Hap.io

Happy to remind your χάπι

Απαιτήσεις Λογισμικού

(Προδιαγραφές Λογισμικού)

Del.1.2

Version 0.5
(draft)

Μηναδάκης Μιχαήλ minadakm@ece.auth.gr

Παπαγγέλης Άρης Ελευθέριος aris.papagelis@gmail.com

Παπαδάμ Στέφανος stefanospapadam@gmail.com

Πετρίδης Περικλής Σάββας periclespetrides@gmail.com

7/5/2019

Ιστορικό Αλλαγών

Όνομα	Ημερομηνία	Αλλαγή	Έκδοση
Α. Συμεωνίδης	17/05/2007	Δημιουργία εγγράφου. Προσαρμογή των προτύπων του K. E. Wiegers[1]♣ και του M. Smialek's.	0.1
Α. Συμεωνίδης	29/3/2014	Προσαρμογή του εγγράφου.	0.1.3
hap.io	3/5/2019	Συγγραφή των πακέτων υψηλής προτεραιότητας.	0.2
hap.io	4/5/2019	Συγγραφή των πακέτων μέσης προτεραιότητας	0.3
hap.io	6/5/2019	Συγγραφή των πακέτων χαμηλής προτεραιότητας και των μή λειτουργικών απαιτήσεων.	0.4
hap.io	7/5/2019	Συγγραφή των προτύπων σχεδίασης και τελική μορφοποίηση εγγράφου.	0.5

Μέλη της Ομάδας Ανάπτυξης

Όνομα	ΟΑ	Email
Α. 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

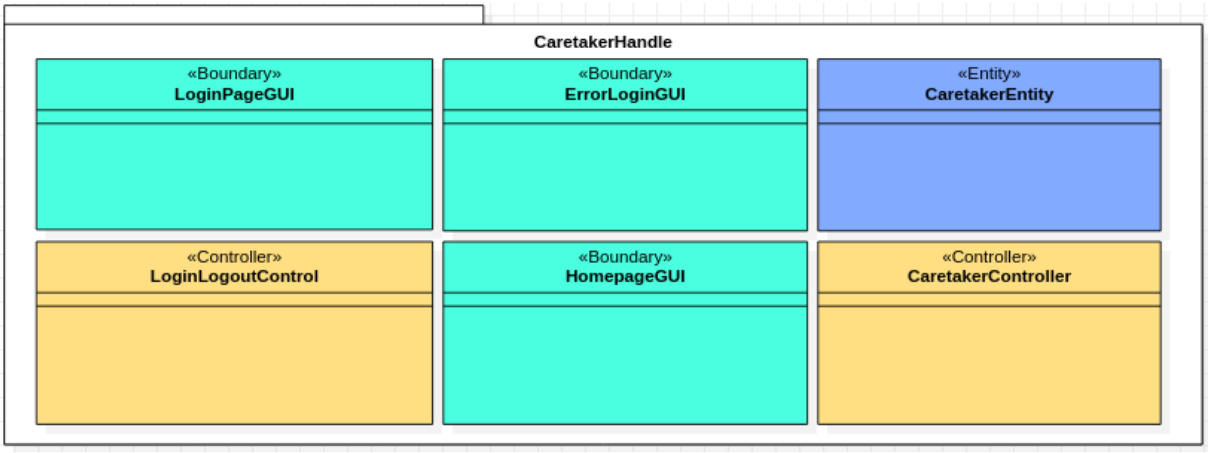
Λίστα Σχημάτων

1 Στατική Μοντελοποίηση

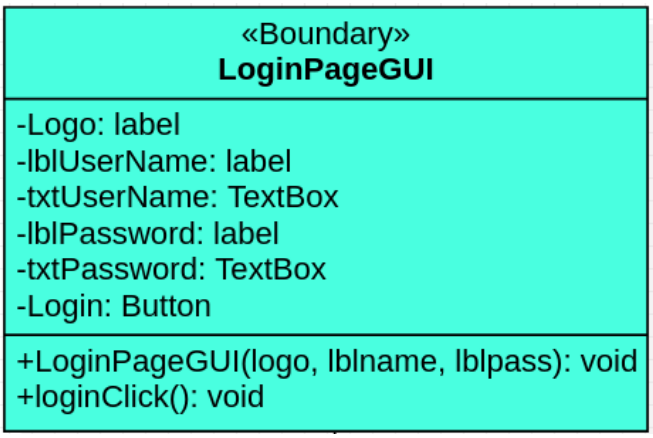
1.1 Πακέτα λεξιλογίου σεναρίων υψηλής προτεραιότητας

1.1.1 Πακέτο CaretakerHandle

Το πακέτο αυτό αναφέρεται στον λογαριασμό του caretaker καθώς και τις οθόνες διεπαφής οι οποίες αφορούν την διαχείριση του λογαριασμού.



Boundary LoginPageGUI



Η κλάση αυτή αναφέρεται στο γραφικό παράθυρο διεπαφής το οποίο θα εμφανίζει στην πλατφόρμα την σελίδα εισαγωγής στον λογαριασμό.

Χαρακτηριστικά της κλάσης:

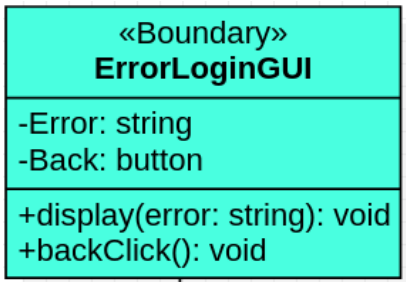
- Logo: Το logo της πλατφόρμας.
- lblUserName: Επιγραφή με το Username του caretaker
- txtUserName: Κουτί εισαγωγής του Username του caretaker
- lblPassword: Επιγραφή με τον κωδικό του caretaker

- txtPassword: Κουτί εισαγωγής του κωδικού του caretaker
- Login: Κουμπί με το οποίο ο caretaker ελέγχει αν εισήγαγε σωστά στοιχεία και εισέρχεται στον λογαριασμό του

Μέθοδοι της κλάσης:

- LoginaPageGUI(logo, lblname, lblpass): Constructor για την κλάση LoginPageGUI.
- loginClick(): Μέθοδος που καλείται όταν ο caretaker θέλει να εισαχθεί στο λογαριασμό του.

Boundary ErrorLoginGUI



Η κλάση αυτή αναφέρεται στο γραφικό παράθυρο διεπαφής το οποίο θα εμφανίζεται στην πλατφόρμα όταν ο caretaker εισάγει λανθασμένα στοιχεία Login.

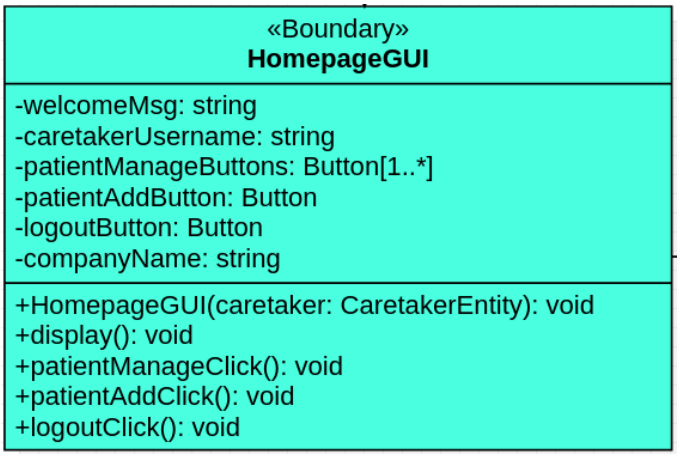
Χαρακτηριστικά της κλάσης:

- Error: Μήνυμα λάθους που εμφανίζεται στην οθόνη.
- Back: Κουμπί με το οποίο ο caretaker επιστρέφει στην σελίδα του Login.

Μέθοδοι της κλάσης:

- display(error:string):Μέθοδος η οποία εμφανίζει την οθόνη διεπαφής με μήνυμα σφάλματος σε περίπτωση που ο caretaker εισήγαγε εσφαλμένα στοιχεία κατά τη σύνδεση του.
- backClick(): Μέθοδος που καλείται όταν ο caretaker θέλει να επιστρέψει στην προηγούμενη οθόνη.

Boundary HomepageGUI



Η κλάση αυτή αναφέρεται στο γραφικό παράθυρο διεπαφής το οποίο θα εμφανίζει στην πλατφόρμα την αρχική οθόνη του λογαριασμού του caretaker.

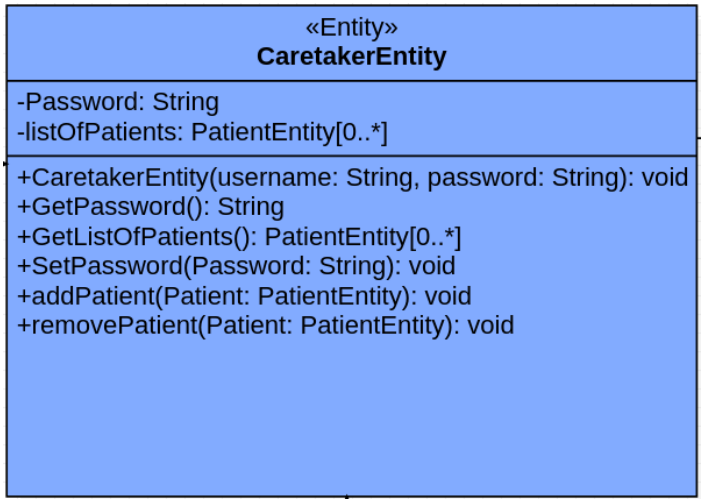
Χαρακτηριστικά της κλάσης:

- welcomeMsg: Μήνυμα καλωσορίσματος στην αρχική οθόνη ως μεταβλητή τύπου string.
- caretakerUsername: Το username του caretaker ως μεταβλητή τύπου string.
- patientManageButtons: Κουμπιά με τα οποία ο caretaker επιλέγει τον ασθενή τον οποίο θέλει να διαχειριστεί.
- patientAddButton: Κουμπί με το οποίο ο caretaker εισάγει καινούργιο ασθενή.
- logoutButton: Κουμπί με το οποίο ο caretaker κάνει εξαγωγή από τον λογαριασμό του.
- companyName: Το όνομα της εταιρείας ως μεταβλητή τύπου string.

Μέθοδοι της κλάσης:

- HomepageGUI(caretaker: CaretakerEntity): Μέθοδος δόμησης της κλάσης HomepageGUI.
- display(): Μέθοδος η οποία εμφανίζει την οθόνη με την αρχική σελίδα.
- patientManageClick(): Μέθοδος η οποία καλείται όταν ο caretaker επιλέγει να επεξεργαστεί τα στοιχεία του ασθενή.
- patientAddClick(): Μέθοδος η οποία καλείται όταν ο caretaker επιθυμεί να προσθέσει κάποιον ασθενή στη λίστα του.
- logoutClick():Μέθοδος η οποία καλείται όταν ο caretaker επιθυμεί να αποσυνδεθεί από την πλατφόρμα.

Entity CaretakerEntity



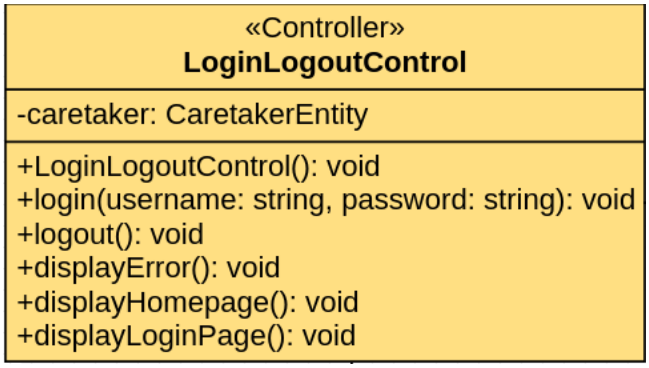
Χαρακτηριστικά της κλάσης:

- Password: μεταβλητή τύπου String που αφορά τον κωδικό που πρέπει να εισάγει ο caretaker για να εισαχθεί στην πλατφόρμα.
- listOfPatients: μεταβλητή τύπου PatientEntity που περιέχει τη λίστα με τους ασθενείς που διαχειρίζεται ο caretaker.

Μέθοδοι της κλάσης:

- CaretakerEntity(username: String, password: String): Μέθοδος δόμησης της κλάσης CaretakerEntity.
- GetPassword(): Μέθοδος που επιστρέφει τον κωδικό που χρησιμοποιεί ο caretaker ως string μεταβλητή.
- GetListOfPatients():Μέθοδος η οποία επιστρέφει τη λίστα ασθενών που διαχειρίζεται ο caretaker σαν μεταβλητή τύπου PatientEntity[0..*].
- SetPassword(Password: String): Μέθοδος που θέτει τον κωδικό του caretaker.
- addPatient(Patient: PatientEntity): Μέθοδος η οποία έχει όρισμα ένα αντικείμενο τύπου PatientEntity το οποίο είναι ο ασθενής που μπορεί να προστεθεί στη λίστα ασθενών που διαχειρίζεται ο caretaker. Η μέθοδος δηλαδή αυτή ανανεώνει τη λίστα ασθενών του caretaker.
- removePatient(Patient: PatientEntity): Μέθοδος η οποία αφαιρεί τον ασθενή που δόθηκε σαν όρισμα (Patient) από τη λίστα ασθενών του caretaker.

Controller LoginLogoutControl



Χαρακτηριστικά της κλάσης:

- caretaker: Ο χρήστης ο οποίος διαχειρίζεται την πλατφόρμα.

Μέθοδοι της κλάσης:

- LoginLogoutControl(): Μέθοδος δόμησης της κλάσης LoginLogoutControl.
- login(username: string, password: string): Μέθοδος η οποία έχει ως ορίσματα το όνομα χρήστη (username) και τον κωδικό (password) την οποία χρησιμοποιεί ο caretaker για να συνδεθεί στην πλατφόρμα.
- logout(): Μέθοδος η οποία χρησιμοποιείται από τον caretaker για να αποσυνδεθεί από την πλατφόρμα.
- displayError(): Μέθοδος η οποία καλεί την συνάρτηση display() της ErrorLoginGUI η οποία εμφανίζει μήνυμα λάθους στην οθόνη σε περίπτωση αποτυχημένης σύνδεσης.
- displayHomepage(): Μέθοδος η οποία καλεί την συνάρτηση display() της HomePageGUI η οποία εμφανίζει την αρχική σελίδα στην οθόνη.
- displayLoginPage(): Μέθοδος η οποία καλεί την συνάρτηση display() της LoginPageGUI η οποία εμφανίζει την οθόνη σύνδεσης στον caretaker.

