|  |
| --- |
|  |

|  |
| --- |
| NEXT LEVEL SOFTWARE |
| ΜyDoctor |
| Έγγραφο Περιγραφής Σχεδίου Λογισμικού |

Ιστορικό

|  |  |  |
| --- | --- | --- |
| Ημερομηνία | Περιγραφή | Συγγραφέας |
| 1/5/18 | Συμπλήρωση του 1.1 , 1.2 | Κυριακίδης Αντώνης,  Χατζηιωαννίδου Μαρία |
| 5/5/18 | Συμπλήρωση του 2.2 ,2.3 | Γιάντσιος Κων/νος |
| 13/5/18 | Συμπλήρωση του 3 | Γιάντσιος Κων/νος,  Καλογερόπουλος Γιώργος,  Μαρίνος Μάριος,  Μπουγιουκλής Δημήτριος |
| 17/5/18 | Συμπλήρωση του 4 | Ακριτίδης Θεόδωρος  Λαινάς Πασχάλης  Μάστορας Γιώργος |
| 20/5/18 | Συμπλήρωση του 5.1 , 5.2 | Γιάντσιος Κων/νος,  Καλογερόπουλος Γιώργος,  Μαρίνος Μάριος,  Μπουγιουκλής Δημήτριος |

Πίνακας Περιεχομένων

1. Εισαγωγή 4

1.1 Σκοπός 4

1.2 Επισκόπηση 4

2. Σχέδιο Δεδομένων 5

2.2 Αρχεία 5

2.3 Δομές Δεδομένων

3. Σχέδιο Μονάδων 6

4. Περιγραφή Διασυνδέσεων 10

5. Παραρτήματα 14

5. 1 Διαγράμματα Κλάσεων 14

5. 2 Πίνακας Ιχνηλάτισης 15

5. 2 Διάγραμμα Ακολουθίας 16

**1. Εισαγωγή**

**1.1 Σκοπός**

α)Σκοπός του Εγγράφου Περιγραφής Σχεδίου Λογισμικού ( ΕΠΣΛ ) είναι ο καθορισμός της αρχιτεκτονικής σχεδίασης του λογισμικού διαχείρισης της σχέσης γιατρού - ασθενή, δηλαδή των μονάδων από τις οποίες αποτελείται και των μεταξύ τους σχέσεων.

β) Το συγκεκριμένο Έγγραφο Περιγραφής Σχεδίου Λογισμικού ( ΕΠΣΛ ) απευθύνεται στην ομάδα ανάπτυξης της εταιρείας NLS που θα προχωρήσει στην υλοποίηση, τον έλεγχο και τη συντήρηση του συγκεκριμένου λογισμικού.

**1.2 Επισκόπηση**

Στα επόμενα κεφάλαια του Εγγράφου Περιγραφής Σχεδίου Λογισμικού αναφέρουμε το Σχέδιο Δεδομένων, το Σχέδιο Μονάδων, την περιγραφή Διασυνδέσεων και τα Παραρτήματα. Πιο συγκεκριμένα, όσο αναφορά το Σχέδιο Δεδομένων περιγράφουμε τα αρχεία δεδομένων που δημιουργήσαμε και χρησιμοποιήσαμε στο λογισμικό , καθώς και τις δομές δεδομένων του συστήματος. Όσο αναφορά το Σχέδιο Μονάδων που μαζί με το διάγραμμα κλάσεων αποτελούν το σημαντικότερο τμήμα του Εγγράφου Περιγραφής Σχεδίου Λογισμικού παρουσιάζουμε το όνομα κάθε κλάσης και της κάθε μεθόδου και δίνοντας μια περιγραφή της λειτουργικότητας τους. Σχετικά με την Περιγραφή των Διασυνδέσεων παραθέτουμε τις συνοριακές κλάσεις του συστήματος που έχουν σκοπό την αλληλεπίδραση με το χρήστη αναφέροντας το όνομα κάθε κλάσης και εξηγώντας τη λειτουργικότητά της. Τέλος, στα Παραρτήματα δίνουμε το διάγραμμα κλάσεων του συστήματος , τον Πίνακα Ιχνηλάτισης, τα Διαγράμματα Ακολουθίας το οποίο καλύπτει την βασική και τις εναλλακτικές ροές, τα Διαγράμματα Αναφορών και Οθονών που συμβάλλουν στην κατανόηση της λειτουργικότητας των κλάσεων και των.

