29 κλάσεις, επομένως για την φάση της ολοκλήρωσης θα χρειαστούν επιπλέον $0,3 * 29 = 8,7$ ανθρωπομήνες.

Συνολικό κόστος ολοκλήρωσης: 8,7 ανθρωπομήνες → 17.400 Ευρώ

Οπότε, συνολικά για το σύστημά μας θα έχουμε:

| | <u>Ανθρωπομήνες</u> | <u>Ευρώ</u> |
|---------------------------|---------------------|-------------|
| Κόστος συγγραφής κώδικα | 13,1 | 26.200 |
| Κόστος ολοκλήρωσης | 8,7 | 17.400 |
| Συνολικό κόστος ανάπτυξης | 21,8 | 43.600 |

Στο παραπάνω κόστος συνολικό κόστος ανάπτυξης, θα προστεθούν τα εξής κόστη:

Κόστος συντήρησης: 65.400 Ευρώ, δηλαδή 1,5 φορές το κόστος ανάπτυξης

Το πλάνο μας εμπεριέχει και ένα κέρδος για την εταιρία μας, της τάξεως του 15% επί του κόστους ανάπτυξης.

Κέρδος: 6.540 Ευρώ

Επομένως, η τελική προσφορά μας προς τον πελάτη θα είναι:

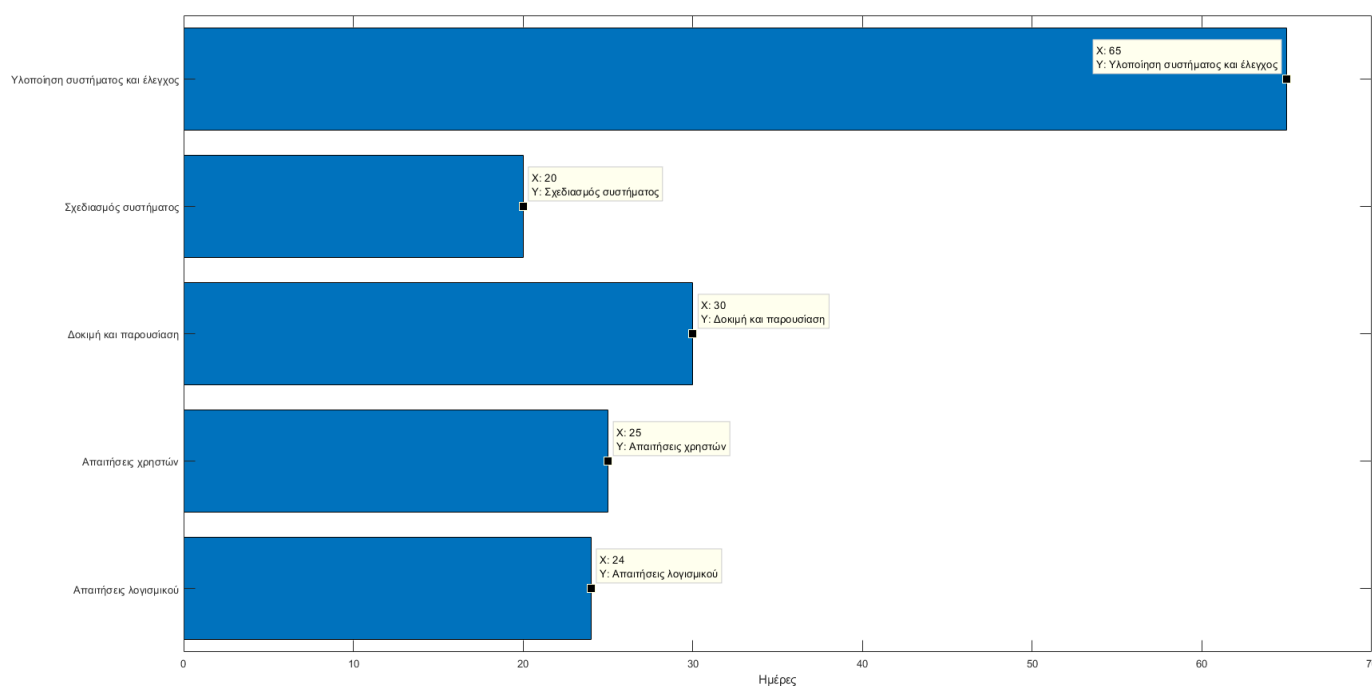
115.540 Ευρώ

5.2 Χρονοδιάγραμμα

Η ομάδα μας αποτελείται από 4 άτομα και το έργο μας θα χρειαστεί 21,8 ανθρωπομήνες για την υλοποίηση του.

