



/gnosis

Giving a piece of advice for your health.

Προδιαγραφές Συστήματος

Version 1.0

28/4/2024

Θεόδωρος Βλαχάκης vlathegeo@ece.auth.gr

Κωνσταντίνος Καλαμάρας kkalamar@ece.auth.gr

Κωνσταντίνος Φώτιος Παπαδάκης krapadak@ece.auth.gr

Νικόλαος Ευαγγελίδης nievange@ece.auth.gr

Μέλη της Ομάδας Ανάπτυξης

Όνομα	ΟΑ	Email
Α. Συμεωνίδης	*	asymeon@issel.ee.auth.gr
Θεόδωρος Βλαχάκης	50	vlathegeo@ece.auth.gr
Κωνσταντίνος Καλαμάρας	50	kkalamar@ece.auth.gr
Κωνσταντίνος Φώτιος Παπαδάκης	50	krapadak@ece.auth.gr
Νικόλαος Ευαγγελίδης	50	nievange@ece.auth.gr

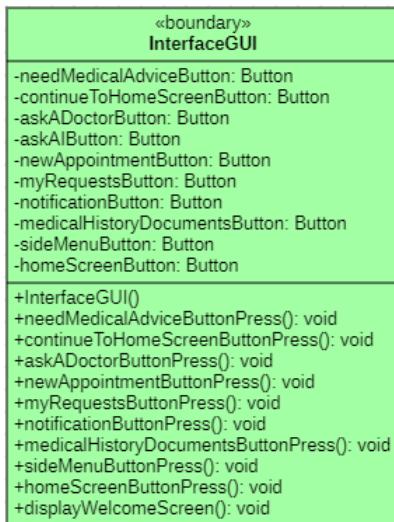
Πίνακας Περιεχομένων

Πίνακας Περιεχομένων.....	3
1 Στατική Μοντελοποίηση.....	5
1.0 <InterfaceGUI>.....	5
1.1 <Πακέτα κλάσεων σχετικών με το Gherkin Feature (Σενάριο Χρήσης) 1: "Patient asks for medical advice">.....	7
1.1.1 Πακέτο HealthCondition.....	7
1.1.1.1 <FillHealthConditionGUI>.....	7
1.1.1.2 <SuccessfulSubmissionDialog>.....	8
1.1.1.3 <UnsuccessfulSubmissionDialog>.....	9
1.1.1.4 <DescriptionCancellationConfirmationDialog>.....	10
1.1.1.5 <ImageSelectionScreen>.....	10
1.1.1.6 <MedHistoryDocSelectionScreen>.....	11
1.1.1.7 <MedicalHistoryDocumentGUI>.....	11
1.1.1.8 <MedicalHistoryDocument>.....	12
1.1.1.9 <HealthRequest>.....	13
1.1.1.10 <Patient>.....	15
1.1.1.11 <MedicalHistoryDocumentController>.....	17
1.1.1.12 <HealthRequestController>.....	18
1.1.1.13 Διάγραμμα κλάσεων.....	20
1.1.2 Πακέτο ChooseDoctor.....	20
1.1.2.1 <ChooseDoctorController>.....	20
1.1.2.2 <ChooseDoctorGUI>.....	22
1.1.2.3 <DoctorConfirmationDialog>.....	23
1.1.2.4 Διάγραμμα κλάσεων.....	24
1.2 <Πακέτα κλάσεων σχετικών με το Gherkin Feature (Σενάριο Χρήσης) 4: "Patient receives a medical diagnosis">.....	25
1.2.1 HealthRequestList Package.....	25
1.2.1.1 <Notification>.....	25
1.2.1.2 <MyHealthRequestList>.....	26
1.2.1.3 <NotificationGUI>.....	27
1.2.1.4 <MyHealthRequestListGUI>.....	27
1.2.1.5 <HealthRequestGUI>.....	28
1.2.1.6 <MyHealthRequestListController>.....	28
1.2.1.7 Class Diagram.....	30
1.2.2 Database Package.....	30
1.2.2.1 <Database>.....	30
1.2.2.2 <DatabaseProxy>.....	31
1.2.2.3 Class Diagram.....	32
1.3 <Πακέτα κλάσεων σχετικών με το Gherkin Feature (Σενάριο Χρήσης) 3: "Patient requests medical appointment">.....	32
1.3.1 Appointment package.....	32
1.3.1.1 <Appointment>.....	33

1.3.1.2 <BookAppointmentGUI>.....	34
1.3.1.3 <AppointmentController>.....	34
1.4 <Πακέτα κλάσεων σχετικών με το Gherkin Feature (Σενάριο Χρήσης) 6: “Doctor gives advice to patient ”>.....	36
1.3.1 Πακέτο Medical Advice.....	36
1.4.1.1 <Doctor>.....	36
1.4.1.2 <Diagnosis>.....	37
1.4.1.3 <DiagnosisController>.....	38
1.4.1.4 <FillDiagnosisGUI>.....	38
1.4.1.5 Διάγραμμα κλάσεων.....	39
2 Πρότυπα Σχεδιασμού που υιοθετήθηκαν.....	39
2.1 Πρότυπο Proxy.....	39
2.2 Πρότυπο Adapter.....	40
2.3 Πρότυπο Balking.....	41
3 Δυναμική Μοντελοποίηση.....	42
3.1 Gherkin Scenario 1 “Successfully describe health condition” (Feature 1): “Patient describes health condition”.....	42
3.1.1 Αφήγηση Σεναρίου.....	42
3.1.2 Sequence Diagram.....	44
3.2 Gherkin Scenario 1 (Feature 4)”Patient receives a medical diagnosis”.....	44
3.2.1 Αφήγηση Σεναρίου.....	44
3.2.2 Sequence Diagram.....	45
3.3 Gherkin Scenario 3 (Feature 6) : ”Doctor gives advice to patient based on the Medical History Document.”	45
3.3.1 Αφήγηση Σεναρίου.....	45
3.3.2 Sequence Diagram.....	46

1 Στατική Μοντελοποίηση

1.0 <InterfaceGUI>



Η κλάση αυτή διαχειρίζεται τις οθόνες μενού οι οποίες προβάλλουν στον χρήστη όλες τις δυνατές επιλογές του εντός της εφαρμογής. Τοποθετείται εδώ γιατί είναι πολύ γενικό και χρησιμοποιείται από πολλά features.

Χαρακτηριστικά Κλάσης

- `needMedicalAdviceButton: Button`
- `continueToHomeScreenButton: Button`
- `askADoctorButton: Button`
- `askAIButton: Button`
- `newAppointmentButton: Button`
- `myRequestsButton: Button`
- `notificationButton: Button`
- `medicalHistoryDocumentsButton: Button`
- `sideMenuButton: Button`
- `homeScreenButton: Button`

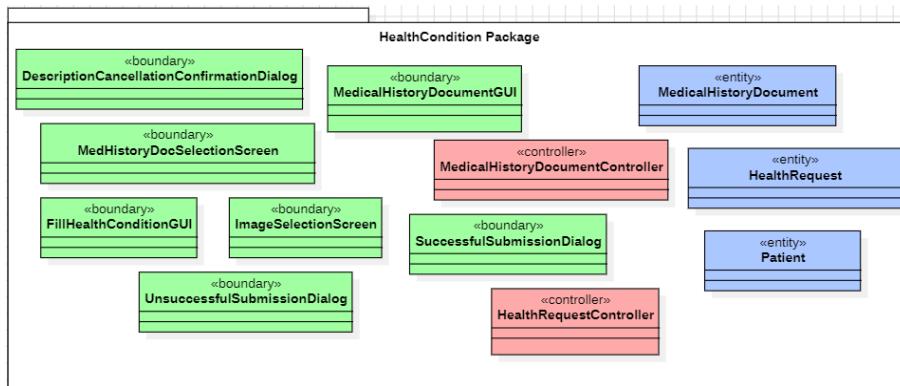
Μέθοδοι Κλάσης

- `InterfaceGUI()`: O constructor της κλάσης.
- `needMedicalAdviceButtonPress()`: Ελέγχει αν έχει πατηθεί το κουμπί `needMedicalAdviceButton`.
- `continueToHomeScreenButtonPress()`: Ελέγχει αν έχει πατηθεί το κουμπί `continueToHomeScreenButton`.
- `askADoctorButtonPress()`: Ελέγχει αν έχει πατηθεί το κουμπί `askADoctorButton`.
- `askAIButtonPress()`: Ελέγχει αν έχει πατηθεί το κουμπί `askAIButton`.
- `newAppointmentButtonPress()`: Ελέγχει αν έχει πατηθεί το κουμπί `newAppointmentButton`.
- `myRequestsButtonPress()`: Ελέγχει αν έχει πατηθεί το κουμπί `myRequestsButton`.
- `notificationButtonPress()`: Ελέγχει αν έχει πατηθεί το κουμπί `notificationButton`.

- medicalHistoryDocumentsButtonPress(): Ελέγχει αν έχει πατηθεί το κουμπί medicalHistoryDocumentsButton.
- sideMenuButtonPress(): Ελέγχει αν έχει πατηθεί το κουμπί sideMenuButton.
- homeScreenButtonPress(): Ελέγχει αν έχει πατηθεί το κουμπί homeScreenButton.
- displayWelcomeScreen(): Απεικονίζει την οθόνη που καλωσορίζει τον χρήστη στην εφαρμογή.
- displayQuickHelpScreen(): Απεικονίζει την οθόνη γρήγορης εξυπηρέτησης του ασθενή δίντας του δύο επιλογές, ask AI και ask a Doctor.
- displayHomeScreen(): Απεικονίζει την οθόνη μενού η οποία λειτουργεί ως σημείο αναφοράς για όλες τις λειτουργίες της εφαρμογής.

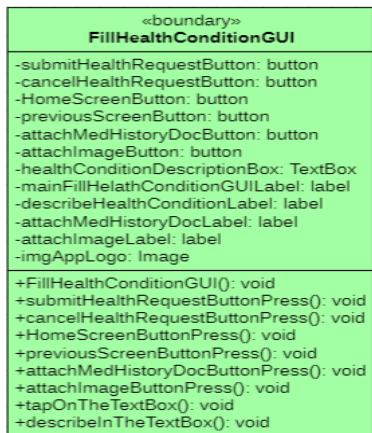
1.1 <Πακέτα κλάσεων σχετικών με το Gherkin Feature (Σενάριο Χρήσης) 1: “Patient asks for medical advice”>

1.1.1 Πακέτο HealthCondition



Το πακέτο περιλαμβάνει τις κλάσεις που σχετίζονται με την δημιουργία ιατρικού αιτήματος σχετικού με την κατάσταση της υγείας του ασθενή.

1.1.1.1 <FillHealthConditionGUI>



Η κλάση αυτή εκφράζει τη διεπαφή της οθόνης περιγραφής της κατάστασης της υγείας/ιατρικού ζητήματος του χρήστη “ασθενή”.

Χαρακτηριστικά Κλάσης

- submitHealthRequestButton: button – Κουμπί υποβολής της κατάστασης της υγείας του χρήστη “ασθενή”.
- cancelHealthRequestButton: button – Κουμπί ακύρωσης και διαγραφής της φόρμας υποβολής της κατάστασης της υγείας του χρήστη “ασθενή”.
- HomeScreenButton: button – Κουμπί μετάβασης στην αρχική οθόνη της εφαρμογής.
- previousScreenButton: button – Κουμπί επιστροφής στην προηγούμενη οθόνη της εφαρμογής.
- attachMedHistoryDocButton: button – Κουμπί επισύναψης εγγράφου ιατρικού ιστορικού.
- attachImageButton: button – Κουμπί επισύναψης εικόνας που συνοδεύει την περιγραφή της κατάστασης της υγείας.

- healthConditionDescriptionBox: TextBox – Πλαίσιο κειμένου για εισαγωγή κειμένου περιγραφής της κατάστασης της υγείας του χρήστη “ασθενή”.
- mainFillHealthConditionGUILabel: label – Επιγραφή ερώτησης προς τον χρήστη “ασθενή” για το ιατρικό του ζήτημα.
- describeHealthConditionLabel: label – Επιγραφή ένδειξης του πλαισίου κειμένου περιγραφής.
- attachMedHistoryDocLabel: label – Επιγραφή ένδειξης της διεπαφής επισύναψης αρχείου ιατρικού ιστορικού.
- attachImageLabel: label – Επιγραφή ένδειξης της διεπαφής επισύναψης εικόνας.
- imgAppLogo: Image – Εικόνα λογοτύπου της εφαρμογής.

Μέθοδοι Κλάσης

- FillHealthConditionGUI(): void – Η μέθοδος FillHealthConditionGUI αποτελεί τη συνάρτηση δόμησης της διεπαφής αυτής.
- submitHealthRequestButtonPress(): void – Η μέθοδος submitHealthRequestButtonPress καλεί τη συνάρτηση createHealthRequest του ελεγκτή HealthRequestController και οδηγεί στον έλεγχο έγκυρης περιγραφής της κατάστασης της υγείας προκειμένου ο ελεγκτής να καλέσει τη συνάρτηση DisplaySuccessfulSubmissionDialog ή τη συνάρτηση DisplayUnsuccessfulSubmissionDialog για προβολή του αντίστοιχου παραθύρου διαλόγου. Εκτελείται όταν ο χρήστης “ασθενής” πατάει το κουμπί submitHealthRequestButton.
- cancelHealthRequestButtonPress(): void – Η μέθοδος cancelHealthRequestButtonPress καλεί τη συνάρτηση DisplayDescriptionCancellationConfirmationDialog του ελεγκτή HealthRequestController για προβολή του αντίστοιχου παραθύρου διαλόγου.
- HomeScreenButtonPress(): void – Η μέθοδος HomeScreenButtonPress καλεί τη συνάρτηση DisplayHomeScreen του ελεγκτή HealthRequestController και οδηγεί στην προβολή της αρχικής οθόνης της εφαρμογής.
- previousScreenButtonPress(): void – Η μέθοδος previousScreenButtonPress καλεί τη συνάρτηση DisplayPreviousScreen του ελεγκτή HealthRequestController και οδηγεί στην προβολή της προηγούμενης οθόνης της εφαρμογής.
- attachMedHistoryDocButtonPress(): void – Η μέθοδος attachMedHistoryDocButtonPress καλεί τη συνάρτηση DisplayMedHistoryDocSelectionScreen του ελεγκτή HealthRequestController και οδηγεί στην προβολή της οθόνης επιλογής αρχείου.
- attachImageButtonPress(): void – Η μέθοδος attachImageButtonPress καλεί τη συνάρτηση DisplayImageSelectionScreen του ελεγκτή HealthRequestController και οδηγεί στην προβολή της οθόνης επιλογής φωτογραφίας/εικόνας.
- tapOnTheTextBox(): void – Η μέθοδος tapOnTheTextBox καλεί τη συνάρτηση enableDescriptionInTheTextBox του ελεγκτή HealthRequestController για ενεργοποίηση της λειτουργίας εισαγωγής κειμένου από τον χρήστη “ασθενή” στο πλαίσιο κειμένου.
- describeInTheTextBox(): void – Η μέθοδος describeInTheTextBox καλεί τη μέθοδο describeInTheTextBox του ελεγκτή HealthRequestController για δημιουργία κειμένου περιγραφής.

1.1.1.2 <SuccessfulSubmissionDialog>

SuccessfulSubmissionDialog

«boundary»
SuccessfulSubmissionDialog
-okButton: button
-successfulSubmissionMessageLabel: label
+SuccessfulSubmissionDialog(): void
+okButtonPress(): void

Η κλάση αυτή εκφράζει τη διεπαφή του παραθύρου διαλόγου επιτυχούς υποβολής της κατάστασης της υγείας από το χρήστη “ασθενή” στο σύστημα.

Χαρακτηριστικά Κλάσης

- okButton: button – Κουμπί επιβεβαίωσης επιτυχούς υποβολής της κατάστασης της υγείας.
- successfulSubmissionMessageLabel: label – Επιγραφή μηνύματος επιτυχούς υποβολής της κατάστασης της υγείας από το χρήστη “ασθενή”.

Μέθοδοι Κλάσης

- SuccessfulSubmissionDialog(): void – Η μέθοδος SuccessfulSubmissionDialog αποτελεί τη συνάρτηση δόμησης της διεπαφής αυτής.
- okButtonPress(): void – Η μέθοδος okButtonPress καλεί τη συνάρτηση closeSuccessfulSubmissionDialog του ελεγκτή HealthRequestController για διακοπή προβολής του παραθύρου διαλόγου. Εκτελείται όταν ο χρήστης “ασθενής” πατάει το κουμπί okButton.

1.1.1.3 <UnsuccessfulSubmissionDialog>

UnsuccessfulSubmissionDialog

«boundary»
UnsuccessfulSubmissionDialog
-okButton: button
-unsuccessfulSubmissionMessageLabel: label
+UnsuccessfulSubmissionDialog(): void
+okButtonPress(): void

Η κλάση αυτή εκφράζει τη διεπαφή του παραθύρου διαλόγου ανεπιτυχούς υποβολής της κατάστασης της υγείας από το χρήστη “ασθενή” στο σύστημα.

Χαρακτηριστικά Κλάσης

- okButton: button – Κουμπί επιβεβαίωσης ανεπιτυχούς υποβολής της κατάστασης της υγείας.
- unsuccessfulSubmissionMessageLabel: label – Επιγραφή μηνύματος ανεπιτυχούς υποβολής της κατάστασης της υγείας από το χρήστη “ασθενή”.

Μέθοδοι Κλάσης

- UnsuccessfulSubmissionDialog(): void – Η μέθοδος UnsuccessfulSubmissionDialog αποτελεί τη συνάρτηση δόμησης της διεπαφής αυτής.
- okButtonPress(): void – Η μέθοδος okButtonPress καλεί τη συνάρτηση closeUnsuccessfulSubmissionDialog του ελεγκτή HealthRequestController για διακοπή

προβολής του παραθύρου διαλόγου. Εκτελείται όταν ο χρήστης “ασθενής” πατάει το κουμπί okButton.

1.1.1.4 <DescriptionCancellationConfirmationDialog>

DescriptionCancellationConfirmationDialog

«boundary»	
DescriptionCancellationConfirmationDialog	
-	YesButton: button
-	NoButton: button
-	confirmationMessageLabel: label
+	DescriptionCancellationConfirmationDialog(): void
+	YesButtonPress(): void
+	NoButtonPress(): void

Η κλάση αυτή εκφράζει τη διεπαφή του παραθύρου διαλόγου επιβεβαίωσης ακύρωσης της φόρμας υποβολής της κατάστασης της υγείας από το χρήστη “ασθενή”.

Χαρακτηριστικά Κλάσης

- YesButton: button – Κουμπί επιβεβαίωσης ακύρωσης της φόρμας υποβολής της κατάστασης της υγείας.
- NoButton: button – Κουμπί άρνησης ακύρωσης της φόρμας υποβολής της κατάστασης της υγείας.
- confirmationMessageLabel: label – Επιγραφή μηνύματος επιβεβαίωσης της ακύρωσης της φόρμας υποβολής της κατάστασης της υγείας από το χρήστη “ασθενή”.

Μέθοδοι Κλάσης

- DescriptionCancellationConfirmationDialog(): void – Η μέθοδος DescriptionCancellationConfirmationDialog αποτελεί τη συνάρτηση δόμησης της διεπαφής αυτής.
- YesButtonPress(): void – Η μέθοδος YesButtonPress καλεί τη συνάρτηση cancelDescription του ελεγκτή HealthRequestController. Εκτελείται όταν ο χρήστης “ασθενής” πατάει το κουμπί YesButton.
- NoButtonPress(): void – Η μέθοδος NoButtonPress καλεί τη συνάρτηση noCancelDesription του ελεγκτή HealthRequestController. Εκτελείται όταν ο χρήστης “ασθενής” πατάει το κουμπί NoButton.

1.1.1.5 <ImageSelectionScreen>

ImageSelectionScreen

«boundary»	
ImageSelectionScreen	
-	imgList: ImageList
+	ImageSelectionScreen()
+	selectImage()

Η κλάση αυτή εκφράζει τη διεπαφή της οθόνης επιλογής εικόνας από το χρήστη “ασθενή” για εισαγωγή στη Βάση Δεδομένων του συστήματος.