**2. Σχέδιο Δεδομένων**

**2.1 Αρχεία**

To λογισμικό χρησιμοποίει ένα αρχείο ονόματι registry. ser για την αποθήκευση των δεδομένων . Για να αποθηκεύσουμε τα δεδομένα μας στο αρχείο χρησιμοποιούμε τον μηχανισμό object serialization .To αντικείμενο είναι της κλάσης Registry που θα αναλύσουμε παρακάτω .

**2.2 Δομές Δεδομένων**

public ArrayList <Patient> Patients της κλάσης Registry :

Χρησιμοποιείται για την αποθήκευση όλων των ασθενών.

public ArrayList <Doctor> Doctors της κλάσης Registry:

Χρησιμοποιείται για την αποθήκευση όλων των γιατρών .

private HashMap<Integer, Integer[]> appointmentMap της κλάσης Doctor:

Χρησιμοποιείται για την αποθήκευση των ραντεβού ενός γιατρού.

private HashMap<Patient, Integer> patientMap της κλάσης Doctor:

Χρησιμοποιείται για την αποθήκευση των ασθενών ενός γιατρού και τις φορές που τον έχει επισκεφθεί.

private ArrayList<Appointment> appointmentList της κλάσης Patient:

Χρησιμοποιείται για την αποθήκευση των ραντεβού ενός ασθενή.

private HashMap<Doctor,Review> doctorsMap της κλάσης Patient:

Χρησιμοποιείται για την αποθήκευση των γιατρών και των review που έχει κάνει ο ασθενής.

private ArrayList<String> prescriptionList της κλάσης Patient:

Χρησιμοποιείται για την αποθήκευση των συνταγών .

**3. Σχέδιο Μονάδων**

**3.1 Registry**

Ταυτότητα Κλάσης: <1>

Η Registry είναι μια στατική κλάση που χρησιμοποιείται ως βάση δεδομένων. Κατά την εκκίνηση του προγράμματος δημιουργεί τις λίστες με όλα τα αντικείμενα που χρησιμοποιούνται στην εφαρμογή (γιατροί, ασθενείς, διαχειριστής/στές), παίρνοντας τα δεδομένα από την κλάση Serialization.

Επιπλέον, στην κλάση υπάρχει και η μέθοδος που κάνει τον έλεγχο κατά την είσοδο των χρηστών (authentication) συγκρίνοντας τα στοιχεία εισόδου με τα δεδομένα που υπάρχουν στις λίστες.

public Object authentication(String username, String password):

Μέθοδος που δέχεται ένα όνομα και ένα κωδικό και επιστέφει το αντίστοιχο object(doctor,admin ή patient).Χρησιμεύει στο login.

**3.2 Doctor**

Ταυτότητα Κλάσης: <2>

Αυτή η κλάση έχει κάποια πεδία που είναι πληροφορίες για τον γιατρό πχ όνομα ,επώνυμο ,ειδικότητα κλπ. Και περιλαμβάνει τις εξής μεθόδους :

public void makeAppointment(Patient aPatient , Date aDate):

Η μέθοδος αυτή δέχεται ως παραμέτρους έναν ασθενή και μία ημερομηνία και καταχωρεί ένα ραντεβού στο map του γιατρού.

public void cancelAppointment(Date selectedDate):

Η μέθοδος αυτή δέχεται ως παράμετρο μία ημερομηνία τύπου Date και διαγράφει το αντίστοιχο ραντεβού από τον ασθενή και τον γιατρό.

public void configureInfo(String newPassword,String newEmail,String newTelephone):