Controller CaretakerController

«Controller» **CaretakerController**
-caretaker: CaretakerEntity -selectedPatient: PatientEntity
+displayPatientProfilePage(Patient: PatientEntity): void +displayPatientEditInfoPage(Patient: PatientEntity): void +displayPatientEditPillsPage(Patient: PatientEntity): void +displayHomepage(): void +displayBiomedicalData(patient: PatientEntity): void +displayNotification(notification: NotificationEntity): void +displayPillNotification(Patient: PatientEntity, Pill: PillEntity, confirmPillFlag: boolean): void +displayAnomalyNotification(Patient: PatientEntity, BioData: BiomedicalDataEntity): void +addPatient(wearableID: integer): void +managePatient(selectedPatient: PatientEntity): void +confirmPill(confirmFlag: boolean): void

Χαρακτηριστικά της κλάσης:

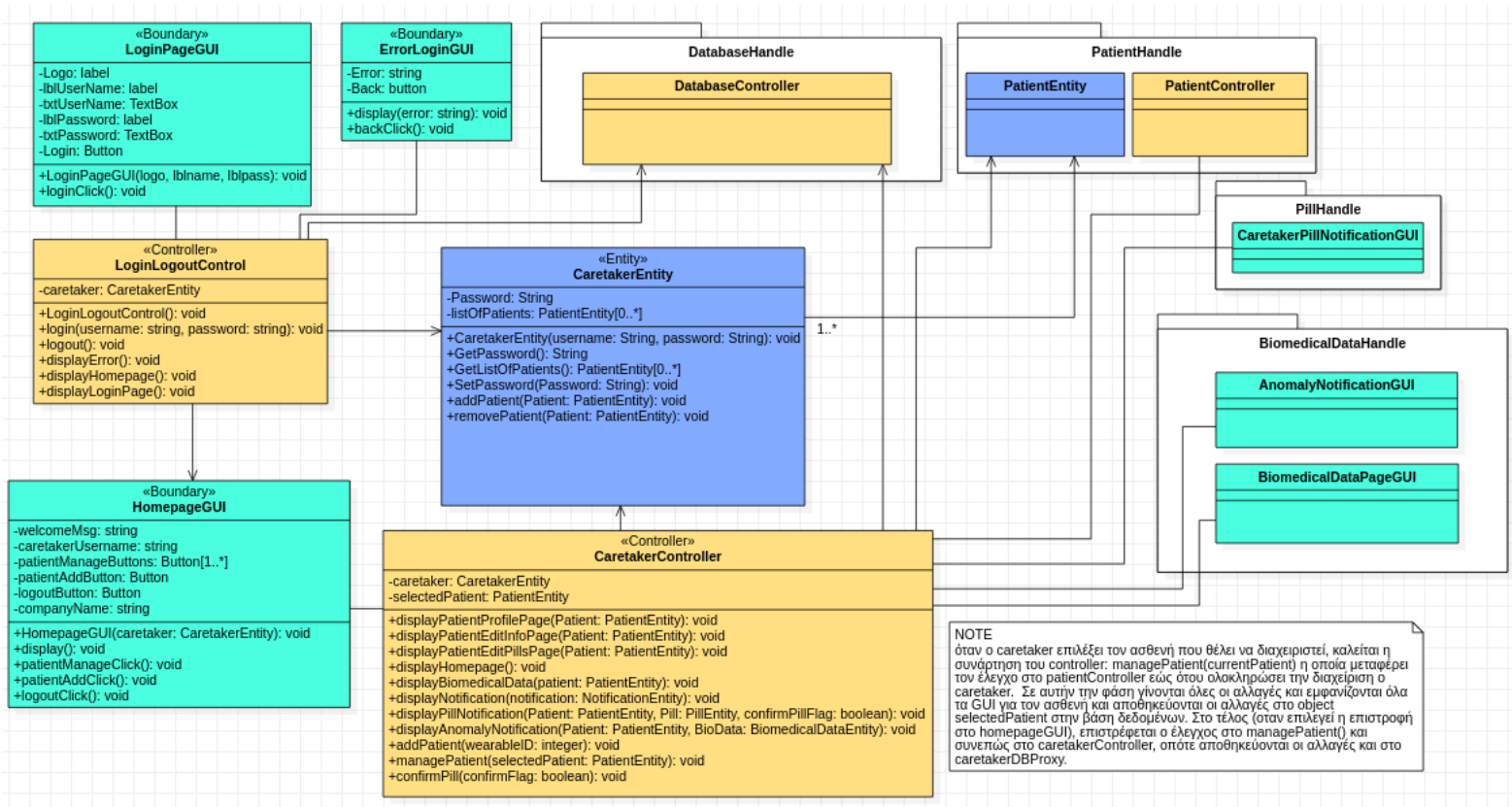
- caretaker: Ο χρήστης ο οποίος διαχειρίζεται την πλατφόρμα .
- selectedPatient: Ο ασθενής ο οποίος επιλέγεται από τον caretaker για επεξεργασία ή προβολή των στοιχείων του.

Μέθοδοι της κλάσης:

- displayPatientProfilePage(Patient: PatientEntity): Μέθοδος η οποία καλεί μέσω του PatientController την μέθοδο display() της PatientProfilePageGUI.
- displayPatientEditInfoPage(Patient: PatientEntity): Μέθοδος η οποία καλεί μέσω του PatientController την μέθοδο display() της PatientEditInfoPageGUI.
- displayPatientEditPillsPage(Patient: PatientEntity): Μέθοδος η οποία καλεί μέσω του PatientController την μέθοδο display() της PatientEditPillsPageGUI.
- displayHomepage(): Μέθοδος η οποία καλεί την συνάρτηση display() της HomePageGUI η οποία εμφανίζει την αρχική σελίδα στην οθόνη.
- displayBiomedicalData(patient: PatientEntity): Μέθοδος η οποία καλεί την μέθοδο display() της κλάσης BiomedicalDataPageGUI.
- addPatient(wearableID: String): Μέθοδος η οποία προσθέτει έναν ασθενή στη λίστα ασθενών ανάλογα με το wearableID που της δίνεται σαν όρισμα.
- managePatient(selectedPatient: PatientEntity): Μέθοδος η οποία καλείται όταν ο caretaker επιθυμεί να διαχειριστεί τα στοιχεία του ασθενή που έχει επιλεχθεί μέσω του ορίσματος selectedPatient.
- confirmPill(confirmFlag: boolean): Μέθοδος η οποία καλείται όταν ο caretaker επιθυμεί να επιβεβαιώσει τη λήψη του φαρμάκου από τον ασθενή.

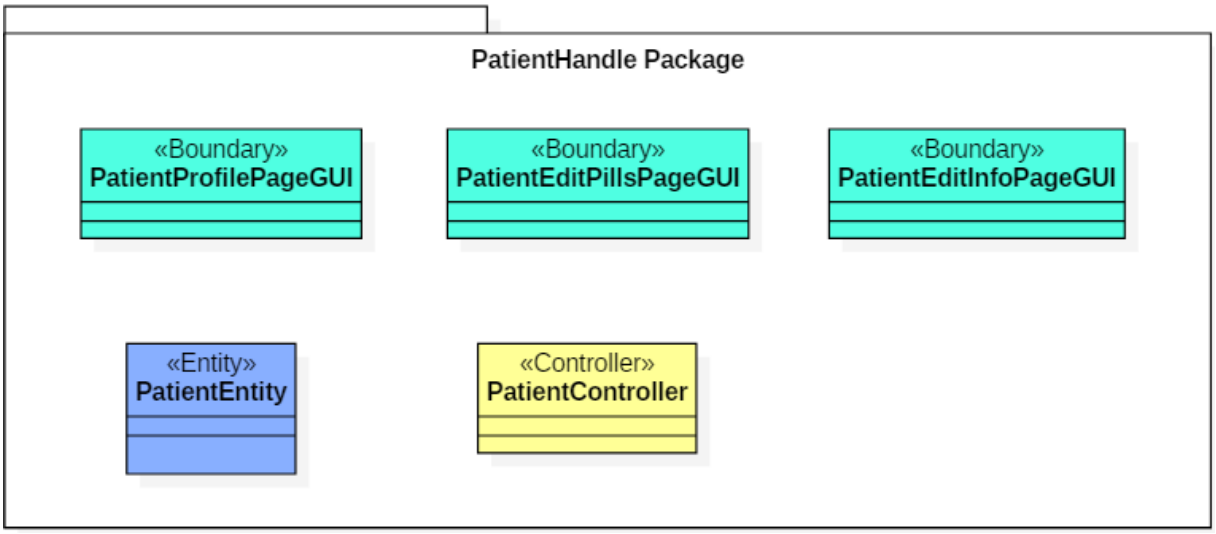
Διάγραμμα Κλάσεων

Σημείωση: Η δεικτοδότηση σε βέλη για ένα προς ένα συσχετίσεις μεταξύ κλάσεων παραλείπεται.

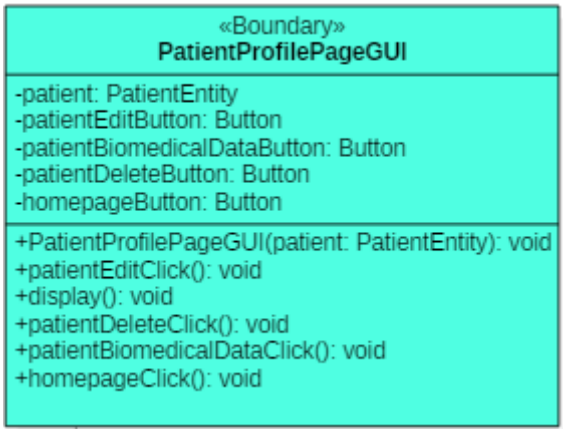


1.1.2 Πακέτο PatientHandle

Το πακέτο αυτό, αναφέρεται στον ασθενή που φοράει και χειρίζεται το wearable και στις οθόνες διεπαφής που βλέπει ο caretaker σχετικά με τη διαχείριση του.



Boundary PatientProfilePageGUI



Η κλάση αυτή αναφέρεται στο γραφικό παράθυρο διεπαφής το οποίο θα εμφανίζει στην πλατφόρμα το προφίλ του ασθενή.

Χαρακτηριστικά της κλάσης:

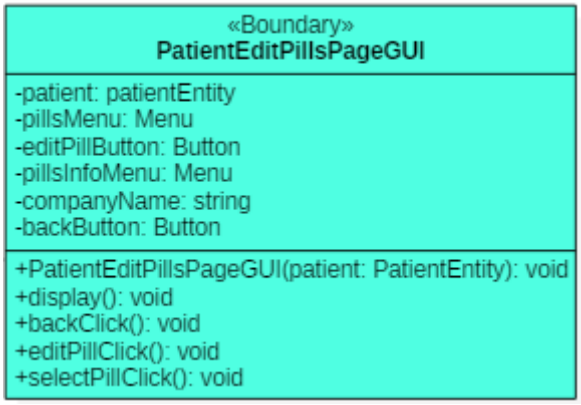
- patient: Ο ασθενής του οποίου το προφίλ προβάλλεται.
- PatientEditButton: Κουμπί με το οποίο ο caretaker επιλέγει να επεξεργαστεί το προφίλ του ασθενή.
- patientBiomedicalDataButton:Κουμπί με το οποίο ο caretaker επιλέγει να προβάλλει τα biomedical data του ασθενή.

- PatientDeleteButton: Κουμπί με το οποίο ο caretaker επιλέγει να διαγράψει το προφίλ του συγκεκριμένου ασθενή από τη λίστα ασθενών.
- homepageButton: Κουμπί με το οποίο ο caretaker επιλέγει να επιστρέψει στην αρχική σελίδα της εφαρμογής.

Μέθοδοι της κλάσης:

- PatientProfilePageGUI(patient:PatientEntity):Μέθοδος δόμησης της κλάσης PatientProfilePageGUI.
- PatientEditClick(): Μέθοδος η οποία καλείται όταν ο caretaker θέλει να επεξεργαστεί το προφίλ του ασθενή.
- display(): Μέθοδος η οποία καλείται για να εμφανίζει την οθόνη διεπαφής που περιέχει το προφίλ του ασθενή.
- patientDeleteClick(): Μέθοδος η οποία καλείται όταν ο caretaker επιθυμεί να διαγράψει κάποιο προφίλ ασθενή από τη λίστα ασθενών.
- patientBiomedicalDataClick(): Μέθοδος η οποία καλείται όταν ο caretaker επιθυμεί να προβάλλει τα biomedical data του ασθενή που έχει επιλεγεί.
- homepageClick(): Μέθοδος η οποία καλείται όταν ο caretaker επιθυμεί να επιστρέψει στην αρχική σελίδα της εφαρμογής.

Boundary PatientEditPillsGUI



Η κλάση αυτή αναφέρεται στο γραφικό παράθυρο διεπαφής το οποίο θα εμφανίζει στην πλατφόρμα τη σελίδα επεξεργασίας των χαπιών του ασθενή.

Χαρακτηριστικά της κλάσης:

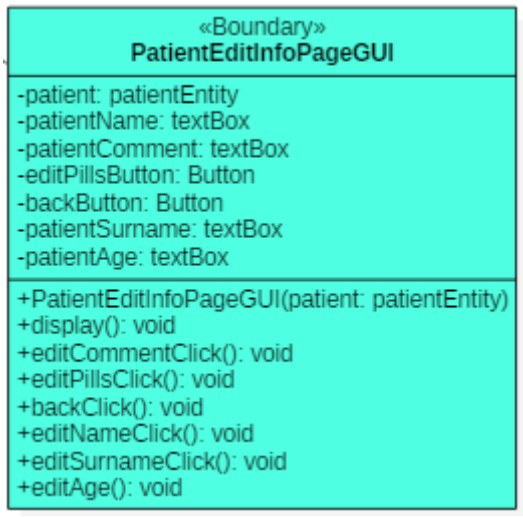
- patient: Ο ασθενής του οποίου η λίστα φαρμάκων βρίσκεται σε επεξεργασία.
- pillsMenu: Μενού το οποίο θα προβάλλει τη λίστα φαρμάκων.
- editPillButton: Κουμπί με το οποίο ο caretaker θα μπορεί να επεξεργαστεί το χάπι που επέλεξε.
- pillsInfoMenu:Μενού το οποίο θα προβάλλει πληροφορίες για το χάπι που επιλέχθηκε.

- backButton: Κουμπί με το οποίο ο caretaker επιστρέφει στην προηγούμενη σελίδα.

Μέθοδοι της κλάσης:

- PatientEditPillsPageGUI(patient:PatientEntity):Μέθοδος δόμησης της κλάσης PatientEditPillsPageGUI.
- display(): Μέθοδος η οποία καλείται για να εμφανίσει τη σελίδα επεξεργασίας των φαρμάκων του ασθενή.
- backClick(): Μέθοδος η οποία καλείται όταν ο caretaker επιθυμεί να επιστρέψει στην προηγούμενη σελίδα από την υπάρχουσα.
- editPillClick(): Μέθοδος η οποία καλείται όταν ο caretaker επιθυμεί να επεξεργαστεί το χάπι που έχει επιλέξει.
- selectPillClick(): Μέθοδος η οποία καλείται όταν ο caretaker επιθυμεί να επιλέξει ένα χάπι για να το επεξεργαστεί.

Boundary PatientEditInfoPageGUI



Η κλάση αυτή αναφέρεται στο γραφικό παράθυρο διεπαφής το οποίο θα εμφανίζει στην πλατφόρμα τη σελίδα επεξεργασίας των στοιχείων του ασθενή.

Χαρακτηριστικά της κλάσης:

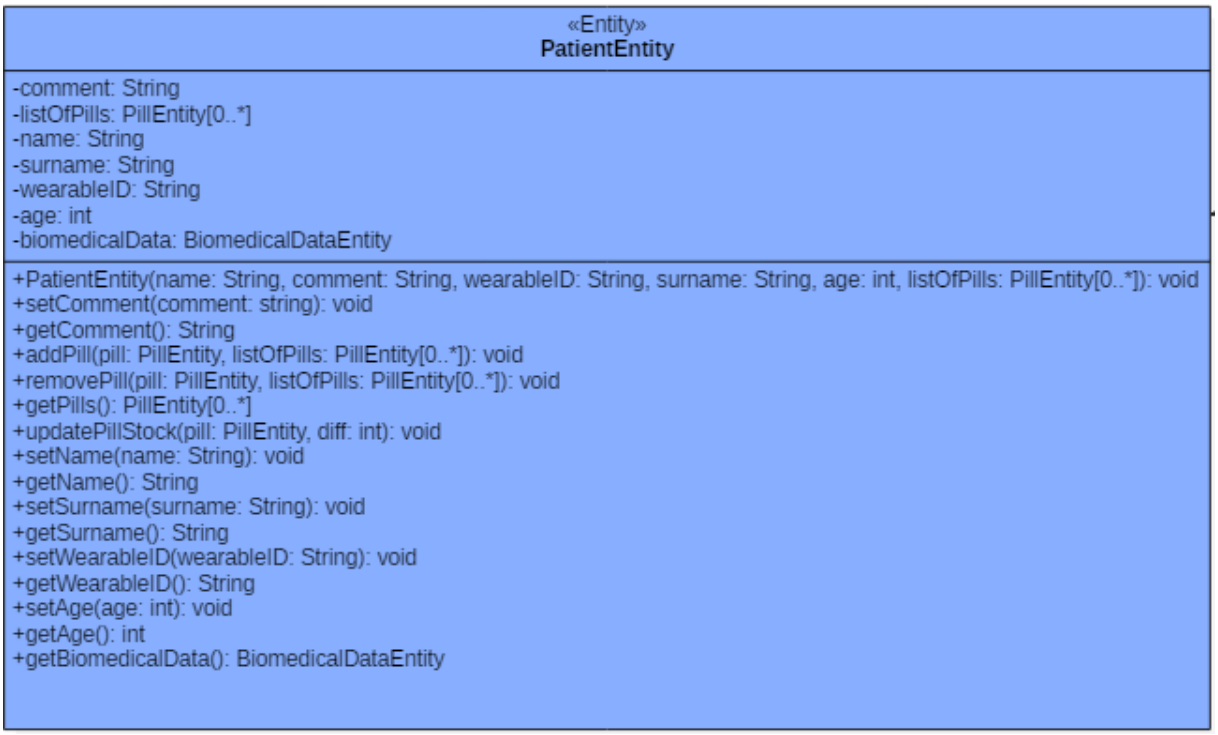
- patient: Ο ασθενής του οποίου τα στοιχεία βρίσκονται σε επεξεργασία.
- patientName: Κουτί για τη συμπλήρωση του ονόματος του ασθενή.
- patientSurname: Κουτί για τη συμπλήρωση του επωνύμου του ασθενή.