Χαρακτηριστικά Κλάσης

- imgList: ImageList – Λίστα εικόνων με τις διαθέσιμες αποθηκευμένες εικόνες στη συσκευή του χρήστη “ασθενή”.

Μέθοδοι Κλάσης

- ImageSelectionScreen(): void – Η μέθοδος ImageSelectionScreen αποτελεί τη συνάρτηση δόμησης της διεπαφής αυτής.
- selectImage(): void – Η μέθοδος selectImage καλεί τη συνάρτηση selectImageToImport του ελεγκτή HealthRequestController. Εκτελείται όταν ο χρήστης “ασθενής” επιλέγει μια εικόνα από τη λίστα εικόνων.

1.1.1.6 <MedHistoryDocSelectionScreen>

MedHistoryDocSelectionScreen

«boundary»	
MedHistoryDocSelectionScreen	
-MedHistoryDocList: FileList	
+MedHistoryDocSelectionScreen()	
+selectMedHistoryDoc()	

Η κλάση αυτή εκφράζει τη διεπαφή της οθόνης επιλογής αρχείου από το χρήστη “ασθενή” για εισαγωγή στη Βάση Δεδομένων του συστήματος.

Χαρακτηριστικά Κλάσης

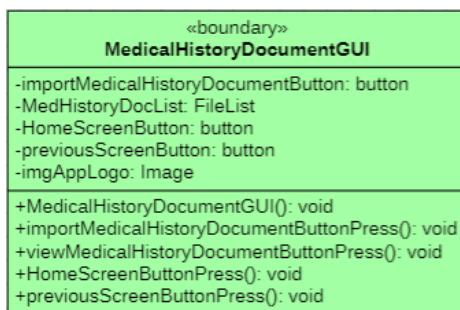
- MedHistoryDocList: FileList – Λίστα αρχείων με τα διαθέσιμα αποθηκευμένα αρχεία στη συσκευή του χρήστη “ασθενή”.

Μέθοδοι Κλάσης

- MedHistoryDocSelectionScreen(): void – Η μέθοδος MedHistoryDocSelectionScreen αποτελεί τη συνάρτηση δόμησης της διεπαφής αυτής.
- selectMedHistoryDoc(): void – Η μέθοδος selectMedHistoryDoc καλεί τη συνάρτηση selectMedHistoryDocToImport του ελεγκτή HealthRequestController, αν ο χρήστης “ασθενής” αλληλεπιδρά με τη διεπαφή της οθόνης επιλογής γιατρού προερχόμενος από τη διεπαφή νέας αίτησης για ιατρικό ζήτημα (request) της Welcome Screen ή της κεντρικής οθόνης της εφαρμογής, ή τη συνάρτηση createMedicalHistoryDocument του ελεγκτή MedicalHistoryDocumentController, αν ο χρήστης “ασθενής” αλληλεπιδρά με τη διεπαφή της οθόνης εγγράφων ιατρικού ιστορικού. Εκτελείται όταν ο χρήστης “ασθενής” επιλέγει ένα αρχείο από τη λίστα αρχείων.

1.1.1.7 <MedicalHistoryDocumentGUI>

MedicalHistoryDocumentGUI



Η κλάση αυτή εκφράζει τη διεπαφή της οθόνης εγγράφων ιατρικού ιστορικού σχετικών με το χρήστη “ασθενή”.

Χαρακτηριστικά Κλάσης

- `importMedicalHistoryDocumentButton: button` – Κουμπί εισαγωγής εγγράφου ιατρικού ιστορικού από το χρήστη “ασθενή”.
- `MedHistoryDocList: FileList` – Λίστα αποθηκευμένων στη Βάση Δεδομένων αρχείων/εγγράφων ιατρικού ιστορικού.
- `HomeScreenButton: button` – Κουμπί μετάβασης στην αρχική οθόνη της εφαρμογής.
- `previousScreenButton: button` – Κουμπί επιστροφής στην προηγούμενη οθόνη της εφαρμογής.
- `imgAppLogo: Image` – Εικόνα λογοτύπου της εφαρμογής.

Μέθοδοι Κλάσης

- `MedicalHistoryDocumentGUI(): void` – Η μέθοδος `MedicalHistoryDocumentGUI` αποτελεί τη συνάρτηση δόμησης της διεπαφής αυτής.
- `importMedicalHistoryDocumentButtonPress(): void` – Η μέθοδος `importMedicalHistoryDocumentButtonPress` καλεί τη συνάρτηση δημιουργίας νέου εγγράφου ιατρικού ιστορικού `createMedicalHistoryDocument` του ελεγκτή `MedicalHistoryDocumentController`. Εκτελείται όταν ο χρήστης “ασθενής” πατάει το κουμπί `importMedicalHistoryDocumentButton`.
- `viewMedicalHistoryDocumentButtonPress(): void` – Η μέθοδος `viewMedicalHistoryDocumentButtonPress` καλεί τη συνάρτηση προβολής του εγγράφου ιατρικού ιστορικού `viewMedicalHistoryDocument` του ελεγκτή `MedicalHistoryDocumentController`. Εκτελείται όταν ο χρήστης “ασθενής” επιλέγει ένα αρχείο από τη λίστα αρχείων που υπάρχουν στην οθόνη.
- `HomeScreenButtonPress(): void` – Η μέθοδος `HomeScreenButtonPress` καλεί τη συνάρτηση `DisplayHomeScreen` του ελεγκτή `MedicalHistoryDocumentController` και οδηγεί στην προβολή της αρχικής οθόνης της εφαρμογής.
- `previousScreenButtonPress(): void` – Η μέθοδος `previousScreenButtonPress` καλεί τη συνάρτηση `DisplayPreviousScreen` του ελεγκτή `MedicalHistoryDocumentController` και οδηγεί στην προβολή της προηγούμενης οθόνης της εφαρμογής.

1.1.1.8 <MedicalHistoryDocument>

MedicalHistoryDocument

«entity»	
MedicalHistoryDocument	
-MedHistoryDocID: integer	
-document: File	
-documentName: String	
-patientID: integer	
+MedicalHistoryDocument(): void	
+setMedHistoryDocID(MedHistoryDocID: integer): void	
+setDocument(document: File): void	
+setDocumentName(documentName: String): void	
+setPatientID(patientID: integer): void	
+getMedHistoryDocID(): integer	
+getDocument(): File	
+getDocumentName(): String	
+getPatientID(): integer	

Η κλάση αυτή περιέχει όλα τα χαρακτηριστικά/δεδομένα και τις συναρτήσεις/μεθόδους που αφορούν την οντότητα “έγγραφο ιατρικού ιστορικού” μέσα στο σύστημα.

Χαρακτηριστικά Κλάσης

- MedHistoryDocID: integer – Ο μοναδικός χαρακτηριστικός ακέραιος αριθμός του εγγράφου ιατρικού ιστορικού.
- document: File – Το αρχείο που περιέχει το πραγματικό (φυσικό) έγγραφο ιατρικού ιστορικού, που εισάγει ο “ασθενής” στο σύστημα.
- documentName: String – Το όνομα του εγγράφου ιατρικού ιστορικού.
- patientID: integer – Ο μοναδικός χαρακτηριστικός ακέραιος αριθμός του (χρήστη-) “ασθενή” της εφαρμογής, στον οποίο αφορά το έγγραφο ιατρικού ιστορικού.

Μέθοδοι Κλάσης

- MedicalHistoryDocument(): void – Η μέθοδος HealthRequest αποτελεί τη συνάρτηση δόμησης της οντότητας MedicalHistoryDocument.
- setMedHistoryDocID(MedHistoryDocID: integer): void – Η μέθοδος setMedHistoryDoc θέτει τον μοναδικό χαρακτηριστικό ακέραιο αριθμό του εγγράφου ιατρικού ιστορικού.
- setDocument(document: File): void – Η μέθοδος setDocument θέτει το αρχείο με το πραγματικό έγγραφο ιατρικού ιστορικού του “ασθενή”.
- setDocumentName(documentName: String): void – Η μέθοδος setDocumentName θέτει το όνομα του εγγράφου ιατρικού ιστορικού.
- setPatientID(patientID: integer): void – Η μέθοδος setPatientID θέτει το μοναδικό χαρακτηριστικό ακέραιο αριθμό του “ασθενή”, που εισάγει το έγγραφο ιατρικού ιστορικού στη Βάση Δεδομένων του συστήματος.
- getMedHistoryDocID(): integer – Η μέθοδος getMedHistoryDoc επιστρέφει τον μοναδικό χαρακτηριστικό ακέραιο αριθμό του εγγράφου ιατρικού ιστορικού.
- getDocument(): File – Η μέθοδος getDocument επιστρέφει το αρχείο με το πραγματικό έγγραφο ιατρικού ιστορικού του “ασθενή”.
- getDocumentName(): String – Η μέθοδος getDocumentName επιστρέφει το όνομα του εγγράφου ιατρικού ιστορικού.
- getPatientID(): integer – Η μέθοδος getPatientID επιστρέφει το μοναδικό χαρακτηριστικό ακέραιο αριθμό του “ασθενή”, που εισάγει το έγγραφο ιατρικού ιστορικού στη Βάση Δεδομένων του συστήματος.

1.1.1.9 <HealthRequest>

HealthRequest

«entity»
HealthRequest
<pre>-HealthRequestID: integer -descriptionText: Text -image: Image -objMedicalHistoryDocument: MedicalHistoryDocument -patientID: integer -doctorID: integer -hasReceivedDiagnosis: boolean +HealthRequest(): void +setHealthRequestID(HealthRequestID: integer): void +setDescriptionText(descriptionText: Text): void +setImage(image: Image): void +setMedHistoryDoc(objMedicalHistoryDocument: MedicalHistoryDocument): void +setPatientID(patientID: integer): void +setDoctorID(doctorID: integer): void +setHasReceivedDiagnosis(hasReceivedDiagnosis: boolean): void +getHealthRequestID(): integer +getDescriptionText(): Text +getImage(): Image +getMedHistoryDoc(): MedicalHistoryDocument +getPatientID(): integer +getHasReceivedDiagnosis(): boolean</pre>

Η κλάση αυτή περιέχει όλα τα χαρακτηριστικά/δεδομένα και τις συναρτήσεις/μεθόδους που αφορούν την οντότητα “ιατρικό αίτημα”, το οποίο αφορά σε ένα ιατρικό ζήτημα το “ασθενή” και για το οποίο αιτείται ιατρική συμβουλή ο “ασθενής”, μέσα στο σύστημα.

Χαρακτηριστικά Κλάσης

- HealthRequestID: integer – Ο μοναδικός χαρακτηριστικός ακέραιος αριθμός του ιατρικού αιτήματος στην εφαρμογή.
- descriptionText: Text – Το κείμενο περιγραφής της κατάστασης της υγείας (ιατρικού ζητήματος) του “ασθενή”.
- image: Image – Η εικόνα που συνοδεύει/συμπληρώνει την περιγραφή της κατάστασης της υγείας του “ασθενή”.
- objMedicalHistoryDocument: MedicalHistoryDocument – Το έγγραφο ιατρικού ιστορικού που συνοδεύει/συμπληρώνει την περιγραφή της κατάστασης της υγείας του “ασθενή”.
- patientID: integer – Ο μοναδικός χαρακτηριστικός ακέραιος αριθμός του (χρήστη-) “ασθενή” της εφαρμογής, που αιτείται ιατρικής συμβουλής.
- doctorID: integer – Ο μοναδικός χαρακτηριστικός ακέραιος αριθμός του (χρήστη-) “γιατρού”, που δέχεται την αίτηση για το ιατρικό ζήτημα του “ασθενή”.
- hasReceivedDiagnosis:boolean – Μια boolean μεταβλητή, που λαμβάνει την τιμή True, όταν υπάρχει συνδεδεμένο αντικείμενο της κλάσης Diagnosis με το αντικείμενο της κλάσης HealthRequest (δηλαδή το ιατρικό αίτημα έχει λάβει διάγνωση) και False, όταν δεν υπάρχει. Αρχικοποιείται στην τιμή False.

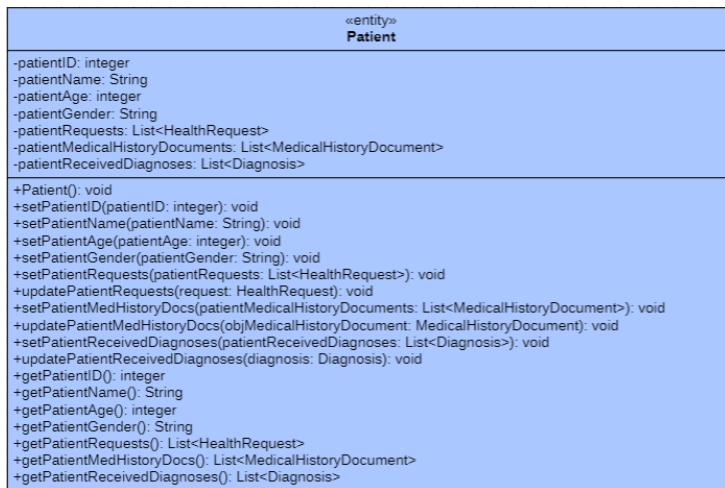
Μέθοδοι Κλάσης

- HealthRequest(): void – Η μέθοδος HealthRequest αποτελεί τη συνάρτηση δόμησης της οντότητας HealthRequest. Αυτή αρχικοποιεί την boolean μεταβλητή hasReceivedDiagnosis στην τιμή False.
- setHealthRequestID(HealthRequestID: integer): void – Η μέθοδος setHealthRequestID θέτει το μοναδικό χαρακτηριστικό ακέραιο αριθμό του ιατρικού αιτήματος.
- setDescriptionText(descriptionText: Text): void – Η μέθοδος setDescriptionText θέτει το κείμενο περιγραφής της κατάστασης της υγείας του “ασθενή”.
- setImage(image: Image): void – Η μέθοδος setImage θέτει την εικόνα που συνοδεύει την περιγραφή της κατάστασης της υγείας του “ασθενή”.
- setMedHistoryDoc(objMedicalHistoryDocument: MedicalHistoryDocument): void – Η μέθοδος setMedHistoryDoc θέτει το έγγραφο ιατρικού ιστορικού που συνοδεύει την περιγραφή της κατάστασης της υγείας του “ασθενή”.

- setPatientID(patientID: integer): void – Η μέθοδος setPatientID θέτει το μοναδικό χαρακτηριστικό ακέραιο αριθμό του “ασθενή”, που αιτείται ιατρικής συμβουλής για το ιατρικό του ζήτημα.
- setDoctorID(doctorID: integer): void – Η μέθοδος setDoctorID θέτει το μοναδικό χαρακτηριστικό ακέραιο αριθμό του “γιατρού” που δέχεται την αίτηση για το ιατρικό ζήτημα του “ασθενή”.
- setHasReceivedDiagnosis(hasReceivedDiagnosis: boolean): void – Η μέθοδος setHasReceivedDiagnosis θέτει την τιμή (True ή False) της boolean μεταβλητής hasReceivedDiagnosis του ιατρικού αιτήματος. Καλείται όταν υπάρξει Διάγνωση για το ιατρικό αίτημα.
- getDescriptionText(): Text – Η μέθοδος getDescriptionText επιστρέφει το κείμενο περιγραφής της κατάστασης της υγείας του “ασθενή”.
- getImage(): Image – Η μέθοδος getImage επιστρέφει την εικόνα που συνοδεύει την περιγραφή της κατάστασης της υγείας του “ασθενή”.
- getMedHistoryDoc(): MedicalHistoryDocument – Η μέθοδος getMedHistoryDoc επιστρέφει το έγγραφο ιατρικού ιστορικού που συνοδεύει την κατάσταση της υγείας του “ασθενή”.
- getPatientID(): integer – Η μέθοδος getPatientID επιστρέφει το μοναδικό χαρακτηριστικό ακέραιο αριθμό του “ασθενή”, που αιτείται ιατρικής συμβουλής για το ιατρικό του ζήτημα.
- getDoctorID(): integer – Η μέθοδος getDoctorID επιστρέφει το μοναδικό χαρακτηριστικό ακέραιο αριθμό του γιατρού που δέχεται την αίτηση για το ιατρικό ζήτημα του “ασθενή”.
- getHasReceivedDiagnosis(): boolean – Η μέθοδος getHasReceivedDiagnosis επιστρέφει την τιμή (True ή False) της boolean μεταβλητής hasReceivedDiagnosis του ιατρικού αιτήματος.

1.1.1.10 <Patient>

Patient



Η κλάση αυτή περιέχει όλα τα χαρακτηριστικά/δεδομένα και τις συναρτήσεις/μεθόδους που αφορούν την οντότητα “ασθενής” μέσα στο σύστημα (μόνο όσα χαρακτηριστικά και συναρτήσεις χρειάζονται για την υλοποίηση των Gherkin features του 1ου Παραδοτέου).

Χαρακτηριστικά Κλάσης

- patientID: integer – Ο μοναδικός χαρακτηριστικός ακέραιος αριθμός του (χρήστη-) “ασθενή” της εφαρμογής.
- patientName: String – Το όνομα του (χρήστη-) “ασθενή” στο σύστημα.
- patientAge: integer – Η ηλικία που έχει εισάγει ο “ασθενής”, για τον εαυτό του, στο σύστημα.
- patientGender: String – Το φύλο που έχει επιλέξει ο “ασθενής”, για τον εαυτό του, στο σύστημα.
- patientRequests: List<HealthRequests> – Λίστα με το σύνολο των ιατρικών ζητημάτων που έχει περιγράψει, αιτηθεί για ιατρική συμβουλή και εισάγει ο “ασθενής” στο σύστημα.
- patientMedicalHistoryDocuments: List<MedicalHistoryDocument> – Λίστα με το σύνολο των εγγράφων/αρχείων ιατρικού ιστορικού που έχει εισάγει ο “ασθενής” στο σύστημα.
- patientReceivedDiagnoses: List<Diagnosis> – Λίστα με το σύνολο των διαγνώσεων που έχει λάβει ο “ασθενής” για ιατρικά του ζητήματα.

Μέθοδοι Κλάσης

- Patient(): void – Η μέθοδος Patient αποτελεί τη συνάρτηση δόμησης της οντότητας Patient.
- setPatientID(patientID: integer): void – Η μέθοδος setPatientID θέτει το μοναδικό χαρακτηριστικό ακέραιο αριθμό του “ασθενή”.
- setPatientName(patientName: String): void – Η μέθοδος setPatientName θέτει το όνομα του “ασθενή”.
- setPatientAge(patientAge: integer): void – Η μέθοδος setPatientAge θέτει την ηλικία του “ασθενή”.
- setPatientGender(patientGender: String): void – Η μέθοδος setPatientGender θέτει το φύλο του “ασθενή”.
- setPatientRequests(patientRequests: List<HealthRequest>): void – Η μέθοδος setPatientRequests θέτει τη λίστα με το σύνολο των ιατρικών ζητημάτων του “ασθενή”.
- updatePatientRequests(request: HealthRequest): void – Η μέθοδος updatePatientRequests ενημερώνει τη λίστα με το σύνολο των ιατρικών ζητημάτων του “ασθενή”, με προσθήκη νέου αιτήματος του ασθενή.
- setPatientMedHistoryDocs(patientMedicalHistoryDocuments: List<MedicalHistoryDocument>): void – Η μέθοδος setPatientMedHistoryDocs θέτει τη λίστα με το σύνολο των εγγράφων ιατρικού ιστορικού του “ασθενή”.
- updatePatientMedHistoryDocs(objMedicalHistoryDocument: MedicalHistoryDocument): void – Η μέθοδος updatePatientMedHistoryDocs ενημερώνει τη λίστα με το σύνολο των εγγράφων ιατρικού ιστορικού του “ασθενή”, με προσθήκη νέου εγγράφου ιατρικού ιστορικού του “ασθενή”.
- setPatientReceivedDiagnoses(patientReceivedDiagnoses: List<Diagnosis>): void – Η μέθοδος setPatientReceivedDiagnoses θέτει τη λίστα με το σύνολο των διαγνώσεων για τα ιατρικά ζητήματα του “ασθενή”.
- updatePatientReceivedDiagnoses(diagnosis: Diagnosis): void – Η μέθοδος updatePatientReceivedDiagnoses ενημερώνει τη λίστα με το σύνολο των διαγνώσεων για τα ιατρικά ζητήματα του “ασθενή”, με προσθήκη νέας διάγνωσης για κάποιο ιατρικό ζήτημα.
- getPatientID(): integer – Η μέθοδος getPatientID επιστρέφει το μοναδικό χαρακτηριστικό ακέραιο αριθμό του “ασθενή”.
- getPatientName(): String – Η μέθοδος getPatientName επιστρέφει το όνομα του “ασθενή”.