Η μέθοδος αυτή δέχεται ως παραμέτρους τις πληροφορίες που μπορεί να τροποποιήσει ο γιατρός, οι οποίες είναι το password , e-mail και το τηλέφωνό του.

public void showHours(Date SelectedDate);

Η μέθοδος αυτή διασχίζει το map του γιατρού για την συγκεκριμένη μέρα και βρίσκει τις διαθέσιμες ώρες του και τις επιστρέφει στον ασθενή.

public void addPatient(Patient aPatient):

H μέθοδος αυτή προσθέτει έναν ασθενή στο map του γιατρού ώστε να έχει μία λίστα με όλους τους ασθενείς και να κρατάει πόσες φορές έχει επισκεφτεί τον γιατρό (το όποιο χρειάζεται για στατιστικούς λόγους) καλείται από την μέθοδο του ασθενή MakeAppointment για να προσθέσει τον ασθενή.

public void writePrescription(Patient aPatient,String prescription):

Η μέθοδος αυτή δέχεται ως παραμέτρους έναν ασθενή και ένα αλφαριθμητικό το οποίο αποτελεί την συνταγή την οποία καταχωρεί στον ασθενή.

**3.3 Patient**

Ταυτότητα Κλάσης: <3>

Η κλάση Patient έχει πολλά πεδία τα οποία λίγα θα χρειαστούν για τις μεθόδους.

public void makeAppointment(Doctor aDoctor,Date aDate) :

Η μέθοδος αυτή δέχεται ως παραμέτρους ένα αντικείμενο τύπου Doctor και μία ημερομηνία τύπου Date , δημιουργεί ένα ραντεβού τύπου Appointment και το καταχωρεί σε ένα arrayList στον ασθενή και σε ένα hashmap στον γιατρό .

public void cancelAppointment(Date selectedDate):

Η μέθοδος αυτή δέχεται ως παράμετρο ένα αντικείμενο τύπου Date ,ψάχνει στο arrayList του ασθενή με τα ραντεβού και αφού βρει το κατάλληλο ραντεβού το διαγράφει και από τον γιατρό και από τον ασθενή.

public void configureInfo(String newPassword,String newEmail,String newTelephone):

H μέθοδος αυτή δέχεται ως παραμέτρους τα στοιχεία που ουσιαστικά έχει πρόσβαση ο ασθενής για να τροποποιήσει που είναι μόνο ο κωδικός το e-mail και το τηλέφωνό του.

public String showPrescription(String selectedPrescription):

Η μέθοδος αυτή δέχεται ως παράμετρο μία συνταγή και επιστρέφει απλά την συνταγή ως αλφαριθμητικό.

public void addDoctor(Doctor aDoctor):

Η μέθοδος αυτή δέχεται έναν γιατρό και τον προσθέτει στην λίστα του ασθενή. Ουσιαστικά η μέθοδος είναι βοηθητική και καλείται από την μέθοδο makeAppointment.

**3.4 Admin**

Ταυτότητα Κλάσης: <4>

Τα αντικείμενα τύπου Admin αντιπροσωπεύουν τους διαχειριστές-χρήστες του συστήματος οι οποίοι έχουν πρόσβαση σε στατιστικά δεδομένα των γιατρών καθώς και κριτικές των ασθενών προς τους γιατρούς. Επιπλέον, οι διαχειριστές έχουν την δυνατότητα προθήκης και διαγραφής γιατρών στο σύστημα καθώς και να μεταφέρουν κάποιο γιατρό σε άλλη πόλη.