- patientAge: Κουτί για τη συμπλήρωση της ηλικίας του ασθενή.
- patientComment: Κουτί για τη συμπλήρωση των σχολίων του ασθενή.
- editPillsButton: Κουμπί με το οποίο ο caretaker μεταφέρεται στη σελίδα επεξεργασίας των χαπιών του ασθενή.
- backButton:Κουμπί με το οποίο ο caretaker μεταφέρεται στην προηγούμενη σελίδα.

Μέθοδοι της κλάσης:

- PatientEditInfoPageGUI(patient:patientEntity):Μέθοδος δόμησης της κλάσης PatientEditInfoPageGUI.
- display(): Μέθοδος η οποία καλείται για να εμφανίσει τη σελίδα επεξεργασίας των στοιχείων του ασθενή.
- editCommentClick():Μέθοδος η οποία καλείται όταν ο caretaker επιθυμεί να επεξεργαστεί το κουτί με τα σχόλια της υγείας του ασθενή.
- editPillsClick(): Μέθοδος η οποία καλείται όταν ο caretaker επιθυμεί να μεταβεί στη σελίδα επεξεργασίας των χαπιών του ασθενή.
- backClick(): Μέθοδος η οποία καλείται όταν ο caretaker επιθυμεί να μεταβεί στην προηγούμενη σελίδα.
- editNameClick():Μέθοδος η οποία καλείται όταν ο caretaker επιθυμεί να επεξεργαστεί το κουτί με το όνομα του ασθενή.
- editSurnameClick(): Μέθοδος η οποία καλείται όταν ο caretaker επιθυμεί να επεξεργαστεί το κουτί με το επώνυμο του ασθενή.
- editAge(): Μέθοδος η οποία καλείται όταν ο caretaker επιθυμεί να επεξεργαστεί το κουτί με την ηλικία του ασθενή.

Entity PatientEntity



Η κλάση αυτή εμπεριέχει όλες τις ιδιότητες και τις μεθόδους που απαιτούνται για να οριστεί πλήρως η οντότητα του ασθενή.

Χαρακτηριστικά της κλάσης:

- comment: μεταβλητή τύπου String που αφορά τα σχόλια σχετικά με την κατάσταση του ασθενή.
- name: μεταβλητή τύπου String που αναφέρεται στο όνομα του ασθενή.
- surname: μεταβλητή τύπου String που αναφέρεται στο επώνυμο του ασθενή.
- wearableID: μεταβλητή τύπου String που αφορά το μοναδικό αναγνωριστικό της συσκευής (wearable) που φοράει ο ασθενής στο χέρι του.
- age: μεταβλητή τύπου String που περιέχει την ηλικία του ασθενή.
- biomedicalData: μεταβλητή τύπου BiomedicalDataEntity που περιέχει τις μετρήσεις στα biomedical data του ασθενή.
- listOfPills: μεταβλητή τύπου PillEntity που περιέχει τη λίστα με τα φάρμακα του ασθενή.

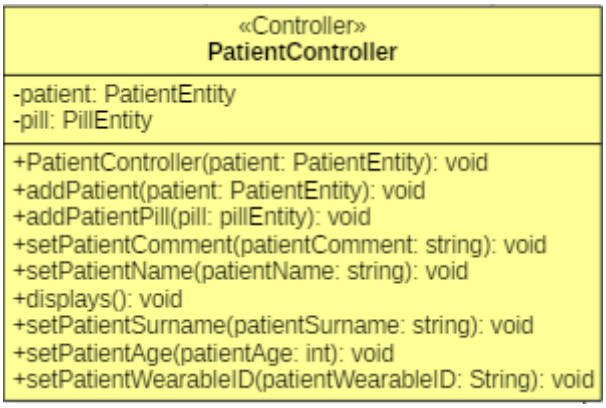
Μέθοδοι της κλάσης:

- PatientEntity(name: String, comment: String, wearableID: String, surname: String, age: int, listOfPills:PillEntity[0..*]): Μέθοδος δόμησης με ορίσματα το όνομα, το επώνυμο, την ηλικία, τα σχόλια για την υγεία,τη λίστα φαρμάκων του ασθενή και το αναγνωριστικό της συσκευής. Τα biomedicalData δεν δίνονται ως όρισμα επειδή θεωρούμε ότι λαμβάνονται από τους

αισθητήρες της συσκευής.

- setComment(comment: String): Μέθοδος που θέτει τα σχόλια στο προφίλ του ασθενή.
- getComment(): Μέθοδος που επιστρέφει τα σχόλια στο προφίλ του ασθενή ως string μεταβλητή.
- addPill(pill:PillEntity,listOfPills:PillEntity[0..*]):Μέθοδος η οποία έχει πρώτο όρισμα ένα αντικείμενο τύπου PillEntity το οποίο είναι το χάπι που μπορεί να προστεθεί στο δεύτερο όρισμα της μεθόδου listOfPills που υποδεικνύει τη λίστα χαπιών του ασθενή. Η μέθοδος δηλαδή αυτή ανανεώνει τη λίστα χαπιών του ασθενή.
- removePill(pill: PillEntity,listOfPills: PillEntity[0..*]): Μέθοδος η οποία αφαιρεί το χάπι το οποίο δόθηκε σαν όρισμα (pill) από τη λίστα χαπιών του ασθενή (listOfPills).
- getPills():Μέθοδος η οποία επιστρέφει τη λίστα χαπιών του ασθενή σαν μεταβλητή τύπου PillEntity[0..*].
- updatePillStock(pill:PillEntity, diff:int): Μέθοδος η οποία ανανεώνει τη διαθεσιμότητα του κάθε χαπιού. Το πρώτο όρισμα που παίρνει είναι το χάπι που πρόκειται να προστεθεί ή να μειωθεί και το δεύτερο ο αριθμός των χαπιών που απομένουν.
- setName(name: String): Μέθοδος που θέτει το όνομα του ασθενή.
- getName():Μέθοδος που επιστρέφει το όνομα του ασθενή ως string μεταβλητή.
- setSurname(surname: String): Μέθοδος που θέτει το επώνυμο του ασθενή.
- getSurname(): Μέθοδος που επιστρέφει το επώνυμο του ασθενή ως String μεταβλητή.
- setWearableID(wearableID: String): Μέθοδος που θέτει το μοναδικό αναγνωριστικό (ID) της συσκευής (wearable).
- getWearableID(): Μέθοδος που επιστρέφει το μοναδικό αναγνωριστικό (ID) της συσκευής (wearable) ως μεταβλητή τύπου String.
- setAge(age:int): Μέθοδος που θέτει την ηλικία του ασθενή.
- getAge(): Μέθοδος που επιστρέφει την ηλικία του ασθενή ως μεταβλητή τύπου integer.
- getBiomedicalData():Μέθοδος που επιστρέφει τις μετρήσεις στα biomedical data του ασθενή ως μεταβλητή τύπου BiomedicalDataEntity.

Controller PatientController



Η κλάση αυτή είναι ο ελεγκτής του ασθενή και περιέχει όλες τις κατάλληλες μεθόδους και τα χαρακτηριστικά που σχετίζονται με την αλληλεπίδραση του ασθενή με το χάπι ή τον caretaker.

Χαρακτηριστικά της κλάσης:

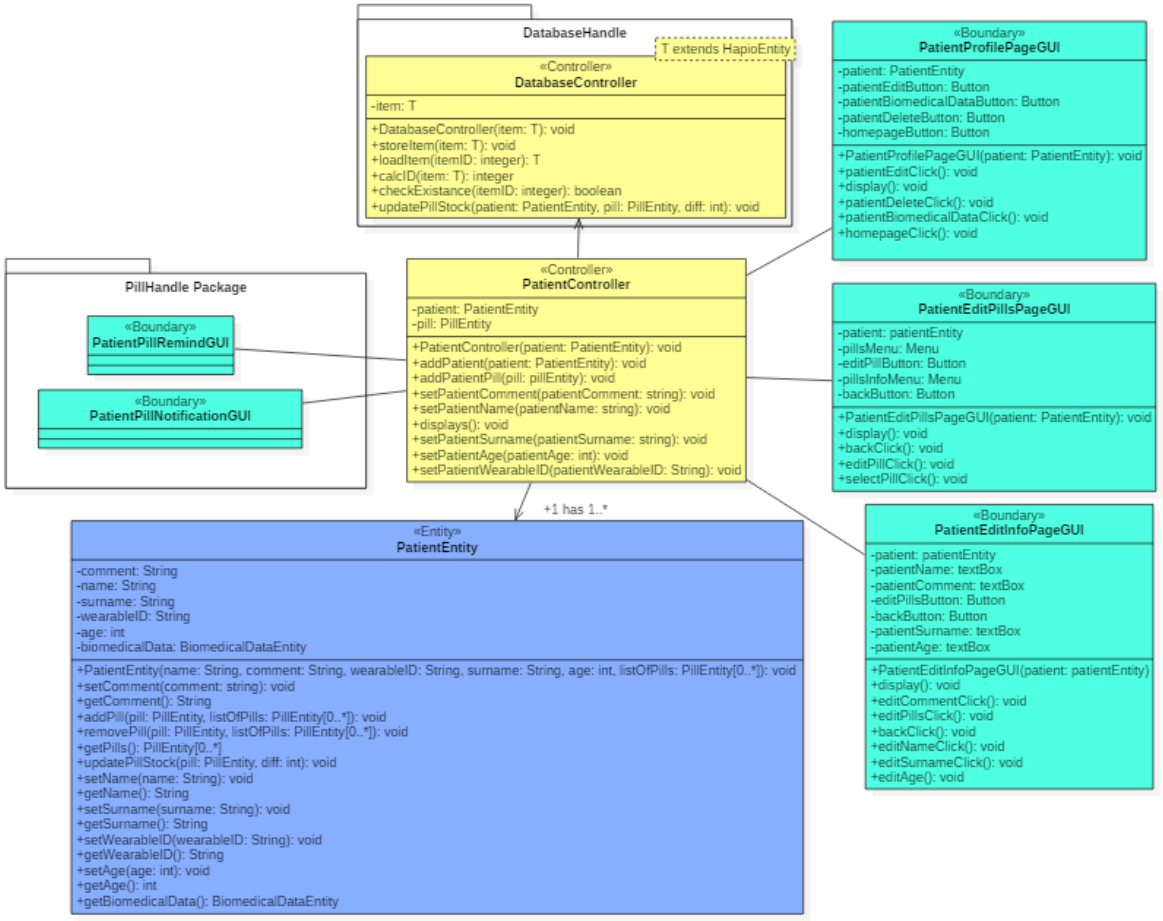
- patient: Ο ασθενής ο οποίος ελέγχεται από τον συγκεκριμένο ελεγκτή.
- Pill: Το χάπι που λαμβάνεται από τον ασθενή.

Μέθοδοι της κλάσης:

- PatientController(patient: PatientEntity): Μέθοδος δόμησης της κλάσης PatientController.
- addPatient(patient: PatientEntity): Μέθοδος η οποία προσθέτει έναν καινούργιο ασθενή με όρισμα τον ασθενή (patient) τύπου PatientEntity.
- addPatientPill(pill: pillEntity): Μέθοδος η οποία προσθέτει ένα καινούργιο χάπι σε κάποιον ασθενή με όρισμα το χάπι (pill) τύπου PillEntity.
- setPatientComment(patientComment:String):Μέθοδος η οποία καλείται όταν ο caretaker ορίζει τα σχόλια για την κατάσταση υγείας του ασθενή, μέσω του αντίστοιχου GUI.
- setPatientName(patientName: String): Μέθοδος η οποία καλείται όταν ο caretaker ορίζει το όνομα του ασθενή που πρόκειται να προστεθεί μέσω του αντίστοιχου GUI.
- displays():Μέθοδος η οποία καλεί τις μεθόδους display() των κλάσεων PatientProfilePageGUI, PatientEditPillsPageGUI, PatientEditInfoPageGUI οι οποίες εμφανίζουν τις αντίστοιχες γραφικές διεπαφές στην οθόνη όπως εξηγείται σε κάθε κλάση.
- setPatientSurname(patientSurname:String):Μέθοδος η οποία καλείται όταν ο caretaker ορίζει το επώνυμο του ασθενή που πρόκειται να προστεθεί μέσω του αντίστοιχου GUI.
- setPatientAge(patientAge: int): Μέθοδος η οποία καλείται όταν ο caretaker ορίζει την ηλικία του ασθενή που πρόκειται να προστεθεί μέσω του αντίστοιχου GUI.
- setPatientWearableID(patientWearableID: String): Μέθοδος η οποία καλείται όταν ο caretaker ορίζει το αναγνωριστικό της συσκευής (id) του ασθενή που πρόκειται να προστεθεί μέσω του αντίστοιχου GUI.

Διάγραμμα Κλάσεων

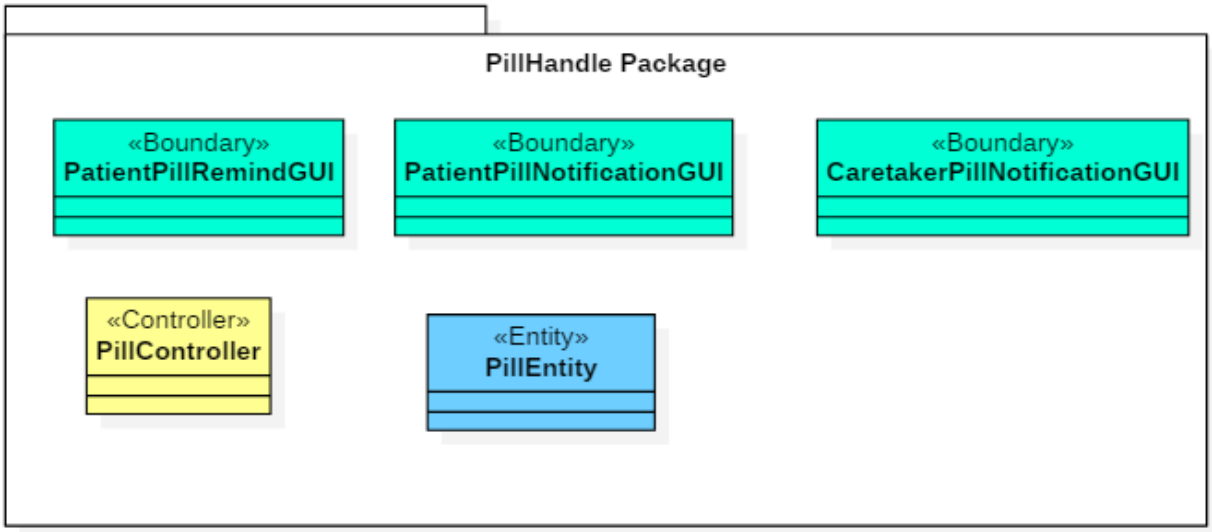
Σημείωση: Η δεικτοδότηση σε βέλη για ένα προς ένα συσχετίσεις μεταξύ κλάσεων παραλείπεται.



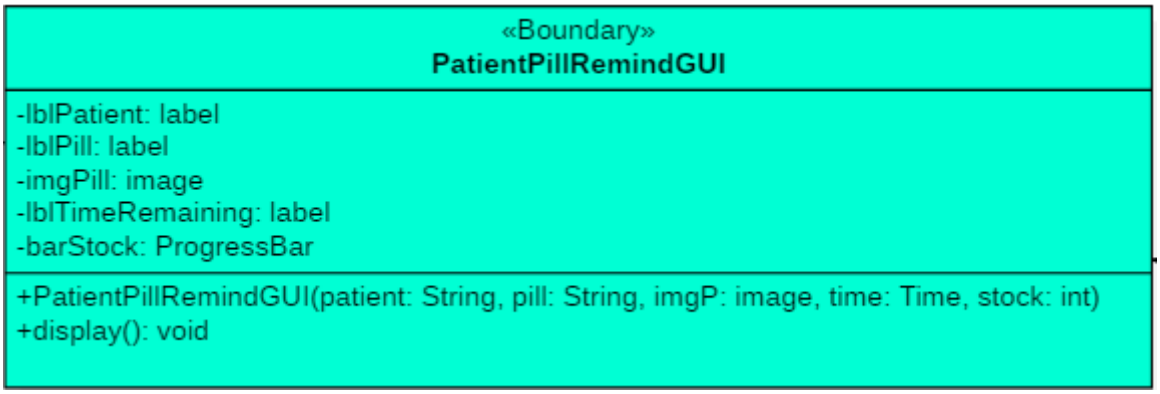
1.2 Πακέτα λεξιλογίου σεναρίων μέσης προτεραιότητας

1.2.1 Πακέτο PillHandle

Το πακέτο αυτό, αναφέρεται στη διαχείριση των χαπιών του ασθενή από την πλατφόρμα, καθώς και στις σχετικές με τα χάπια οθόνες που βλέπει ο ασθενής και ο caretaker.