- getPatientAge(): integer – Η μέθοδος getPatientAge επιστρέφει την ηλικία του “ασθενή”.
- getPatientGender(): String – Η μέθοδος getPatientGender επιστρέφει το φύλο του “ασθενή”
- getPatientRequests(): List<HealthRequest> – Η μέθοδος getPatientRequests επιστρέφει τη λίστα με το σύνολο των ιατρικών ζητημάτων του “ασθενή” (που έχει αιτηθεί για ιατρική συμβουλή).
- getPatientMedHistoryDocs(): List<MedicalHistoryDocument> – Η μέθοδος getPatientMedHistoryDocs επιστρέφει τη λίστα με το σύνολο των εγγράφων ιατρικού ιστορικού του “ασθενή”.
- getPatientReceivedDiagnoses: List<Diagnosis> – Η μέθοδος επιστρέφει τη λίστα με το σύνολο των διαγνώσεων για ιατρικά ζητήματα του “ασθενή”.

1.1.1.11 <MedicalHistoryDocumentController>

MedicalHistoryDocumentController

«controller»	
MedicalHistoryDocumentController	
+MedicalHistoryDocumentController(): void	
+createMedicalHistoryDocument(MedHistoryDocID: integer, patientID: integer, document: File, documentName: String): MedicalHistoryDocument	
+importMedicalHistoryDocumentToDB(objMedHistoryDoc: MedicalHistoryDocument): void	
+DisplayMedHistoryDocSelectionScreen(): void	
+viewMedicalHistoryDocument(): void	
+DisplayHomeScreen(): void	
+DisplayPreviousScreen(): void	

Η κλάση αυτή μοντελοποιεί τον ελεγκτή (κλάση ελέγχου) για την εισαγωγή και διαχείριση των εγγράφων ιατρικού ιστορικού (οντότητες της κλάσης MedicalHistoryDocument) στο σύστημα. Ο ελεγκτής περιέχει όλες τις συναρτήσεις/μεθόδους που χρειάζονται για την υλοποίηση της λειτουργικότητας σχετικά με την εισαγωγή και διαχείριση των εγγράφων ιατρικού ιστορικού μέσα στο σύστημα.

Μέθοδοι Κλάσης

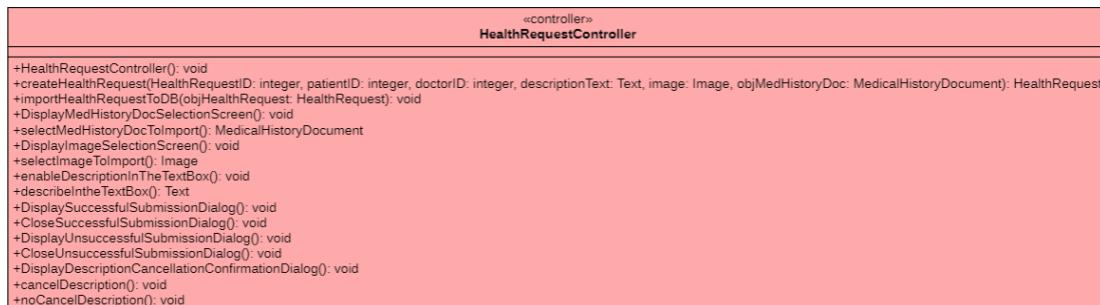
- MedicalHistoryDocumentController(): void – Η μέθοδος MedicalHistoryDocumentController αποτελεί τη συνάρτηση δόμησης του ελεγκτή.
- createMedicalHistoryDocument(MedHistoryDocID: integer, patientID: integer, document: File, documentName: String): MedicalHistoryDocument – Η μέθοδος createMedicalHistoryDocument δημιουργεί ένα αντικείμενο της κλάσης MedicalHistoryDocument, δηλαδή ένα έγγραφο ιατρικού ιστορικού, λαμβάνοντας ως ορίσματα το αρχείο που περιέχει το πραγματικό έγγραφο ιατρικού ιστορικού, το όνομα του εγγράφου, τον μοναδικό χαρακτηριστικό ακέραιο αριθμό του εγγράφου ιατρικού ιστορικού και τον μοναδικό χαρακτηριστικό ακέραιο αριθμό του χρήστη “ασθενή”. Η μέθοδος στην υλοποίηση της, καλεί τη συνάρτηση δόμησης της κλάσης MedicalHistoryDocument και έπειτα τις μεθόδους setDocument, setDocumentName, setPatientID της ίδιας κλάσης, προκειμένου να δομήσει το αντικείμενο που κατασκευάζεται. Η μέθοδος επιστρέφει το έγγραφο ιατρικού ιστορικού που δημιούργησε.
- importMedicalHistoryDocumentToDB(objMedHistoryDoc: MedicalHistoryDocument): void – Η μέθοδος importMedicalHistoryDocumentToDB εισάγει ένα αντικείμενο της κλάσης MedicalHistoryDocument, δηλαδή ένα έγγραφο ιατρικού ιστορικού, το οποίο δέχεται ως ορισμα, στη Βάση Δεδομένων του συστήματος. Η μέθοδος στην υλοποίηση

της καλεί τη μέθοδο importMedicalHistoryDocumentToDB της κλάσης DBProxy, προκειμένου να εισαχθεί στη Βάση Δεδομένων.

- DisplayMedHistoryDocSelectionScreen(): void – Η μέθοδος DisplayMedHistoryDocSelectionScreen καλείται για την προβολή της οθόνης επιλογής αρχείου, που θα αποτελεί το έγγραφο ιατρικού ιστορικού εντός της εφαρμογής, από το χρήστη “ασθενή” για εισαγωγή στη Βάση Δεδομένων του συστήματος.
- viewMedicalHistoryDocument(): void – Η μέθοδος viewMedicalHistoryDocument καλείται για την προβολή του αρχείου που αντιστοιχεί στο έγγραφο ιατρικού ιστορικού, που επιλέγεται από το χρήστη, μέσω της γραφικής διεπαφής στην οθόνη εγγράφων ιατρικού ιστορικού σχετικών με των χρήστη “ασθενή”.
- DisplayHomeScreen(): void – Η μέθοδος DisplayHomeScreen καλείται για την προβολή της αρχικής οθόνης της εφαρμογής.
- DisplayPreviousScreen(): Text – Η μέθοδος DisplayPreviousScreen καλείται για την προβολή της προηγούμενης οθόνης της εφαρμογής από την οποία ο χρήστης προήλθε στην τρέχουσα προβαλλόμενη.

1.1.1.12 <HealthRequestController>

HealthRequestController



Η κλάση αυτή μοντελοποιεί τον ελεγκτή (κλάση ελέγχου) για την εισαγωγή και διαχείριση των ιατρικών αιτημάτων του χρήστη “ασθενή” (οντότητες της κλάσης HealthRequest), για ιατρικά ζητήματα, στο σύστημα. Ο ελεγκτής περιέχει όλες τις συναρτήσεις/μεθόδους που απαιτούνται για την υλοποίηση της λειτουργικότητας σχετικά με την εισαγωγή και διαχείριση αιτημάτων για ιατρικά ζητήματα μέσα στο σύστημα. η οποία αφορά στη λειτουργικότητα για την εισαγωγή της περιγραφής της κατάστασης της υγείας του χρήστη “ασθενή” στο σύστημα και την επιλογή γιατρού, στον οποίο ο “ασθενής” αιτείται να εξετάσει το ιατρικό του ζήτημα. Η κλάση ελέγχου HealthRequestController κληρονομεί την κλάση ελέγχου ChooseDoctorController.

Μέθοδοι Κλάσης

- HealthRequestController(): void – Η μέθοδος HealthRequestController αποτελεί τη συνάρτηση δόμησης του ελεγκτή.
- createHealthRequest(HealthRequestId: integer, patientID: integer, doctorID: integer, descriptionText: Text, image: Image, objMedHistoryDoc: MedicalHistoryDocument): HealthRequest – Η μέθοδος createHealthRequest δημιουργεί ένα αντικείμενο της κλάσης HealthRequest, δηλαδή ένα ιατρικό αίτημα για κάποιο ιατρικό ζήτημα, λαμβάνοντας ως ορίσματα τον μοναδικό χαρακτηριστικό ακέραιο αριθμό του ιατρικού αιτήματος που δημιουργείται, τον μοναδικό χαρακτηριστικό ακέραιο αριθμό του χρήστη “ασθενή”, που αιτείται, τον μοναδικό χαρακτηριστικό ακέραιο αριθμό του χρήστη “γιατρός”, στον οποίο

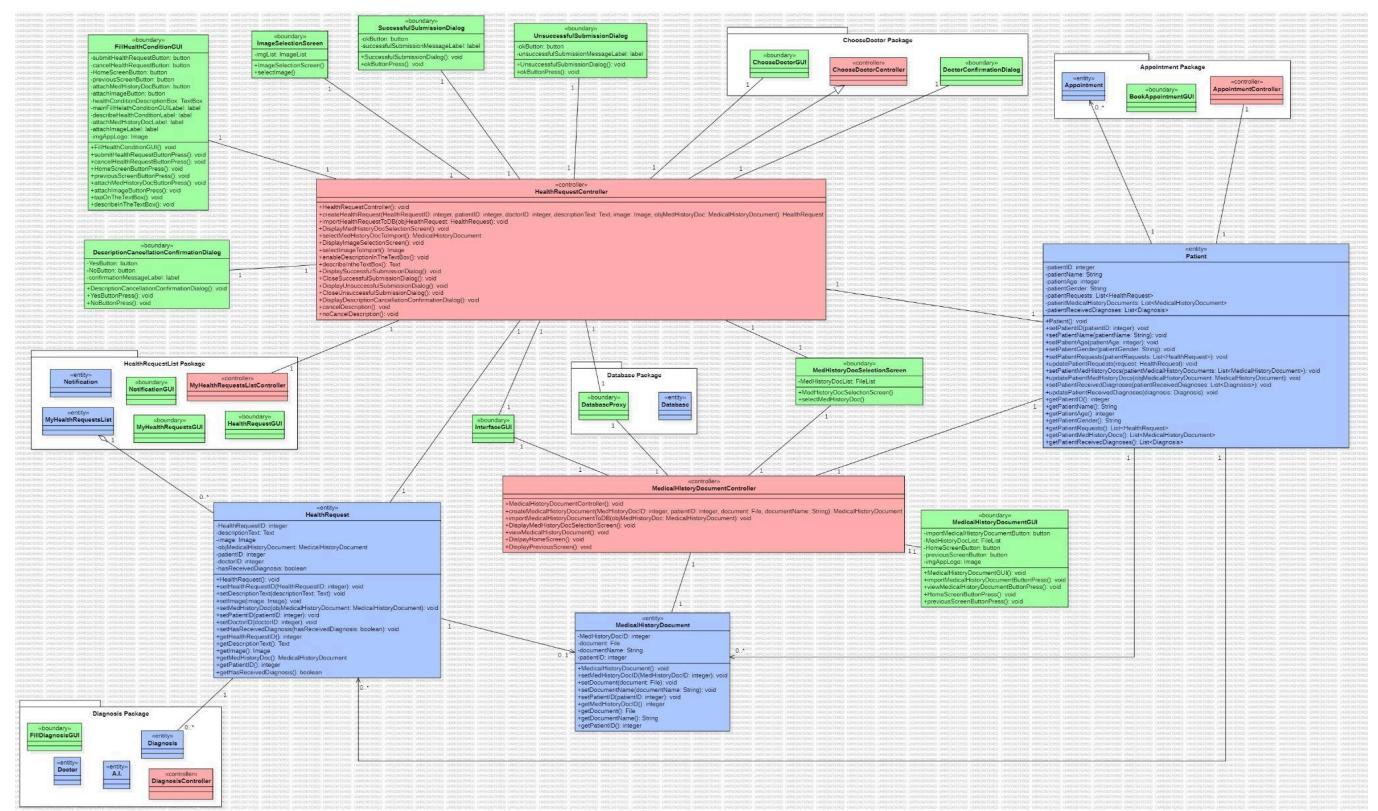
αιτείται, το κείμενο στο οποίο περιγράφεται η κατάσταση της υγείας του “ασθενή”, η εικόνα που συνοδεύει την περιγραφή της κατάστασης της υγείας και ένα αντικείμενο της κλάσης MedicalHistoryDocument που αντιστοιχεί σε ένα έγγραφο ιατρικού ιστορικού. Στην υλοποίηση της μεθόδου, τα ορίσματα image και objMedHistoryDoc μπορούν να είναι null ή να παραλειφθούν, ενώ τα ορίσματα HealthRequestID, patientID, doctorID και descriptionText δεν μπορούν. Η μέθοδος επιστρέφει το ιατρικό αίτημα που δημιούργησε.

- importHealthRequestToDB(objHealthRequest: HealthRequest): void – Η μέθοδος importHealthRequestToDB εισάγει ένα αντικείμενο της κλάσης HealthRequest, δηλαδή ένα ιατρικό αίτημα, το οποίο δέχεται ως όρισμα, στη Βάση Δεδομένων του συστήματος. Η μέθοδος στην υλοποίηση της καλεί τη μέθοδο importHealthRequestToDB της κλάσης DBProxy, προκειμένου να εισαχθεί στη Βάση Δεδομένων.
- DisplayMedHistoryDocSelectionScreen(): void – Η μέθοδος DisplayMedHistoryDocSelectionScreen καλείται για την προβολή της οθόνης επιλογής αρχείου, που θα αποτελεί το έγγραφο ιατρικού ιστορικού εντός της εφαρμογής, από το χρήστη “ασθενή” για εισαγωγή στη Βάση Δεδομένων του συστήματος.
- selectMedHistoryDocToImport(): MedicalHistoryDocument – Η μέθοδος selectMedHistoryDocToImport καλείται από τη συνάρτηση selectMedHistoryDoc της κλάσης MedHistoryDocSelectionScreen. Η μέθοδος δημιουργεί ένα αντικείμενο της κλάσης MedicalHistoryDocument, δηλαδή ένα έγγραφο ιατρικού ιστορικού, που έχει ως χαρακτηριστικά το αρχείο που επέλεξε ο χρήστης “ασθενής”, το όνομα του αρχείου αυτού και τον μοναδικό χαρακτηριστικό ακέραιο αριθμό του χρήστη “ασθενή” αυτού. Όταν η μέθοδος καλείται κλείνει η οθόνη επιλογής αρχείου. Η μέθοδος επιστρέφει αυτό το έγγραφο ιατρικού ιστορικού.
- DisplayImageSelectionScreen(): void – Η μέθοδος DisplayImageSelectionScreen καλείται για την προβολή της οθόνης επιλογής εικόνας, που θα αποτελεί την εικόνα που συνοδεύει το κείμενο περιγραφής της κατάστασης της υγείας από το χρήστη “ασθενή”.
- selectImageToImport(): Image – Η μέθοδος selectImageToImport καλείται από τη συνάρτηση selectImage της κλάσης ImageSelectionScreen. Η μέθοδος δημιουργεί ένα αντικείμενο της κλάσης Image, δηλαδή μια εικόνα, που αντιστοιχεί στην εικόνα που ο χρήστης “ασθενής” επέλεξε. Όταν η μέθοδος καλείται κλείνει η οθόνη επιλογής εικόνας. Η μέθοδος επιστρέφει αυτή την εικόνα.
- enableDescriptonInTheTextBox(): void – Η μέθοδος enableDescriptionInTheTextBox καλείται από τη συνάρτηση tapOnTheTextBox της κλάσης FillHealthConditionGUI, για την ενεργοποίηση της δυνατότητας εισαγωγής χαρακτήρων στο πλαίσιο/κουτί κειμένου, οι οποίοι θα αποτελέσουν το σώμα κειμένου που εισάγει ο χρήστης “ασθενής” στο σύστημα για την περιγραφή της κατάστασης της υγείας.
- describeInTheTextBox():Text – Η μέθοδος describeInTheTextBox καλείται για να δημιουργήσει ένα αντικείμενο της κλάσης Text, δηλαδή το κείμενο που δομούν οι ακολουθίες χαρακτήρων που πληκτρολόγησε ο χρήστης “ασθενής” στο πλαίσιο κειμένου της οθόνης περιγραφής της κατάστασης της υγείας. Η μέθοδος επιστρέφει αυτό το κείμενο.
- DisplaySuccessfulSubmissionDialog(): void – Η μέθοδος DisplaySuccessfulSubmissionDialog καλείται για την προβολή του παραθύρου διαλόγου επιτυχούς υποβολής της κατάστασης της υγείας από το χρήστη “ασθενή” στο σύστημα.
- CloseSuccessfulSubmissionDialog(): void – Η μέθοδος CloseSuccessfulSubmissionDialog καλείται για την διακοπή προβολής του παραθύρου διαλόγου επιτυχούς υποβολής της κατάστασης της υγείας από το χρήστη “ασθενή” στο σύστημα.
- DisplayUnsuccessfulSubmissionDialog(): void – Η μέθοδος DisplayUnsuccessfulSubmissionDialog καλείται για την προβολή του παραθύρου διαλόγου

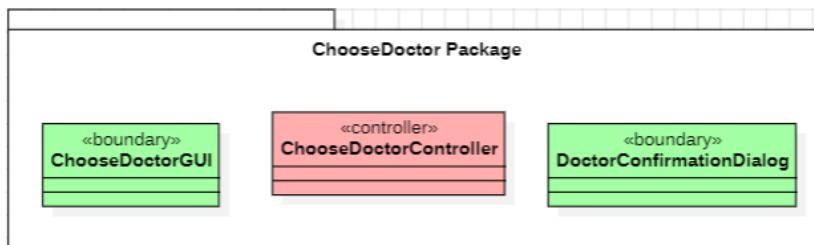
ανεπιτυχούς υποβολής της κατάστασης της υγείας από το χρήστη “ασθενή” στο σύστημα.

- CloseUnsuccessfulSubmissionDialog(): void – Η μέθοδος CloseUnsuccessfulSubmissionDialog καλείται για τη διακοπή προβολής του παραθύρου διαλόγου ανεπιτυχούς υποβολής της κατάστασης της υγείας από το χρήστη “ασθενή” στο σύστημα.
- DisplayDescriptionCancellationConfirmationDialog(): void – Η μέθοδος DisplayDescriptionCancellationConfirmationDialog καλείται για την προβολή του παραθύρου διαλόγου επιβεβαίωσης ακύρωσης της φόρμας υποβολής της κατάστασης της υγείας από το χρήστη “ασθενή”.
- cancelDescription(): void – Η μέθοδος cancelDescription καλείται για την ακύρωση της φόρμας υποβολής της κατάστασης της υγείας του χρήστη “ασθενή” και διαγραφή όλων των αντικειμένων που δημιουργήθηκαν από ενέργειες του “ασθενή” στη συγκεκριμένη φόρμα, χωρίς εισαγωγή τους στη Βάση Δεδομένων του συστήματος. Η μέθοδος καλείται από τη συνάρτηση YesButtonPress της κλάσης DescriptionCancellationConfirmationDialog.
- noCancelDescription(): void – Η μέθοδος noCancelDescription καλείται για τη διακοπή προβολής του παραθύρου διαλόγου επιβεβαίωσης ακύρωσης της φόρμας υποβολής της κατάστασης της υγείας από το χρήστη “ασθενή” και επιστροφή στην οθόνη περιγραφής της κατάστασης της υγείας του “ασθενή”, όπως ακριβώς την άφησε ο χρήστης, χωρίς την διαγραφή ή την εισαγωγή αντικειμένων στο σύστημα. Η μέθοδος καλείται από τη συνάρτηση NoButtonPress της κλάσης DescriptionCancellationConfirmationDialog.

