



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ**  
**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ**  
**ΕΠΙΚΟΙΝΩΝΙΩΝ**  
**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**ΤΕΧΝΟΛΟΓΙΑ ΛΟΓΙΣΜΙΚΟΥ: ΑΠΑΛΛΑΚΤΙΚΗ ΕΡΓΑΣΙΑ**

**Ακαδ. Έτος 2020-2021**

**ΓΑΛΙΑΤΣΑΤΟΥ ΑΙΚΑΤΕΡΙΝΗ | Π18022**

**ΖΑΙΜΗΣ ΓΕΩΡΓΙΟΣ | Π18041**

**ΖΩΤΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ | Π18046**

**ΚΑΛΑΧΑΝΗΣ ΑΓΓΕΛΟΣ | Π18051**

**ΣΗΜΙΡΙΩΤΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ | Π18135**

## ΠΕΡΙΕΧΟΜΕΝΑ

1. Εισαγωγή
  - 1.1 Στόχοι της εργασίας
  - 1.2 Ορισμός του προβλήματος προς επίλυση
2. Σύντομη παρουσίαση της RUP
3. Φάση: Έναρξη (Inception)
  - 3.1 Σύλληψη απαιτήσεων
  - 3.2 Ανάλυση-Σχεδιασμός
    - 3.2.1 Διαγράμματα Περιπτώσεων Χρήσης (1<sup>η</sup> έκδοση)
    - 3.2.2 Διαγράμματα Τάξεων (1<sup>η</sup> έκδοση)
4. Φάση: Εκπόνηση Μελέτης (Elaboration)
  - 4.1 Ανάλυση-Σχεδιασμός
    - 4.1.1 Διαγράμματα Περιπτώσεων Χρήσης (2<sup>η</sup> έκδοση)
    - 4.1.2 Διαγράμματα Τάξεων (2<sup>η</sup> έκδοση)
    - 4.1.3 Διαγράμματα Αντικειμένων (1<sup>η</sup> έκδοση)
    - 4.1.4 Διαγράμματα Συνεργασίας (1<sup>η</sup> έκδοση)
    - 4.1.5 Διαγράμματα Σειράς (1<sup>η</sup> έκδοση)
    - 4.1.6 Διαγράμματα Δραστηριοτήτων (1<sup>η</sup> έκδοση)
    - 4.1.7 Διαγράμματα Καταστάσεων (1<sup>η</sup> έκδοση)
    - 4.1.8 Διαγράμματα Εξαρτημάτων (1<sup>η</sup> έκδοση)
    - 4.1.9 Διαγράμματα Διανομής (1<sup>η</sup> έκδοση)
  - 4.2 Υλοποίηση-Έλεγχος
    - 4.2.1 Υλοποίηση: 1<sup>η</sup> εκτελέσιμη έκδοση
    - 4.2.2 Αναφορά ελέγχου για την 1<sup>η</sup> εκτελέσιμη έκδοση
5. Φάση: Κατασκευή (Construction)
  - 5.1 Ανάλυση-Σχεδιασμός
    - 5.1.1 Διαγράμματα Περιπτώσεων Χρήσης (3<sup>η</sup> έκδοση)
    - 5.1.2 Διαγράμματα Τάξεων (3<sup>η</sup> έκδοση)
    - 5.1.3 Διαγράμματα Αντικειμένων (2<sup>η</sup> έκδοση)
    - 5.1.4 Διαγράμματα Συνεργασίας (2<sup>η</sup> έκδοση)
    - 5.1.5 Διαγράμματα Σειράς (2<sup>η</sup> έκδοση)
    - 5.1.6 Διαγράμματα Δραστηριοτήτων (2<sup>η</sup> έκδοση)
    - 5.1.7 Διαγράμματα Καταστάσεων (2<sup>η</sup> έκδοση)
    - 5.1.8 Διαγράμματα Εξαρτημάτων (2<sup>η</sup> έκδοση)
    - 5.1.9 Διαγράμματα Διανομής (2<sup>η</sup> έκδοση)
  - 5.2 Υλοποίηση-Έλεγχος
    - 5.2.1 Υλοποίηση: Τελική εκτελέσιμη έκδοση
    - 5.2.2 Αναφορά ελέγχου για την τελική εκτελέσιμη έκδοση
6. Εγχειρίδιο Χρήστη
  - 6.1 Σύντομη παρουσίαση του προγράμματος
  - 6.2 Παρουσίαση σεναρίων λειτουργίας

## 1. Εισαγωγή

Η εργασία αυτή εκπονήθηκε στα πλαίσια του μαθήματος <<Τεχνολογία Λογισμικού>>. Το θέμα που κληθήκαμε να αναπτύξουμε είναι μια web εφαρμογή καταχώρησης ραντεβού ασθενών-ιατρών με τη βοήθεια του αντικειμενοστραφούς μοντέλου ανάπτυξης λογισμικού Rational Unified Process.

Οι στόχοι της Τεχνολογίας Λογισμικού είναι να βελτιώσει την ποιότητα των προϊόντων λογισμικού και να αυξήσει την παραγωγικότητα των μηχανικών λογισμικού.

Κατά τη διαδικασία ανάπτυξης λογισμικού προσδιορίζονται τρεις αρχικές διαδικασίες:

- Απαιτήσεις: αποφασίζει τι πρέπει να κάνει ένα σύστημα, τις δραστηριότητές του, τους κινδύνους και το δοκιμαστικό σχέδιό του.
- Σχεδιασμός: καθορίζει πώς υπολογίζει ένα σύστημα τις συγκεκριμένες λειτουργίες και τη δομή του.
- Υλοποίηση: παράγει τον πηγαίο κώδικα, την τεκμηρίωση και τις δοκιμές (επικυρώνει και ελέγχει).

### 1.1 Στόχοι της εργασίας

- Καταχωρημένος κατάλογος γιατρών με τα στοιχεία τους.
- Εγγραφή χρήστη.
- Δυνατότητα προγραμματισμού ενός ραντεβού.
- Αναζήτηση γιατρού ανά ειδικότητα και περιοχή.
- Δυνατότητα προβολής της διαθεσιμότητας ραντεβού με τους γιατρούς.
- Καταχώρηση ημέρας, ώρας και λόγου επίσκεψης στο γιατρό.
- Ακύρωση προγραμματισμένου ραντεβού.
- Προβολή καταλόγου με προγραμματισμένα ραντεβού του ασθενή.

### 1.2 Ορισμός του προβλήματος προς επίλυση

Με την υλοποίηση μίας τέτοιας εφαρμογής, η αναζήτηση για ραντεβού απλοποιείται και ο ασθενής μπορεί μέσα σε μικρό χρονικό διάστημα να κλείσει το ραντεβού με τον ιατρό που επιθυμεί. Μία τέτοια εφαρμογή όχι μόνο επισπεύδει την διαδικασία για τον ασθενή αλλά και δίνει την δυνατότητα σε ασθενή και ιατρό να δουν όλα τα προγραμματισμένα τους ραντεβού σε μία πλατφόρμα.

Στην εφαρμογή θα πρέπει να κρατούνται στοιχεία (ΑΜΚΑ, ονοματεπώνυμο, ειδικότητα, στοιχεία επικοινωνίας, περιοχή που έχουν ιατρείο) για τους ιατρούς που είναι εγγεγραμμένοι στην εφαρμογή.

Ο χρήστης που θα είναι ασθενής θα πρέπει να εγγράφεται στο σύστημα με την χρήση ενός username και να επιλέγει ένα password.

Κάθε φορά που εισέρχεται στο σύστημα για τον προγραμματισμό ενός ραντεβού, ο ασθενής θα επιλέγει την ειδικότητα που επιθυμεί ή την περιοχή που τον εξυπηρετεί. Έπειτα, θα επιλέγει τον ιατρό με τον οποίο θέλει να κλείσει ραντεβού.

Στη συνέχεια, θα μπορεί να ενημερώνεται για την διαθεσιμότητα του ιατρού και ποιες ημέρες-ώρες είναι διαθέσιμες προκειμένου να κλείσει ραντεβού μαζί του.

Στη συνέχεια θα μπορεί να καταχωρεί-δεσμεύει την ημέρα/ώρες για να κλείσει ραντεβού διευκρινίζοντας το λόγο του ραντεβού.