Boundary PatientPillRemindGUI



Η κλάση αυτή εκφράζει την οθόνη υπενθύμισης λήψης χαπιού που εμφανίζεται στο wearable.

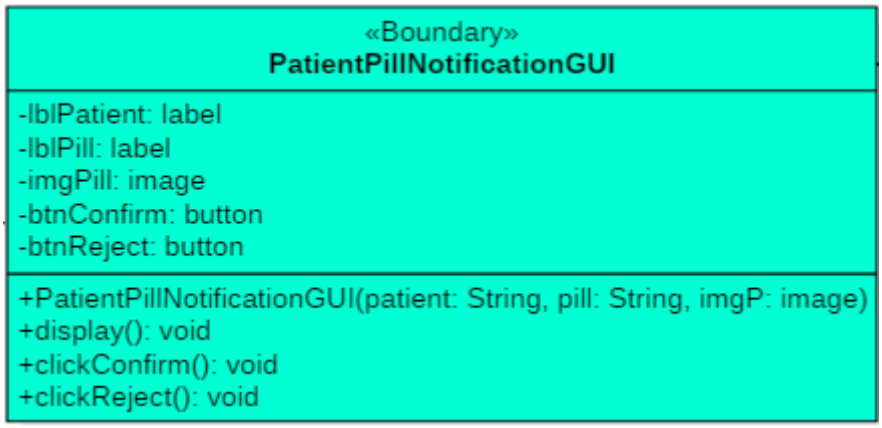
Χαρακτηριστικά της κλάσης:

- lblPatient: Επιγραφή με το όνομα του ασθενή
- lblPill: Επιγραφή με το όνομα του χαπιού προς λήψη
- imgPill: Εικόνα του χαπιού προς λήψη
- lblTimeRemaining: Χρόνος που απομένει μέχρι την λήψη του επόμενου χαπιού, σε μορφή H:M
- barStock: Μπάρα που δείχνει πόσο αποθέματα απομένουν στον ασθενή από το εν λόγω χάπι

Μέθοδοι της κλάσης:

- PatientPillRemindGUI(patient:String,pill:String, imgP:image,time:Time, stock:int): Constructor για την κλάση PatientPillRemindGUI
- display(): Η μέθοδος που καλείται για να εμφανίσει την οθόνη υπενθύμισης λήψης χαπιού

Boundary PatientPillNotificationGUI



Η κλάση αυτή εκφράζει την οθόνη ειδοποίησης του ασθενή που εμφανίζεται στο wearable, για την επιβεβαίωση ή μη της λήψης του χαπιού από τον ασθενή.

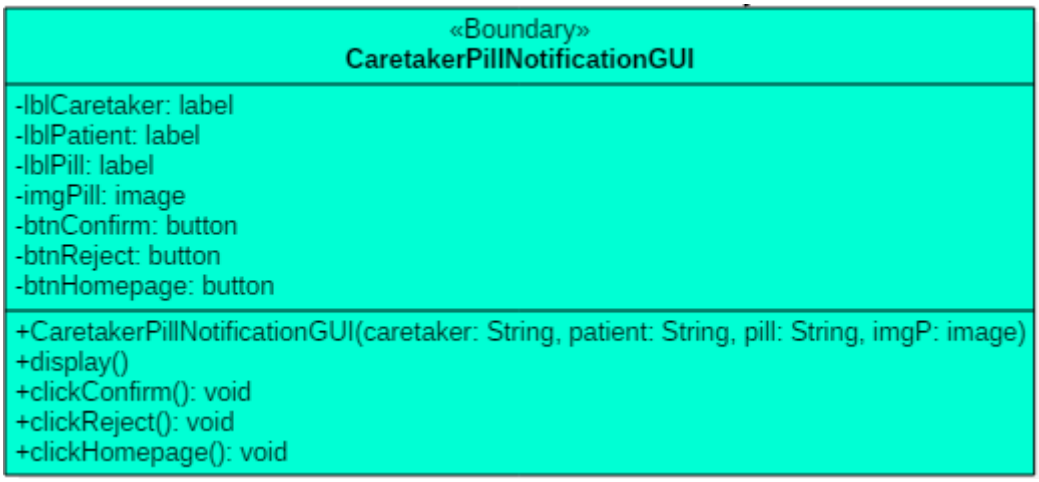
Χαρακτηριστικά της κλάσης:

- lblPatient: Επιγραφή με το όνομα του ασθενή
- lblPill: Επιγραφή με το όνομα του χαπιού προς λήψη
- imgPill: Εικόνα του χαπιού προς λήψη
- btnConfirm: Κουμπί με το οποίο ο ασθενής επιβεβαιώνει την λήψη του χαπιού
- btnReject: Κουμπί με το οποίο ο ασθενής απορρίπτει την λήψη του χαπιού

Μέθοδοι της κλάσης:

- PatientPillNotificationGUI(patient:String,pill:String,imgP:image): Constructor για την κλάση PatientPillNotificationGUI
- display(): Η μέθοδος που καλέιται για να εμφανίσει την οθόνη ειδοποίησης του ασθενή για την επιβεβαίωση ή μη της λήψης του χαπιού
- clickConfirm(): Η μέθοδος που καλείται για την επιβεβαίωση της λήψης του χαπιού από τον ασθενή
- clickReject(): Η μέθοδος που καλείται για την απόρριψη της λήψης του χαπιού από τον ασθενή

Boundary CaretakerPillNotificationGUI



Η κλάση αυτή εκφράζει την οθόνη ειδοποίησης του caretaker που εμφανίζεται στην εφαρμογή διαχείρισης, για την επιβεβαίωση ή μη της λήψης του χαπιού από τον ασθενή και από τον ίδιο τον caretaker.

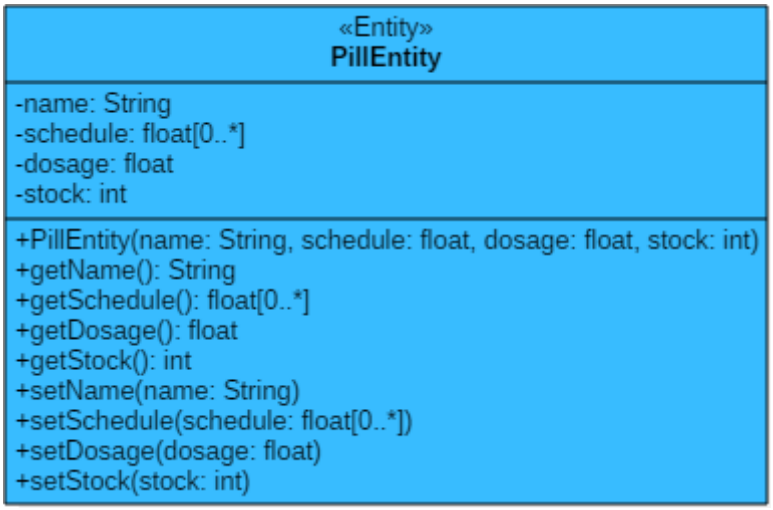
Χαρακτηριστικά της κλάσης:

- lblCaretaker: Επιγραφή με το όνομα του caretaker.
- lblPatient: Επιγραφή με το όνομα του ασθενή
- lblPill: Επιγραφή με το όνομα του χαπιού προς λήψη
- imgPill: Εικόνα του χαπιού προς λήψη
- btnConfirm: Κουμπί με το οποίο ο caretaker επιβεβαιώνει την λήψη του χαπιού από τον ασθενή
- btnReject: Κουμπί με το οποίο ο caretaker απορρίπτει την λήψη του χαπιού από τον ασθενή

Μέθοδοι της κλάσης:

- CaretakerPillNotificationGUI(caretaker:String,patient:String,pill:String,imgP:image): Constructor για την κλάση CaretakerPillNotificationGUI
- display(): Η μέθοδος που καλέιται για να εμφανίσει την οθόνη ειδοποίησης του caretaker για την επιβεβαίωση ή μη της λήψης του χαπιού από τον ασθενή
- clickConfirm(): Η μέθοδος που καλείται για την επιβεβαίωση από τον caretaker της λήψης του χαπιού από τον ασθενή
- clickReject(): Η μέθοδος που καλείται για την απόρριψη από τον caretaker της λήψης του χαπιού από τον ασθενή
- clickHomepage(): Η μέθοδος που καλείται για να επιστρέψει ο caretaker στην αρχική σελίδα της εφαρμογής

Entity PillEntity



Η κλάση αυτή είναι η κλάση οντότητας του χαπιού, και περιλαμβάνει όλα τα χαρακτηριστικά που αυτό εμπεριέχει.

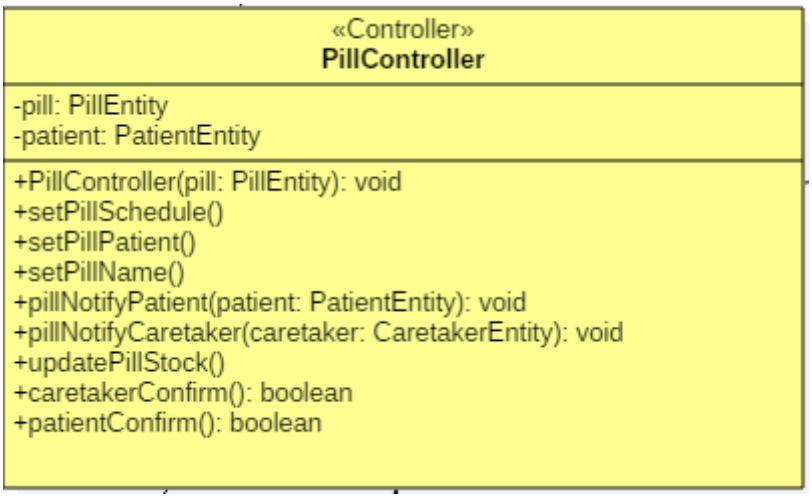
Χαρακτηριστικά της κλάσης:

- name: Το όνομα του χαπιού, που αποθηκεύεται σε String
- schedule: Το πρόγραμμα λήψης του χαπιού από τον ασθενή
- dosage: Η δοσολογία του χαπιού, δηλαδή η ποσότητα που λαμβάνει ο ασθενής κάθε φορά
- stock: Το απόθεμα του συγκεκριμένου χαπιού που παραμένει, πριν χρειαστεί ο χρήστης να προμηθευτεί άλλα

Μέθοδοι της κλάσης:

- PillEntity(name:String, schedule:float, dosage:float, stock:int): Constructor για την κλάση PillEntity
- getName(): Η συνάρτηση αυτή επιστρέφει το όνομα του χαπιού
- getSchedule(): Η συνάρτηση αυτή επιστρέφει το πρόγραμμα λήψης του χαπιού
- getDosage(): Η συνάρτηση αυτή επιστρέφει τη δοσολογία του χαπιού
- getStock(): Η συνάρτηση αυτή επιστρέφει τα αποθέματα του χαπιού
- setName(name: String): Η συνάρτηση αυτή θέτει τη μεταβλητή name ίση με το όρισμα
- setSchedule(schedule: float[0..*]): Η συνάρτηση αυτή θέτει τη μεταβλητή schedule ίση με το όρισμα
- setDosage(dosage: float): Η συνάρτηση αυτή θέτει τη μεταβλητή dosage ίση με το όρισμα
- setStock(stock: int): Η συνάρτηση αυτή θέτει τη μεταβλητή stock ίση με το όρισμα

Controller PillController



Η κλάση αυτή είναι ο ελεγκτής του χαπιού, που περιέχει όλες τις μεθόδους που σχετίζονται με την αλληλεπίδραση του χαπιού με τον ασθενή ή τον caretaker.

Χαρακτηριστικά της κλάσης:

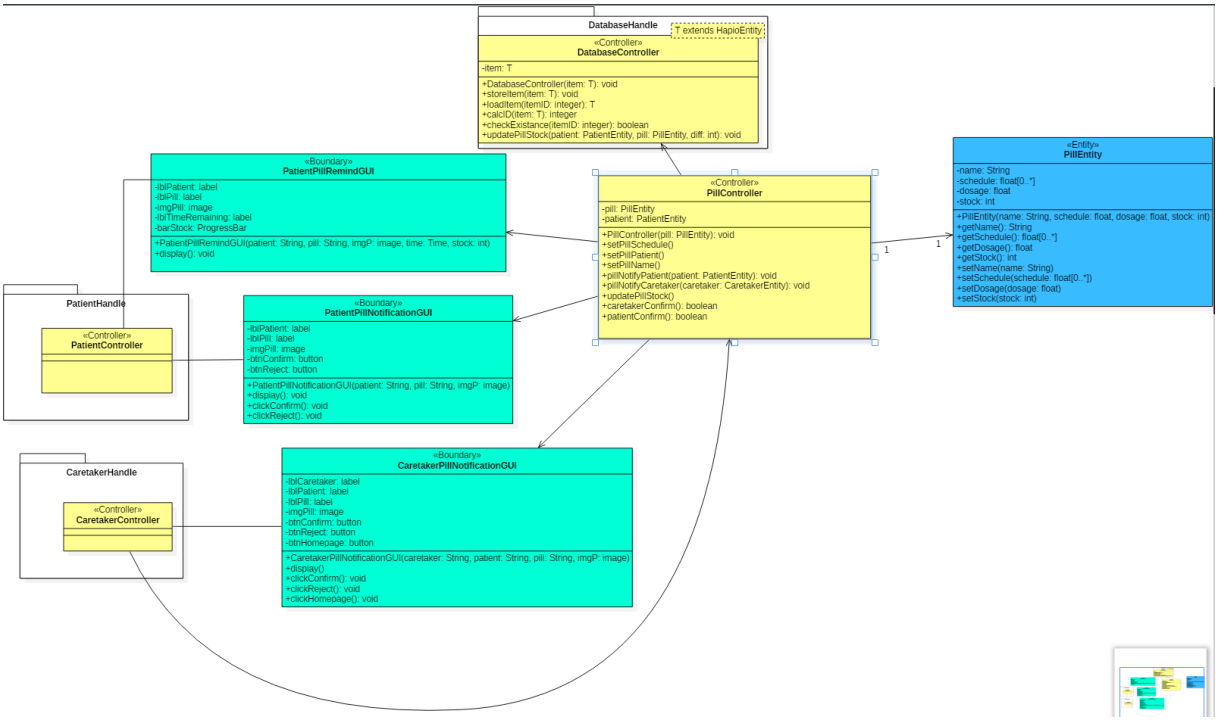
- pill: Το χάπι το οποίο ελέχγει ο εν λόγω ελεγκτής
- patient: Ο ασθενής που λαμβάνει το εν λόγω χάπι

Μέθοδοι της κλάσης:

- PillController(pill: PillEntity): Ο constructor της κλάσης PillController
- setPillSchedule(): Η μέθοδος που καλείται όταν ο caretaker ορίζει το πρόγραμμα λήψης του χαπιού, μέσω του αντίστοιχου GUI
- setPillPatient(): Η μέθοδος που καλείται όταν ο caretaker ορίζει τον ασθενή που λαμβάνει το χάπι, μέσω του αντίστοιχου GUI
- setPillName(): Η μέθοδος που καλείται όταν ο caretaker δίνει όνομα στο χάπι, μέσω του αντίστοιχου GUI
- pillNotifyPatient(patient: PatientEntity): Η μέθοδος που καλείται όταν στέλνεται ειδοποίηση στον ασθενή για επιβεβαίωση ή μη της λήψης χαπιού
- pillNotifyCaretaker(caretaker: CaretakerEntity): Η μέθοδος που καλείται όταν στέλνεται ειδοποίηση στον caretaker για επιβεβαίωση ή μη της λήψης χαπιού από τον ασθενή
- updatePillStock(): Η μέθοδος που καλείται όταν αλλάζουν τα αποθέματα του χαπιού, είτε μέσω της προμήθειας νέων από τον caretaker, είτε μέσω της μείωσης τους μετά από επιτυχή λήψη
- caretakerConfirm(): Η μέθοδος που καλείται, όταν ο caretaker επιβεβαίωσε επιτυχώς τη λήψη του χαπιού από τον ασθενή
- patientConfirm(): Η μέθοδος που καλείται, όταν ο ασθενής επιβεβαίωσε επιτυχώς τη λήψη του χαπιού του

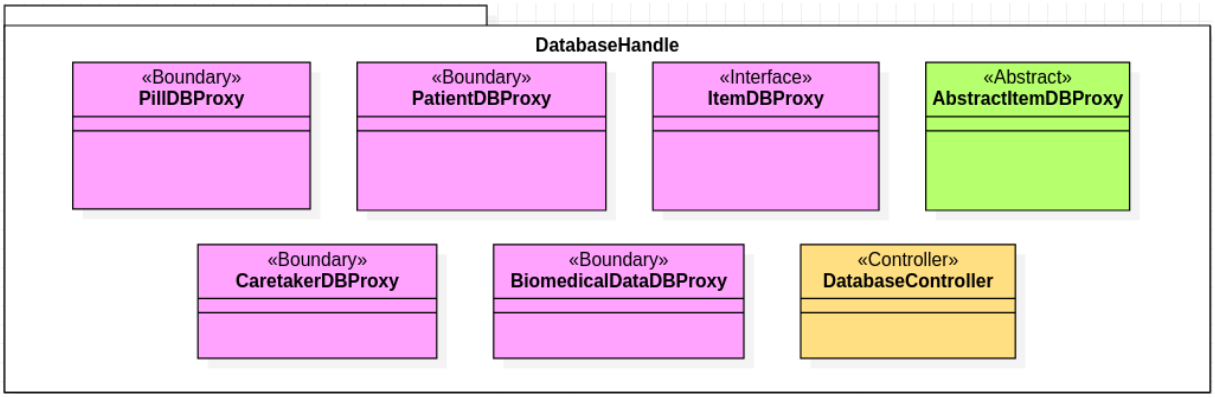
Διάγραμμα Κλάσεων

Σημείωση: Η δεικτοδότηση σε βέλη για ένα προς ένα συσχετίσεις μεταξύ κλάσεων παραλείπεται.