1.1.1.13 Διάγραμμα κλάσεων



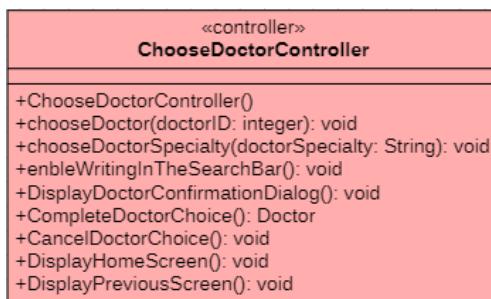
1.1.2 Πακέτο ChooseDoctor



Το πακέτο περιλαμβάνει τις κλάσεις που σχετίζονται με την επιλογή γιατρού.

1.1.2.1 <ChooseDoctorController>

ChooseDoctorController



Η κλάση αυτή μοντελοποιεί τον ελεγκτή (κλάση ελέγχου) για την επιλογή γιατρού από τον χρήστη “ασθενή”, στον οποίο αιτείται να εξετάσει το ιατρικό του ζήτημα ή αιτείται ένα ιατρικό ραντεβού. Ο ελεγκτής περιέχει όλες τις συναρτήσεις/μεθόδους που χρειάζονται για την υλοποίηση της λειτουργικότητας σχετικά με την επιλογή γιατρού μέσα στο σύστημα. Η κλάση αυτή με τις μεθόδους και τις συσχετίσεις της με άλλες κλάσεις, όπως παρουσιάζονται στο διάγραμμα κλάσεων, κληρονομείται από τις κλάσεις HealthRequestController και AppointmentController.

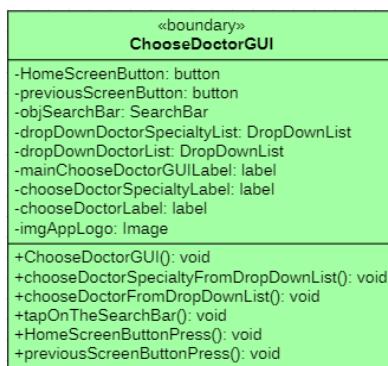
Μέθοδοι Κλάσης

- ChooseDoctorController():void – Η μέθοδος ChooseDoctorController αποτελεί τη συνάρτηση δόμησης του ελεγκτή.
- chooseDoctor(doctorID: integer): Doctor – Η μέθοδος chooseDoctor δημιουργεί ένα αντικείμενο της κλάσης Doctor λαμβάνοντας ως όρισμα τον μοναδικό χαρακτηριστικό ακέραιο αριθμό του χρήστη “γιατρός”, τον οποίο επιλέγει ο χρήστης “ασθενής” μέσω κάποιας γραφικής διεπαφής. Η μέθοδος στην υλοποίηση της, καλεί τη συνάρτηση δόμησης της κλάσης Doctor και έπειτα τις μεθόδους ανάθεσης τιμών (setters) στα χαρακτηριστικά αυτού του αντικειμένου, προκειμένου να δημιουργηθεί ένα ίδιο αντικείμενο με αυτό στο οποίο αντιστοιχεί ο μοναδικός χαρακτηριστικός ακέραιος αριθμός, που δέχεται η μέθοδος chooseDoctor ως όρισμα. Η μέθοδος επιστρέφει τον γιατρό που δημιούργησε.
- chooseDoctorSpecialty(doctorSpecialty: String): void – Η μέθοδος chooseDoctorSpecialty καλείται από τη συνάρτηση chooseDoctorSpecialtyFromDropDownList της κλάσης ChooseDoctorGUI και δέχεται ως όρισμα την ακολουθία χαρακτήρων που αφορά στο χαρακτηριστικό specialty της κλάσης Doctor. Η μέθοδος φιλτράρει την αναπτυσσόμενη λίστα με τους διαθέσιμους γιατρούς (στη Βάση Δεδομένων της εφαρμογής) στην οθόνη επιλογής γιατρού από το χρήστη “ασθενή”, με βάση την τιμή του χαρακτηριστικού specialty, δηλαδή της ειδικότητας που επιλέχθηκε.

- enableWritingInTheSearchBar(): void – Η μέθοδος enableWritingInTheSearchBar καλείται από τη συνάρτηση tapOnTheSearchBar της κλάσης ChooseDoctorGUI, για την ενεργοποίηση της δυνατότητας εισαγωγής χαρακτήρων στην γραμμή αναζήτησης, για την αναζήτηση γιατρού με βάση την τιμή του χαρακτηριστικού name της κλάσης Doctor.
- DisplayDoctorConfirmationDialog(): void – Η μέθοδος DisplayDoctorConfirmationDialog καλείται για την προβολή του παραθύρου διαλόγου επιβεβαίωσης της επιλογής γιατρού από το χρήστη “ασθενή” για εξέταση του ιατρικού του ζητήματος ή για αίτημα ραντεβού.
- CompleteDoctorChoice(): Doctor – Η μέθοδος CompleteDoctorChoice καλείται από τη συνάρτηση confirmButtonPress της κλάσης DoctorConfirmationDialog και επιστρέφει ένα αντικείμενο της κλάσης Doctor, το οποίο είναι εκείνη η οντότητα του συστήματος, που αντιστοιχεί στον γιατρό που επέλεξε ο χρήστης ασθενής, μέσω των γραφικών διεπαφών της οθόνης επιλογής γιατρού από το χρήστη “ασθενή”.
- CancelDoctorChoice(): void – Η μέθοδος CancelDoctorChoice καλείται από τη συνάρτηση cancelButtonPress της κλάσης DoctorConfirmationDialog για την ακύρωση της επιλογής του συγκεκριμένου γιατρού από το χρήστη “ασθενή”.
- DisplayHomeScreen(): void – Η μέθοδος DisplayHomeScreen καλείται για την προβολή της αρχικής οθόνης της εφαρμογής.
- DisplayPreviousScreen(): void – Η μέθοδος DisplayPreviousScreen καλείται για την προβολή της προηγούμενης οθόνης της εφαρμογής από την οποία ο χρήστης προήλθε στην τρέχουσα προβαλλόμενη.

1.1.2.2 <ChooseDoctorGUI>

ChooseDoctorGUI



Η κλάση αυτή εκφράζει τη διεπαφή της οθόνης επιλογής γιατρού από το χρήστη “ασθενή” για εξέταση του ιατρικού του ζητήματος ή για αίτημα ραντεβού.

Χαρακτηριστικά Κλάσης

- HomeScreenButton: button – Κουμπί μετάβασης στην αρχική οθόνη της εφαρμογής.
- previousScreenButton: button – Κουμπί επιστροφής στην προηγούμενη οθόνη της εφαρμογής.
- objSearchBar: SearchBar – Πλαίσιο αναζήτησης για εισαγωγή ονόματος γιατρού επιλογής από το χρήστη “ασθενή”.
- dropDownDoctorSpecialtyList: DropDownList – Αναπτυσσόμενη λίστα με τις διαθέσιμες ειδικότητες γιατρών στη Βάση Δεδομένων της εφαρμογής.
- dropDownDoctorList: DropDownList – Αναπτυσσόμενη λίστα με τους διαθέσιμους γιατρούς στη Βάση Δεδομένων της εφαρμογής.
- mainChooseDoctorLabel: label – Κεντρική επιγραφή της οθόνης επιλογής γιατρού από τον χρήστη “ασθενή”.
- chooseDoctorSpecialtyLabel: label – Επιγραφή ένδειξης της διεπαφής προβολής της λίστας ειδικοτήτων γιατρών.
- chooseDoctorLabel: label – Επιγραφή ένδειξης της διεπαφής προβολής της λίστας ειδικοτήτων γιατρών.
- imgAppLogo: Image – Εικόνα λογοτύπου της εφαρμογής.

Μέθοδοι Κλάσης

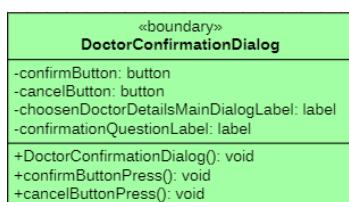
- ChooseDoctorGUI(): void – Η μέθοδος ChooseDoctorGUI αποτελεί τη συνάρτηση δόμησης της διεπαφής αυτής.
- chooseDoctorSpecialtyFromDropDownList(): void – Η μέθοδος chooseDoctorSpecialtyFromDropDownList καλεί τη συνάρτηση chooseDoctorSpecialty του ελεγκτή HealthRequestController, αν ο χρήστης “ασθενής” αλληλεπιδρά με τη διεπαφή της οθόνης επιλογής γιατρού προερχόμενος από τη διεπαφή νέας αίτησης για ιατρικό ζήτημα (request) της Welcome Screen ή της κεντρικής οθόνης της εφαρμογής. Η μέθοδος chooseDoctorSpecialtyFromDropDownList καλεί τη συνάρτηση chooseDoctorSpecialty του ελεγκτή AppointmentController, αν ο χρήστης “ασθενής” αλληλεπιδρά με τη διεπαφή της οθόνης επιλογής γιατρού προερχόμενος από τη διεπαφή νέας αίτησης για ιατρικό ραντεβού της κεντρικής οθόνης της εφαρμογής. Εκτελείται όταν ο χρήστης “ασθενής” πατάει την αντίστοιχη γραφική διεπαφή μιας ειδικότητας γιατρού από την αναπτυσσόμενη λίστα.
- chooseDoctorFromDropDownList(): void – Η μέθοδος chooseDoctorFromDropDownList καλεί τη συνάρτηση chooseDoctor του ελεγκτή HealthRequestController, αν ο χρήστης “ασθενής” αλληλεπιδρά με τη διεπαφή της οθόνης επιλογής γιατρού προερχόμενος

από τη διεπαφή νέας αίτησης για ιατρικό ζήτημα (request) της Welcome Screen ή της κεντρικής οθόνης της εφαρμογής. Η μέθοδος chooseDoctorFromDropDownList καλεί τη συνάρτηση chooseDoctor του ελεγκτή AppointmentController, αν ο χρήστης “ασθενής” αλληλεπιδρά με τη διεπαφή της οθόνης επιλογής γιατρού προερχόμενος από τη διεπαφή νέας αίτησης για ιατρικό ραντεβού της κεντρικής οθόνης της εφαρμογής. Εκτελείται όταν ο χρήστης “ασθενής” πατάει την αντίστοιχη γραφική διεπαφή ενός γιατρού από την αναπτυσσόμενη λίστα.

- tapOnTheSearchBar(): void – Η μέθοδος tapOnTheSearchBar καλεί τη συνάρτηση enableWritingInTheSearchBar του ελεγκτή HealthRequestController, αν ο χρήστης “ασθενής” αλληλεπιδρά με τη διεπαφή της οθόνης επιλογής γιατρού προερχόμενος από τη διεπαφή νέας αίτησης για ιατρικό ζήτημα (request) της Welcome Screen ή της κεντρικής οθόνης της εφαρμογής, ή τη συνάρτηση enableWritingInTheSearchBar του ελεγκτή AppointmentController, αν ο χρήστης “ασθενής” αλληλεπιδρά με τη διεπαφή της οθόνης επιλογής γιατρού προερχόμενος από τη διεπαφή νέας αίτησης για ιατρικό ραντεβού της κεντρικής οθόνης της εφαρμογής, για ενεργοποίηση της λειτουργίας εισαγωγής ονόματος γιατρού επιλογής στο πλαίσιο αναζήτησης, από το χρήστη “ασθενή”.
- HomeScreenButtonPress(): void – Η μέθοδος HomeScreenButtonPress καλεί τη συνάρτηση DisplayHomeScreen του ελεγκτή HealthRequestController, αν ο χρήστης “ασθενής” αλληλεπιδρά με τη διεπαφή της οθόνης επιλογής γιατρού προερχόμενος από τη διεπαφή νέας αίτησης για ιατρικό ζήτημα (request) της Welcome Screen ή της κεντρικής οθόνης της εφαρμογής, ή τη συνάρτηση DisplayHomeScreen του ελεγκτή AppointmentController, αν ο χρήστης “ασθενής” αλληλεπιδρά με τη διεπαφή της οθόνης επιλογής γιατρού προερχόμενος από τη διεπαφή νέας αίτησης ιατρικού ραντεβού της κεντρικής οθόνης της εφαρμογής, και οδηγεί στην προβολή της αρχικής οθόνης της εφαρμογής.
- previousScreenButtonPress(): void – Η μέθοδος previousScreenButtonPress καλεί τη συνάρτηση DisplayPreviousScreen του ελεγκτή HealthRequestController, αν ο χρήστης “ασθενής” αλληλεπιδρά με τη διεπαφή της οθόνης επιλογής γιατρού προερχόμενος από τη διεπαφή νέας αίτησης για ιατρικό ζήτημα (request) της Welcome Screen ή της κεντρικής οθόνης της εφαρμογής, ή τη συνάρτηση DisplayPreviousScreen του ελεγκτή AppointmentController, αν ο χρήστης “ασθενής” αλληλεπιδρά με τη διεπαφή της οθόνης επιλογής γιατρού προερχόμενος από τη διεπαφή νέας αίτησης ιατρικού ραντεβού της κεντρικής οθόνης της εφαρμογής, και οδηγεί στην προβολή της προηγούμενης οθόνης της εφαρμογής.

1.1.2.3 <DoctorConfirmationDialog>

DoctorConfirmationDialog



Η κλάση αυτή εκφράζει τη διεπαφή του παραθύρου διαλόγου επιβεβαίωσης της επιλογής γιατρού από το χρήστη “ασθενή” για εξέταση του ιατρικού του ζητήματος ή για αίτημα ραντεβού.

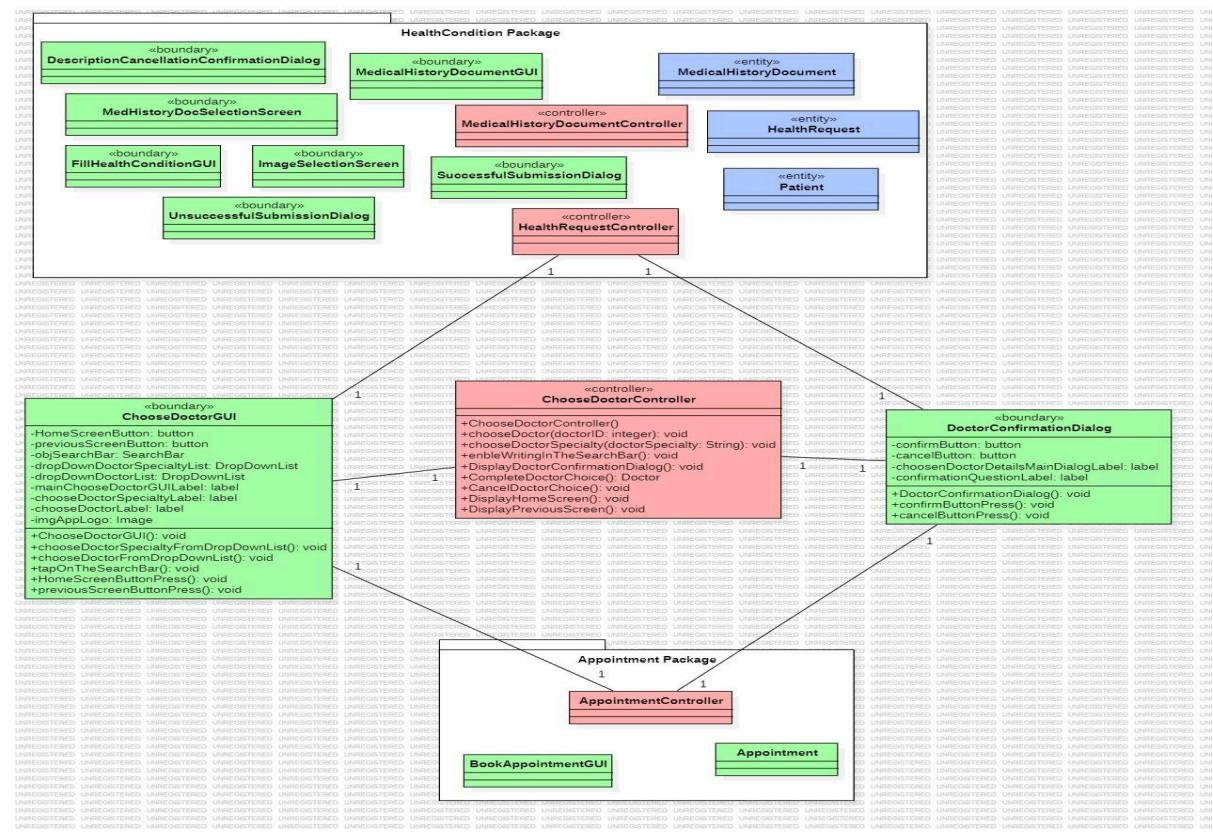
Χαρακτηριστικά Κλάσης

- confirmButton: button – Κουμπί επιβεβαίωσης της επιλογής γιατρού.
- cancelButton: button – Κουμπί ακύρωσης της επιλογής γιατρού.
- chosenDoctorDetailsMainDialogLabel: label – Κεντρική επιγραφή του παραθύρου διαλόγου επιβεβαίωσης, με το όνομα και την ειδικότητα του γιατρού επιλογής.
- confirmationQuestionLabel: label – Επιγραφή ερώτησης επιβεβαίωσης της επιλογής γιατρού.

Μέθοδοι Κλάσης

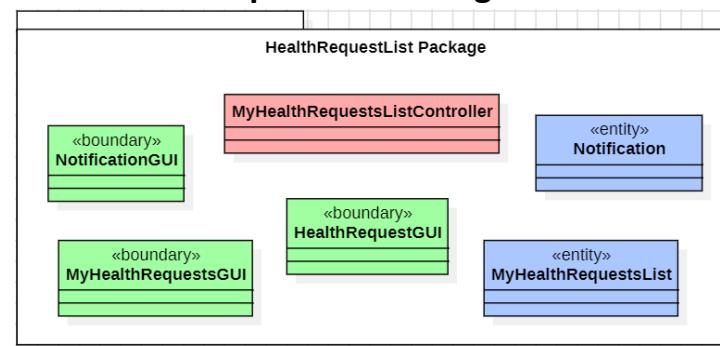
- DoctorConfirmationDialog(): void – Η μέθοδος DoctorConfirmationDialog αποτελεί τη συνάρτηση δόμησης της διεπαφής αυτής.
- confirmButtonPress(): void – Η μέθοδος confirmButtonPress καλεί τη συνάρτηση completeDoctorChoice του ελεγκτή HealthRequestController, αν ο χρήστης “ασθενής” αλληλεπιδρά με τη διεπαφή της οθόνης επιλογής γιατρού προερχόμενος από τη διεπαφή νέας αίτησης για ιατρικό ζήτημα (request) της Welcome Screen ή της κεντρικής οθόνης της εφαρμογής, ή τη συνάρτηση completeDoctorChoice του ελεγκτή AppointmentController, αν ο χρήστης “ασθενής” αλληλεπιδρά με τη διεπαφή της οθόνης επιλογής γιατρού προερχόμενος από τη διεπαφή νέας αίτησης για ιατρικό ζήτημα (request) της Welcome Screen ή της κεντρικής οθόνης της εφαρμογής, ή τη συνάρτηση completeDoctorChoice του ελεγκτή AppointmentController, αν ο χρήστης “ασθενής” αλληλεπιδρά με τη διεπαφή της οθόνης επιλογής γιατρού προερχόμενος από τη διεπαφή νέας αίτησης για ιατρικό ζήτημα (request) της Welcome Screen ή της κεντρικής οθόνης της εφαρμογής. Εκτελείται όταν ο χρήστης “ασθενής” πατάει το κουμπί confirmButton.
- cancelButtonPress(): void – Η μέθοδος cancelButtonPress καλεί τη συνάρτηση cancelDoctorChoice του ελεγκτή HealthRequestController, αν ο χρήστης “ασθενής” αλληλεπιδρά με τη διεπαφή της οθόνης επιλογής γιατρού προερχόμενος από τη διεπαφή νέας αίτησης για ιατρικό ζήτημα (request) της Welcome Screen ή της κεντρικής οθόνης της εφαρμογής, ή τη συνάρτηση cancelDoctorChoice του ελεγκτή AppointmentController, αν ο χρήστης “ασθενής” αλληλεπιδρά με τη διεπαφή της οθόνης επιλογής γιατρού προερχόμενος από τη διεπαφή νέας αίτησης για ιατρικό ζήτημα (request) της Welcome Screen ή της κεντρικής οθόνης της εφαρμογής. Εκτελείται όταν ο χρήστης “ασθενής” πατάει το κουμπί cancelButton.