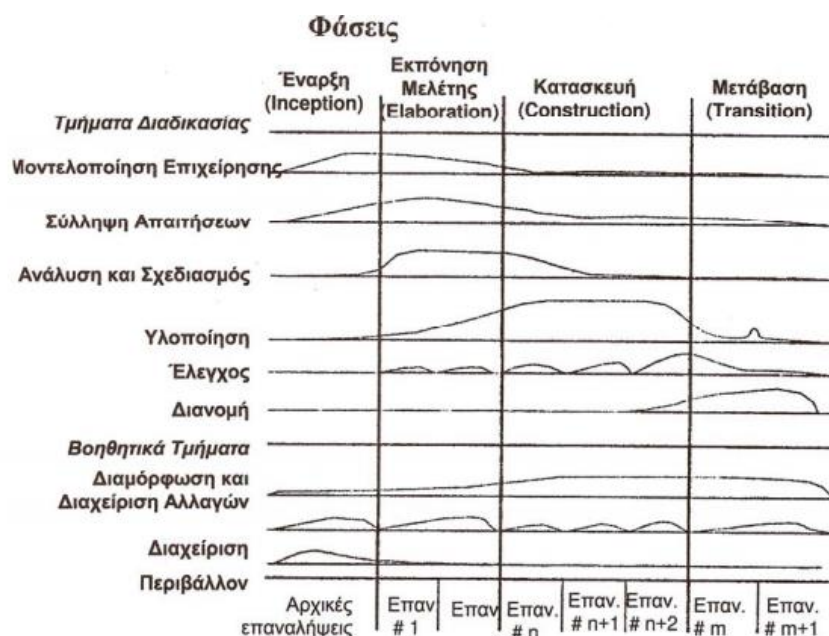
Ο χρήστης (ασθενής ή ιατρός) θα μπορεί να ακυρώσει κάποιο ραντεβού. Επίσης, η εφαρμογή θα πρέπει να έχει την δυνατότητα να προβάλει τα ραντεβού που έχει συνολικά ο χρήστης ανά εβδομάδα και ανά μήνα.

## 2. Σύντομη παρουσίαση της RUP

Στην RUP (Rational Unified Process), ο κύκλος ζωής του λογισμικού είναι επαναληπτικός. Η ανάπτυξη, δηλαδή, προχωρεί σε μια σειρά επαναλήψεων μέχρι να εξελιχθεί το τελικό προϊόν.

Η διαδικασία Rational Unified Process αποτελείται από ένα σύνολο οδηγιών σχετικά με τις τεχνικές και οργανωτικές απόψεις της ανάπτυξης λογισμικού. Η διαδικασία αυτή αφορά κυρίως στην Ανάλυση Απαιτήσεων και στο Σχεδιασμό.

Παρακάτω φαίνεται ο κύκλος ζωής του λογισμικού.



Η διαδικασία δομείται σε δύο διαστάσεις: στον χρόνο, δηλαδή στον χωρισμό του κύκλου ζωής σε φάσεις και επαναλήψεις, και σε τμήματα διαδικασίας, δηλαδή σε καλά ορισμένες εργασίες.

Η δόμηση του έργου σε σχέση με το χρόνο ακολουθεί τις εξής φάσεις:

- Έναρξη (Inception) : καθορίζει την προοπτική του έργου
- Εκπόνηση Μελέτης (Elaboration) : σχεδιασμός των απαιτούμενων δραστηριοτήτων και πόρων. Καθορισμός των χαρακτηριστικών και σχεδιασμός της αρχιτεκτονικής.
- Κατασκευή (Construction): ανάπτυξη προϊόντος σε μία σειρά βημάτων επαναλήψεων.
- Μετάβαση (Transition): προμήθεια του προϊόντος στη κοινότητα χρηστών

Η δομή του έργου σύμφωνα με τη διάσταση των τμημάτων διαδικασίας περιλαμβάνει τις ακόλουθες φάσεις:

- Σύλληψη Απαιτήσεων (Requirements Capture): μια αφήγηση του τι πρέπει να κάνει το σύστημα

- Ανάλυση και Σχεδιασμός (Analysis and Design): μια περιγραφή το πώς θα υλοποιηθεί το σύστημα
- Υλοποίηση (Implementation): η παραγωγή κώδικα
- Έλεγχος (Test): η επαλήθευση του συστήματος

### 3. Φάση: Έναρξη (Inception)

#### 3.1 Σύλληψη Απαιτήσεων

Η διαδικασία απαιτήσεων εστιάζει στο τι πρέπει να κάνει ένα σύστημα λογισμικού, και παρέχει μια μηχανική περιγραφή των αντικειμένων, των λειτουργιών, και των καταστάσεων ενός συστήματος λογισμικού.

Κατά τη διάρκεια της διαδικασίας των απαιτήσεων αναπτύσσονται οι προτεραιότητες, η ολοκλήρωση του λογισμικού και οι ανάγκες της διεπαφής, τα μοντέλα ροής δεδομένων πληροφοριών, η λεπτομερής ανάλυση των κινδύνων, και οι δοκιμές και σχέδια εγκατάστασης.

Η ανάλυση απαιτήσεων είναι μια πολύ δύσκολη διαδικασία και αυτό γιατί οι χρήστες δεν είναι απαραίτητα καλοί γνώστες της Πληροφορικής, οπότε δυσκολεύονται πολύ να ορίσουν τι ακριβώς θέλουν από το λογισμικό.

Από την άλλη πλευρά, οι μηχανικοί λογισμικού μπορεί να θεωρούν κάποια θέματα της Πληροφορικής στοιχειώδη και δεν αντιλαμβάνονται ότι οι τελικοί χρήστες μπορεί να μην τα καταλαβαίνουν.

Σε αυτό το σημείο παρουσιάζονται οι λειτουργίες οι οποίες θα παρέχει η εφαρμογή.

Η εγγραφή στην εφαρμογή πρέπει να είναι δυνατή τόσο για τους ασθενείς όσο και για τους ιατρούς. Με την εγγραφή των ιατρών, δίνεται η δυνατότητα εισαγωγής του ιατρού τους στον κατάλογο και η προβολή όλων των προγραμματισμένων ραντεβού. Με την εγγραφή των ασθενών, δίνεται η δυνατότητα προγραμματισμού ενός ραντεβού με τους διαθέσιμους ιατρούς, καθώς και η προβολή όλων των ραντεβού που έχει κλείσει.

Εφόσον ένας χρήστης έχει ήδη δημιουργήσει έναν λογαριασμό, τότε θα πρέπει να είναι δυνατή η σύνδεση στο λογαριασμό του με την εισαγωγή των απαραίτητων στοιχείων. Με την σύνδεση των ιατρών, δίνεται η δυνατότητα προβολής όλων των προγραμματισμένων ραντεβού, καθώς και η διαγραφή του ιατρού τους από τον κατάλογο. Με την σύνδεση των ασθενών, δίνεται η δυνατότητα κλήσης ενός ραντεβού με τους διαθέσιμους ιατρούς, καθώς και η προβολή όλων των ραντεβού που έχει κλείσει.

Η αναζήτηση των στοιχείων των ιατρών πρέπει να γίνεται όσο το δυνατόν ταχύτερα. Η εφαρμογή θα εμφανίζει το ονοματεπώνυμο του εκάστοτε ιατρού μαζί με τα προσωπικά του στοιχεία. Στα προσωπικά στοιχεία συμπεριλαμβάνονται το τηλέφωνο επικοινωνίας του, η διεύθυνση του ιατρού του και επιπλέον πληροφορίες που θεωρεί απαραίτητες ο κάθε ιατρός να φαίνονται (όπως το email).

Μία επιπλέον λειτουργία που θεωρήθηκε απαραίτητη να συμπεριληφθεί στην εφαρμογή είναι η δυνατότητα διαγραφής ενός ιατρού από τον ίδιο τον ιατρό. Με την διαγραφή του επαγγελματικού προφίλ, αυτόματα αφαιρείται και το ιατρείο από τον κατάλογο των διαθέσιμων ιατρών.

Σημαντική λειτουργία είναι και αυτή του λανθασμένου κωδικού πρόσβασης. Η εφαρμογή οφείλει να ενημερώνει τον χρήστη με σχετικό μήνυμα για την εισαγωγή λανθασμένου κωδικού.

Για την αποτελεσματικότερη περιήγηση στην πλατφόρμα, η εφαρμογή οφείλει να περιέχει ένα μενού επιλογών στο πάνω μέρος της σελίδας, βάση του οποίου μπορούν να μεταφερθούν οι χρήστες της εφαρμογής στις διαφορετικές λειτουργίες που προσφέρονται.

Βασικός στόχος της εφαρμογής είναι η ευχρηστία. Αυτή είναι μια βασική προϋπόθεση διότι δεν υπάρχει ηλικιακό όριο από τη πλευρά του χρήστη. Έτσι, το user interface οφείλει να είναι απλό και κατανοητό για κάθε είδος χρήστη, καθώς και επεκτάσιμο για μελλοντικές προσθήκες που μπορούν να γίνουν στην εφαρμογή.

Γενικές λειτουργικές απαιτήσεις της εφαρμογής:

- Η εφαρμογή είναι web.
- Τα χρώματα που επιλέγονται στο user interface είναι ενιαία και συνεπή για κάθε σελίδα.
- Το μενού επιλογών είναι σταθερό και στο ίδιο σημείο για κάθε σελίδα.
- Η γραμματοσειρά είναι η ίδια σε κάθε σελίδα.
- Υπάρχει σε κάθε σελίδα κουμπί για online help το οποίο δίνει οδηγίες στον χρήστη για το ποιες ενέργειες μπορεί να εκτελέσει στην εκάστοτε σελίδα.
- Όλα τα κουμπιά και οι clickable περιοχές της εφαρμογής έχουν το ίδιο χρώμα για να τα αναγνωρίζει ο χρήστης.