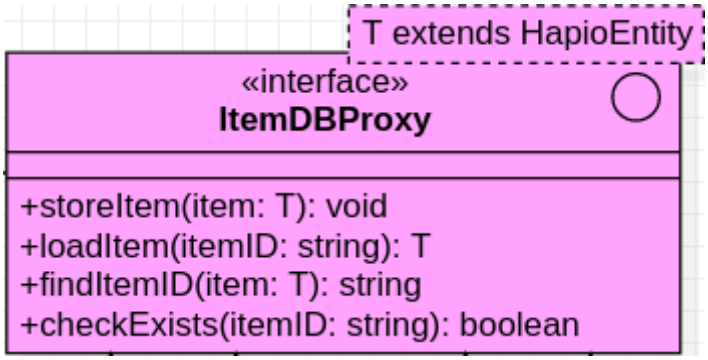


1.2.2 Πακέτο DatabaseHandle

Το πακέτο αυτό διαχειρίζεται ό,τι αφορά τις βάσεις δεδομένων του συστήματος, και χρησιμοποιείται το bridge design pattern σε συνδυασμό με Generics για επεκτασιμότητα για μελλοντικά entities.



Interface ItemDBProxy

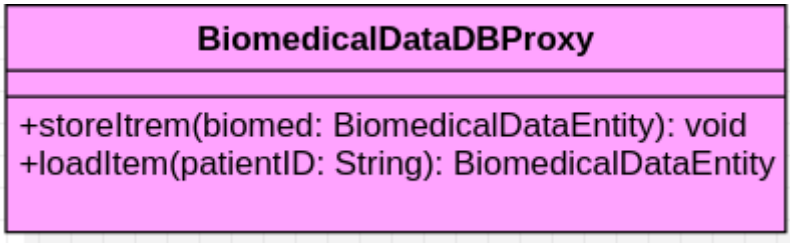


Η διεπαφή αυτή είναι η βάση του bridge design pattern μέσω της οποίας υλοποιούνται διαφορετικά proxy για διαφορετικά entity.

Μέθοδοι της διεπαφής:

- storeItem(item: T): Μέθοδος που καλείται για την αποθήκευση αντικειμένου τύπου T.
- loadItem(itemID: string): Μέθοδος που καλείται για την φόρτωση του αντικειμένου μέσω το αντίστοιχο ID του.
- findItemID(item: T): Μέθοδος που καλείται για την εύρεση του ID του αντικειμένου τύπου T.
- checkExists(itemID: string): Μέθοδος που καλείται για να ελέγξουμε αν υπάρχει το αντικειμένου μέσω του θεωρητικού ID του.

Boundary BiomadicalDataDBProxy

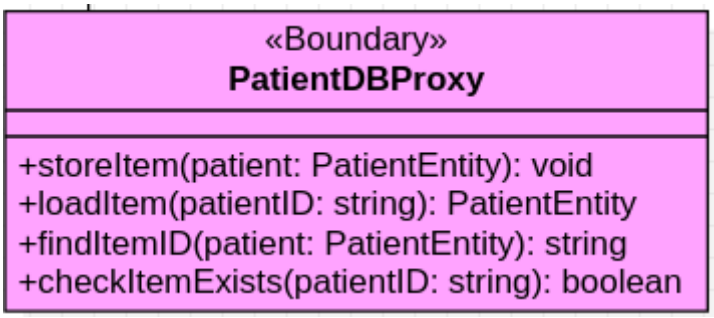


Η κλάση αυτή εκφράζει την σύνδεση του συστήματος με τη βάση δεδομένων που σχετίζεται με τα biomedical data. Υλοποιεί την διεπαφή ItemDBProxy.

Μέθοδοι της κλάσης:

- storeItem(biomed: BiomedicalDataEntity): Μέθοδος η οποία καλείται όταν αποθηκεύεται στη βάση δεδομένων ένα αντικείμενο τύπου BiomedicalDataEntity το οποίο δίνεται ως όρισμα στη μέθοδο.
- loadItem(patientID: String): Μέθοδος η οποία καλείται όταν θέλουμε να φορτώσουμε τα Biomedical Data κάποιου ασθενή, δίνοντας ως όρισμα το ID του ασθενή για να αναζητηθεί στη βάση.

Boundary PatientDBProxy

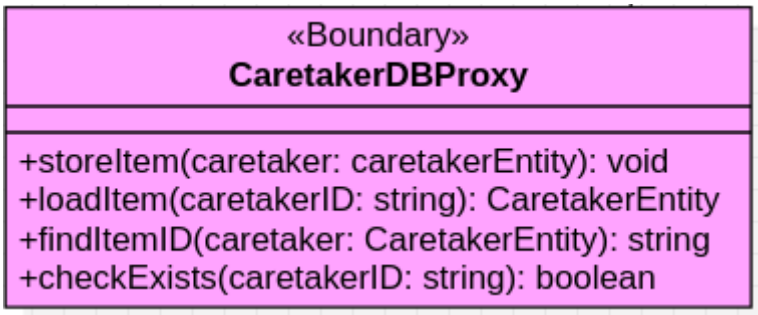


Η κλάση αυτή εκφράζει την σύνδεση του συστήματος με τη βάση δεδομένων που σχετίζεται με τα στοιχεία του ασθενή.

Μέθοδοι της κλάσης:

- storeItem(patient: PatientEntity): Μέθοδος η οποία καλείται όταν αποθηκεύεται στη βάση δεδομένων ένα αντικείμενο τύπου PatientEntity το οποίο δίνεται ως όρισμα στη μέθοδο.
- loadItem(patientID: String): Μέθοδος η οποία καλείται όταν φορτώνεται ένα αντικείμενο τύπου PatientEntity από τη βάση δεδομένων δίνοντας ως όρισμα το ID του ασθενή για να αναζητηθεί στη βάση.
- findItemID(patient: PatientEntity): Μέθοδος η οποία καλείται όταν γίνεται αναζήτηση στη βάση δεδομένων για το ID του αντικειμένου patient τύπου PatientEntity που δίνεται ως όρισμα στη μέθοδο.
- checkItemExists(patientID: string): Μέθοδος η οποία καλείται όταν ελέγχεται αν ο ασθενής με το ID που δίνεται ως όρισμα υπάρχει στη βάση δεδομένων.

Boundary CaretakerDBProxy

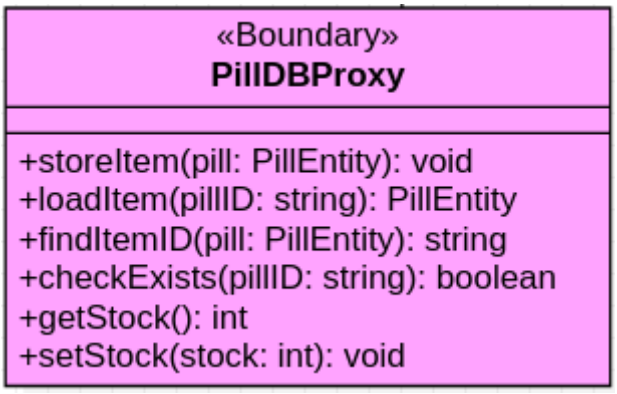


Η κλάση αυτή εκφράζει την διεπαφή του συστήματος για τη βάση δεδομένων που σχετίζεται με τα στοιχεία του caretaker.

Μέθοδοι της κλάσης:

- storeItrem(patient: CaretakerEntity): Μέθοδος η οποία καλείται όταν αποθηκεύεται στη βάση δεδομένων ένα αντικείμενο τύπου CaretakerEntity το οποίο δίνεται ως όρισμα στη μέθοδο.
- loadItem(patientID: String): Μέθοδος η οποία καλείται όταν φορτώνεται ένα αντικείμενο τύπου CaretakerEntity από τη βάση δεδομένων δίνοντας ως όρισμα το ID του caretaker για να αναζητηθεί στη βάση.
- findItemID(caretaker: CaretakerEntity): Μέθοδος η οποία καλείται όταν γίνεται αναζήτηση στη βάση δεδομένων για το ID του αντικειμένου caretaker τύπου CaretakerEntity που δίνεται ως όρισμα στη μέθοδο.
- checkItemExists(caretakerID: string): Μέθοδος η οποία καλείται όταν ελέγχεται αν ο caretaker με το ID που δίνεται ως όρισμα υπάρχει στη βάση δεδομένων.

Boundary PillDBProxy

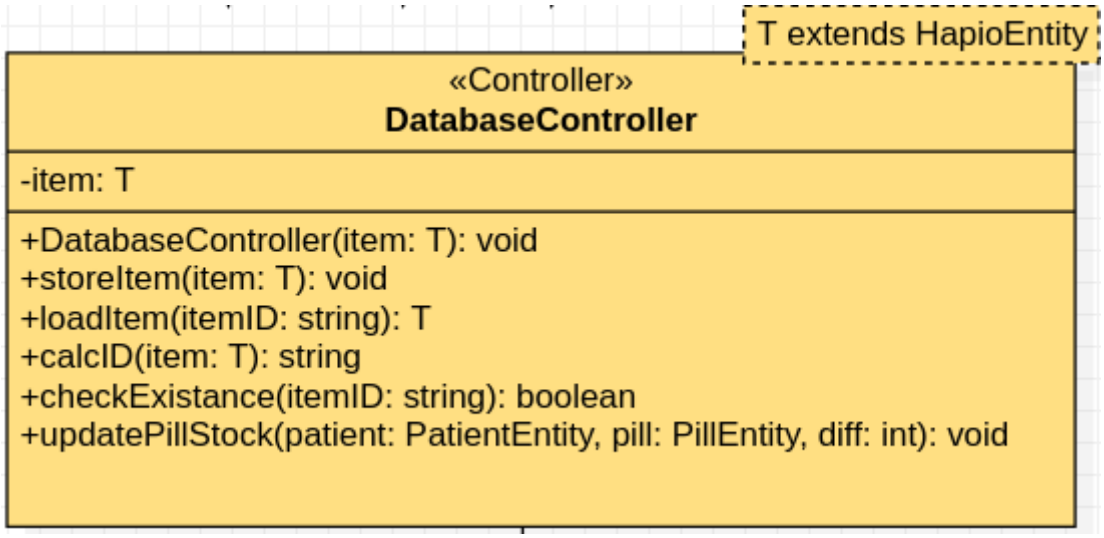


Η κλάση αυτή εκφράζει την σύνδεση του συστήματος με τη βάση δεδομένων που σχετίζεται με τα φάρμακα του ασθενή.

Μέθοδοι της κλάσης:

- storeItem(pill: PillEntity): Μέθοδος η οποία καλείται όταν αποθηκεύεται στη βάση δεδομένων ένα αντικείμενο τύπου PillEntity το οποίο δίνεται ως όρισμα στη μέθοδο.
- loadItem(pillID: String): Μέθοδος η οποία καλείται όταν φορτώνεται ένα αντικείμενο τύπου PillEntity από τη βάση δεδομένων δίνοντας ως όρισμα το ID του pill για να αναζητηθεί στη βάση.
- findItemID(pill: PillEntity): Μέθοδος η οποία καλείται όταν γίνεται αναζήτηση στη βάση δεδομένων για το ID του αντικειμένου pill τύπου PillEntity που δίνεται ως όρισμα στη μέθοδο.
- checkItemExists(pillID: string): Μέθοδος η οποία καλείται όταν ελέγχεται αν το φάρμακο με το ID που δίνεται ως όρισμα υπάρχει στη βάση δεδομένων.
- getStock(): Η συνάρτηση αυτή επιστρέφει το απόθεμα των φαρμάκων.
- setStock(stock: int): Η συνάρτηση αυτή θέτει τη μεταβλητή Stock ίση με το όρισμα.

Controller DatabaseController



Η κλάση αυτή είναι ο ελεγκτής της σύνδεσης του συστήματος με τις βάσεις δεδομένων και περιέχει όλες τις μεθόδους που σχετίζονται με την αποθήκευση, φόρτωση και αναζήτηση των διαφόρων entities από και προς τις βάσεις δεδομένων. Χρησιμοποιεί Generics τα οποία περιορίζονται στις υποκλάσεις του HapioEntity, ώστε να υπάρχει ενιαία διεπαφή με τα υπόλοιπα πακέτα και να μειώνεται η πολυπλοκότητα του συστήματος

Χαρακτηριστικά της κλάσης:

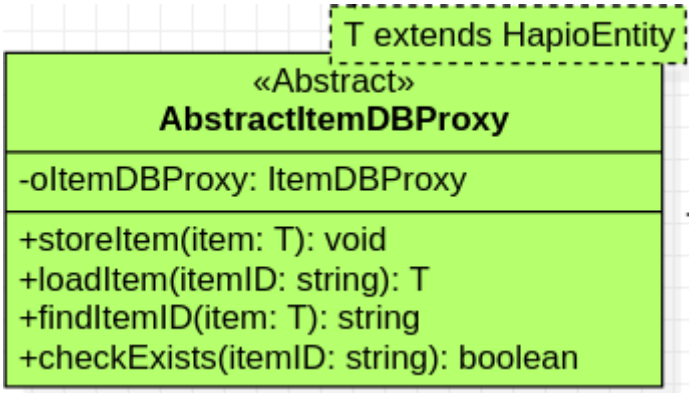
- item: Πρόκεται για οποιοδήποτε από τα entity που είναι υπο-κλάσεις του hapioEntity και χρησιμοποιούνται στο σύστημα. Χρησιμοποιώντας Generics, μπορούμε να χρησιμοποιήσουμε το ίδιο σημείο εισόδου για όλα τα hapioEntities.

Μέθοδοι της κλάσης:

- DatabaseController(item: T): Ο constructor της κλάσης DatabaseController
- storeItem(item: T): Μέθοδος η οποία καλείται όταν αποθηκεύεται στη βάση δεδομένων ένα αντικείμενο υποκλάση του hapioEntity το οποίο δίνεται ως όρισμα στη μέθοδο.

- loadItem(itemID: string): Μέθοδος η οποία καλείται όταν φορτώνεται ένα αντικείμενο τύπου itemID από τη βάση δεδομένων δίνοντας ως όρισμα το ID του item για να αναζητηθεί στη βάση.
- calcID(item: T): Μέθοδος που υπολογίζει το ID του αντικειμένου.
- checkExistance(itemID: string): Μέθοδος που ελέγχει αν το αντικειμενο υπάρχει στην βάση δεδομένων, δεδομένου του ID του.
- updatePillStock(patient: PatientEntitym pill: PillEntity, diff: int): Μέθοδος συγκεκριμένα για να ανανεώνει τα αποθέματα του δοθέντος φαρμάκου κάποιου ασθενούς.