1.1.2.4 Διάγραμμα κλάσεων



1.2 <Πακέτα κλάσεων σχετικών με το Gherkin Feature (Σενάριο Χρήστης) 4: “Patient receives a medical diagnosis”>

1.2.1 HealthRequestList Package



1.2.1.1 <Notification>

«entity»
Notification
-NotificationId: Int
-NotificationText: String
+Notification(): Notification
+setNotificationId(Diagnosis): void
+setNotificationText(Notification): void
+getNotificationText(Notification): String
+editNotification(Notification): void

Η κλάση αυτή περιγράφει το αντικείμενο Notification.

Χαρακτηριστικά Κλάσης

- notificationId: Η τιμή Id που δίνουμε στην κάθε ειδοποίηση για να τις διακρίνουμε ανα μεταξύ τους.
- notificationText: Το κείμενο το οποίο προβάλλουμε στον ασθενή όταν θέλουμε να του γνωστοποιήσουμε ότι έχει λάβει μία ειδοποίηση.

Μέθοδοι Κλάσης

- Notification(): Ο constructor του αντικειμενου Notification.
- setNotificationId(Diagnosis): Δίνει στο νέο Notification μια τιμή id ίδια με την τιμή του id του Diagnosis και κατ' επέκταση και του HealthRequest για το οποίο δημιουργήθηκε.
- setNotificationText(Notification): Δημιουργεί ένα τυποποιημένο μήνυμα για να προβληθεί έπειτα στον χρήστη το οποίο θα περιέχει το id του HealthRequest. Π.χ. "Your Health Request #12 has received a diagnosis." όπου το μόνο μεταβλητό στοιχείο είναι το HealthRequestId.
- getNotificationText(): Δίνει το κείμενο που προβάλλεται στον χρήστη όταν λαμβανει μια ειδοποίηση. Π.χ. "Your Health Request #12 has received a diagnosis."
- editNotification(Notification): Τροποποιεί την κατάσταση του εκάστοτε Notification αλλάζοντας το status σε μια από τις εξής τιμές: sent, seen.

1.2.1.2 <MyHealthRequestList>

«entity»
MyHealthRequestsList
-HealthRequest: HealthRequest
-Diagnosis: Diagnosis
-SeqId: int
+HealthRequestList(): HealthRequestList
+getHealthRequestList(HealthRequestList): Array{HealthRequestId hasReceivedDiagnosis SeqId}

Η κλάση αυτή περιγράφει το αντικείμενο MyHealthRequestList.

Χαρακτηριστικά Κλάσης

- HealthRequest: Η λίστα HealthRequestList αποτελείται από HealthRequests όπως προϊδεάζει και η ονομασία της κλάσης.
- Diagnosis: Ως απάντηση σε αυτά τα HealthRequests η λίστα επίσης περιλαμβάνει τις διαγνώσεις από τους γιατρούς και το AI.
- SeqId: Ένας αριθμός που συνοδεύει κάθε HealthRequest, και Diagnosis αν υπάρχει, σε μία λίστα. Εκφράζει την φθίνουσα ταξινομημένη σειρά των HealthRequestId που

βρίσκονται εντός της λίστας επιτυγχάνοντας ουσιαστικά μια χρονολογική ταξινόμηση των HealthRequests.

Μέθοδοι Κλάσης

- `HealthRequestList()`: O constructor του αντικειμενου `HealthRequestList`.
- `getHealthRequestList(HealthRequestList)`: Δίνει ως αποτέλεσμα ένα matrix με τρεις στήλες. Η πρώτη απεικονίζει τα id των `HealthRequests`, η δεύτερη σημειώνει την ύπαρξη ή όχι `Diagnosis` με το ίδιο Id και η τρίτη αριθμεί τα `HealthRequest` βάσει τα `HealthRequestId` τους με φθίνουσα σειρά. Η αρίθμηση ξεκινάει από το ένα και χρησιμεύει στο να μπουν όλα τα `Requests` στην ίδια σειρά με αυτήν που ο ασθενής τα υπέβαλε.

1.2.1.3 <NotificationGUI>

«boundary»	NotificationGUI
-	<code>displayNotificationButton: Button</code>
+	<code>NotificationGUI()</code>
+	<code>displayNotification(Notification): void</code>
+	<code>clickNotification(Notification): void</code>

Η κλάση αυτή προβάλλει το αντικείμενο `Notification`.

Χαρακτηριστικά Κλάσης

- `displayNotificationButton`: Το κουμπί της ειδοποίησης που πατάει ο χρήστης αφού σύρει τη μπάρα ειδοποίσεων προς τα κάτω.

Μέθοδοι Κλάσης

- `NotificationGUI()`: O constructor της κλάσης.
- `displayNotification(Notification)`: Η συνάρτηση που προβάλλει την ειδοποίηση στον χρήστη.
- `clickNotification()`: Η συνάρτηση η οποία δηλώνει ότι η ειδοποίηση έχει πατηθεί έτσι ώστε να παραπεμφεί ο χρήστης στην κατάλληλη οθόνη της εφαρμογής.

1.2.1.4 <MyHealthRequestListGUI>

«boundary»	MyHealthRequestListGUI
-	<code>displayRequestButton: Button</code>
-	<code>homeScreenButton: Button</code>
-	<code>sideMenuButton: Button</code>
+	<code>MyHealthRequestListGUI()</code>
+	<code>displayHealthRequestList(): void</code>
+	<code>clickRequestButton(): void</code>
+	<code>homeScreenButtonPress(): void</code>
+	<code>sideMenuButtonPress(): void</code>

Η κλάση αυτή προβάλλει το αντικείμενο `MyHealthRequestList`.

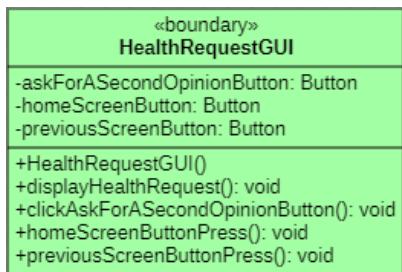
Χαρακτηριστικά Κλάσης

- `MyHealthRequestListGUI()`: Ο constructor της κλάσης.
- `displayRequestButton`: Το κουμπί που πατάει ο χρήστης για να μεταβεί στο συγκεκριμένο `HealthRequest` εντός της λίστας του.
- `homeScreenButton`: Το κουμπί που επιστρέφει τον χρήστη στην αρχική οθόνη.
- `sideMenuButton`: Το κουμπί που δίνει στον χρήστη πρόσβαση στο μενού γρήγορης μετάβασης από τη μια ενότητα της εφαρμογής στην άλλη.

Μέθοδοι Κλάσης

- `displayHealthRequestList()`: Παίρνει την έξοδο της συνάρτησης `getHealthRequestList()`, προσθέτει γραφικά στοιχεία και τη δίνει στο χρήστη στη μορφή μίας οθόνης-λίστας.
- `clickRequestButton(HealthRequestId)`: Παραπέμπει τον ασθενή στην οθόνη η οποία υλοποιείται από τη συνάρτηση `displayRequest()`.
- `homeScreenButtonPress()`: Ελέγχει αν έχει πατηθεί το κουμπί `homeScreenButton`.
- `sideMenuButtonPress()`: Ελέγχει αν έχει πατηθεί το κουμπί `sideMenuButton`.

1.2.1.5 <HealthRequestGUI>



Η κλάση αυτή προβάλλει το κάθε `HealthRequest` εντός του `MyHealthRequestList`.

Χαρακτηριστικά Κλάσης

- `HealthRequestGUI()`: Ο constructor της κλάσης.
- `askForASecondOpinionButton`: Το κουμπί που δηλώνει στο σύστημα ότι ο ασθενής θέλει να λάβει μια δεύτερη γνώμη.
- `homeScreenButton`: Το κουμπί που επιστρέφει τον χρήστη στην αρχική οθόνη.
- `previousScreenButton`: Το κουμπί που επιστρέφει τον χρήστη στην προηγούμενη οθόνη.

Μέθοδοι Κλάσης

- `HealthRequestGUI()`:
- `displayHealthRequest()`: Διαμορφώνει την οθόνη προβολής του `HealthRequest`.
- `clickAskForASecondOpinionButton(HealthRequestId)`: Σηματοδοτεί την έναρξη της διαδικασίας προώθησης του προκείμενου `HealthRequest` στον `HealthRequestController` έτσι ώστε να επιλεγεί διαφορετικός γιατρός από τον χρήστη και να λάβει μια δεύτερη γνώμη.
- `homeScreenButtonPress()`: Ελέγχει αν έχει πατηθεί το κουμπί `homeScreenButton`.
- `previousScreenButtonPress()`: Ελέγχει αν έχει πατηθεί το κουμπί `previousScreenButton`.

1.2.1.6 <MyHealthRequestListController>

MyHealthRequestListController
+MyHealthRequestListController()
+getRequestFromList(HealthRequestList, HealthRequestId): void
+duplicateHealthRequest(HealthRequest): HealthRequest
+continueRequestCreation(HealthRequest): HealthRequest
+Notification(): Notification
+setNotificationId(Diagnosis): void
+setNotificationText(Notification): void
+getNotificationText(): String
+editNotification(Notification): void
+displayNotification(Notification): void
+fetchList(patientId): HealthRequestList
+getHealthRequestList(HealthRequestList): Array{HealthRequestId hasReceivedDiagnosis SeqId}
+displayHealthRequestList(): void
+displayHealthRequest(): void

Η κλάση αυτή συντονίζει όλες τις λειτουργίες των κλάσεων σχετικών με Notification και MyHealthRequestList.

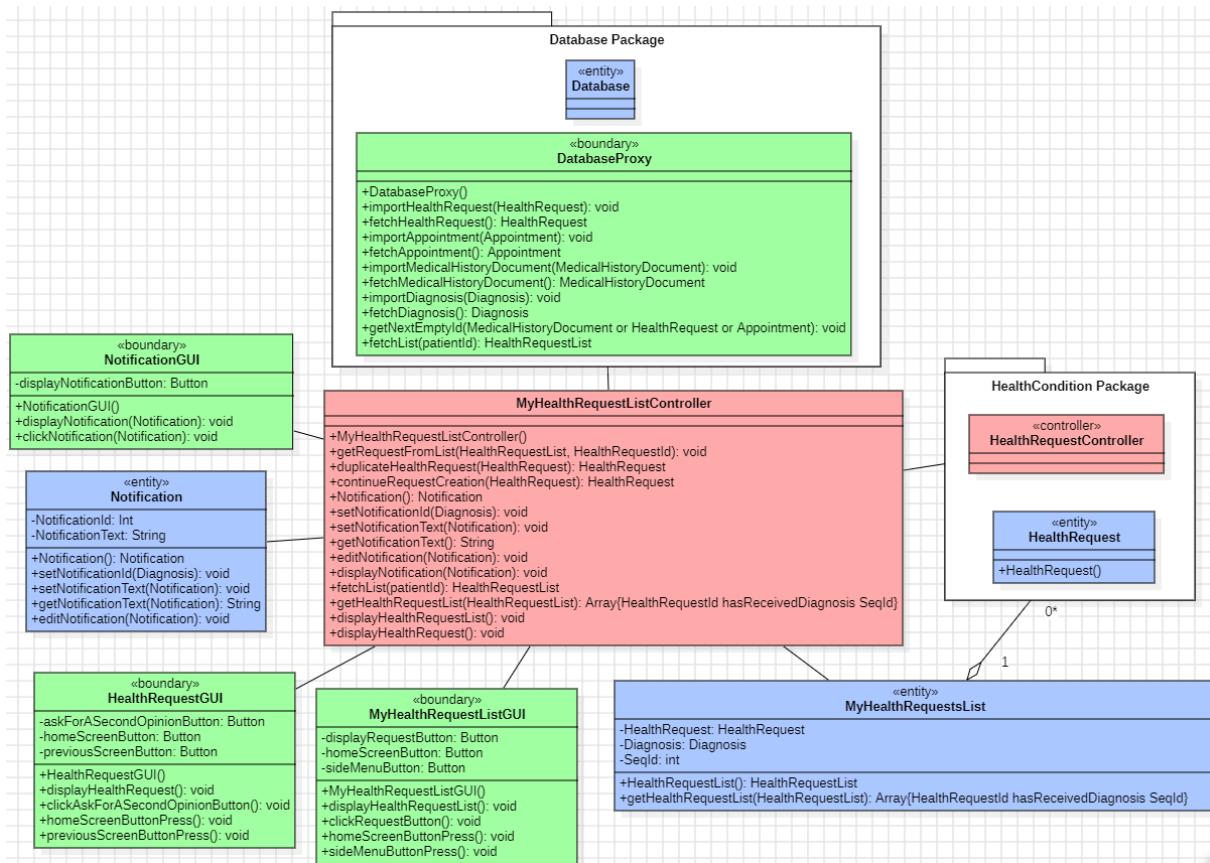
Μέθοδοι Κλάσης

- MyHealthRequestListController(): Ο constructor της κλάσης.
- getRequestFromList(HealthRequestList, HealthRequestId): Ψάχνει εντός της λίστας την σειρά με το ίδιο HealthRequestId για να προωθήσει την πληροφορία αυτή στη συνάρτηση προβολής του HealthRequest.
- DuplicateHealthRequest(HealthRequest): Παίρνει ένα HealthRequest από τη λίστα και αντιγράφει όλες τις παραμέτρους εκτός από εκείνες της επιλογής γιατρού. Επιστρέφει ένα ημιτελές HealthRequest έτοιμο να προωθηθεί στον χρήστη για τη συνέχεια της διαδικασίας AskForASSecondOpinion().
- continueRequestCreation(HealthRequest): Στέλνει το προαναφερθέν ημιτελές HealthRequest στον σχετικό controller για περαιτέρω επεξεργασία μέσω της αλληλεπίδρασης με τον χρήστη ασθενή.

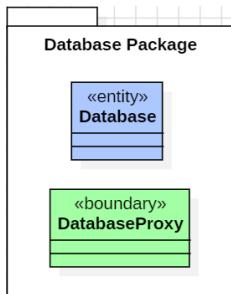
Οι παρακάτω συναρτήσεις καλώνται από τον controller αλλά περιγράφονται αναλυτικά κάτω από τις σχετικές κλάσεις.

- Notification()
- setNotificationId(Diagnosis)
- setNotificationText(Notification)
- getNotificationText()
- displayNotification(Notification)
- editNotification(Notification)
- fetchList(patientId)
- getHealthRequestList(HealthRequestList)
- displayHealthRequestList()
- displayHealthRequest()

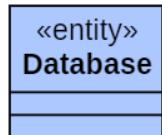
1.2.1.7 Class Diagram



1.2.2 Database Package



1.2.2.1 <Database>



Η κλάση Database είναι μία εξωτερική κλάση στην οποία βρίσκονται αποθηκευμένα όλα τα αντικείμενα μας. Καθώς είναι εξωτερική δεν θα την αναλύσουμε σε αυτό το παραδοτέο. Θα αναλύσουμε όμως την κλάση δίσιυλο μέσω της οποίας επικοινωνούμε για να αποκτήσουμε πρόσβαση στα αντικείμενα της βάσης.

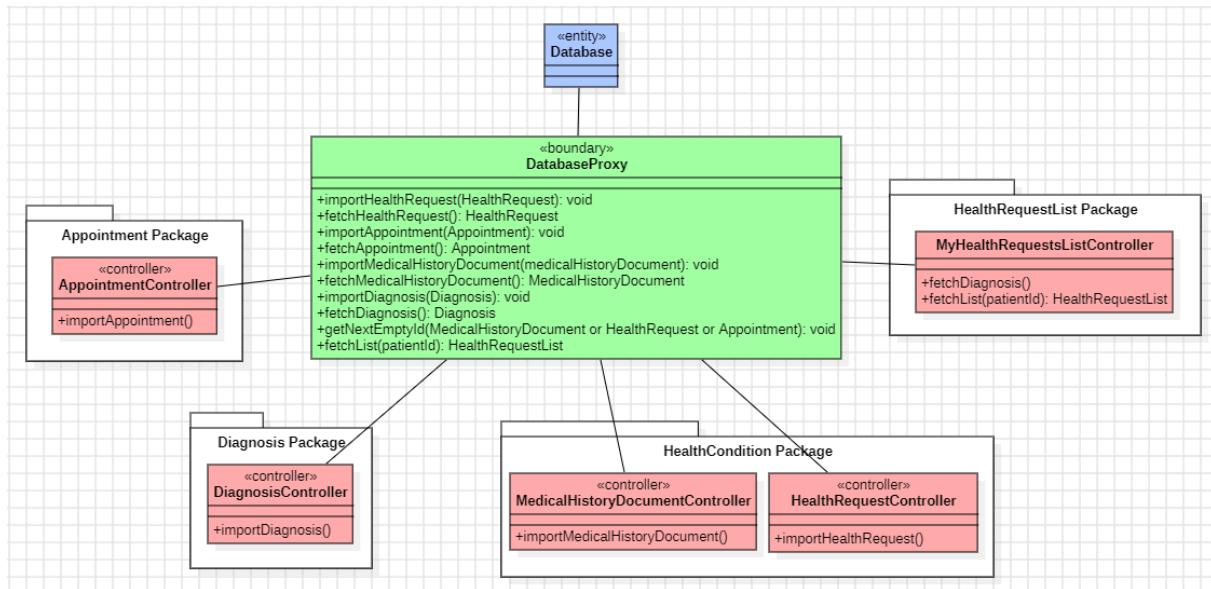
1.2.2.2 <DatabaseProxy>

«boundary»	DatabaseProxy
+DatabaseProxy()	
+importHealthRequest(HealthRequest): void	
+fetchHealthRequest(): HealthRequest	
+importAppointment(Appointment): void	
+fetchAppointment(): Appointment	
+importMedicalHistoryDocument(MedicalHistoryDocument): void	
+fetchMedicalHistoryDocument(): MedicalHistoryDocument	
+importDiagnosis(Diagnosis): void	
+fetchDiagnosis(): Diagnosis	
+getNextEmptyId(MedicalHistoryDocument or HealthRequest or Appointment): void	
+fetchList(patientId): HealthRequestList	

Μέθοδοι Κλάσης

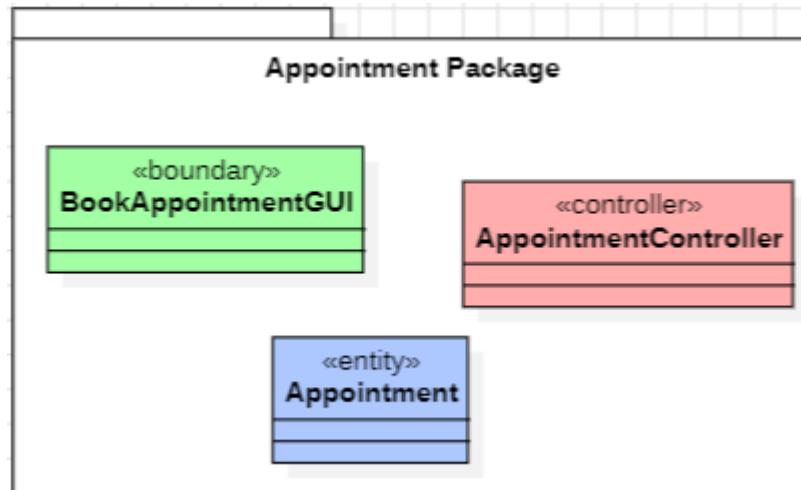
- DatabaseProxy(): Ο constructor της κλάσης.
- importHealthRequest(HealthRequest): Ο ασθενής κάνει import στο Database το HealthRequest. Η συνάρτηση getNextEmptyId() συνεισφέρει στην παραγωγή του HealthRequestId που το HealthRequest λαμβάνει.
- fetchHealthRequest(): Ο γιατρός όσο και ο ασθενής, κατά την προβολή του σχετικού HealthRequest, καλούν την συγκεκριμένη συνάρτηση η οποία παρέχει το αντικείμενο HealthRequest.
- importAppointment(Appointment): Ο ασθενής κάνει import στο Database το Appointment. Η συνάρτηση getNextEmptyId() συνεισφέρει στην παραγωγή του AppointmentId που το Appointment λαμβάνει.
- fetchAppointment(): Ο γιατρός ή/και ο ασθενής καλούν έμμεσα αυτή τη συνάρτηση για να προβάλουν στις οθόνες τους το Appointment.
- importMedicalHistoryDocument(medicalHistoryDocument): Ο ασθενής κάνει import στο Database το MedicalHistoryDocument. Η συνάρτηση getNextEmptyId() συνεισφέρει στην παραγωγή του MedicalHistoryDocumentId που το MedicalHistoryDocument λαμβάνει.
- fetchMedicalHistoryDocument(): Ο γιατρός ή/και ο ασθενής καλούν έμμεσα αυτή τη συνάρτηση για να προβάλουν στις οθόνες τους το MedicalHistoryDocument.
- importDiagnosis(Diagnosis): Ο γιατρός κάνει import στο Database την διάγνωση που έχει γράψει για ένα HealthRequest. Η διάγνωση αυτή είναι συνδεδεμένη με το αρχικό HealthRequest το οποίο κληροδοτεί και το Id του στο Diagnosis (HealthRequestId=DiagnosisId).
- fetchDiagnosis(): Όταν ο γιατρός κάνει importDiagnosis() τότε ενεργοποιείται αυτόματα η συνάρτηση fetchDiagnosis() της κλάσης DatabaseProxy έτσι ώστε να πυροδοτηθεί η διαδικασία δημιουργίας ειδοποίησης και οπτικής αναπαράστασης του HealthRequest σε συνδυασμό με το Diagnosis στο HealthRequestList.
- getNextEmptyId(MedicalHistoryDocument or HealthRequest or Appointment): Παράγει ένα μοναδικό αριθμό Id για όποιο import το χρειάζεται ενώ παράλληλα διασφαλίζει πως κάθε Id είναι μοναδικό.
- fetchList(patientId): Με βάση το patientId ψάχνει στη βάση δεδομένων HealthRequests και Diagnoses με ίδιο patientId με τον χρήστη που κάνει αυτή την αίτηση και τα στέλνει όλα μαζί ως HealthRequestList.