Οι διαχειριστές συνδέονται στο σύστημα από την οθόνη εισόδου όπως όλοι οι χρήστες αλλά δεν μπορούν να δημιουργηθούν ή να διαγραφούν από την διεπαφή χρήστη. Συγκεκριμένα, τα αντικείμενα τύπου Admin είναι αποθηκευμένα σε δυαδικό αρχείο και κατά την εκκίνηση του προγράμματος δημιουργούνται μόνο μία φορά και δεν τροποποιούνται.

public double calAvgGeneralRating(Doctor aDoctor):

Μέθοδος που υπολογίζει τον μέσο όρο της γενικής βαθμολογίας του γιατρού που δέχεται ως παράμετρο.

public double calAvgCleanliness(Doctor aDoctor)

public double calAvgStaffCooperation(Doctor aDoctor)

public double calAvgDignityAndRespect(Doctor aDoctor)

public double calAvgInvolvementInDecisions(Doctor aDoctor):

Ομοίως, οι παραπάνω μέθοδοι υπολογίζουν τον αντίστοιχο μέσο όρο των βαθμολογιών που αναγράφονται.

**3.5 Appointment**

Ταυτότητα Κλάσης: <5>

Τα αντικείμενα τύπου Appointment περιλαμβάνουν τα εξής πεδία : ένα αντικείμενο τύπου Patient, ένα αντικείμενο τύπου Doctor ,μία ημερομηνία τύπου Date και ένα κλειδί που θα είναι μοναδικό για το κάθε αντικείμενο τύπου Integer.

**3.6** **Statistics**

Ταυτότητα Κλάσης: <6>

Είναι η κλάση που αποθηκεύει τα στατιστικά στοιχεία του γιατρού (αξιολογήσεις και επισκέψεις). Κάθε αντικείμενο γιατρού έχει ένα αντικείμενο Statistics το οποίο αποθηκεύει τα δεδομένα του αντίστοιχου γιατρού. Η κλάση αυτή περιέχει τις εξής μεθόδους:

public void updateNumberOfVisitsPerDay():

Ανανεώνει τις επισκέψεις της τρέχουσας ημέρας στην πεδίο numberOfVisitsPerDay του γιατρού.

**3.7 Review**

Ταυτότητα Κλάσης: <7>

Αυτή η κλάση χρησιμεύει για τον αποθήκευση συγκεκριμένων πεδίων(id, Cleanliness,StaffCo\_operation,Dignity\_and\_respect,Involvement\_in\_decisions,General\_Rating) και για τον υπολογισμό του general\_rating που χρησιμεύει στην κλάση Statistics.

public double calcGeneral\_Rating () :

Υπολογίζει το general\_rating βάσει των μεταβλητών που προανέφερα και το επιστρέφει.

# 4. Περιγραφή Διασυνδέσεων

# 4.1 Auth\_GUI

Ταυτότητα Κλάσης: <8>

Μέσω της συγκεκριμένης κλάσης ο χρήστης έχει τη δυνατότητα να συνδεθεί στον ήδη υπάρχοντα λογαριασμό(ασθενής, γιατρός ή admin) πληκτρολογώντας το ΑΜΚΑ και τον κωδικό του ή να δημιουργήσει έναν καινούργιο λογαριασμό(ασθενής) γεμίζοντας τα απαραίτητα πεδία.

**4.2 Patient\_GUI**

Ταυτότητα Κλάσης: <9>

Αυτή η κλάση εμφανίζει το βασικό παράθυρο του ασθενή στο οποίο αναγράφονται οι πληροφορίες του και του δίνει έξι επιλογές κουμπιών(Make an Appointment, Appointments, Prescriptions, Review, Configure, Sign Out).Ανάλογα με το κουμπί που θα πατήσει ο χρήστης εμφανίζεται το αντίστοιχο παράθυρο.

# 4.3 MakeAppointment\_Patient\_GUI

Ταυτότητα Κλάσης: <10>

Αυτή η κλάση εμφανίζει το παράθυρο μέσω του οποίου ο ασθενής κλείνει ραντεβού. Του δίνεται η δυνατότητα αρχικά να επιλέξει την πόλη στην οποία βρίσκεται καθώς και την ειδικότητα του γιατρού που επιθυμεί και πατώντας ένα κουμπί(Search) εμφανίζονται στην οθόνη οι διαθέσιμοι γιατροί και μέρες. Ο χρήστης τα επιλέγει και πατάει το κουμπί Show Hours για να εμφανιστούν οι ώρες. Επιλέγοντας ώρα έχει τη δυνατότητα να πατήσει το κουμπί Make the Appointment το οποίο θα κλείσει το ραντεβού .Εφόσον έχουν επιλεχθεί ο γιατρός και η μέρα εμφανίζει τις διαθέσιμες ώρες.