AbstractItemDBProxy



Είναι η αφαίρεση που χρησιμοποιείται για το bridge pattern για την υλοποίηση των διαφορετικών Database proxy

Χαρακτηριστικά της κλάσης:

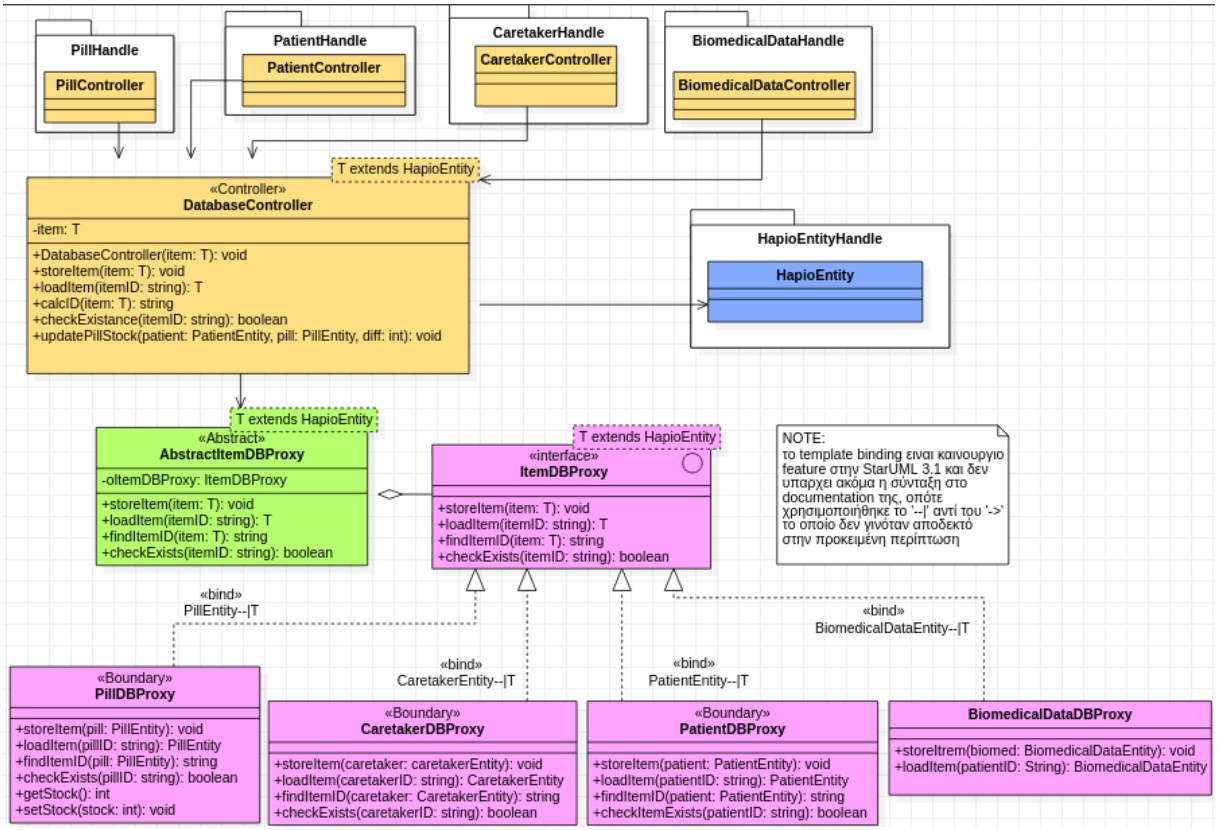
- ο ItemDBProxy: Είναι ο implementor της αφαίρεσης.

Μέθοδοι της κλάσης:

- storeItem(item: T): Μέθοδος η οποία καλείται όταν αποθηκεύεται στη βάση δεδομένων ένα αντικείμενο υποκλάση του hapioEntity το οποίο δίνεται ως όρισμα.
- loadItem(itemID: string): Μέθοδος η οποία καλείται για να φορτωθεί ένα αντικείμενο τύπου itemID από τη βάση δεδομένων δίνοντας ως όρισμα το ID του item για να αναζητηθεί στη βάση.
- checkExists(itemID: string): Μέθοδος που ελέγχει αν το αντικειμενο υπάρχει στην βάση δεδομένων, δεδομένου του ID του.
- findItemID(item: T): Μέθοδος η οποία καλείται όταν γίνεται αναζήτηση στη βάση δεδομένων για το ID του αντικειμένου hapioEntity που δίνεται ως όρισμα στη μέθοδο.
- updatePillStock(patient: PatientEntitym pill: PillEntity, diff: int): Μέθοδος συγκεκριμένα για να ανανεώνει τα αποθέματα του δοθέντος φαρμάκου κάποιου ασθενούς.

Διάγραμμα Κλάσεων

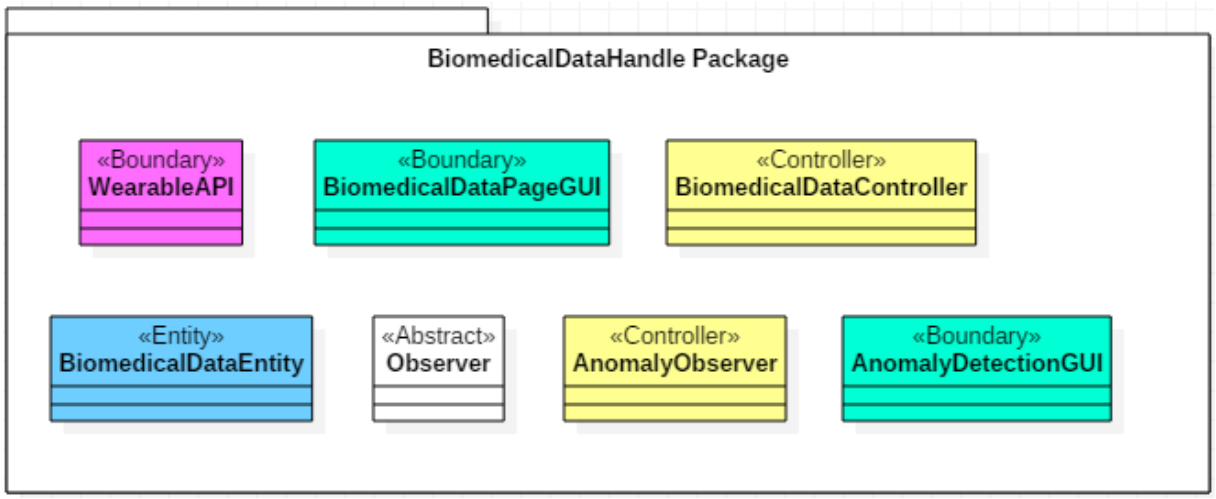
Σημείωση: Η δεικτοδότηση σε βέλη για ένα προς ένα συσχετίσεις μεταξύ κλάσεων παραλείπεται.



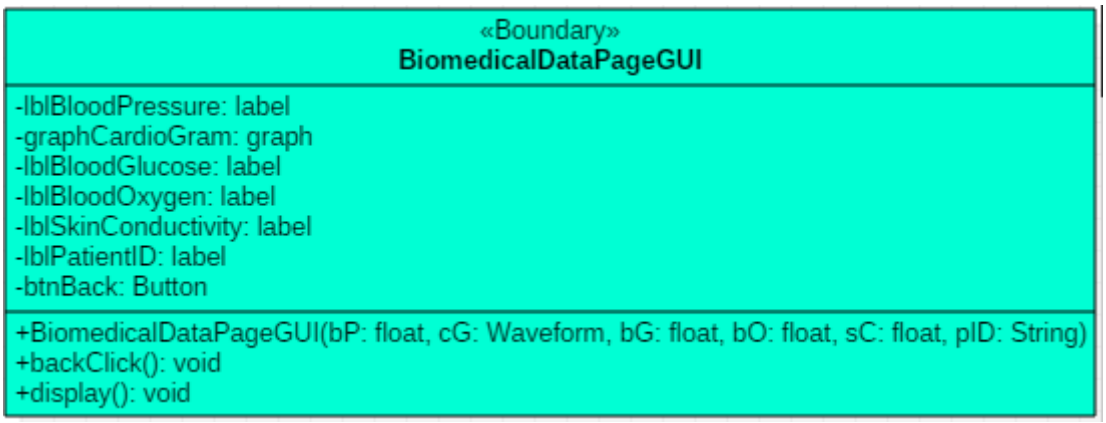
1.3 Πακέτα λεξιλογίου σεναρίων χαμηλής προτεραιότητας

1.3.1 Πακέτο BiomedicalDataHandle

Το πακέτο αυτό αναφέρεται στην προβολή των βιοϊατρικών δεδομένων του ασθενούς, καθώς και στην ενημέρωση του caretaker για τυχόν ανωμαλίες που μπορεί να προκύψουν σε αυτά, δηλαδή αποκλίσεις από τις φυσιολογικές τιμές.



Boundary BiomedicalDataPageGUI



Η κλάση αυτή εκφράζει την οθόνη προβολής των βιοϊατρικών δεδομένων του ασθενούς από τον caretaker του.

Χαρακτηριστικά της κλάσης:

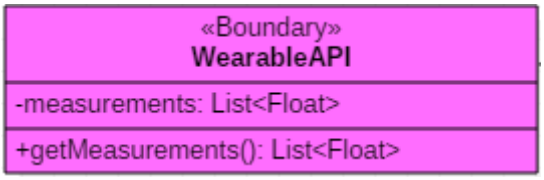
- lblBloodPressure: Επιγραφή με την αρτηριακή πίεση του ασθενούς
- graphCardioGram: Γράφημα με τους καρδιακούς παλμούς του ασθενούς
- lblBloodGlucose: Επιγραφή με την συγκέντρωση γλυκόζης στο αίμα του ασθενούς

- lblSkinConductivity: Επιγραφή με την αγωγιμότητα του δέρματος του ασθενούς
- lblPatientID: Επιγραφή με το όνομα του ασθενούς, στον οποίο ανήκουν τα εν λόγω δεδομένα
- btnBack: Κουμπί για να επιστρέψει ο caretaker στην προηγούμενη οθόνη

Μέθοδοι της κλάσης:

- BiomedicalDataPageGUI(bP:float,cG:Waveform,bG:float,bO:float,sC:float,pID:String): Constructor για την κλάση BiomedicalDataPageGUI
- display(): Η μέθοδος που καλείται για να εμφανίσει την οθόνη προβολής των βιοϊατρικών δεδομένων του ασθενούς
- backClick(): Η μέθοδος που καλείται για να επιστρέψει ο caretaker στην προηγούμενη οθόνη, πατώντας το κατάλληλο κουμπί

Boundary WearableAPI



Η κλάση αυτή εκφράζει τη διεπαφή ανάμεσα στο εξωτερικό σύστημα του wearable, που λαμβάνει τις μετρήσεις, και την πλατφόρμα του hap.io.

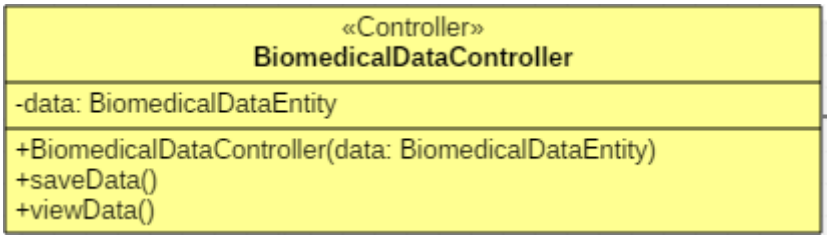
Χαρακτηριστικά της κλάσης:

- measurements: Λίστα από float, στην οποία περιέχονται οι πρωτογενείς μετρήσεις που λαμβάνει το wearable. Μετασχηματίζονται σε μεταγενέστερο στάδιο στην κατάλληλη μορφή για χρήση από την εφαρμογή

Μέθοδοι της κλάσης:

- getMeasurements(): Η μέθοδος που καλείται, ώστε να σταλούν οι μετρήσεις από το wearable στην εφαρμογή

Controller BiomedicalDataController



Η κλάση αυτή είναι ο ελεγκτής της οντότητας BiomedicalDataEntity, και περιέχει τις μεθόδους που σχετίζονται με την αποθήκευση και προβολή των βιοϊατρικών δεδομένων.

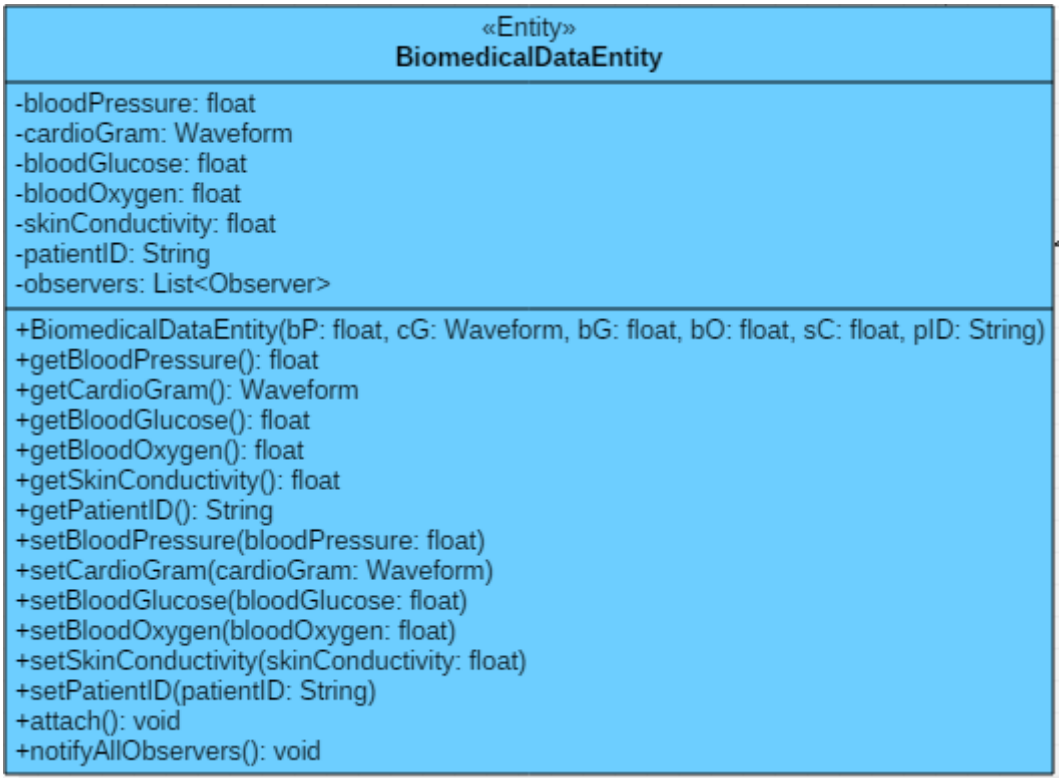
Χαρακτηριστικά της κλάσης:

- data: Η οντότητα BiomedicalDataEntity την οποία ελέγχει το εν λόγω controller

Μέθοδοι της κλάσης:

- BiomedicalDataController(data: BiomedicalDataEntity): Constructor για την κλάση BiomedicalDataController
- saveData(): Μέθοδος που καλείται ώστε να αποθηκευτούν τα δεδομένα που ελήφθησαν από το wearable στο Entity και στη βάση δεδομένων
- viewData(): Μέθοδος που καλείται ώστε να προβληθούν τα βιοϊατρικά δεδομένα, δηλαδή να μεταβεί ο χρήστης στο BiomedicalDataPageGUI

Entity BiomedicalDataEntity



Η κλάση αυτή είναι η κλάση οντότητας των βιοϊατρικών δεδομένων του χρήστη, και περιλαμβάνει όλες τις σχετικές μετρήσεις, καθώς και τα στοιχεία σχετικά με το observer pattern που εφαρμόστηκε.