1.2.2.3 Class Diagram



1.3 <Πακέτα κλάσεων σχετικών με το Gherkin Feature (Σενάριο Χρήσης) 3: “Patient requests medical appointment”

1.3.1 Appointment package



1.3.1.1 <Appointment>

«entity»	
Appointment	
-doctorId: int	
-date: int	
-time: int	
-patientId: int	
-appointmentStatus	
-AppointmentId: int	
+Appointment(doctorId, date, time, patientId): void	
+setDate(): void	
+getTime(): void	
+getPatientID(): int	
+getDoctorID(): int	
+setPatientID(): void	
+setDoctorID(): void	
+getDate(): int	
+getTime(): int	
+setDoctorName(): void	
+setPatientName(): void	
+getPatientName(): string	
+getDoctorName(): string	

Η κλάση αυτή προβάλλει το αντικείμενο Appointment

Χαρακτηριστικά Κλάσης:

- doctorId: Ο μοναδικός αριθμός ταυτοποίησης του γιατρού που συμμετέχει στο εν λόγω ραντεβού
- date: Η ημερομηνία του ραντεβού
- time: Η ώρα του ραντεβού
- patientId: Ο μοναδικός αριθμός ταυτοποίησης του γιατρού που συμμετέχει στο εν λόγω ραντεβού
- appointmentStatus:string : Η κατάσταση του ραντεβού, πχ: ολοκληρωμένο, αναβληθέν, ακυρωμένο Κ.Ο.Κ.
- AppointmentId:int : Η τιμή Id που αναθέτουμε στο αντικείμενο για να τα διακρίνουμε ανά μεταξύ τους

Μέθοδοι Κλάσης

- Appointment(in doctorId, in date, in time, in patientId): void - Ο constructor της κλάσης, δέχεται ως παραμέτρους τους μοναδικούς αριθμούς γιατρού και ασθενή τύπου int, όπως και της ημερομηνίας και ώρας.
- setDate(): void – Θέτει την ημερομηνία του ραντεβού σύμφωνα με την αρχικοποίηση
- setTime(): void – Θέτει την ώρα του ραντεβού σύμφωνα με την αρχικοποίηση
- getPatientID(): int – Επιστρέφει τον μοναδικό αριθμό του ασθενή, patientId:int
- getDoctorID(): int – Επιστρέφει τον μοναδικό αριθμό του ιατρού, doctorId :int
- setPatientID(): void - Θέτει τον μοναδικό αριθμό του ασθενή σύμφωνα με την αρχικοποίηση
- setDoctorID(): void - Θέτει τον μοναδικό αριθμό του ιατρού σύμφωνα με την αρχικοποίηση
- getDate(): int – Επιστρέφει την ημερομηνία του ραντεβού σύμφωνα με την αρχικοποίηση, date: int
- getTime(): int - Επιστρέφει την ώρα του ραντεβού σύμφωνα με την αρχικοποίηση, time: int

- setPatientName(): void - Θέτει το όνομα του ασθενή
- setDoctorName(): void – Θέτει το όνομα του ιατρού
- getPatientName(): string – Επιστρέφει το όνομα του ασθενή
- getDoctorName(): string – Επιστρέφει το όνομα του ιατρού
- setAppointmentStatus(): void - Θέτει το στάδιο του ραντεβού
- getAppointmentStatus(): string – Επιστρέφει το στάδιο του ραντεβού, AppointmentStatus: string

1.3.1.2 <BookAppointmentGUI>

«boundary»	
BookAppointmentGUI	
-SearchDoctor: Textbox	
-Cancel: Button	
-Confirm: Button	
-Doctors: Array	
-DoctorSpecialty: Browse	
+BookAppointmentGUI(): void	
+SearchClick(): void	
+CancelClick(): void	
+ConfirmClick(): void	

Η κλάση αυτή εκφράζει την διεπαφή της οθόνης ιατρικών ραντεβού που αφορά τον χρήστη ασθενή

Χαρακτηριστικά Κλάσης

- SearchDoctor: Textbox: Περιοχή εισαγωγής κειμένου για την αναζήτηση ιατρού
- Cancel: Button: Κουμπί ακύρωσης διαδικασίας
- Confirm: Button: Κουμπί επιβεβαίωσης διαδικασίας
- Doctors: Array: Πίνακας προβολής διαθέσιμων ιατρών
- DoctorSpecialty: Browse: Αναζήτηση σε λίστα ιατρικών ειδικοτήτων

Μέθοδοι Κλάσης

- BookAppointmentGUI(): void: Ο constructor της διεπαφής
- SearchClick(): void: Η συνάρτηση SearchClick αναζητά διαθέσιμους ιατρούς με βάση το όνομά
- CancelClick(): void: Η συνάρτηση CancelClick ακυρώνει την διαδικασία υποβολής ενός ραντεβού
- ConfirmClick(): void: Η συνάρτηση ConfirmClick επιβεβαιώνει την διαδικασία υποβολής ενός ραντεβού

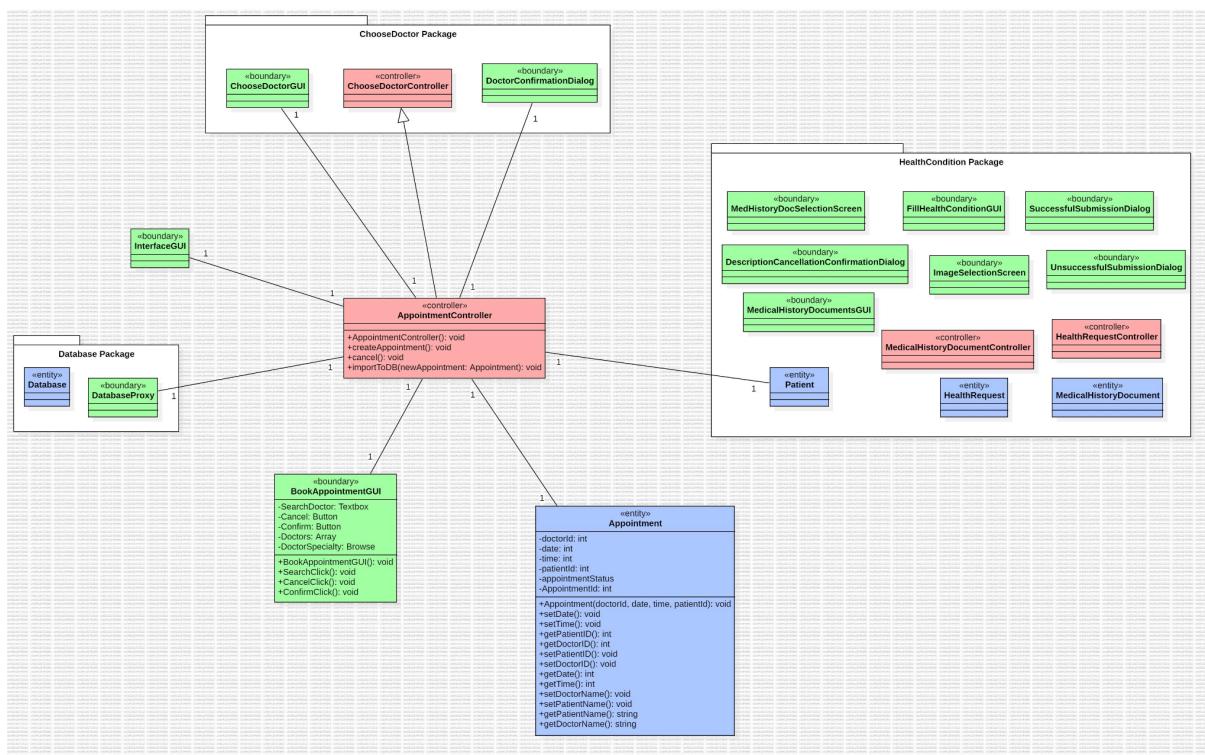
1.3.1.3 <AppointmentController>

«controller»
AppointmentController
+AppointmentController(): void
+createAppointment(): void
+cancel(): void
+importAppointment(newAppointment: Appointment): void

Η κλάση αυτή εκφράζει τον ελεγκτή (κλάση ελέγχου) για την εισαγωγή και διαχείριση των ιατρικών ραντεβού (οντότητες της κλάσης Appointment) στο σύστημα. Ο ελεγκτής περιέχει όλες τις συναρτήσεις/μεθόδους που χρειάζονται για την υλοποίηση της λειτουργικότητας σχετικά με την εισαγωγή και διαχείριση των ιατρικών ραντεβού μέσα στο σύστημα.

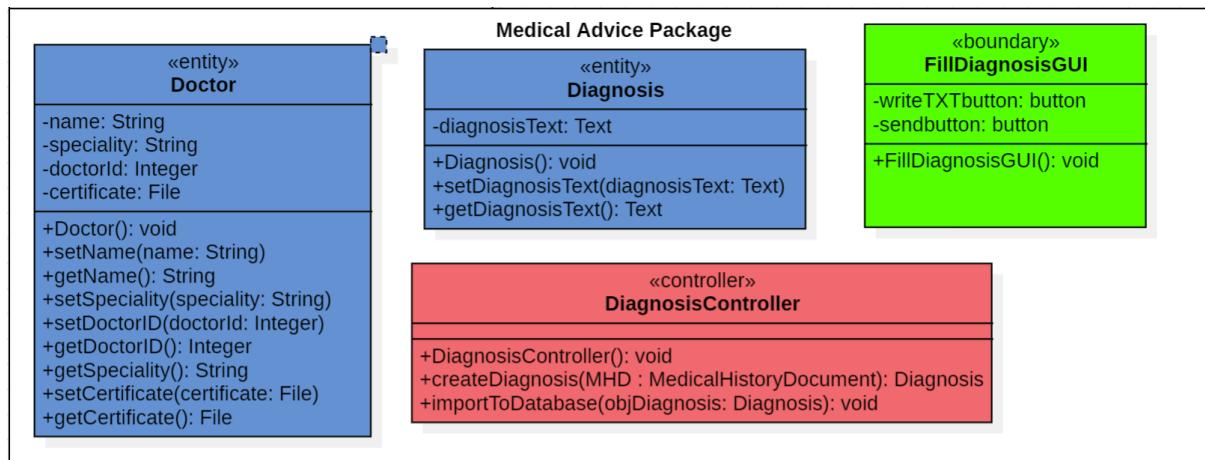
Μέθοδοι Κλάσης

- AppointmentController: void: Η μέθοδος AppointmentController αποτελεί τη συνάρτηση δόμησης του ελεγκτή.
- createAppointment(doctorId: int, date: int, time: int, patientId: int) AppointmentId - Η μέθοδος createAppointment δημιουργεί ένα αντικείμενο της κλάσης Appointment, δηλαδή ένα ιατρικό ραντεβού, λαμβάνοντας ως ορίσματα τους χαρακτηριστικούς αριθμούς του ιατρού και του ασθενή, καθώς και την ημερομηνία και ώρα του ραντεβού. Η μέθοδος στην υλοποίηση της, καλεί τη συνάρτηση δόμησης της κλάσης Appointment και έπειτα τις μεθόδους setDate, setTime, setPatientID, setDoctorId της ίδιας κλάσης, προκειμένου να δομήσει το αντικείμενο που κατασκευάζεται. Η μέθοδος επιστρέφει τον χαρακτηριστικό αριθμό του ραντεβού.
- cancelAppointment (): void - Η μέθοδος cancelAppointment καλείται για την ακύρωση μιας φόρμας ραντεβού, και διαγραφή όλων των αντικειμένων που δημιουργήθηκαν από ενέργειες του “ασθενή” στη συγκεκριμένη.
- importAppointment(newAppointment: Appointment) :void- Η μέθοδος importAppointmentToDB εισάγει ένα αντικείμενο της κλάσης Appointment, δηλαδή ένα ραντεβού, το οποίο δέχεται ως όρισμα, στη Βάση Δεδομένων του συστήματος.



1.4 <Πακέτα κλάσεων σχετικών με το Gherkin Feature (Σενάριο Χρήσης) 6: “Doctor gives advice to patient ”>

1.3.1 Πακέτο Medical Advice



Το πακέτο περιλαμβάνει τις κλάσεις που σχετίζονται με την παροχή ιατρικής συμβουλής από τον γιατρό στον ασθενή.

1.4.1.1 <Doctor>

«entity»	
Doctor	
-name:	String
-speciality:	String
-doctorId:	Integer
-certificate:	File
+Doctor():	void
+setName(name: String)	
+getName():	String
+setSpeciality(speciality: String)	
+setDoctorID(doctorId: Integer)	
+getDoctorID():	Integer
+getSpeciality():	String
+setCertificate(certificate: File)	
+getCertificate():	File
+readMHD(MHD: Medical History Document):	void

Η κλάση αυτή περιέχει όλα τα χαρακτηριστικά/δεδομένα και τις συναρτήσεις/μεθόδους που αφορούν την οντότητα “γιατρός”.

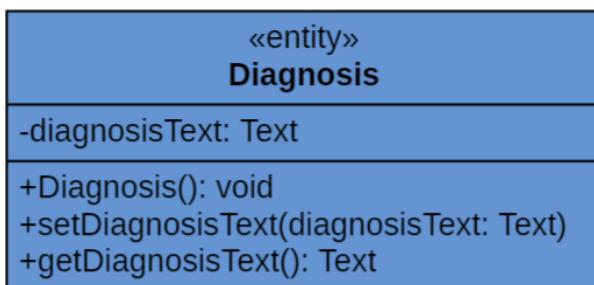
Χαρακτηριστικά Κλάσης

- name: String - Το όνομα του “γιατρού” στην εφαρμογή.
- speciality: String - Η ιατρική ειδικότητα του “γιατρού”.
- DoctorId : integer - Ο μοναδικός χαρακτηριστικός ακέραιος αριθμός του “γιατρού” στην εφαρμογή.
- certificate: File - Το έγγραφο του πτυχίου του “γιατρού” , για τυχόν εξακρίβωση από τον “ασθενή”.

Μέθοδοι Κλάσης

- Doctor(): void – Η μέθοδος Doctor αποτελεί τη συνάρτηση δόμησης της οντότητας Doctor.
- setName(name: String): void – Η μέθοδος setName θέτει το όνομα του “γιατρού”.
- setSpeciality(speciality: String): void – Η μέθοδος setSpeciality θέτει την ιατρική ειδικότητα του “γιατρού”.
- setDoctorID(doctorID : integer): void – Η μέθοδος setDoctorID θέτει το μοναδικό χαρακτηριστικό ακέραιο αριθμό του “γιατρού”.
- setCertificate(certificate : Text): void – Η μέθοδος setCertificate θέτει το έγγραφο του πτυχίου του “γιατρού”.
- getName(): String– Η μέθοδος getName επιστρέφει το όνομα του “γιατρού”.
- getSpeciality(): String– Η μέθοδος getSpeciality επιστρέφει την ιατρική ειδικότητα του “γιατρού”.
- getDoctorID(): integer– Η μέθοδος getDoctorID επιστρέφει το μοναδικό χαρακτηριστικό ακέραιο αριθμό του “γιατρού”.
- getCertificate(): File– Η μέθοδος getCertificate επιστρέφει το έγγραφο του πτυχίου του “γιατρού”.
- ReadMHD(MHD: Medical History Document): void - Η μέθοδος μέσω της οποίας ο “γιατρός” διαβάζει ένα Medical History Document κάποιου ασθενή.

1.4.1.2 <Diagnosis>



Η κλάση αυτή περιέχει όλα τα χαρακτηριστικά/δεδομένα και τις συναρτήσεις/μεθόδους που αφορούν την οντότητα “Ιατρική συμβουλή” μέσα στο σύστημα.

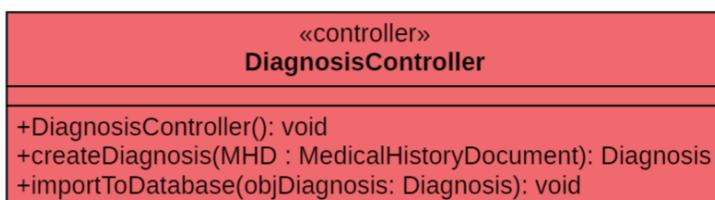
Χαρακτηριστικά Κλάσης

- diagnosisText : Text - Το κείμενο που αποτελεί την “Ιατρική συμβουλή” του χρήστη “γιατρού” στον “ασθενή”.

Μέθοδοι Κλάσης

- Diagnosis(): void – Η μέθοδος Diagnosis αποτελεί τη συνάρτηση δόμησης της οντότητας Diagnosis.
- setDiagnosisText (DiagnosisText: Text): void – Η μέθοδος setDiagnosisText θέτει την “Ιατρική συμβουλή” του χρήστη “γιατρού” στον “ασθενή”.
- getDiagnosisText (): Text – Η μέθοδος getDiagnosisText επιστρέφει την “Ιατρική συμβουλή” του χρήστη “γιατρού” στον “ασθενή”.

1.4.1.3 <DiagnosisController>



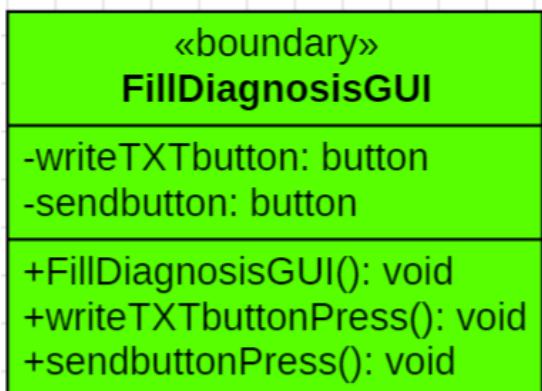
Η κλάση αυτή μοντελοποιεί τον ελεγκτή (κλάση ελέγχου) για την εισαγωγή και διαχείριση των εγγράφων “Ιατρικής συμβουλής” (οντότητες της κλάσης Diagnosis) στο σύστημα. Ο ελεγκτής περιέχει όλες τις συναρτήσεις/μεθόδους που χρειάζονται για την υλοποίηση της λειτουργικότητας σχετικά με την εισαγωγή και διαχείριση των εγγράφων “Ιατρικών συμβουλών” μέσα στο σύστημα.