Λειτουργικές Απαιτήσεις για τον Χρήστη-Ασθενή:

- Δημιουργία λογαριασμού χρήστη με την εισαγωγή των προσωπικών του στοιχείων και αποδοχή των όρων χρήσης της εφαρμογής και πολιτικής απορρήτου.
- Σύνδεση του χρήστη με την εισαγωγή του username του και τον κωδικό χρήστη που έχει οριστεί κατά την εγγραφή.
- Πλοήγηση στην web εφαρμογή.
- Προβολή ιστορικού ραντεβού (ενεργά, παλαιότερα και ακυρωμένα).
- Αναζήτηση ιατρικών και πληροφοριών σχετικά με αυτά.
- Διαχείριση προφίλ: αλλαγή προσωπικών στοιχείων, αλλαγή κωδικού, διαγραφή προφίλ.
- Διαδικασία προγραμματισμού ενός ραντεβού: επιλογή ειδικότητας ιατρού, ημέρας, ώρας, περιοχής και προβολή των διαθέσιμων ιατρών.
- Επιλογή του ιατρού με τον οποίο επιθυμεί να κλείσει ραντεβού.
- Μήνυμα επιβεβαίωσης ραντεβού.
- Ακύρωση ραντεβού από την λίστα ιστορικού και εφόσον το ραντεβού είναι ενεργό (δηλαδή δεν έχει περάσει η ημέρα και ώρα του ραντεβού).

Λειτουργικές Απαιτήσεις για τον Χρήστη-Ιατρό:

- Δημιουργία λογαριασμού χρήστη με την εισαγωγή των προσωπικών του στοιχείων και αποδοχή των όρων χρήσης της εφαρμογής και πολιτικής απορρήτου.
- Σύνδεση του χρήστη με την εισαγωγή του username του και τον κωδικό χρήστη που έχει οριστεί κατά την εγγραφή.
- Πλοήγηση στην web εφαρμογή.
- Εισαγωγή του ιατρείου στον κατάλογο.
- Διαγραφή του ιατρείου από τον κατάλογο.
- Προβολή ιστορικού ραντεβού (ενεργά, παλαιότερα και ακυρωμένα).
- Διαχείριση προφίλ: αλλαγή προσωπικών στοιχείων, αλλαγή κωδικού, διαγραφή προφίλ.
- Ακύρωση ραντεβού από την λίστα ιστορικού και εφόσον το ραντεβού είναι ενεργό (δηλαδή δεν έχει περάσει η ημέρα και ώρα του ραντεβού).

### Μη λειτουργικές απαιτήσεις της εφαρμογής:

- Απόδοση: ο χρόνος απόκρισης του συστήματος θα πρέπει να είναι λιγότερος των 10 δευτερολέπτων.
- Ευχρηστία: σε κάθε ενέργεια που κάνει ο χρήστης θα πρέπει να υπάρχει ανταπόκριση από το σύστημα.
- Εγχειρίδιο χρήσης: πρόσβαση από την αρχική σελίδα της εφαρμογής.
- Online Help: διαθέσιμο σε κάθε σελίδα της εφαρμογής με οδηγίες για την αποτελεσματική χρήση της.
- Ιδιωτικότητα: τα προσωπικά στοιχεία του χρήστη είναι διαθέσιμα μόνο στον ίδιο.
- Αξιοπιστία και Διαθεσιμότητα: η εφαρμογή είναι διαθέσιμη 24/7. Η εφαρμογή είναι επεκτάσιμη και συμβατή με κάθε web browser.

## **3.2 Ανάλυση – Σχεδιασμός**

Με γνώμονα τις παραπάνω απαιτήσεις, θα πραγματοποιηθούν τα Διαγράμματα UML. Η UML είναι γλώσσα μοντελοποίησης η οποία βοηθάει στην περιγραφή του σχεδιασμού.

Η UML ορίζει 9 είδη διαγραμμάτων για να αναπαραστήσει τις διαφορετικές απόψεις μοντελοποίησης:

- Διαγράμματα Περιπτώσεων Χρήσης (Use case diagrams): Αναπαριστούν τις λειτουργίες ενός συστήματος από την οπτική γωνία του χρήστη.
- Διαγράμματα Τάξεων (Class Diagrams): Αναπαριστούν τη στατική δομή όσον αφορά στις τάξεις και τις σχέσεις τους.
- Διαγράμματα αντικειμένων (Object Diagrams): Αναπαριστούν αντικείμενα και τις σχέσεις τους και αντιστοιχούν σε απλοποιημένα διαγράμματα συνεργασίας που δεν αναπαριστούν μεταδόση μηνυμάτων.
- Διαγράμματα συνεργασίας (Collaboration Diagrams): Η αναπαράσταση των αντικειμένων, συνδέσεων και αλληλεπιδράσεων.
- Διαγράμματα σειράς (Sequence diagrams): Χρονική αναπαράσταση των αντικειμένων και των αλληλεπιδράσεών τους.
- Διαγράμματα καταστάσεων (Statechart diagrams): Αναπαριστούν τη συμπεριφορά της τάξης όσον αφορά στις καταστάσεις της.
- Διαγράμματα δραστηριοτήτων (Activity diagrams): Αναπαριστούν τη συμπεριφορά μιας λειτουργίας ως σύνολο ενεργειών.
- Διαγράμματα εξαρτημάτων (Component diagrams): Αναπαριστούν τα φυσικά εξαρτήματα μιας εφαρμογής.
- Διαγράμματα διανομής (Deployment diagrams): Αναπαριστούν τη διανομή των εξαρτημάτων σε συγκεκριμένα τεμάχια του hardware (υλικού).

### **3.2.1 Διάγραμμα Περιπτώσεων Χρήσης – Use Case Diagram (1<sup>η</sup> έκδοση)**

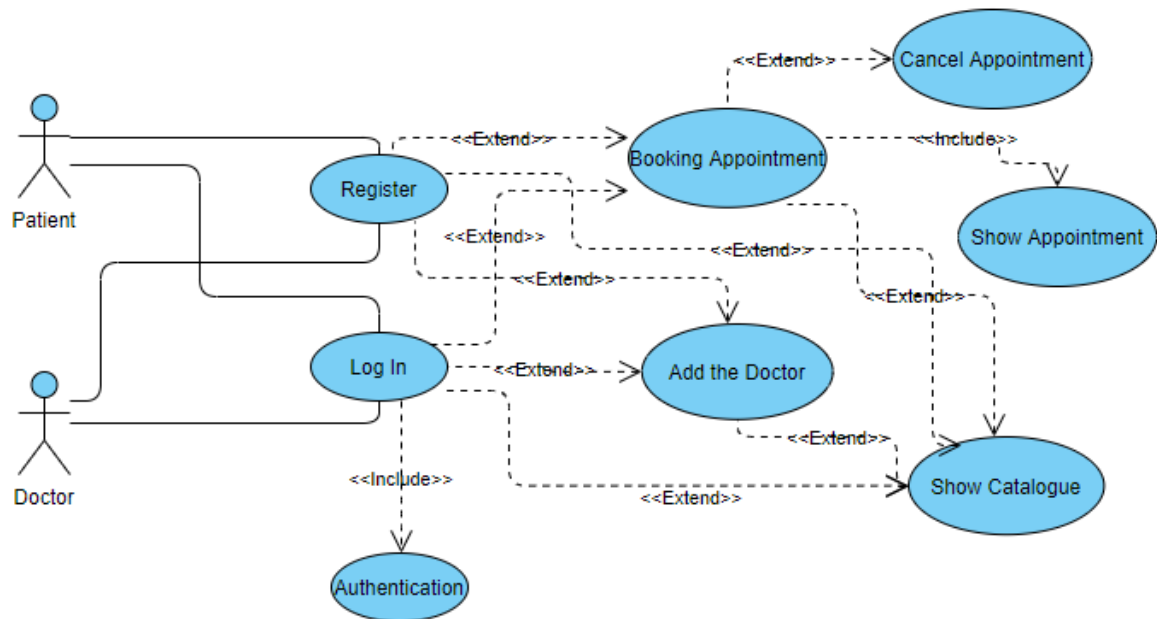
Τα Διαγράμματα Περιπτώσεων Χρήσης περιγράφουν τη συμπεριφορά ενός συστήματος από την οπτική γωνία ενός χρήστη. Επιτρέπουν τον ορισμό των ορίων του συστήματος και του περιβάλλοντος. Μία περίπτωση χρήσης αντιστοιχεί σε ένα συγκεκριμένο είδος χρήσης του συστήματος. Είναι μια εικόνα της λειτουργικότητας ενός συστήματος το οποίο ενεργοποιείται για να ανταποκριθεί σε έναν εξωτερικό ενεργοποιό (actor). Το μοντέλο περιπτώσεων χρήσης περιλαμβάνει : τις ίδιες τις περιπτώσεις χρήσης, τους ενεργοποιούς (actors).

Ένας ενεργοποιός αναπαριστά ένα ρόλο που παίζεται από ένα άτομο ή πράγμα που αλληλεπιδρά με το σύστημα. Οι ενεργοποιοί προσδιορίζονται παρατηρώντας τους άμεσους χρήστες του συστήματος (αυτούς που το χρησιμοποιούν και το συντηρούν). Επίσης παρατηρώντας κάθε άλλο σύστημα που αλληλεπιδρά με αυτό που αναπτύσσεται. Το ίδιο φυσικό πρόσωπο μπορεί να παίζει το ρόλο πολλών ενεργοποιών. Ενώ πολλοί άνθρωποι μπορεί να παίζουν τον ίδιο ρόλο. Το όνομα του ενεργοποιού περιγράφει το ρόλο που παίζει ο χρήστης.