# 4.4 ActiveAppointments\_GUI

Ταυτότητα Κλάσης: <11>

Αυτή η κλάση εμφανίζει το παράθυρο των ραντεβού που έχουν προγραμματιστεί(μέρα,ώρα) και δίνει στο χρήστη τη δυνατότητα να ακυρώσει κάποιο ραντεβού επιλέγοντας το ραντεβού που επιθυμεί και πατώντας το κουμπί Delete.

**4.5 Prescription\_GUI**

Ταυτότητα Κλάσης :<12>

Αυτή η κλάση εμφανίζει τους τίτλους των συνταγών(ημερομηνία, γιατρός) και αφού επιλεχθεί τίτλος και πατηθεί το κουμπί watch prescription εμφανίζεται αναλυτικά η συνταγή.

# 4.6 Review\_GUI

Ταυτότητα Κλάσης: <13>

Αυτή η κλάση δίνει στο χρήστη τη δυνατότητα να αξιολογήσει το γιατρό που επιλέγει, απαντώντας σε κάποιες συγκεκριμένες ερωτήσεις με έναν αριθμό από 1-5.

# 4.7 Doctor\_GUI

Ταυτότητα Κλάσης: <14>

Αυτή η κλάση εμφανίζει το βασικό παράθυρο του γιατρού στο οποίο αναγράφονται οι πληροφορίες του και του δίνει δύο επιλογές κουμπιών. Συμπληρώνοντας το ΑΜΚΑ του ασθενή έχει τη δυνατότητα να πατήσει το κουμπί Patient Information για να δει τις πληροφορίες του ασθενή ενώ μπορεί να πατήσει και το κουμπί Cancel Appointments αν επιθυμεί να ακυρώσει το ραντεβού με αυτόν.

# 4.8 PatientInformation\_GUI

Ταυτότητα Κλάσης:< 15>

Αυτή η κλάση εμφανίζει στον γιατρό το παράθυρο με τις πληροφορίες του ασθενή και του δίνει τη δυνατότητα να κλείσει ή να ακυρώσει ραντεβού καθώς και να γράψει συνταγή για τον συγκεκριμένο ασθενή.

# 4.8 MakeAppointment\_Doctor\_GUI

Ταυτότητα Κλάσης: <16>

Αυτή η κλάση εμφανίζει το παράθυρο μέσω του οποίου ο γιατρός κλείνει ραντεβού με τον ασθενή. Αρχικά επιλέγει την μέρα, πατάει το κουμπί Show Hours και του εμφανίζονται οι κενές ώρες της συγκεκριμένης μέρας. Με το πάτημα του κουμπιού save αποθηκεύεται το ραντεβού στο σύστημα .Ανάλογα με τη μέρα που έχει επιλεχθεί εμφανίζει στο χρήστη τις διαθέσιμες ώρες για το κλείσιμο του ραντεβού.

# 4.9 CancelAppointment\_GUI

Ταυτότητα Κλάσης: <17>

Αυτή η κλάση εμφανίζει ένα παράθυρο το οποίο εμφανίζει το οποίο δίνει στο γιατρό τη δυνατότητα να διαγράψει ραντεβού. Αρχικά επιλέγει τη μέρα από την οποία επιθυμεί να διαγράψει το ραντεβού και στη συνέχει πατάει το κουμπί Show Hours. Στη συνέχεια εμφανίζονται οι ώρες στις οποίες υπάρχει κλεισμένο ραντεβού. Ο γιατρός επιλέγει την ώρα και πατώντας το κουμπί Delete διαγράφεται το ραντεβού από το σύστημα.Εφόσον έχει επιλεχθεί κάποια μέρα εμφανίζει τις κλεισμένες ώρες της.