Χαρακτηριστικά της κλάσης:

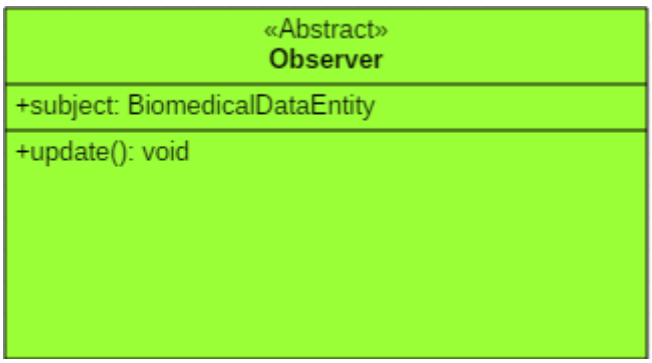
- bloodPressure: Η αρτηριακή πίεση του ασθενούς
- cardioGram: Το καρδιογράφημα του ασθενούς
- bloodGlucose: Η συγκέντρωση γλυκόζης στο αίμα του ασθενούς
- bloodOxygen: Η συγκέντρωση οξυγόνου στο αίμα του ασθενούς
- skinConductivity: Η αγωγιμότητα του δέρματος του ασθενούς
- patientID: Το αναγνωριστικό του ασθενούς στον οποίο αντιστοιχούν τα παραπάνω δεδομένα
- observers: Λίστα από τους observers που παρακολουθούν το entity. Έχει να κάνει με το observer design pattern, που χρησιμοποιείται για την ανίχνευση και ειδοποίηση για ανωμαλίες

Μέθοδοι της κλάσης:

- BiomedicalDataEntity(bP:float,cG:Waveform,bG:float,bO:float,sC:float,pID:String): Constructor για την κλάση BiomedicalDataEntity
- getBloodPressure(): Η συνάρτηση αυτή επιστρέφει την αρτηριακή πίεση του ασθενούς
- getCardioGram(): Η συνάρτηση αυτή επιστρέφει το καρδιογράφημα του ασθενούς

- getBloodGlucose(): Η συνάρτηση αυτή επιστρέφει τη συγκέντρωση γλυκόζης στο αίμα του ασθενούς
- getBloodOxygen(): Η συνάρτηση αυτή επιστρέφει τη συγκέντρωση οξυγόνου στο αίμα του ασθενούς
- getSkinConductivity(): Η συνάρτηση αυτή επιστρέφει την αγωγιμότητα του δέρματος του ασθενούς
- getPatientID(): Η συνάρτηση αυτή επιστρέφει το αναγνωριστικό του ασθενούς στον οποίο αντιστοιχούν τα δεδομένα
- setBloodPressure(bloodPressure: float): Η συνάρτηση αυτή θέτει τη μεταβλητή bloodPressure ίση με το όρισμα
- setCardioGram(cardioGram: Waveform): Η συνάρτηση αυτή θέτει τη μεταβλητή cardioGram ίση με το όρισμα
- setBloodGlucose(bloodGlucose: float): Η συνάρτηση αυτή θέτει τη μεταβλητή bloodGlucose ίση με το όρισμα
- setBloodOxygen(bloodOxygen: float): Η συνάρτηση αυτή θέτει τη μεταβλητή bloodOxygen ίση με το όρισμα
- setSkinConductivity(skinConductivity: float): Η συνάρτηση αυτή θέτει τη μεταβλητή skinConductivity ίση με το όρισμα
- setPatientID(patientID: String): Η συνάρτηση αυτή θέτει τη μεταβλητή patientID ίση με το όρισμα
- attach(): Μέθοδος προσκόλλησης του entity στον observer του
- notifyAllObservers(): Μέθοδος ειδοποίησης των observer για την κατάσταση του Entity

Abstract Observer



Ο αφηρημένος observer του entity BiomedicalDataEntity.

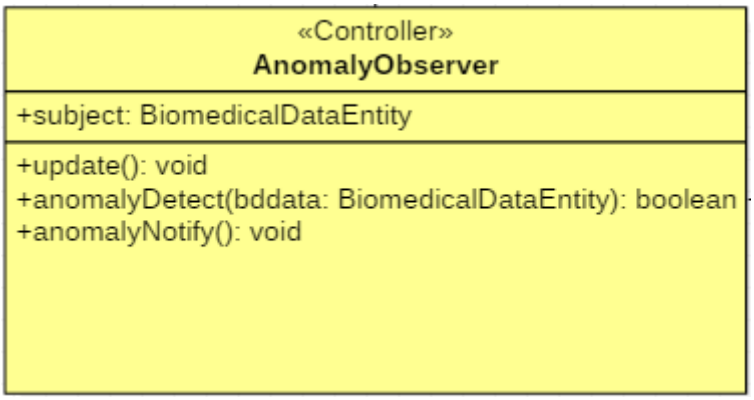
Χαρακτηριστικά της κλάσης:

- subject: Το entity το οποίο παρατηρεί ο observer

Μέθοδοι της κλάσης:

- update(): Η μέθοδος που καλείται για να ενημερωθεί ο observer. Αυτή είναι η abstract υλοποίηση

Controller AnomalyObserver



Η κλάση αυτή είναι ο concrete observer της οντότητας BiomedicalDataEntity, για την ανίχνευση ανωμαλίων που μπορεί να προκύψουν.

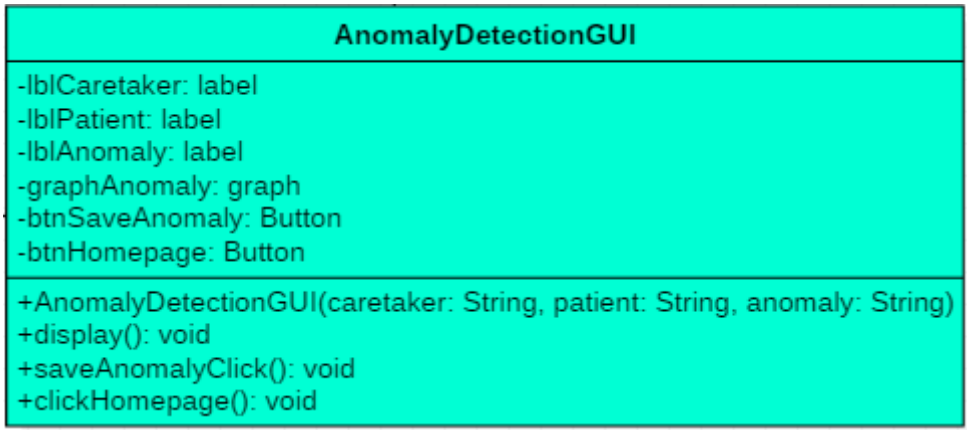
Χαρακτηριστικά της κλάσης:

- subject: Το entity το οποίο παρατηρεί ο observer

Μέθοδοι της κλάσης:

- update(): Η μέθοδος που καλείται για να ενημερωθεί ο observer. Αυτή είναι η πρακτική υλοποίηση της μεθόδου για τη συγκεκριμένη περίπτωση
- anomalyDetect(bddata: BiomedicalDataEntity): Η μέθοδος αυτή ελέγχει τα BiomedicalData και ανιχνεύει τυχόν αποκλίσεις από τις φυσιολογικές τιμές των δεδομένων
- anomalyNotify(): Η μέθοδος αυτή ειδοποιεί τον caretaker αν ανιχνευθεί κάποια ανωμαλία, μέσω του AnomalyDetectionGUI

Boundary AnomalyDetectionGUI



Η κλάση αυτή εκφράζει την οθόνη ειδοποίησης του caretaker για κάποια ανωμαλία στα βιοϊατρικά δεδομένα του ασθενούς.

Χαρακτηριστικά της κλάσης:

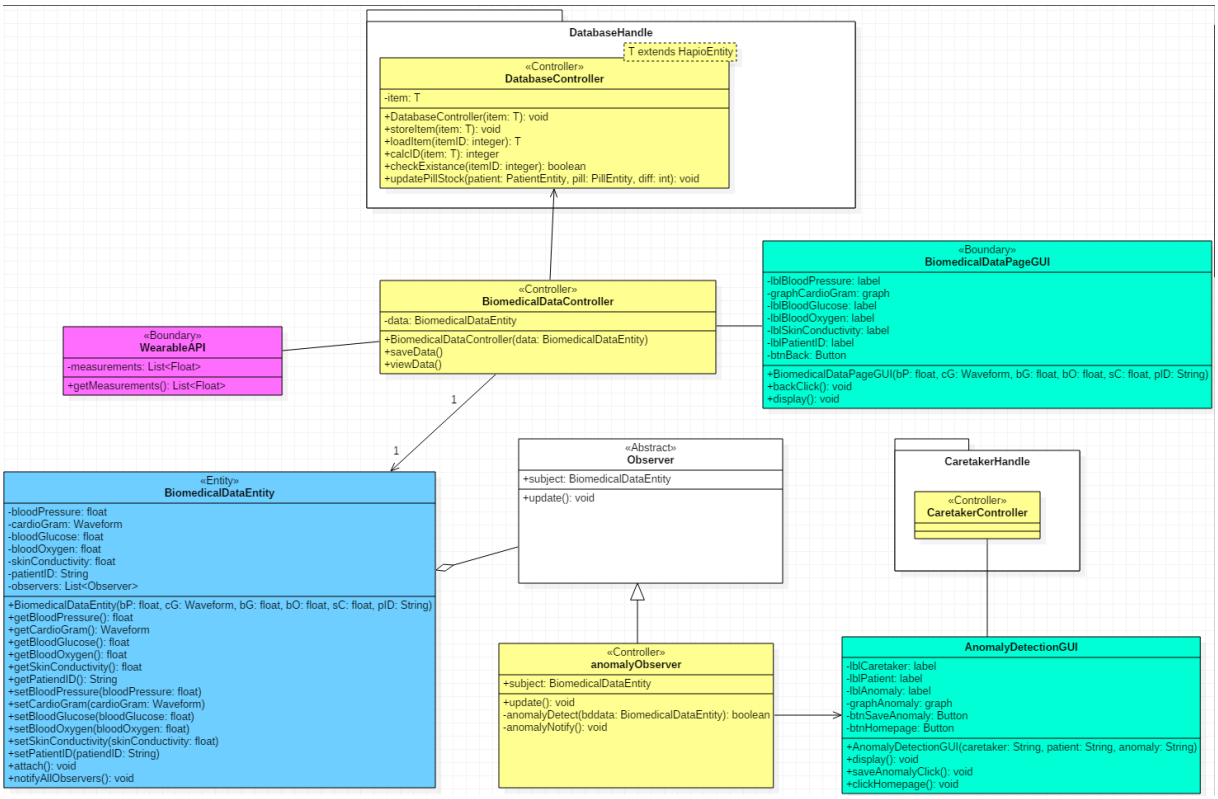
- lblCaretaker: Επιγραφή με το όνομα του Caretaker
- lblPatient: Επιγραφή με το όνομα του ασθενούς
- lblAnomaly: Επιγραφή με το όνομα της ανωμαλίας, δηλαδή από ποιά από τις επιμέρους μετρήσεις προέκυψε ανωμαλία
- graphAnomaly: Γράφημα που οπτικοποιεί την ανωμαλία σε σχέση με προηγούμενες, φυσιολογικές μετρήσεις
- btnSaveAnomaly: Κουμπί που πατάει ο caretaker σε περίπτωση που θέλει να αποθηκεύσει log αρχείο με την ανωμαλία, για μεταγενέστερη μελέτη από τον ιατρό του ασθενούς
- btnHomepage: Κουμπί που πατάει ο caretaker για να επιστρέψει στην αρχική οθόνη της εφαρμογής

Μέθοδοι της κλάσης:

- AnomalyDetectionGUI(caretaker:String,patient:String,anomaly:String): Constructor της κλάσης AnomalyDetectionGUI
- display(): Μέθοδος που καλείται για την εμφάνιση της οθόνης ειδοποίησης για ανωμαλία
- saveAnomalyClick(): Μέθοδος που καλείται όταν ο caretaker πατήσει το κουμπί αποθήκευσης ανωμαλίας
- clickHomepage(): Μέθοδος που καλείται όταν ο caretaker πατήσει το κουμπί επιστροφής στην αρχική οθόνη

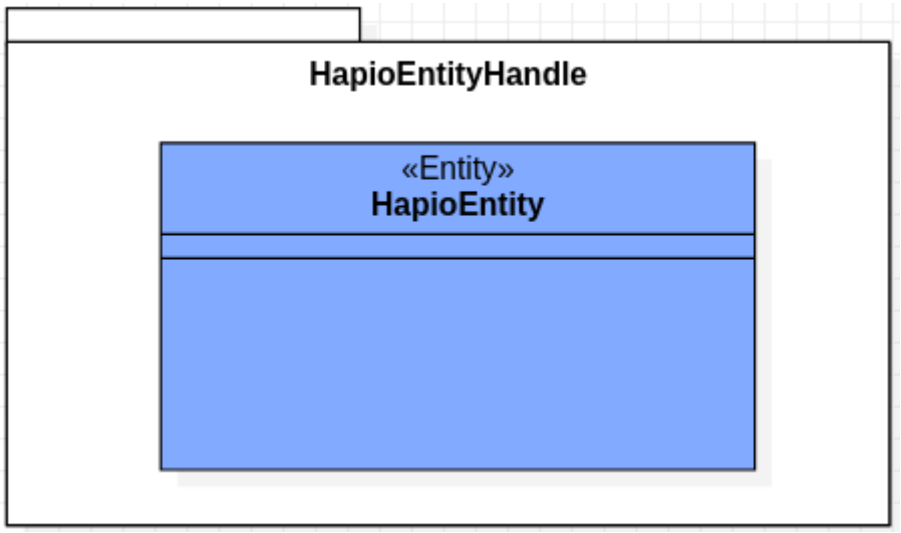
Διάγραμμα Κλάσεων

Σημείωση: Η δεικτοδότηση σε βέλη για ένα προς ένα συσχετίσεις μεταξύ κλάσεων παραλείπεται.

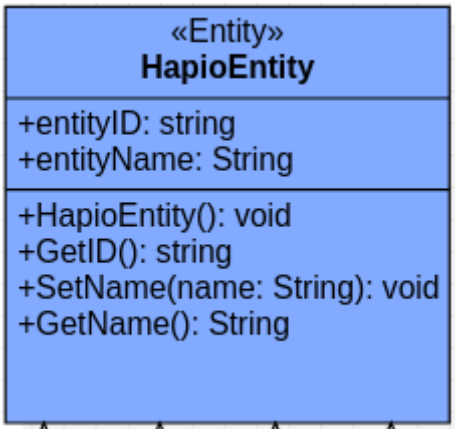


1.3.2 Πακέτο HapioEntityHandle

Το πακέτο αυτό εξυπηρετεί τον σκοπό της γενίκευσης των entities ώστε σε κάποια σημεία να μπορούν να ομαδοποιηθούν και να χρησιμοποιηθούν γενικές μέθοδοι για την διαχείριση τους, όπως για παράδειγμα στο DatabaseHandle.



Entity HapioEntity



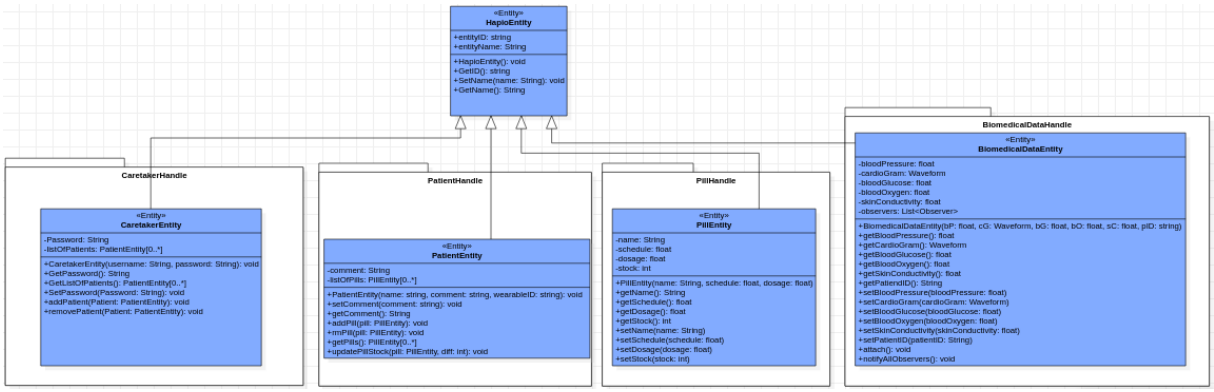
Χαρακτηριστικά της κλάσης:

- entityID: Το χαρακτηριστικό ID του entity.
- entityName: Το όνομα του entity.

Μέθοδοι της κλάσης:

- HapioEntity(): Ο constructor της κλασης HapioEntity. Ο constructor ορίζει και το ID του entity οπότε και δεν υπάρχει setID().
- GetID(): Η συνάρτηση αυτή επιστρέφει το ID του αντίστιχου entity.
- SetName(name: String): Η συνάρτηση αυτή θέτει τη μεταβλητή entityName ίση με το όρισμα.
- GetName(): Η συνάρτηση αυτή επιστρέφει το όνομα του αντίστοιχου entity.

Διάγραμμα κλάσεων



2 Μη λειτουργικές απαιτήσεις

2.1 Απαιτήσεις επίδοσης

<ΜΛΑ-3>

Το σύστημα πρέπει να ανιχνεύει ανωμαλίες στα biomedical data του ασθενούς εντός 30 δευτερολέπτων από την εμφάνιση τους.