Τα διαγράμματα περιπτώσεων χρήσης αναπαριστούν περιπτώσεις χρήσης, ενεργοποιούς και σχέσεις μεταξύ περιπτώσεων χρήσης και ενεργοποιών. Η UML ορίζει 3 τρόπους συνδέσμων μεταξύ ενεργοποιών και περιπτώσεων – χρήσης :

- Η σχέση “επικοινωνεί” (communicate) : Η συμμετοχή του ενεργοποιού σηματοδοτείται από μία γραμμή μεταξύ του ενεργοποιού και της περίπτωσης χρήσης. Αυτή είναι η μόνη σχέση που μπορεί να υπάρξει μεταξύ ενεργοποιών και περιπτώσεων χρήσης.
- Η σχέση “χρησιμοποιεί” (uses) : Η σχέση αυτή ορίζεται μεταξύ περιπτώσεων χρήσης και σημαίνει ότι ένα στιγμιότυπο της πηγής (περίπτωσης χρήσης) συμπεριλαμβάνει τη συμπεριφορά του στόχου (περίπτωσης χρήσης).
- Η σχέση “επεκτείνει” (extends) : Μία σχέση “extends” μεταξύ δύο περιπτώσεων χρήσης σημαίνει ότι η πηγή (περίπτωσης χρήσης) επεκτείνει τη συμπεριφορά του στόχου (περίπτωσης χρήσης).

Στην περίπτωση της εφαρμογής, υπάρχουν δύο είδη ενεργοποιών - χρηστών, ο ασθενής (Patient) και ο ιατρός (Doctor).



### 3.2.2 Διάγραμμα Τάξεων – Class Diagram (1<sup>η</sup> έκδοση)

Στα Διαγράμματα Τάξεων, οι τάξεις αναπαρίστανται ως ορθογώνια τα οποία περιέχουν διαχωρισμούς. Το πρώτο τμήμα του ορθογωνίου περιέχει το όνομα της τάξης, το δεύτερο περιέχει τα χαρακτηριστικά της τάξης και το τρίτο τις λειτουργίες της.

Υπάρχουν τρία είδη ορατότητας για τα χαρακτηριστικά και τις λειτουργίες μίας κλάσης:



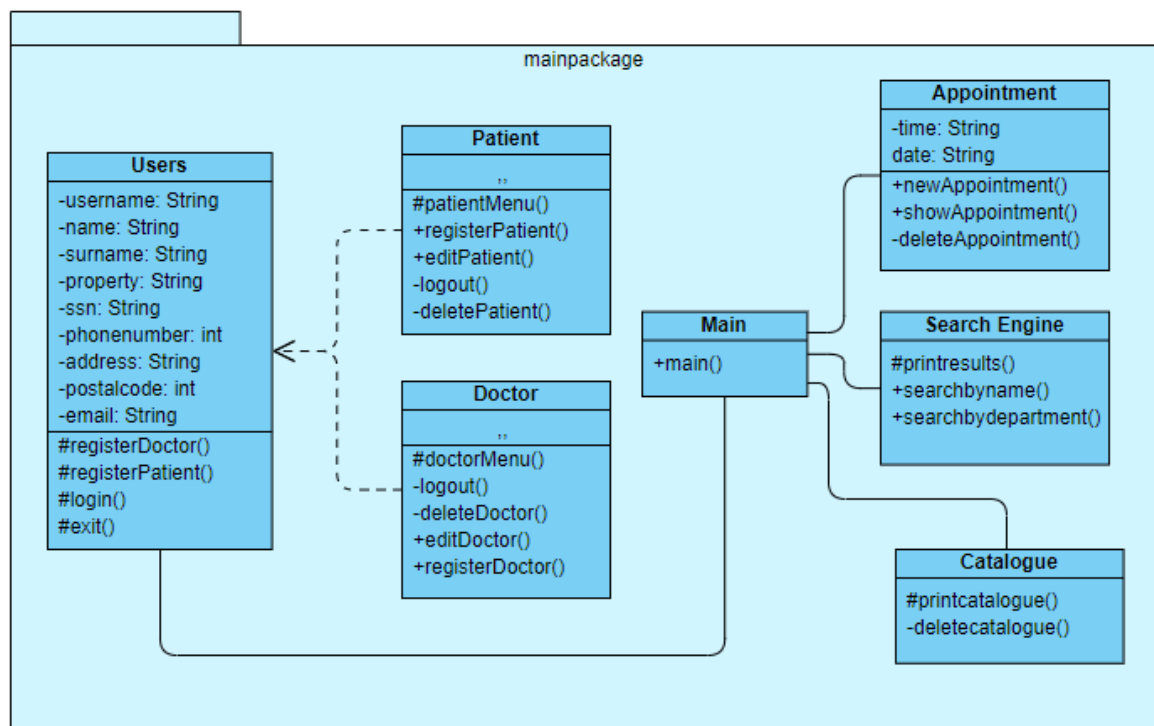
- Δημόσιο (public): Το στοιχείο είναι ορατό από όλους τους πελάτες της τάξης. Συμβολίζεται με +.
- Προστατευόμενο (protected): Το στοιχείο είναι ορατό στις υποτάξεις της τάξης. Συμβολίζεται με #.
- Ιδιωτικό (private): Το στοιχείο είναι ορατό μόνο στην τάξη. Συμβολίζεται με -.

Μερικά χαρακτηριστικά και λειτουργίες μπορεί να είναι ορατά σφαιρικά σε ολόκληρη την ισχύ της τάξης. Αυτά τα στοιχεία ονομάζονται μεταβλητές τάξης (Class variables) και λειτουργίες τάξης (Class Operations). Μια μεταβλητή τάξης μοιάζει σαν ένα αντικείμενο το οποίο μοιράζονται τα στιγμιότυπα μιας τάξης. Επειδή μοιάζουν με αντικείμενα συμβολίζονται με υπογράμμιση.

Η UML αναπαριστά τις διεπαφές χρησιμοποιώντας μικρούς κύκλους που συνδέονται με μια γραμμή στο στοιχείο που παρέχει τις υπηρεσίες που περιγράφει η διεπαφή.

Οι συσχετισμοί αναπαριστούν δομικές σχέσεις μεταξύ τάξεων των αντικειμένων. Ένας συσχετισμός συμβολίζει μια πληροφορία με κύκλο ζωής. Οι περισσότεροι συσχετισμοί είναι δυαδικοί, δηλαδή συνδέουν δύο τάξεις. Οι συσχετισμοί συμβολίζονται με ευθείες γραμμές.

Παρακάτω φαίνεται η πρώτη έκδοση του διαγράμματος τάξεων και αποτελεί μια πιο αφηρημένη έννοια των τάξεων που θα δημιουργηθούν.



## 4. Φάση: Εκπόνηση Μελέτης (Elaboration)

Η εκπόνηση μελέτης είναι ο σχεδιασμός των απαιτούμενων δραστηριοτήτων και πόρων και καθορισμός των χαρακτηριστικών και σχεδιασμός της αρχιτεκτονικής.

### 4.1 Ανάλυση – Σχεδιασμός

Σε αυτή τη φάση της εφαρμογής, προέκυψαν κάποιες επιπλέον λειτουργίες οι οποίες θα προστεθούν για να ικανοποιηθούν όλες οι απαιτήσεις:

- Για την πρόσβαση του χρήστη θα πρέπει να υπάρχει κάλυψη δικτύου από την μεριά του χρήστη.
- Η εφαρμογή θα πρέπει να διαθέτει μία βάση δεδομένων, η οποία θα ενημερώνεται αυτόματα για την προσθήκη, ενημέρωση και διαγραφή στοιχείων από την εφαρμογή.
- Η εφαρμογή θα πρέπει να ειδοποιεί τον χρήστη σε περίπτωση που δεν υπάρχουν διαθέσιμα αποτελέσματα για την αναζήτηση που πραγματοποιήσε.
- Επίσης, ο χρήστης δεν θα πρέπει να έχει την δυνατότητα να αλλάξει το ΑΜΚΑ του και το ονοματεπώνυμό του, εφόσον κάθε ΑΜΚΑ είναι μοναδικό και ανήκει σε ένα ονοματεπώνυμο. Παραμετροποίηση θα επιτρέπεται μόνο για τα πεδία : τηλέφωνο, διεύθυνση, ταχυδρομικός κώδικας, email και κωδικός χρήστη.
- Για την εισαγωγή του ιατρού στην εφαρμογή, επισημαίνεται ότι κατά τη δημιουργία του λογαριασμού, θα εισάγεται αυτόματα και στην λίστα.
- Κατά την επιτυχή σύνδεση ή εγγραφή στην σελίδα, ο χρήστης θα μεταφέρεται στην σελίδα όπου θα μπορεί να διαβάσει τα προσωπικά του στοιχεία (profile information) και να τα τροποποιήσει, όπου αυτό είναι δυνατόν.
- Θα υπάρχει σελίδα η οποία θα ονομάζεται Catalogue και στην οποία θα αναφέρονται όλες οι πληροφορίες σχετικά με τα προγραμματισμένα του ραντεβού. Σε αυτήν την σελίδα, θα μπορεί να γίνει και η ακύρωση ενός ενεργού ραντεβού, τόσο από την μεριά του ασθενή όσο και από την μεριά του ιατρού.
- Θα υπάρχει σελίδα με όνομα Book Appointment στην οποία ο χρήστης θα μπορεί να αναζητήσει ανά περιοχή ή ειδικότητα τον ιατρό που επιθυμεί για να προγραμματίσει ραντεβού. Αφού διαλέξει τον ιατρό με τον οποίο επιθυμεί να κλείσει ραντεβού, θα πατά το αντίστοιχο κουμπί που θα βρίσκεται δίπλα από κάθε ιατρό με όνομα 'Book Appointment'. Έστερα, θα του ζητείται η εισαγωγή της επιθυμητής ώρας, μέρας του ραντεβού, καθώς και λόγου επίσκεψης και εφόσον υπάρχει διαθεσιμότητα, θα του ζητείται επιβεβαίωση του ραντεβού. Η σελίδα αυτή είναι διαθέσιμη μόνο για χρήστες με την ιδιότητα του ασθενούς. Εφόσον ο χρήστης επικυρώσει το ραντεβού, θα μεταφέρεται αυτόματα στην σελίδα με όνομα Catalogue προκειμένου να επιβεβαιώσει ότι το ραντεβού που μόλις προγραμματίσε εμφανίζεται στην λίστα με τα προγραμματισμένα του ραντεβού.
- Ο χρήστης θα μπορεί να αποσυνδεθεί από το προφίλ του πατώντας το αντίστοιχο κουμπί Log Out που θα βρίσκεται στο μενού της εφαρμογής. Με την αποσύνδεσή του, αυτόματα μεταφέρεται στην αρχική σελίδα της εφαρμογής όπου του ζητείται η εισαγωγή των στοιχείων του αν επιθυμεί να ξανασυνδεθεί ή η εγγραφή ενός νέου χρήστη.

Ύστερα της προσθήκης επιπλέον απαιτήσεων και διευκρινήσεων, τέθηκε απαραίτητη η επανεξέταση των διαγραμμάτων περιπτώσεων χρήσης και κλάσεων και η υλοποίηση πρώτων εκδόσεων από τα υπόλοιπα διαγράμματα που αναφέρονται στην UML.

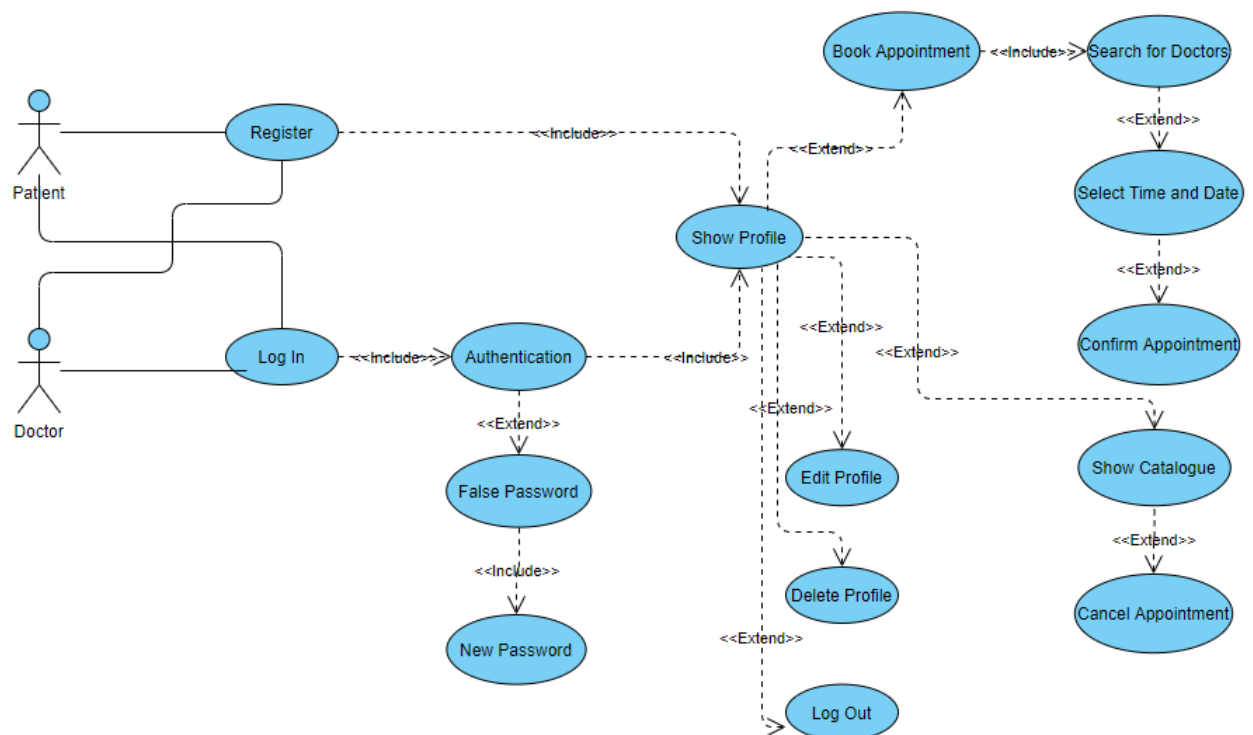
Σε αυτό το σημείο παρατηρείται και η ανάγκη για επαναληπτικότητα, η οποία θεσπίζει την Rational Unified Process.

#### 4.1.1 Διάγραμμα Περιπτώσεων Χρήσης – Use Case Diagram (2<sup>η</sup> έκδοση)

Παρακάτω φαίνεται η δεύτερη έκδοση του διαγράμματος περιπτώσεων χρήσης. Αυτή η έκδοση δεν είναι η τελική. Προστέθηκαν παραπάνω ενέργειες από αυτές που περιείχε το αρχικό διάγραμμα, όπως η αυθεντικοποίηση ενός χρήστη (authentication) η οποία δίνει τη δυνατότητα αλλαγής του κωδικού πρόσβασης σε περίπτωση που ο χρήστης της εφαρμογής τον έχει ξεχάσει.

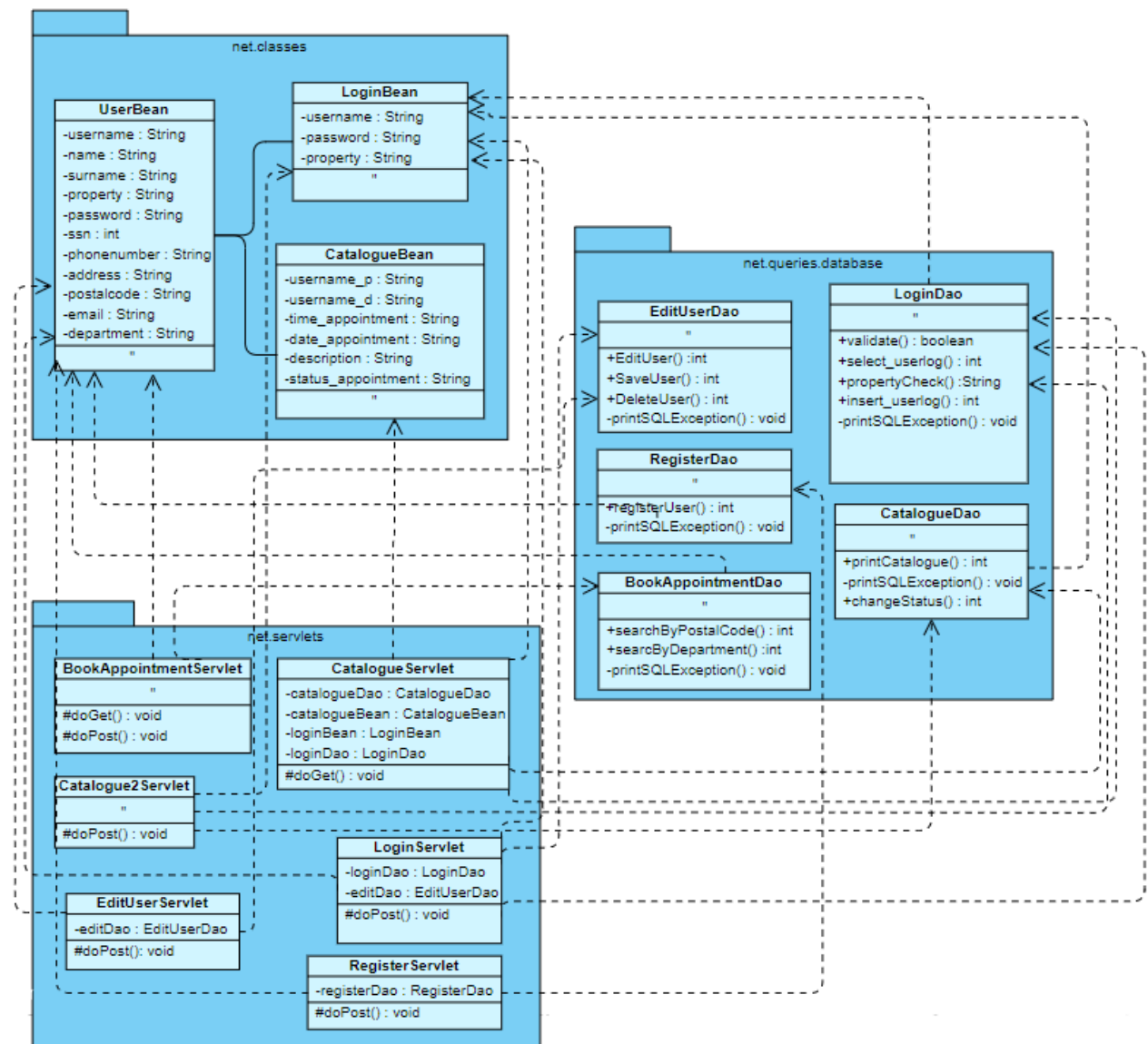
Επίσης, παρουσιάστηκε και η ανάγκη για μία αρχική σελίδα κατόπιν εισόδου του χρήστη στην εφαρμογή η οποία αναφέρεται ως Show Profile. Από εκεί ο χρήστης μπορεί να τελέσει όποια διαθέσιμη λειτουργία υπάρχει στην εφαρμογή.