# 4.10 WritePrescription\_GUI

Ταυτότητα Κλάσης: <18>

Αυτή η κλάση δίνει στο γιατρό τη δυνατότητα να γράψει συνταγή στον επιλεγμένο ασθενή.

**4.11 Admin\_GUI**

Ταυτότητα Κλάσης: <19>

Αυτή η κλάση εμφανίζει το βασικό παράθυρο του Administrator στο οποίο αναγράφονται οι πληροφορίες του και του δίνει πέντε επιλογές κουμπιών. Γεμίζοντας το πεδίο με το όνομα του γιατρού έχει τη δυνατότητα να διαγράψει κάποιον γιατρό από το σύστημα πατώντας το κουμπί delete, να μετακινήσει ένα γιατρό σε κάποια άλλη πόλη ή να δει τις πληροφορίες του γιατρού. Επίσης μπορεί να προσθέσει ένα γιατρό στο σύστημα καθώς και να δει τα στατιστικά των γιατρών με βάση την ειδικότητα και το χρονικό διάστημα που επιθυμεί .Εμφανίζει στο χρήστη μήνυμα με δύο επιλογές(yes, no). Αν η επιλογή είναι yes, διαγράφει το γιατρό από το σύστημα, αλλιώς απλά κλείνει το μήνυμα.

# 4.12 Statistics\_GUI

Ταυτότητα Κλάσης: <20>

Αυτή η κλάση εμφανίζει ένα ραβδόγραμμα με τα στατιστικά των γιατρών της επιλεγμένης ειδικότητας, πόλης και χρονικού διαστήματος.

# 4.13 Information\_GUI

Ταυτότητα Κλάσης: <21>

Αυτή η κλάση εμφανίζει παράθυρο με τις πληροφορίες του γιατρού καθώς και το μέσο όρο των απαντήσεων των πελατών στην αξιολόγησή του.

# 4.14 AddDoctor\_GUI

Ταυτότητα Κλάσης: <22>

Αυτή η κλάση δίνει στον Administrator τη δυνατότητα να προσθέσει έναν καινούργιο γιατρό στο σύστημα αφού συμπληρώσει όλα τα απαραίτητα πεδία.

# 4.15 Transfer\_GUI

Ταυτότητα Κλάσης: <23>

Αυτή η κλάση δίνει τη δυνατότητα στον Administrator να μεταφέρει ένα γιατρό σε μία διαφορετική πόλη.

**4.16 Serialization**

Ταυτότητα Κλάσης: <24>

Αυτή η κλάση χρησιμοποιείται για την διασύνδεση του λογισμικού με το αρχείο registry.ser και την αποθήκευση των δεδομένων .Οι δύο μέθοδοι παρακάτω είναι στατικές γιατί θέλουμε να καλούνται χωρίς να έχει δημιουργηθεί κάποιο αντικείμενο αυτής της κλάσης.

public static void SaveToFile(Registry reg) :