Μέθοδοι Κλάσης

- DiagnosisController(): void – Η μέθοδος DiagnosisController αποτελεί τη συνάρτηση δόμησης του ελεγκτή.
- CreateDiagnosis(MHD : MedicalHistoryDocument) : Diagnosis - Η μέθοδος CreateDiagnosis δημιουργεί ένα αντικείμενο της κλάσης Diagnosis, λαμβάνοντας ως ορίσμα το αρχείο που περιέχει το πραγματικό έγγραφο ιατρικού ιστορικού του χρήστη “ασθενή”. Η μέθοδος στην υλοποίηση της, καλεί τη συνάρτηση δόμησης της κλάσης Diagnosis και έπειτα την μέθοδο setDiagnosisText της ίδιας κλάσης, προκειμένου να δομήσει το αντικείμενο που κατασκευάζεται. Η μέθοδος επιστρέφει την “Ιατρική συμβουλή” που δημιούργησε.

- importToDatabase(objDiagnosis: Diagnosis): void – Η μέθοδος importToDatabase εισάγει ένα αντικείμενο της κλάσης Diagnosis, δηλαδή ένα έγγραφο “ιατρικής συμβουλής”, το οποίο δέχεται ως όρισμα, στη Βάση Δεδομένων του συστήματος. Η μέθοδος στην υλοποίηση της καλεί τη μέθοδο importDiagnosis της κλάσης DBProxy, προκειμένου το αντικείμενο να εισαχθεί στη Βάση Δεδομένων.

1.4.1.4 <FillDiagnosisGUI>



Η κλάση αυτή εκφράζει τη διεπαφή της οθόνης σύνταξης της “ιατρικής συμβουλής” από τον χρήστη “γιατρό” και την αποστολή της στον “ασθενή”.

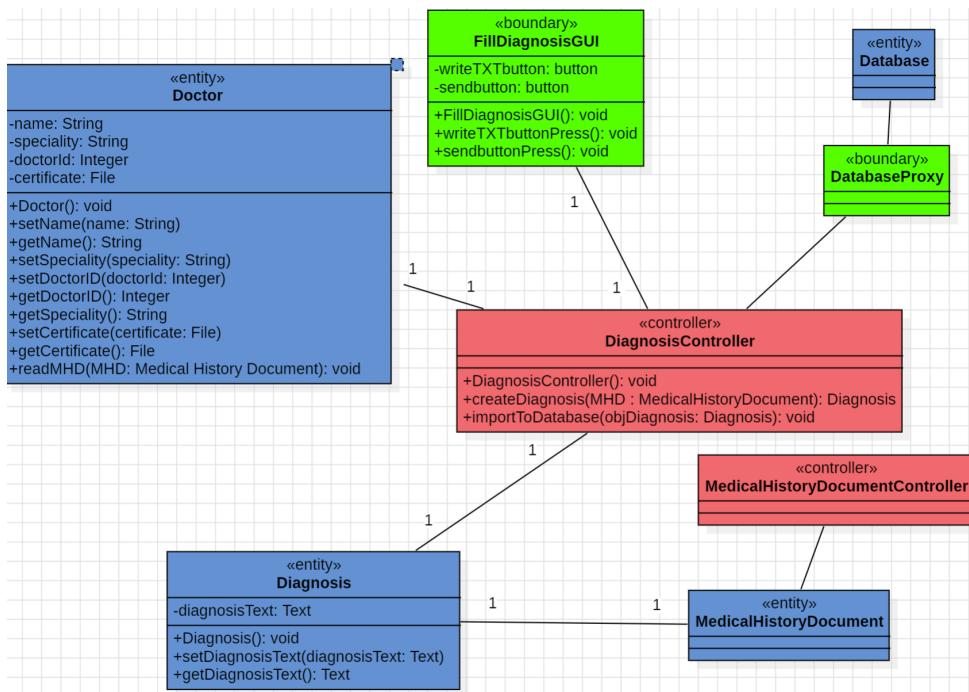
Χαρακτηριστικά Κλάσης

- writeTXTbutton: button – Κουμπί για έναρξη της σύνταξης του κειμένου “ιατρικής συμβουλής”.
- sendbutton: button – Κουμπί αποστολής του κειμένου “ιατρικής συμβουλής” στον “ασθενή”.

Μέθοδοι Κλάσης

- FillDiagnosisGUI(): void – Η μέθοδος FillDiagnosisGUI αποτελεί τη συνάρτηση δόμησης της διεπαφής αυτής.
- writeTXTbuttonPress() : void - Η μέθοδος writeTXTbuttonPress αποτελεί τη συνάρτηση που πραγματοποιεί το πάτημα του κουμπιού “write TXT” και μεταφέρει τον “γιατρό” στον κειμενογράφο για την σύνταξη του κειμένου της “ιατρικής συμβουλής”.
- sendbuttonPress() : void - Η μέθοδος sendbuttonPress αποτελεί τη συνάρτηση που πραγματοποιεί το πάτημα του κουμπιού “send” και στέλνει στον “ασθενή” το κείμενο της “ιατρικής συμβουλής” που συνέταξε ο γιατρός μέσω της μεθόδου createDiagnosis του ελεγκτή DiagnosisController.

1.4.1.5 Διάγραμμα κλάσεων

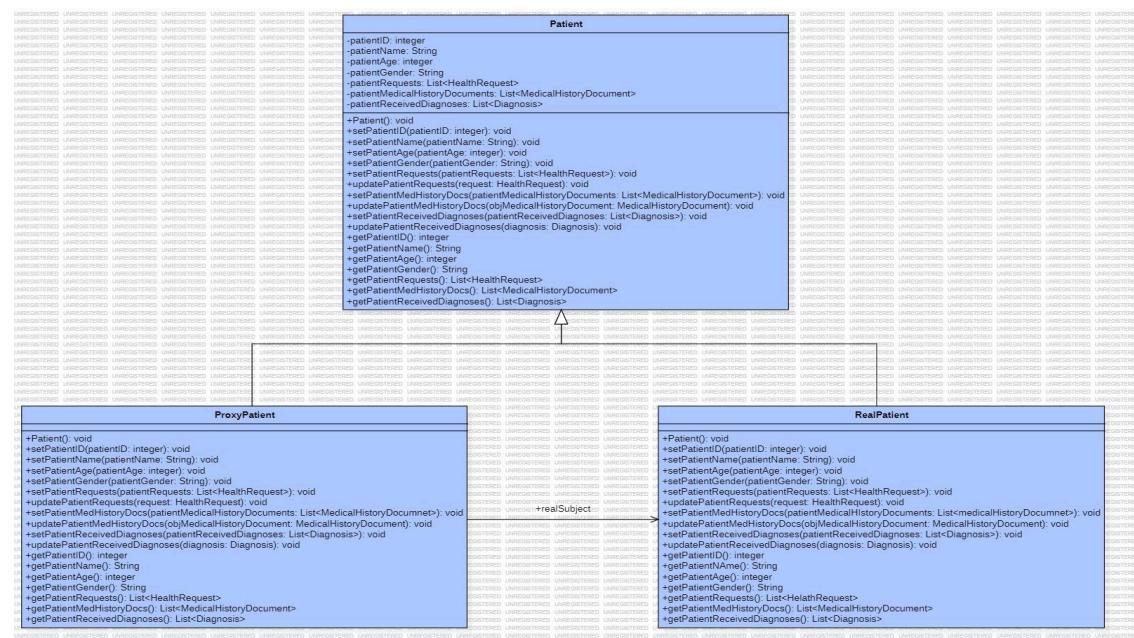


2 Πρότυπα Σχεδίασμού που υιοθετήθηκαν

2.1 Πρότυπο Proxy

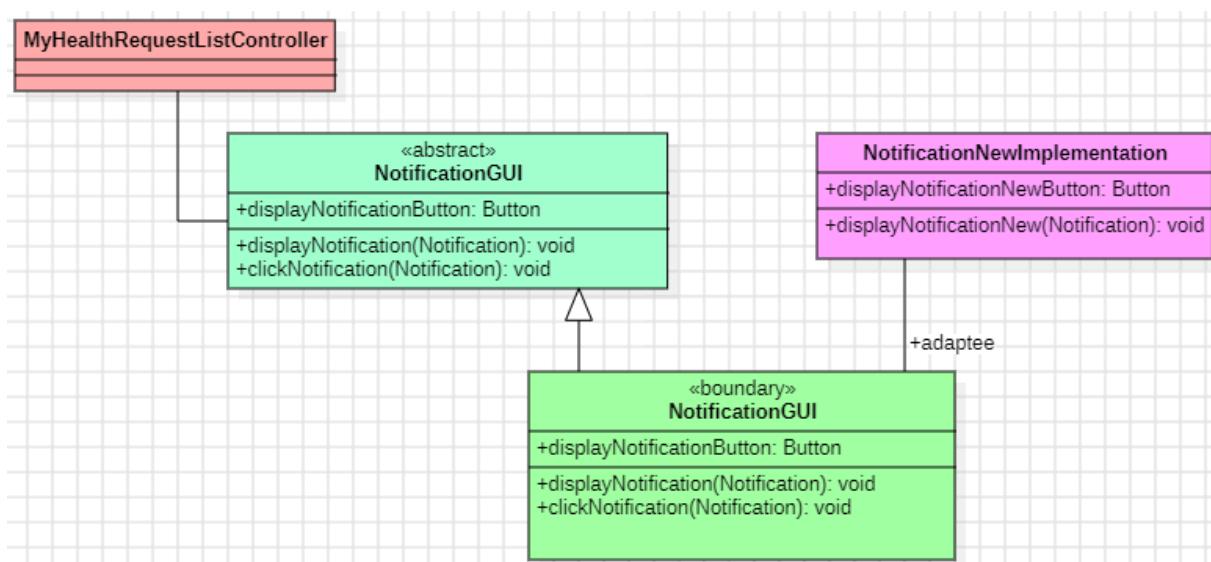
Το πρότυπο Proxy είναι ένα δομικό πρότυπο σχεδίασης. Γενικότερα, σαν πρότυπο σχεδίασης χρησιμοποιεί ένα “proxy” αντικείμενο, το οποίο λειτουργεί σαν αντικαταστάτης του πραγματικού αντικειμένου που ορίζεται εντός του συστήματος λογισμικού. Το proxy αντικείμενο είναι αυτό, που είναι συνεχώς προσπελάσιμο εντός του συστήματος, ενώ το πραγματικό αντικείμενο όχι. Το proxy δημιουργεί το πραγματικό αντικείμενο μόνον όταν απαιτείται από τον χρήστη ή την λειτουργικότητα του συστήματος. Στο σύστημα /gnosis, το συγκεκριμένο πρότυπο σχεδίασης εφαρμόζεται με στόχο την κάλυψη της ΜΛΑ-3, η οποία ορίστηκε στο πλαίσιο των απαιτήσεων χρηστών, αφορά στις απαιτήσεις ασφάλειας-πολιτικής του συστήματος και περιγράφει την απαίτηση ασφάλειας των προσωπικών δεδομένων των “ασθενών” εντός της εφαρμογής, κατά τους κανόνες GDPR που θεστίζονται από την Ε.Ε. (ΜΛΑ-3: Το σύστημα πρέπει να συμμορφώνεται με τους κανονισμούς της Ευρωπαϊκής Ένωσης GDPR). Κάθε χρήστης “ασθενής”, στο σύστημα, είναι μια οντότητα της κλάσης Patient, η οποία περιλαμβάνει χαρακτηριστικά και μεθόδους που σχετίζονται με τα πραγματικά προσωπικά στοιχεία, ιατρικά δεδομένα όπως ζητήματα, έγγραφα, ραντεβού, τα αφορούν τον ασθενή ως φυσικό πρόσωπο. Τα στοιχεία αυτά πρέπει να προστατεύονται και συνεπώς ένα αντικείμενο της κλάσης Patient, που έχει δημιουργηθεί και υπάρχει στο σύστημα, θα πρέπει να είναι προσβάσιμο μόνο από τον ίδιο τον χρήστη “ασθενή” και κατά περίπτωση από τον χρήστη “γιατρό”, στον οποίο ο “ασθενής” αιτείται. Όλοι οι δράστες του συστήματος, συνεπώς, δεν έχουν ίδια δικαιώματα πρόσβασης (ή και καθόλου) σε ένα συγκεκριμένο αντικείμενο της κλάσης Patient. Για τον λόγο αυτό, αξιοποιείται το πρότυπο σχεδίασης Proxy και συγκεκριμένα εφαρμόζουμε ένα Protection

Proxy στην κλάση Patient. Σύμφωνα με το Protection Proxy, χρησιμοποιείται ένα proxy αντικείμενο ProxyPatient και το πραγματικό αντικείμενο RealPatient, τα οποία κληρονομούν την κλάση Patient προκειμένου να καθοριστούν οι διεπαφές που χρησιμοποιούνται από αυτά. Το proxy αντικείμενο είναι αυτό που είναι συνεχώς προσπελάσιμο στο σύστημα, ενώ δικαιώματα πρόσβασης στο πραγματικό αντικείμενο RealPatient, που μοντελοποιεί τον “ασθενή” με τα προσωπικά του δεδομένα, δίνονται μέσω της εξουσιοδότησης από το proxy στο πραγματικό.



2.2 Πρότυπο Adapter

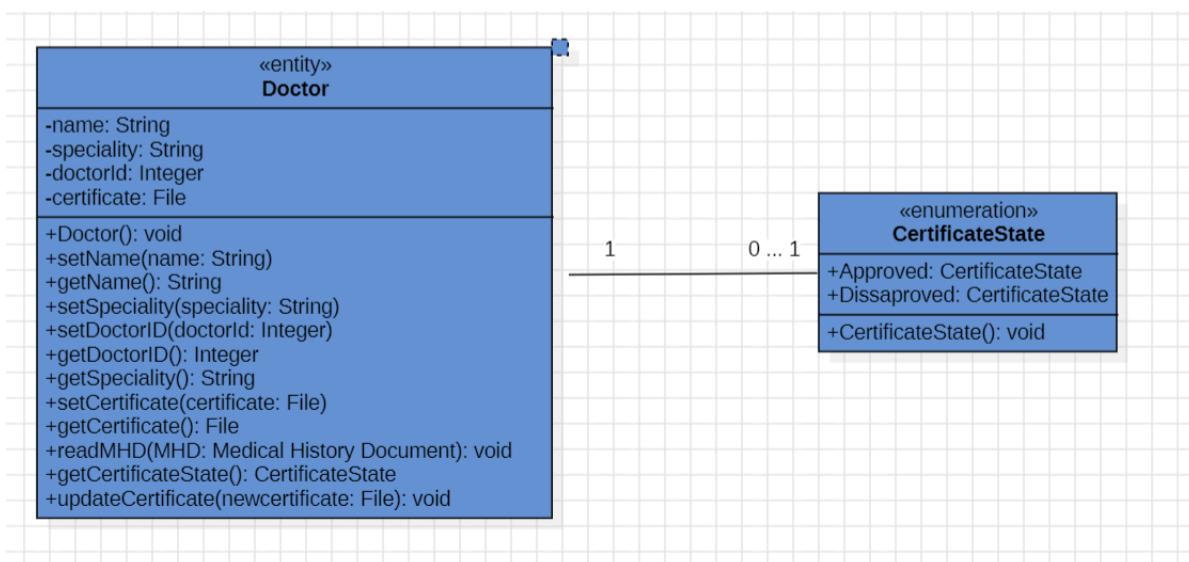
Η εφαρμογή μας είναι μια εφαρμογή κινητού. Καθώς τα κινητά λαμβάνουν ενημερώσεις λογισμικού είναι σημαντικό για την διασφάλισης μιας θετικής και ανανεωμένης εμπειρίας χρήστη να προσαρμόζουμε το User Interface μας ανάλογα. Σκοπός είναι να ταιριάζει περισσότερο με τα νέα δεδομένα και να αξιοποιεί ενδεχομένως νέες δυνατότητες απεικόνισης της εφαρμογής, όπως το “dynamic island” στα iPhone. Με αυτόν τον τρόπο απευθυνόμαστε έμεσα στις προκλήσεις της ΜΛΑ-2. Ως παράδειγμα θα αναφερθούμε στο NotificationGUI το οποίο είναι ένα, αλλά σίγουρα όχι το μόνο, GUI της εφαρμογής που μπορεί να επιωφεληθεί από διαφορετικές αναπαραστάσεις. Ακόμη η συγκεκριμένη boundary κλάση μπορεί να επεκταθεί και να προβάλει ειδοποιήσεις και σε άλλα μέρη του UI του κινητού ενώ μπορούν να προστεθούν και άλλα είδη δεδομένων στην εξίσωση εκτός του NotificationText. Στο διάγραμμα φαίνεται πως το adapter design pattern μας επιτρέπει να υλοποιούμε καινούριες συναρτήσεις απεικόνισης των γραφικών στοιχείων. Μπορούμε να αναπαραστήσουμε την ίδια πληροφορία με διαφορετικό τρόπο χρησιμοποιώντας ίσως καινούριες βιβλιοθήκες σε γλώσσες διαφορετικές από αυτήν που φτιάχτηκε η εφαρμογή. Επομένως το Adapter design pattern μας προσδίδει προσαρμοστικότητα όσον αφορά το πως υλοποιούμε τα γραφικά στοιχεία και σε τι γλώσσα τα υλοποιούμε και ευκολία στο να αλλάξουμε ριζικά την εμφάνιση χωρίς να χαλάμε την δομή της εφαρμογής.



2.3 Πρότυπο Balking

Το συγκεκριμένο πρότυπο σχεδίασης , γενικά χρησιμοποιείται για τον αποκλεισμό ενεργειών κάποιων κλάσεων , όταν αυτές δεν βρίσκονται σε μια συγκεκριμένη κατάσταση. Στο σύστημα /gnosis, το συγκεκριμένο πρότυπο σχεδίασης εφαρμόζεται με στόχο την κάλυψη της ΜΛΑ-1, δηλαδή την διασφάλιση της ύπαρξης επικυρωμένης ιατρικής άδειας που να συμμορφώνεται με την ευρωπαϊκή νομοθεσία για κάθε “γιατρό” της εφαρμογής. Σε περίπτωση που κάτι τέτοιο δεν ισχύει για κάποιον “γιατρό” , θα πρέπει να αποκλείεται η συμμετοχή του από την εφαρμογή , με τους χρήστες να μην του απευθύνουν ιατρικά ερωτήματα και να μην λαμβάνουν υπόψιν τους τυχόν “ιατρικές συμβουλές” του. Για την υλοποίηση , τώρα , αυτού του προτύπου σχεδίασης , τροποποιούμε στην ουσία την κλάση Doctor της οντότητας “γιατρός” ως εξής :

Δημιουργούμε δυο συμπληρωματικές μεθόδους εντός της κλάσης Doctor , την `getCertificateState() : CertificateState` , που επιστρέφει την κατάσταση της ιατρικής άδειας του εκάστοτε “γιατρού” και την `updateCertificate(newcertificate : File) : void` για τυχόν εισαγωγή νέου επικαιροποιημένου εγγράφου(βασική αρχή του balking design pattern είναι πως μπορεί να αλλάζει η κατάσταση σε άγνωστο χρονικό διάστημα και δεν παραμένει πάντα σταθερή) . Επιπλέον , δημιουργούμε την συμπληρωματική κλάση `CertificateState` , που αποτελεί την κλάση μοντελοποίησης των πιθανών καταστάσεων του εγγράφου της ιατρικής άδειας , αποτελούμενη από δύο μεταβλητές κατάστασης και έναν constructor. Αναλυτικά το διάγραμμα κλάσεων :



3 Δυναμική Μοντελοποίηση

3.1 Gherkin Scenario 1 “Successfully describe health condition” (Feature 1): “Patient describes health condition”

Το πακέτο χρήστης που θα παρουσιαστεί παρακάτω περιλαμβάνει δύο διαφορετικές εκδοχές του gherkin scenario “Successfully describe health condition” του gherkin feature “Patient describes health condition”, τα οποία αφορούν τα scenarios: “successfully describe health condition”, “unsuccessfully describe health condition”, δηλαδή το happy και ένα unhappy path αυτής της λειτουργικότητας. Το πακέτο χρήστης αυτό αφορά την περιγραφή και εισαγωγή της κατάστασης της υγείας του χρήστη “ασθενή” στο σύστημα και περιλαμβάνει την οθόνη περιγραφής της κατάστασης της υγείας του “ασθενή”, στην οποία ο χρήστης αυτός συμπληρώνει την φόρμα υποβολής του ιατρικού του ζητήματος. Στο πακέτο χρήστης αυτό παρουσιάζονται η επιτυχής περίπτωση περιγραφής, μέσω κειμένου και, ακόμη, εισαγωγή φωτογραφίας που συνοδεύει την περιγραφή και η ανεπιτυχής περίπτωση περιγραφής μόνο με την εισαγωγή φωτογραφίας στο σύστημα και υποβολή χωρίς την περιγραφή μέσω κειμένου. Το πακέτο χρήστης περιλαμβάνει, επίσης, την οθόνη επιλογής εικόνας, καθώς και τα παράθυρα διαλόγου επιτυχούς και ανεπιτυχούς υποβολής της κατάστασης της υγείας από το χρήστη “ασθενή”.