Επιπλέον, συμπεριλήφθηκε και η δυνατότητα εξόδου (Log Out) από την εφαρμογή.



#### 4.1.2 Διάγραμμα Τάξεων – Class Diagram (2<sup>η</sup> έκδοση)

Παρακάτω φαίνεται η δεύτερη έκδοση του διαγράμματος τάξεων.



#### 4.1.3 Διάγραμμα Αντικειμένων – Object Diagram (1<sup>η</sup> έκδοση)

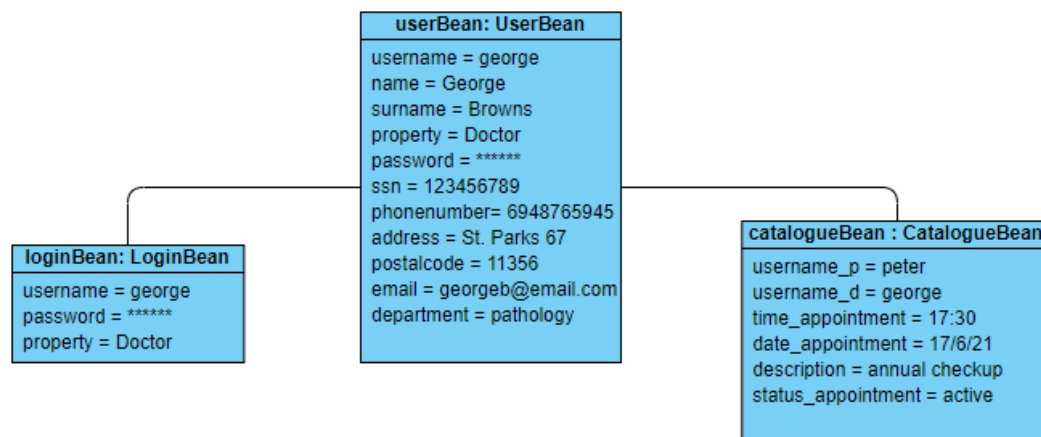
Το Διάγραμμα Αντικειμένων είναι ένα στιγμιότυπο των αντικειμένων του συστήματος κάποια συγκεκριμένη χρονική στιγμή. Χρησιμοποιείται για τον σχεδιασμό στατικής δομής της εφαρμογής και απεικονίζει τα αντικείμενα μαζί με τους συνδέσμούς τους.

Κάθε αντικείμενο αναπαρίστανται με ένα ορθογώνιο, το οποίο περιέχει είτε το όνομα του αντικειμένου, είτε το όνομα και την τάξη του αντικειμένου (χωρισμένα με άνω και κάτω τελεία), ή μόνο την τάξη του αντικειμένου (σε αυτή την περίπτωση το αντικείμενο θεωρείται ότι είναι ανώνυμο). Το όνομα από μόνο του αντιστοιχεί σε ένα ελλειπές μοντέλο, στο οποίο η τάξη του αντικειμένου δεν έχει ακόμα καθοριστεί.

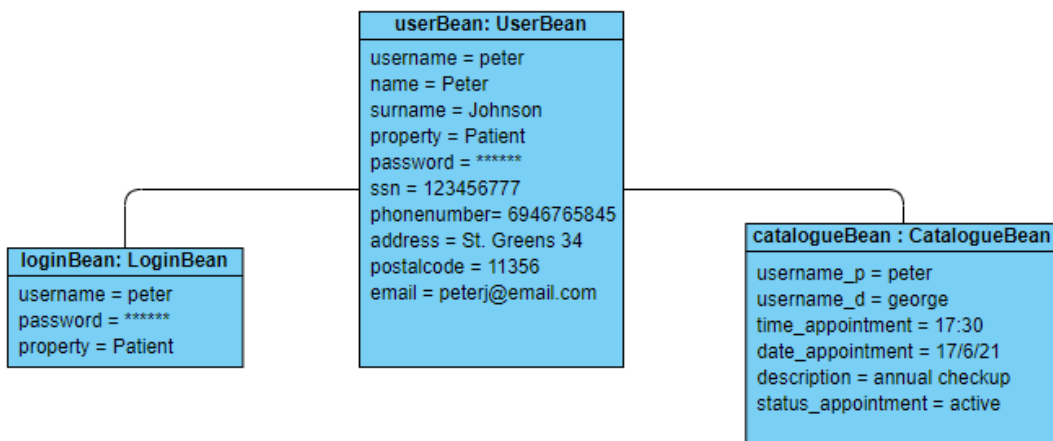
Τα ορθογώνια που συμβολίζουν αντικείμενα μπορούν επίσης να περιλαμβάνουν ένα δεύτερο τμήμα που περιέχει τις τιμές των χαρακτηριστικών. Ο τύπος του χαρακτηριστικού είναι ήδη καθορισμένος στην τάξη, έτσι δεν είναι απαραίτητο να εμφανίζεται στην αναπαράσταση των αντικειμένων. Τα αντικείμενα συνδέονται μέσω συνδέσμων, οι οποίοι είναι στιγμιότυπα των συσχετισμών μεταξύ των τάξεων των αντικειμένων που μελετώνται. Οι περισσότεροι σύνδεσμοι είναι δυαδικοί.

Παρακάτω φαίνεται η πρώτη έκδοση του διαγράμματος αντικειμένων για την περίπτωση ενός ιατρού και στη συνέχεια για την περίπτωση ενός ασθενή.

Ιατρός:



Ασθενής:

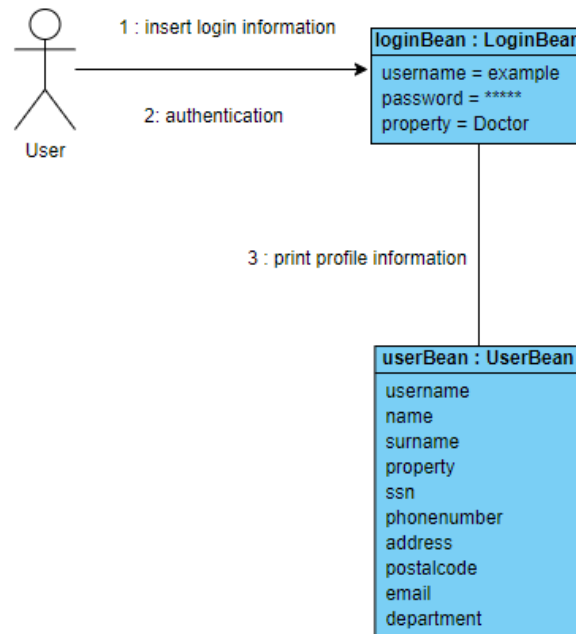


#### 4.1.4 Διάγραμμα Συνεργασίας – Collaboration Diagram (1<sup>η</sup> έκδοση)

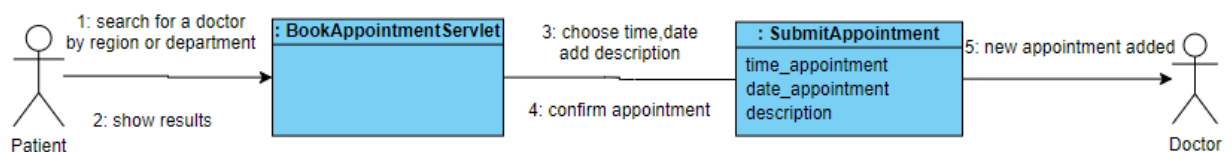
Τα διαγράμματα συνεργασίας απεικονίζουν τις αλληλεπιδράσεις ανάμεσα στα αντικείμενα, χρησιμοποιώντας μία στατική χωρική δομή, που διευκολύνει την αναπαράσταση της συνεργασίας ενός συνόλου αντικειμένων. Τα διαγράμματα συνεργασίας εκφράζουν τόσο το περιβάλλον ενός συνόλου αντικειμένων (μέσω των αντικειμένων και των συνδέσμων) όσο και την αλληλεπίδραση ανά μ ε σε αυτά τα αντικείμενα (αναπαριστώντας εκπομπές μηνυμάτων). Αυτά τα διαγράμματα είναι μία επέκταση των διαγραμμάτων αντικειμένων.