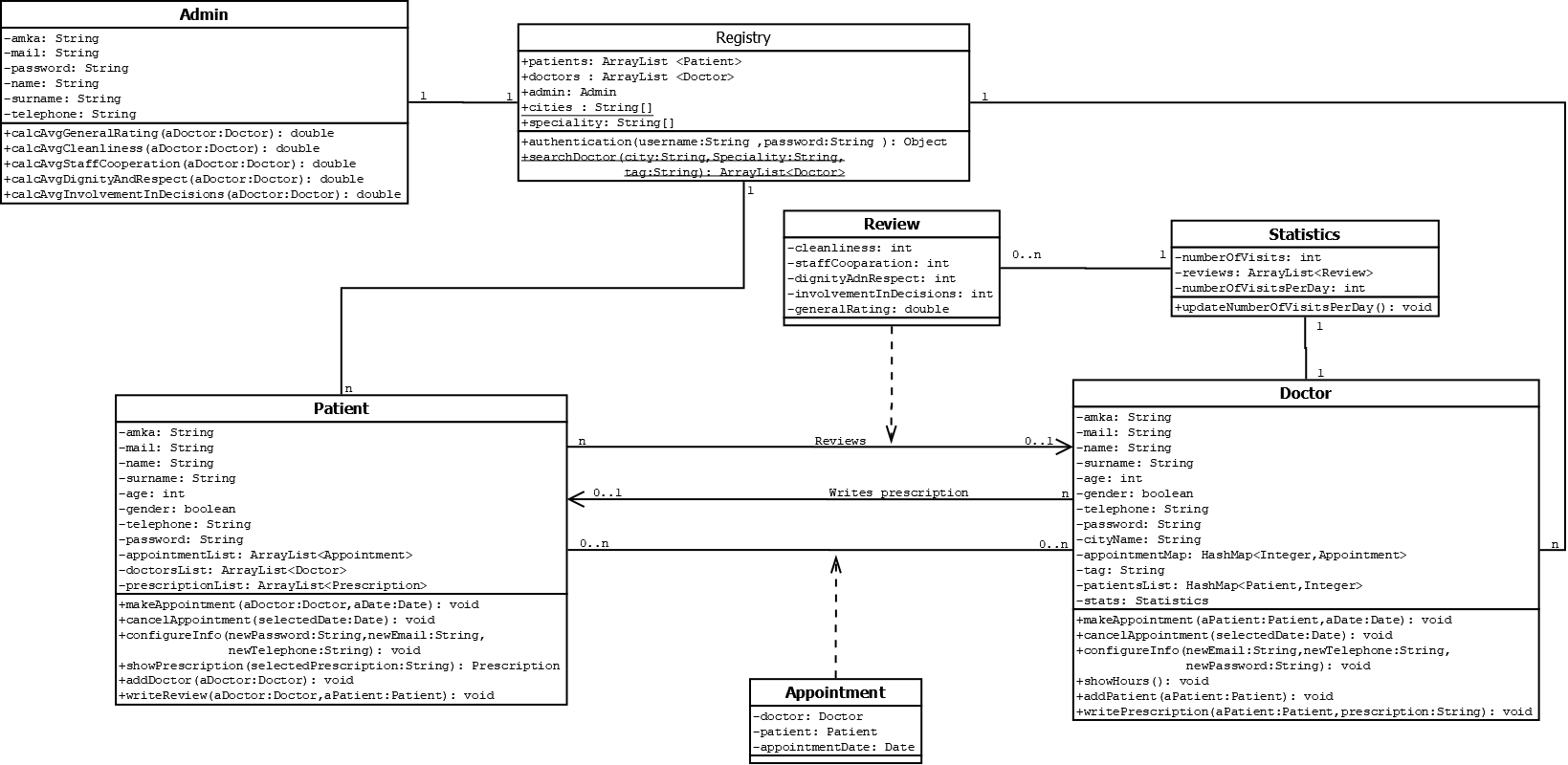
Μέθοδος που δέχεται ένα αντικείμενο της κλάσης Registry και το αποθηκεύει στο registry.ser .

public static Registry ReadFromFile() :

Μέθοδος που διαβάζει δεδομένα από το registry .ser και επιστρέφει ένα αντικείμενο της κλάσης Registry.