Επομένως, με 4 άτομα ομάδα θα κρατήσει 5,45 μήνες → 164 ημέρες

Πιο αναλυτικά, έχουμε το εξής γράφημα:

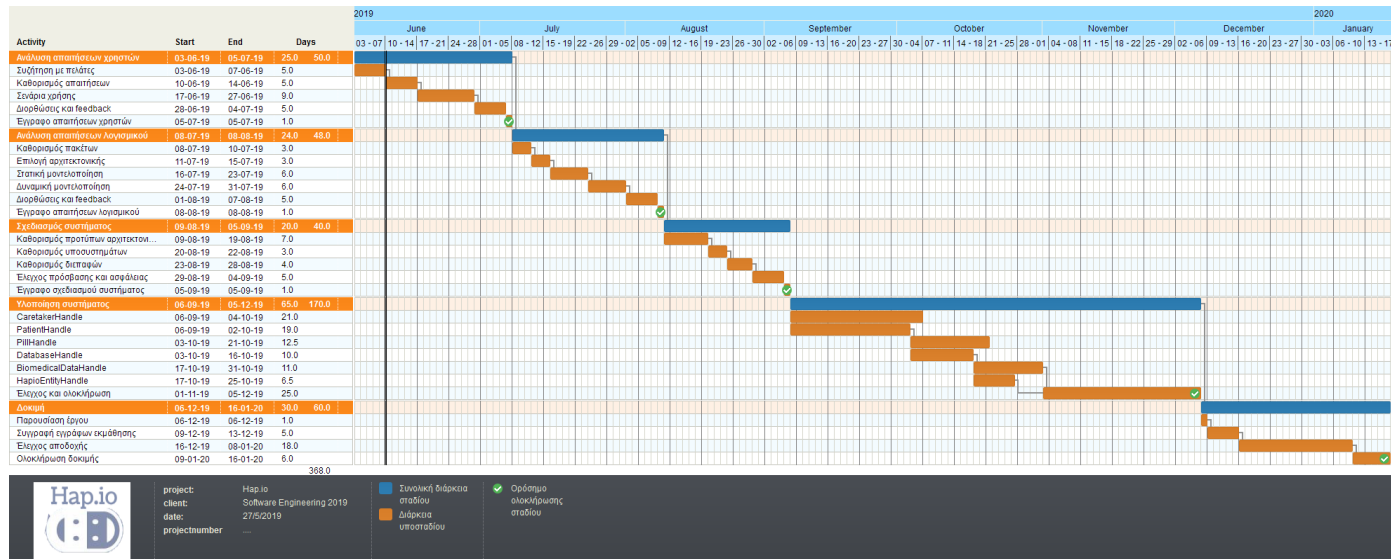


Σχημα 16. Χρονοδιάγραμμα

Δηλαδή, οι 164 ημέρες θα καταμεριστούν με τον τρόπο που φαίνεται στο παραπάνω γράφημα.

5.3 Διάγραμμα Gantt

Το διάγραμμα Gantt φαίνεται παρακάτω:



Σχημα 17. Διάγραμμα Gantt

Παράρτημα Ι – Ανοιχτά Θέματα

- Το διάγραμμα ανάπτυξης μπορεί να αλλάξει ως προς τη συνδεσιμότητα μεταξύ client και server, καθώς και μεταξύ server και wearable, με την πάροδο του χρόνου καθώς αναπτύσσονται νέα πρωτόκολλα επικοινωνίας.
- Τα minimum και recommended requirements ενδέχεται να αλλάξουν στο μέλλον, καθώς προστίθενται νέες λειτουργίες στην εφαρμογή ή αφαιρούνται λειτουργίες που δεν είναι τόσο χρήσιμες για τους πελάτες.
- Δημιουργία server που θα λειτουργεί ως backup στην βάση δεδομένων μας, για να προστατέψουμε το σύστημα μας από περιπτώσεις απώλειας και βλάβης.