Το περιεχόμενο μίας αλληλεπίδρασης περιλαμβάνει τα ορίσματα, τις τοπικές μεταβλητές που δημιουργήθηκαν κατά την διάρκεια της εκτέλεσης, και τους συνδέσμους ανάμεσα στα αντικείμενα που συμμετέχουν στην αλληλεπίδραση. Μία αλληλεπίδραση εκτελείται από ένα σύνολο αντικειμένων που συνεργάζονται ανταλλάσσοντας μηνύματα. Αυτά τα μηνύματα εμφανίζονται κατά μήκος των συνδέσμων που συνδέουν τα αντικείμενα, χρησιμοποιώντας βέλη που δείχνουν προς τον παραλήπτη του μηνύματος.

Παρακάτω φαίνεται η πρώτη έκδοση του διαγράμματος συνεργασίας για την περίπτωση εισόδου στην εφαρμογή.



Επίσης, παρακάτω φαίνεται η πρώτη έκδοση του διαγράμματος συνεργασίας για την περίπτωση του προγραμματισμού ενός ραντεβού.



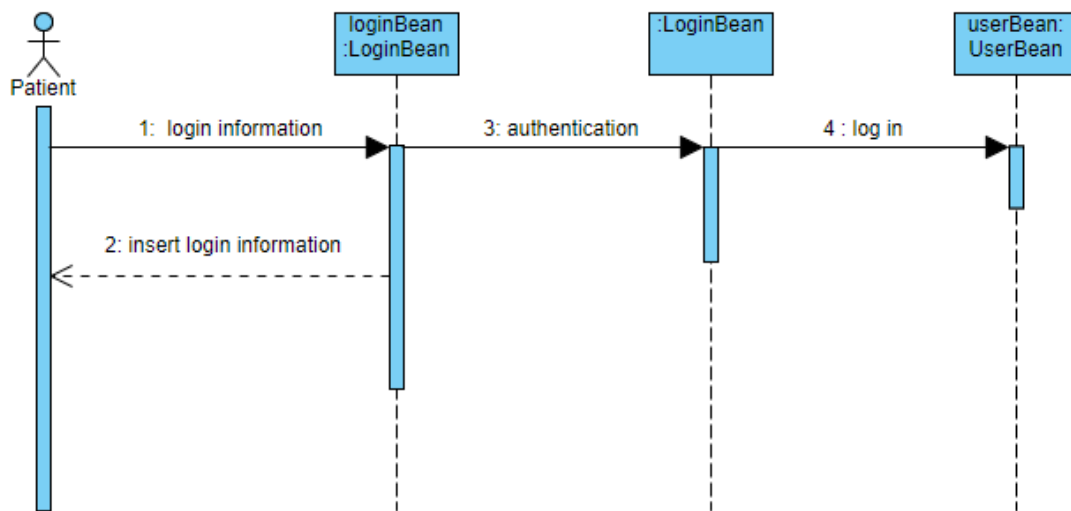
#### 4.1.5 Διάγραμμα Σειράς – Sequence Diagram (1<sup>η</sup> έκδοση)

Τα διαγράμματα σειράς αναπαριστούν αλληλεπιδράσεις ανάμεσα στα αντικείμενα από μία χρονική άποψη. Σε αντίθεση με τα διαγράμματα συνεργασίας, το περιβάλλον των αντικειμένων δεν αναπαρίσταται σαφώς. Η αναπαράσταση επικεντρώνεται στην έκφραση των αλληλεπιδράσεων.

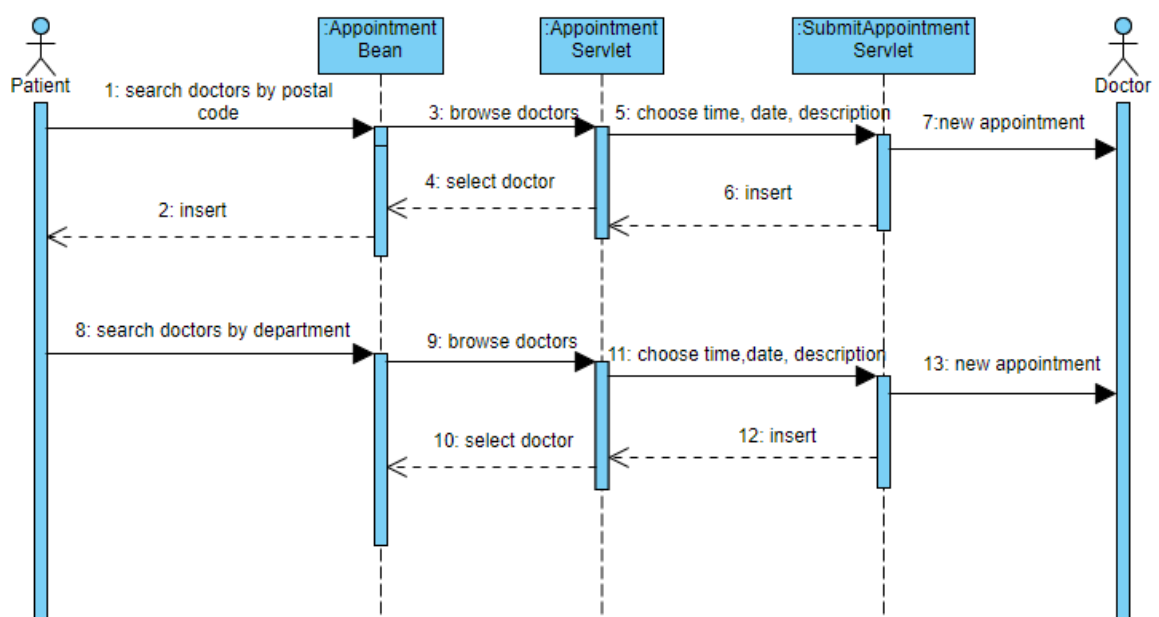
Σε ένα διάγραμμα σειράς αναπαρίσταται μία αλληλεπίδραση ανάμεσα σε αντικείμενα, που επικεντρώνεται στη χρονολογία της εκπομπής μηνυμάτων. Ένα αντικείμενο αναπαρίσταται με ένα ορθογώνιο και μία κάθετη γραμμή, που καλείται γραμμή ζωής του αντικειμένου.

Τα αντικείμενα επικοινωνούν ανταλλάσσοντας μηνύματα, τα οποία αναπαριστώνται με οριζόντια βέλη σχεδιασμένα από τον αποστολέα του μηνύματος προς τον παραλήπτη του μηνύματος. Η σειρά αποστολής μηνύματος καθορίζεται από τη θέση του μηνύματος στον κάθετο άξονα. Ο κάθετος άξονας μπορεί να περιγράφεται με ετικέτες για να εκφράζει ακριβώς τους χρονικούς περιορισμούς.

Παρακάτω φαίνεται η πρώτη έκδοση του διαγράμματος σειράς για την αλληλεπίδραση μεταξύ των αντικειμένων, όταν ο χρήστης-ασθενής θέλει να κάνει Log In στην εφαρμογή.



Παρακάτω φαίνεται η πρώτη έκδοση του διαγράμματος σειράς για την αλληλεπίδραση των αντικειμένων, όταν ο χρήστης-ασθενής θέλει να προγραμματίσει ένα ραντεβού.