Οι λειτουργικές απαιτήσεις που σχετίζονται με αυτό το πακέτο χρήστης:

- <ΛΑ-1> Ο ασθενής πρέπει να μπορεί να περιγράψει την κατάσταση της υγείας του μέσω κειμένου.
- <ΛΑ-2> Ο ασθενής πρέπει να μπορεί να εισάγει εικόνες στη βάση δεδομένων.

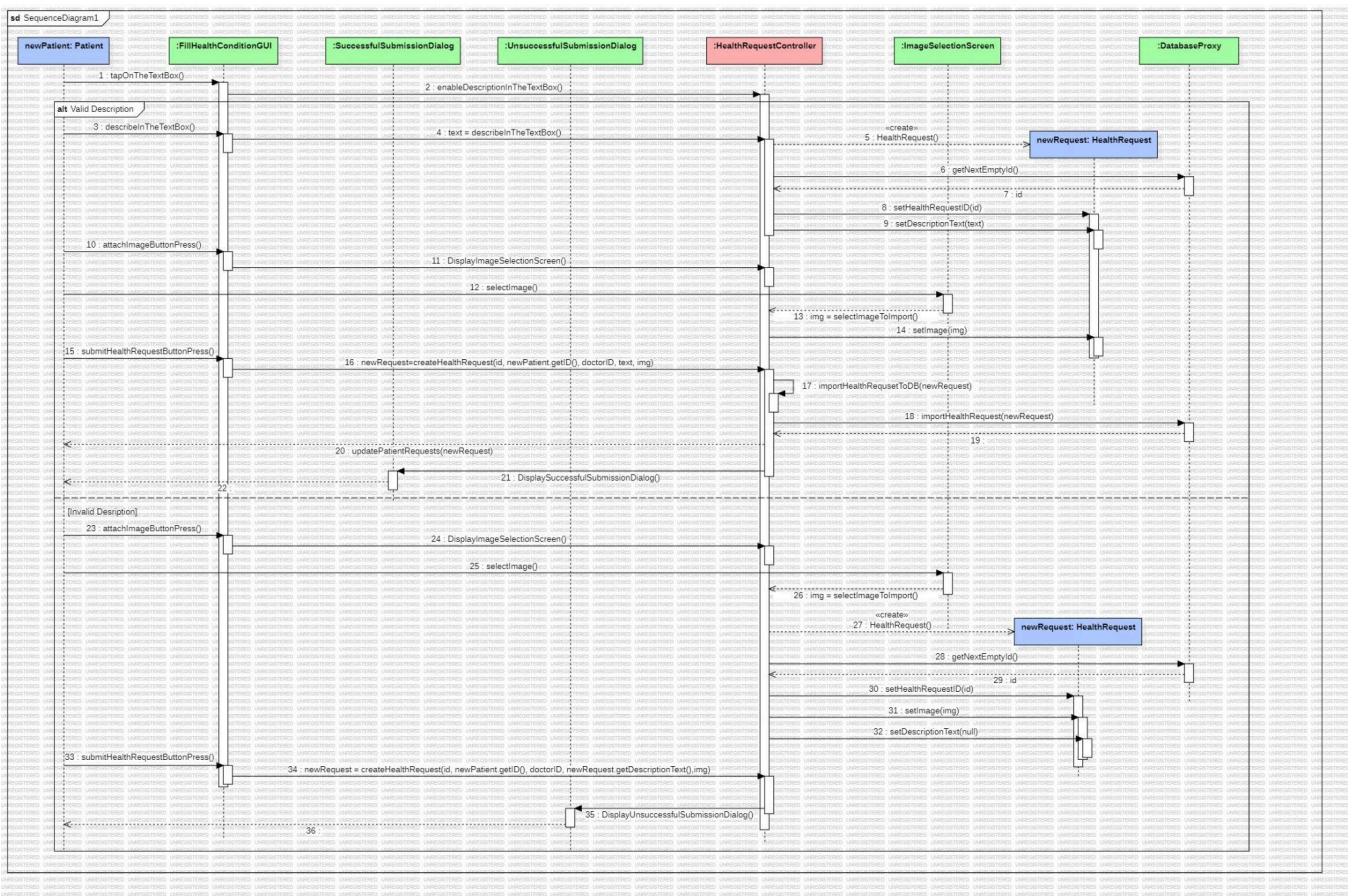
3.1.1 Αφήγηση Σεναρίου

Ο χρήστης “ασθενής” έχει επιλέξει “Αίτηση για ιατρική συμβουλή” και η τρέχουσα οθόνη που εμφανίζεται είναι η οθόνη περιγραφής της κατάστασης της υγείας του “ασθενή”, η οποία είναι ένα αντικείμενο της κλάσης FillHealthConditionGUI. Ο χρήστης «αγγίζει εντός του πλαισίου κειμένου», οπότε καλείται η συνάρτηση tapOnTheTextBox της FillHealthConditionGUI, η οποία καλεί την μέθοδο enableDescriptionInTheTextBox του ελεγκτή HealthRequestController και ο

χρήστης μπορεί να εισάγει κείμενο περιγραφής του ιατρικού του ζητήματος στο πλαίσιο κειμένου. Στην περίπτωση της επιτυχούς περιγραφής της κατάστασης της υγείας του “ασθενή”, ο χρήστης εισάγει κείμενο περιγραφής του ιατρικού του ζητήματος, οπότε καλείται η συνάρτηση describeInTextBox της FillHealthConditionGUI, η οποία καλεί την συνάρτηση describeInTextBox του ελεγκτή HealthRequestController, δημιουργώντας ένα αντικείμενο text (κείμενο) της κλάσης Text. Ο ελεγκτής αυτός δημιουργεί τότε ένα αντικείμενο newRequest της κλάσης HealthRequest («Ιατρικό Αίτημα») μέσω της μεθόδου HealthRequest της εν λόγω κλάσης και αμέσως καλείται η μέθοδος getNextEmptyId της κλάσης DatabaseProxy, προκειμένου να επιστραφεί στον ελεγκτή ένας διαθέσιμος χαρακτηριστικός ακέραιος αριθμός id, ο οποίος ανατίθεται ως χαρακτηριστικό του αντικειμένου newRequest, μέσω της μεθόδου setHealthRequestId της κλάσης HealthRequest. Το text (κείμενο) ανατίθεται ως χαρακτηριστικό του αντικειμένου αυτού, μέσω της μεθόδου setDescriptionText της κλάσης HealthRequest.

Έπειτα, ο χρήστης επιλέγει «Επισύναψη Εικόνας», οπότε καλείται η συνάρτηση attachImageButtonPress της FillHealthConditionGUI, η οποία καλεί την μέθοδο DisplayImageSelectionScreen του ίδιου ελεγκτή για προβολή της οθόνης επιλογής εικόνας, που είναι ένα αντικείμενο της κλάσης ImageSelectionScreen. Ο χρήστης «επιλέγει την εικόνα» που επιθυμεί για εισαγωγή στο σύστημα μέσω της αντίστοιχης γραφικής διεπαφής της οθόνης επιλογής εικόνας, οπότε καλείται η συνάρτηση selectImage της ImageSelectionScreen, η οποία καλεί τη μέθοδο selectImageToImport του ελεγκτή HealthRequestController, που δημιουργεί ένα αντικείμενο img (εικόνα) της κλάσης Image και κλείνει την οθόνη επιλογής εικόνας, επιστρέφοντας τον χρήστη στην οθόνη περιγραφής της κατάστασης της υγείας. Η εικόνα img τίθεται ως χαρακτηριστικό του αντικειμένου newRequest, μέσω της μεθόδου setImage. Στη συνέχεια, ο χρήστης επιλέγει «Υποβολή», οπότε καλείται η συνάρτηση submitHealthRequestButtonPress της FillHealthConditionGUI, η οποία καλεί τη μέθοδο createHealthRequest του ίδιου ελεγκτή (το όρισμα doctorID αναφέρεται στον ID αριθμό του γιατρού που έχει επιλέξει ο χρήστης σε προηγούμενη οθόνη, gherkin scenario του feature “Patient asks for medical advice”, το οποίο δεν εξετάζεται στο συγκεκριμένο πακέτο χρήσης) που δημιουργεί ένα αντικείμενο newRequest τύπου HealthRequest με όλα τα χαρακτηριστικά του newRequest που είχε δημιουργηθεί και το ID του “γιατρού” επιλέχθηκε και του “ασθενή” που δρά. Ο ελεγκτής εισάγει αυτό το «Ιατρικό Αίτημα» στη βάση δεδομένων μέσω της μεθόδου ImportHealthRequestToDB, που καλεί την ImportHealthRequest της κλάσης DatabaseProxy. Η συνάρτηση updatePatientRequests καλείται και ενημερώνει τα «Ιατρικά Αιτήματα» του “ασθενή”. Ο ελεγκτής καλεί τη μέθοδο DisplaySuccessfulSubmissionDialog και προβάλει το παράθυρο διαλόγου επιτυχούς υποβολής της κατάστασης της υγείας στον “ασθενή”. Στην περίπτωση της μη επιτυχούς περιγραφής, ο χρήστης “ασθενής” ακολουθεί όλα τα παραπάνω βήματα που περιγράφηκαν παραλείποντας στην αρχή να εισάγει κείμενο περιγραφής του ιατρικού του ζητήματος. Στην περίπτωση αυτή ο ελεγκτής HealthRequestController δημιουργεί το αντικείμενο newRequest τύπου HealthRequest, όταν ο “ασθενής” επιλέγει «Επισύναψη Εικόνας» οπότε στην ακολουθία ενεργειών για «Επισύναψη Εικόνας», που ήδη περιγράφηκε, έπειτα από την κλήση της setImage καλείται και η συνάρτηση setDescriptionText θέτοντας «κενό» (null) στο descriptionText χαρακτηριστικό του newRequest. Έπειτα, ο χρήστης επιλέγει «Υποβολή» και η ακολουθία ενεργειών είναι ίδια έως ότου ο ελεγκτής καλέσει τη μέθοδο createHealthRequest οπότε το αντικείμενο τύπου HealthRequest που δημιουργεί έχει «κενό» (null) στο descriptionText χαρακτηριστικό. Τότε, δεν αποθηκεύεται κάποιο αντικείμενο στη βάση δεδομένων, αλλά ο ελεγκτής καλεί τη μέθοδο DisplayUnsuccessfulSubmissionDialog και προβάλει το παράθυρο διαλόγου ανεπιτυχούς υποβολής της κατάστασης της υγείας στον “ασθενή”.

3.1.2 Sequence Diagram



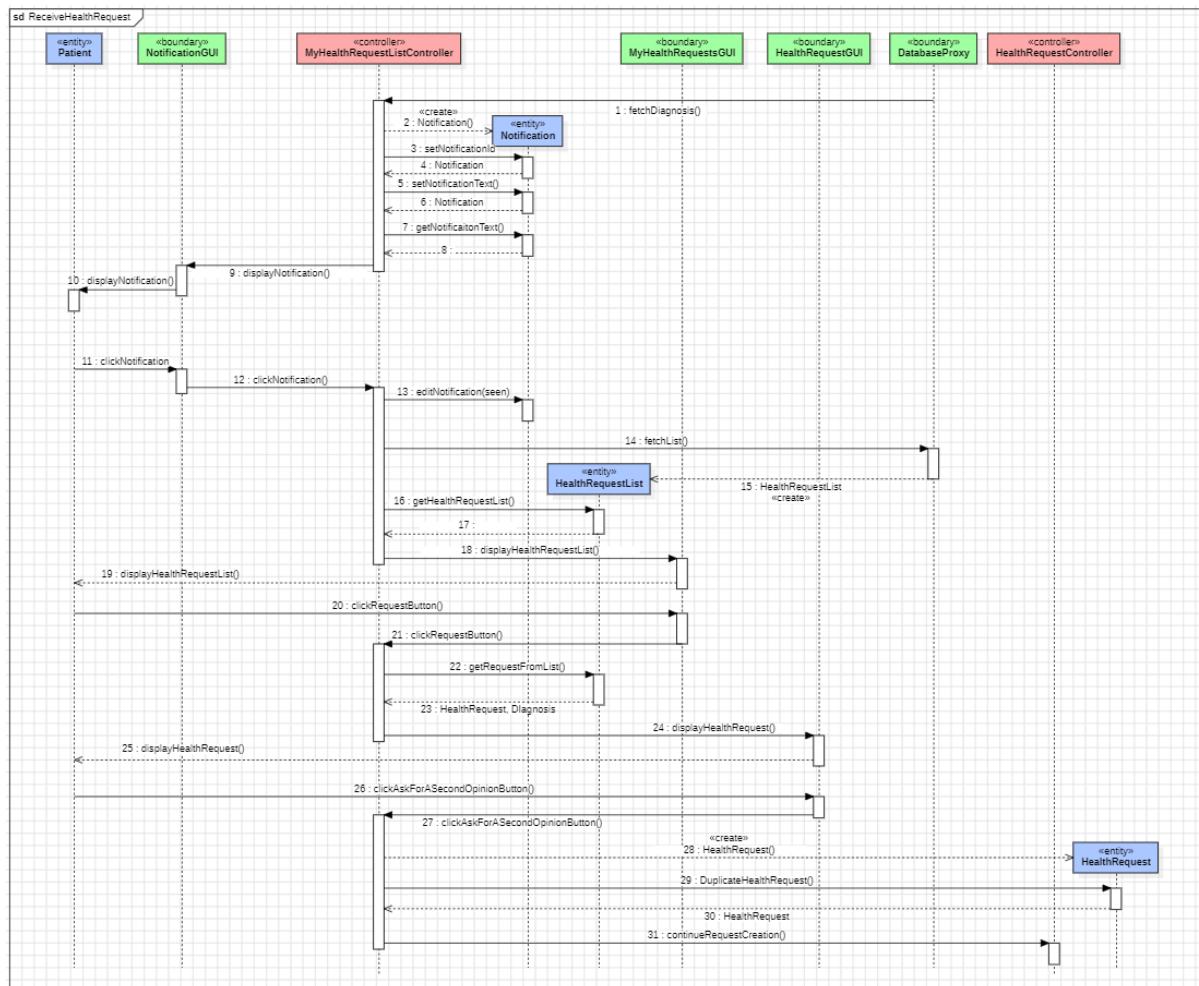
3.2 Gherkin Scenario 1 (Feature 4) "Patient receives a medical diagnosis"

3.2.1 Αφήγηση Σεναρίου

Η διαδικασία παραλαβής Diagnosis ξεκινάει από όταν ο DatabaseProxy λαμβάνει την διάγνωση από τον γιατρό. Εκείνη τη στιγμή στέλνεται απευθείας η διάγνωση στον MyHealthRequestListController όπου ξεκινάει η σύνθεση μιας ειδοποίησης. Καλείται ο constructor του Notification και ορίζονται οι παράμετροι του αντικειμένου έτσι ώστε να προβληθεί η ειδοποίηση στον ασθενή με τη συνάρτηση displayNotification() της κλάσης NotificationGUI. Όταν ο ασθενής πατήσει την ειδοποίηση το NotificationGUI τρέχει τη συνάρτηση clickNotification() γεγονός το οποίο πυροδοτεί την προβολή της λίστας με τα HealthRequests του χρήστη. Καταρχάς θέτουμε την ειδοποίηση ως seen και μετά ζητάμε από το DatabaseProxy τη λίστα συνδεδεμένη με το patientId του χρήστη κάνοντας χρήση της συνάρτησης fetchList(patientId). Έπειτα ομαδοποιούμε την πληροφορία της λίστας που λάβαμε με τη συνάρτηση getHealthRequestList() η οποία μας δίνει ένα πίνακα με τρεις στήλες (1. HealthRequestId, 2. hasReceivedDiagnosis, 3. SeqId). Αυτός ο πίνακας στέλνεται έπειτα στη συνάρτηση displayHealthRequestList() της κλάσης MyHealthRequestsGUI για να προβληθεί στον χρήστη η λίστα. Στη συνέχεια όταν ο χρήστης πατήσει το κουμπί που αναπαριστά ένα HealthRequest ο MyHealthRequestListController ψάχνει το HealthRequest και ,to Diagnosis αν υπάρχει, για να το στείλει στο HealthRequestGUI το οποίο είναι η οθόνη προβολής του HealthRequest με την

απάντηση από τον γιατρό ή το AI όταν γίνουν διαθέσιμα. Τέλος αφού προβληθεί στο χρήστη αυτή η οθόνη ο ίδιος έχει τη δυνατότητα επιλογής AskForASencondOpinion. Στην περίπτωση που την επιλέξει δημιουργείται ένα HealthRequest από τον MyHealthRequestListController, συμπληρώνεται μερικώς αφήνοντας εκτός τις παραμέτρους σχετικές με τον γιατρό και προωθείται στον HealthRequestController έτσι ώστε να επιλέξει ο ασθενής από ποιόν γιατρό επιθυμεί να λάβει την δεύτερη γνώμη.

3.2.2 Sequence Diagram



3.3 Gherkin Scenario 3 (Feature 6) : "Doctor gives advice to patient based on the Medical History Document."

3.3.1 Αφήγηση Σεναρίου

Το σενάριο που θα αναλύσουμε παρακάτω , αφορά την παροχή ιατρικής συμβουλής από τον γιατρό στον ασθενή , σε μορφή κειμένου , με βάση την ύπαρξη ενός Medical History Document για αυτόν τον ασθενή.

Αρχικά , ο “γιατρός” λαμβάνει το έγγραφο αυτό μέσω της μεθόδου fetchMHD() του boundary DatabaseProxy (έστω ότι το ονομάζουμε MHD), έτσι ώστε να το διαβάσει με την συνάρτηση

readMHD(MHD) . Στην συνέχεια με το πάτημα του writeTXTbutton , καλείται η μέθοδος writeTXTbuttonPress που μεταφέρει τον “γιατρό” στην οθόνη εισαγωγής του κειμένου της διάγνωσης , αρχικοποιώντας ένα κενό αντικείμενο Diagnosis. Μόλις ο χρήστης δώσει το κείμενο της διάγνωσης , αυτό επιστρέφεται και μέσω του ελεγκτή DiagnosisController και ειδικότερα της μεθόδου του createDiagnosis(MHD) δημιουργείται η “ιατρική συμβουλή” , η οποία και τέλος αποθηκεύεται στην Βάση Δεδομένων μας , ώστε μέσω της συνάρτησης fetchDiagnosis() της DatabaseProxy να μπορεί να “φτάσει στα χέρια” του “ασθενή”.

3.3.2 Sequence Diagram