Περιγραφή:

Το σύστημα πρέπει να ανιχνεύει άμεσα αν υπάρχει κάποια ανωμαλία, εφόσον υπάρχει περίπτωση να κινδυνεύει η υγεία του ασθενούς.

User Priority (5/5): Ο caretaker θα θέλει να ενημερώνεται όσο πιο άμεσα γίνεται αν υπάρχει κάποια ανωμαλία. Επομένως έχει υψηλή προτεραιότητα.

Technical Priority (5/5): Η απαίτηση είναι πολύ σημαντική για την αξιοπιστία του συστήματος. Η λειτουργία χάνει το νόημα της όταν η ανταπόκριση στα biomedical data είναι τόσο αργή που δεν επιτρέπει άμεσα τον caretaker να δράσει.

Stability (4/5): Με την βελτίωση της τεχνολογίας, ίσως υπάρχει δυνατότητα περαιτέρω μείωσης του χρόνου ανίχνευσης ανωμαλιών. Επιπλέον, με την λήψη διαγνωστικών δεδομένων, μπορεί ίσως να αποδειχθεί ότι δεν απαιτείται τόσο αυστηρή ταχύτητα απόκρισης.

2.2 Απαιτήσεις ασφάλειας (Security)

<ΜΛΑ-1>

Το σύστημα πρέπει να τηρεί τις αρχές που διέπει τον κανονισμό GDPR (General Data Protection Regulation).

Περιγραφή:

Το σύστημα πρέπει να προστατεύει στοιχεία όπως το όνομα, επίθετο, ιατρικές ασθένειες και biomedical data του ασθενή από πιθανές απόπειρες hacking και υποκλοπές στοιχείων συμφωνα με το GDPR. Επιπλέον πρέπει να υπάρχει διαφάνεια προς τους χρήστες ως προς το τι δεδομένα συλλέγει το σύστημα, πως τα χρησιμοποιεί, ποιός έχει πρόσβαση σε αυτά και πόσο καιρό παραμένουν στο σύστημα, καθώς επίσης και να ζητείται η ρητή συγκατάθεση των χρηστών σε όλα τα παραπάνω.

User Priority (5/5): Η προστασία των προσωπικών δεδομένων του ασθενή είναι υψίστης σημασίας ειδικά σε ιατρικά ζητήματα. Η τήρηση του GDPR είναι απολύτως πρώτη προτεραιότητα.

Technical Priority (5/5): Το σύστημα είναι απαγορευτικό να λειτουργεί χωρίς τις κατάλληλες προφυλάξεις για τα δεδομένα των χρηστών. Το σύστημα, εφόσον πρόκειται για ιατρική εφαρμογή, πρέπει να σχεδιαστεί εξαρχής ώστε να τηρεί τον κανονισμό GDPR και είναι πρώτη προτεραιότητα.

Stability (4/5): Υπάρχει η περίπτωση να γίνουν αλλαγές στον GDPR παρόλο ότι είναι πανευρωπαϊκός κανονισμός. Παρόλαυτα, θεωρούμε ότι ο κανονισμός θα παραμείνει σημαντικός για πολύ καιρό.

2.3 Απαιτήσεις Χρηστικότητας (Usability)

<ΜΛΑ-5>

Οι ασθενείς πρέπει να είναι σε θέση να χρησιμοποιούν το σύστημα μετά από συνολική εκπαίδευση μισής ώρας. Μετά από αυτή την εκπαίδευση, ο μέσος αριθμός των λαθών που διαπράττονται από ασθενείς θα πρέπει να μην υπερβαίνει το ένα ημερησίως.

Περιγραφή:

Για ευκολότερη χρήση, η μοναδική ενέργεια που θα έχει να κάνει ο ασθενής θα είναι η επιβεβαίωση της λήψης του φαρμάκου. Επιπλέον, ακόμα και η επιβεβαίωση της λήψης του φαρμάκου θα πρέπει να είναι σχεδιασμένη ώστε να είναι πολύ φιλική προς τους ασθενείς, δηλαδή θα πρέπει να γίνεται με το πάτημα ενός κουμπιού και οι ενδείξεις στην οθόνη να είναι απόλυτα κατανοητές. Λάθος θεωρείται η εσφαλμένη επιβεβαίωση ή μη της λήψης ενος φαρμάκου.

User Priority (5/5): Ο ασθενής θεωρείται ότι θα είναι μεγαλύτερης ηλικίας και ψηφιακά αναλφάβητος. Για να προτιμήσει ο ασθενής το σύστημα μας από τον τρόπο που διαχειρίζονταν τα φάρμακα του προηγουμένως, θα πρέπει το σύστημα να είναι πολύ εύκολο στην χρήση. Επομένως, θεωρείται ζωτικής σημασίας η ευκολία στην χρήση από την πλευρά των ασθενών.

Technical Priority (5/5): Θα πρέπει εξαρχής να γίνει σχεδιασμός του συστήματος ώστε ο ασθενής να διαθέτει μόνο μία ενέργεια και τις περισσότερες λειτουργίες να τις εκτελεί ο caretaker. Επομένως έχει μεγάλη προτεραιότητα.

Stability (4/5): Θα πρέπει να κρίνουμε από το user feedback κατά πόσο χρειάζονται αλλαγές σε αυτό το κομμάτι. Ίσως υπάρχει δυνατότητα να βελτιωθεί περαιτέρω η ευκολία χρήσης της εφαρμογής.

2.4 Απαιτήσεις Φορητότητας (Portability)

<ΜΛΑ-2>

Το σύστημα πρέπει να τρέχει σε mobile λειτουργικά iOS και Android, καθώς και σε websites.

Περιγραφή:

Το σύστημα πρέπει να τρέχει στις περισσότερες δημοφιλείς πλατφόρμες κινητών αλλα και υπολογιστών, ώστε να μπορεί να το χρησιμοποιεί ο μέγιστος αριθμός χρηστών.

User Priority (4/5): Είναι σημαντικό για το χρήστη να μπορεί να χρησιμοποιήσει την εφαρμογή, ανεξαρτήτως της πλατφόρμας του.

Technical Priority (4/5): Η μεγιστοποίηση του αριθμού των χρηστών είναι πολύ σημαντική για το σύστημα. Παρόλα Αυτά, μπορεί να προστεθεί υποστήριξη για κάποια από τις πλατφόρμες σε αργότερη φάση οπότε δεν είναι υψίστης σημασίας.

Stability (3/5): Υπάρχει η πιθανότητα ανάπτυξης καινούργιων δημοφιλών λογισμικών και η εφαρμογή θα πρέπει να τρέχει και σε αυτά.

2.5 Τεχνικές Απαιτήσεις περιβάλλοντος

<ΜΛΑ-4>

Το πρόγραμμα που θα τρέχει στο wearable θα πρέπει να απαιτεί λιγότερο από 1MB μνήμης προγράμματος και τα δεδομένα που αποθηκεύει να απαιτούν λιγότερο από 128KB μνήμης.

Περιγραφή:

Το σύστημα πρέπει κυρίως να τρέχει σε wearable, το οποίο ως ενσωματωμένο σύστημα θα έχει μικρή μνήμη. Επομένως, πρέπει να δοθεί προσοχή ώστε το τελικό πρόγραμμα να χωράει στην μνήμη του wearable και να είναι οικονομικό στην μνήμη δεδομένων.

User Priority (1/5): Ο ασθενής δεν επηρεάζεται με κανέναν τρόπο από τις απαιτήσεις μνήμης του συστήματος στο wearable. Επομένως δεν έχει προτεραιότητα για τον ασθενή.

Technical Priority (5/5): Το κεντρικό σημείο του συστήματος είναι η λειτουργία του στο wearable και η επίβλεψη του ασθενούς. Αν δεν ικανοποιεί τις απαιτήσεις μνήμης του wearable δεν θα μπορεί να χρησιμοποιηθεί εξαρχής. Επομένως, είναι υψίστης σημασίας αυτή η απαίτηση από τεχνική σκοπιά.

Stability (4/5): Με την βελτίωση της τεχνολογίας και την αύξηση των δεδομένων, ίσως υπάρχει δυνατότητα η απαίτηση μνήμης να αυξηθεί περαιτέρω.

3 Πρότυπα Σχεδιασμού που υιοθετήθηκαν

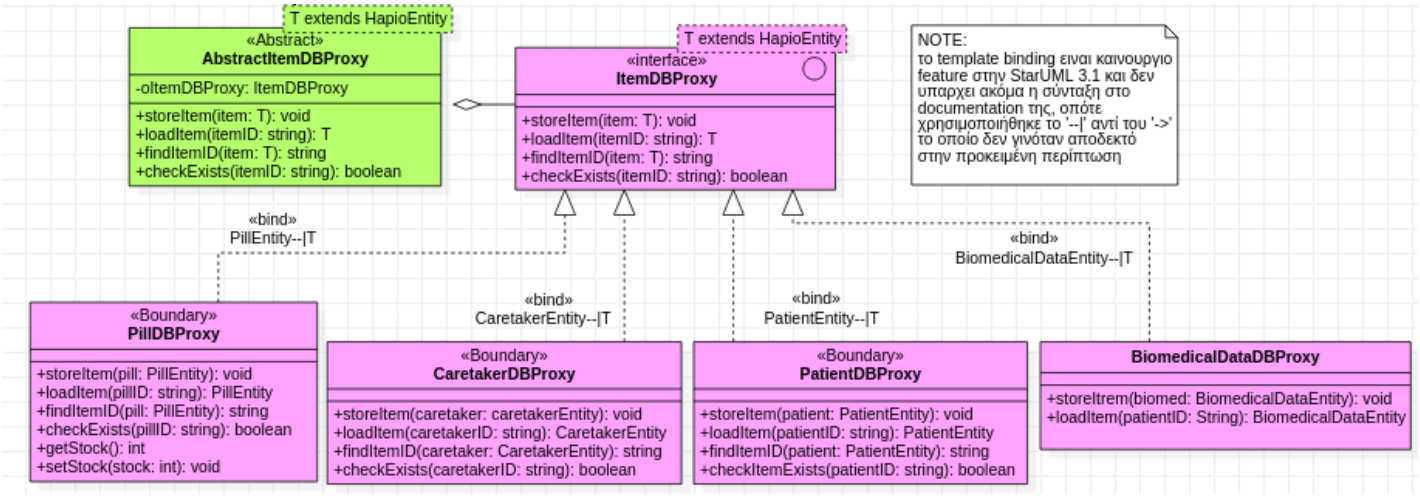
3.1 Δομικά πρότυπα

Τα δομικά πρότυπα εμπεριέχουν σύνθετες δομές, εισάγουν μια abstract κλάση για μελλοντικές επεκτάσεις του συστήματος και επιτυγχάνουν τη μείωση της σύζευξης ανάμεσα σε κλάσεις.

Bridge Design Pattern

Το πρότυπο του γέφυρας επιτρέπει την αποσύζευξη μιας αφαίρεσης από την υλοποίησή της, ώστε να μπορούν να υπάρχουν ανεξάρτητα. Επομένως, χρησιμοποιείται για να παρέχει πολλές υλοποιήσεις κάτω από την ίδια διεπαφή και εφαρμόζεται εκ των προτέρων για να διαχωρίσει τις αφαιρέσεις από τις υλοποιήσεις. Κατ' αυτό τον τρόπο, το σύστημά μας καθίσταται συμβατό με εξωτερικά συστήματα.

- Πρόβλημα που αντιμετωπίστηκε: Το πρότυπο αυτό χρησιμοποιήθηκε για να σχεδιαστεί η διεπαφή του συστήματός μας με το εξωτερικό σύστημα των βάσεων δεδομένων, το οποίο χρησιμοποιείται για την αποθήκευση, φόρτωση, εύρεση και συγχρονισμό των δεδομένων μεταξύ caretaker και ασθενών. Ο λόγος που υιοθετήθηκε το πρότυπο αυτό ήταν για απλοποίηση του συστήματος, ώστε να μπορούν όλοι οι controllers να χρησιμοποιούν το ίδιο σημείο εισόδου για διεπαφή με τα κατάλληλα database.



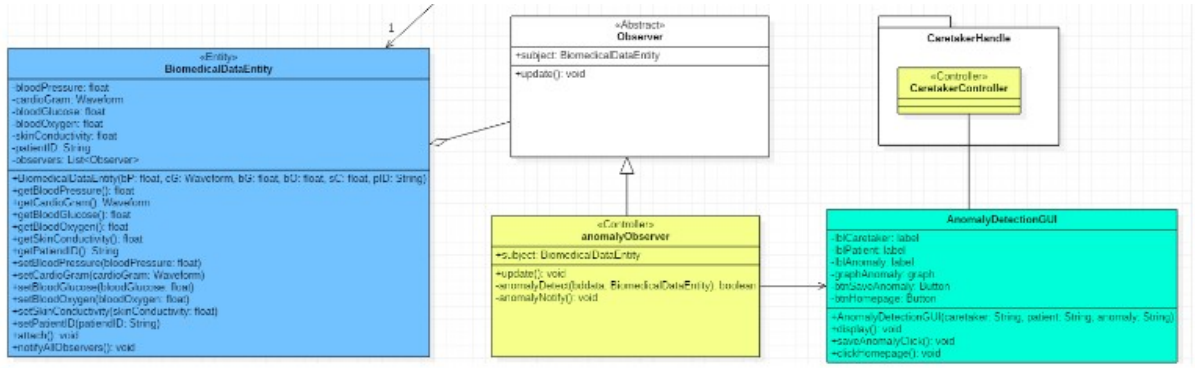
3.2 Πρότυπα Συμπεριφοράς

Τα πρότυπα συμπεριφοράς επιτρέπουν επιλογή ανάμεσα σε αλγορίθμους και ορισμό ευθυνών ανάμεσα σε αντικείμενα. Επιπροσθέτως, αποφεύγουν τη στενή σύζευξη σε συγκεκριμένες λύσεις και χαρακτηρίζουν σύνθετες ροές ελέγχου, οι οποίες είναι δύσκολο να παρακολουθηθούν κατά την εκτέλεση. Κατ' αυτόν τον τρόπο βελτιστοποιείται η ροή πληροφοριών μεταξύ των υποσυστημάτων και επιτυγχάνεται ένα ποσοστό απόζευξης μεταξύ τους.

Observer Pattern

Το παρόν πρότυπο χρησιμοποιείται από τον ανιχνευτή ανωμαλιών στα biomedical data του ασθενούς. Είναι πολύ εύκολο να καταλάβει κάποιος την χρησιμότητα καθώς πρέπει το σύστημα διαρκώς να παρατηρεί αλλαγές στα biomedical data και να ειδοποιήσει τον caretaker σε περίπτωση που χρειαστεί.

- <u>Πρόβλημα που αντιμετωπίστηκε:</u> Ήταν κρίσιμο για το σύστημα να μπορεί να παρακολουθεί αυτόματα την εξέλιξη των biomedical data του ασθενούς, χωρίς ρητή εντολή από κάποιον γενικότερο controller ώστε η σύζευξη να μην είναι με τον controller αλλά με τα ίδια τα δεδομένα. Το design pattern αυτό μας δίνει μια πολύ βολική και εύκολη λύση σε αυτό.



Παράρτημα Ι – Πίνακας Ιχνηλασιμότητας

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

Επομένως, δεν ορίζουμε κάποιον πίνακα ιχνηλασιμότητας.

Παράρτημα ΙΙ – Ανοιχτά Θέματα

- Οι συναρτήσεις οι οποίες έχουν δημιουργηθεί περιγράφουν τις βασικές λειτουργικότητες που εντοπίστηκαν από την ομάδα ανάπτυξης. Είναι πιθανό κατά την υλοποίηση του συστήματος να εντοπιστούν επιπλέον συναρτήσεις απαραίτητες για την ορθή λειτουργία του συστήματος.
- Οι μεταβλητές οι οποίες έχουν αναφέρονται στα διαγραμμάτα κλάσεων είναι ενδεικτικές. Πιθανόν να χρειαστεί να αλλάξει ο τύπος τους ή να προστεθούν νέες.
- Χρειάζεται να δοθεί μεγαλύτερη έμφαση στην διπλή επιβεβαίωση της λήψης ενός χαπιού μεταξύ caretaker και ασθενή. Η παραπάνω λειτουργία είναι ένα προσχέδιο της πραγματικής.
- Οι πολλαπλότητες στα σχεδιαγράμματα οι οποίες είναι 1 προς 1 παραλείπονται.