**5. Παραρτήματα**

**5.1 Διάγραμμα Κλάσεων**

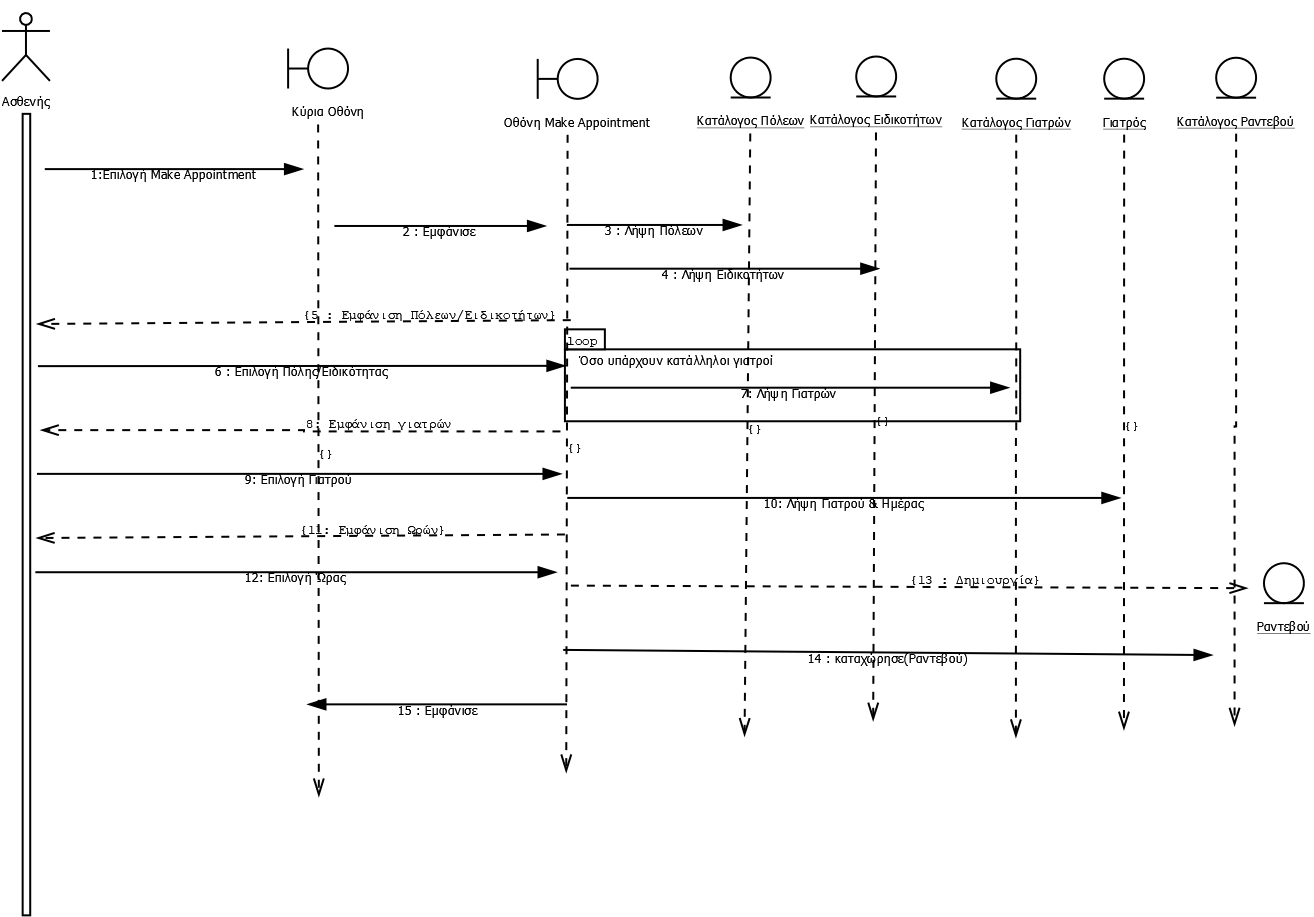
****

**5.2 Πίνακας Ιχνηλάτισης**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | login/register user | Appoint-ment | Prescription | review |
| Patient |  | **×** | **×** | **×** |
| Doctor |  | **×** | **×** | **×** |
| Admin | **×** |  |  |  |
| Registry | **×** |  |  |  |
| Appointment |  | **×** |  |  |
| Review |  |  |  | **x** |
| Prescription |  |  | **x** |  |
| Statistics |  | **x** |  | **x** |

**5.3 Διάγραμμα Ακολουθίας**

Το παρακάτω διάγραμμα ακολουθίας περιγράφει την λειτουργία κλεισίματος ραντεβού από τον ασθενή.

****