#### 4.1.6 Διάγραμμα Δραστηριοτήτων – Activity Diagram (1<sup>η</sup> έκδοση)

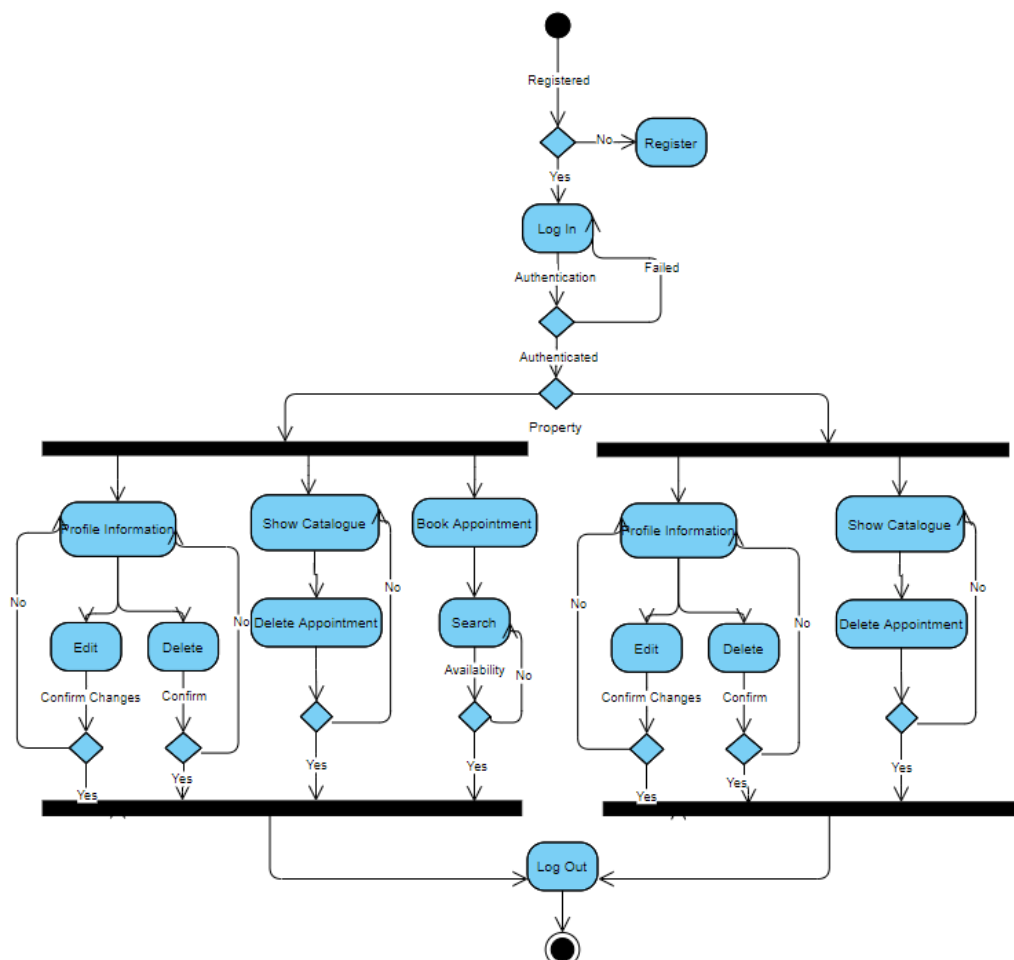
Ένα διάγραμμα δραστηριοτήτων είναι μία παραλλαγή των διαγραμμάτων καταστάσεων οργανωμένο σύμφωνα με ενέργειες, και στοχεύει κυρίως στην αναπαράσταση της εσωτερικής συμπεριφοράς μίας μεθόδου (την υλοποίηση μίας λειτουργίας) ή μίας περίπτωσης χρήσης. Αναπαριστά την κατάσταση εκτέλεσης ενός μηχανισμού σαν μία σειρά βημάτων που ομαδοποιούνται σειριακά σαν παράλληλες διακλαδώσεις ροής ελέγχου.

Κάθε δραστηριότητα αναπαριστά μία συγκεκριμένη κατάσταση στην εκτέλεση της περικλείουσας μεθόδου. Οι δραστηριότητες συνδέονται με αυτόματες μεταβάσεις, που αναπαριστώνται με βέλη, με τον ίδιο τρόπο όπως οι μεταβάσεις στα διαγράμματα καταστάσεων. Όταν μία δραστηριότητα τερματίζει, η μετάβαση ενεργοποιείται και η επόμενη δραστηριότητα αρχίζει. Οι δραστηριότητες δεν έχουν εσωτερικές μεταβάσεις ή μεταβάσεις που ενεργοποιούνται από γεγονότα.

Οι μεταβάσεις ανάμεσα στις δραστηριότητες μπορούν να φυλαχτούν με αμοιβαία αποκλειόμενες συνθήκες που λαμβάνουν λογικές τιμές 0 ή 1. Οι φύλακες φαίνονται δίπλα στις μεταβάσεις την ενεργοποίηση των οποίων επικυρώνουν.

Τα διαγράμματα δραστηριοτήτων δείχνουν συγχρονισμούς ανάμεσα στις ροές ελέγχου χρησιμοποιώντας ράβδους συγχρονισμού. Μία ράβδος συγχρονισμού κάνει δυνατό το άνοιγμα και το κλείσιμο παράλληλων διακλαδώσεων μέσα στη ροή εκτέλεσης μίας μεθόδου ή μίας περίπτωσης χρήσης. Οι μεταβάσεις που αρχίζουν από τη μπάρα συγχρονισμού ενεργοποιούνται ταυτόχρονα.

Παρακάτω φαίνεται η πρώτη έκδοση του διαγράμματος δραστηριοτήτων.





#### 4.1.7 Διάγραμμα Καταστάσεων –State Chart Diagram (1<sup>η</sup> έκδοση)

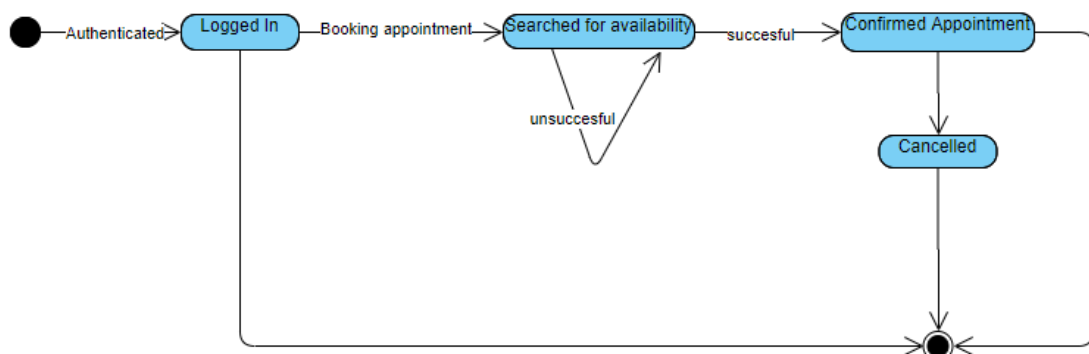
Τα διαγράμματα καταστάσεων αναπαριστούν μηχανές καταστάσεων από την άποψη των καταστάσεων και των μεταβάσεων.

Η συμπεριφορά των αντικειμένων μίας τάξης μπορεί να περιγραφεί τυπικά με όρους καταστάσεων και γεγονότων, χρησιμοποιώντας μία μηχανή καταστάσεων συνδεδεμένη με την τάξη που μελετάται. Αντικείμενα που δεν παρουσιάζουν μία πολύ αντιδραστική συμπεριφορά μπορεί να θεωρηθεί ότι μένουν πάντα στην ίδια κατάσταση. Σε αυτή την περίπτωση οι τάξεις τους δεν κατέχουν μία μηχανή καταστάσεων.

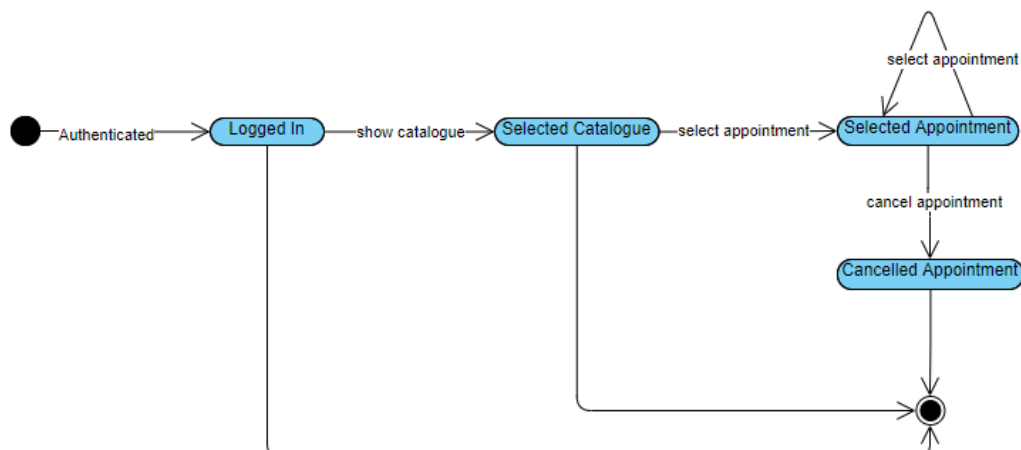
Μία μηχανή καταστάσεων είναι μία αφαίρεση όλων των πιθανών συμπεριφορών, όμοια με τον τρόπο που τα διαγράμματα τάξεων είναι αφαιρέσεις της στατικής δομής. Κάθε αντικείμενο ακολουθεί τη συμπεριφορά που περιγράφεται στη μηχανή καταστάσεων και συσχετίζεται με την τάξη του και είναι, σε μία δεδομένη χρονική στιγμή, σε μία κατάσταση που χαρακτηρίζει τις δυναμικές του καταστάσεις.

Οι μηχανές καταστάσεων και τα σενάρια είναι συμπληρωματικά. Τα σενάρια αναπαριστώνται με μία συνεργασία μεταξύ αντικειμένων. Ο τύπος της αλληλεπίδρασης που συμβαίνει μεταξύ αντικειμένων που συνεργάζονται μέσα σε ένα σενάριο καθορίζεται από τις αντίστοιχες καταστάσεις των διάφορων αντικειμένων. Οι μηχανές καταστάσεων μπορούν να χρησιμοποιηθούν για να περιγράψουν τη συμπεριφορά συνόλων αντικειμένων συσχετίζοντας μία μηχανή καταστάσεων με ένα σύνθετο αντικείμενο, ή ακόμα και με μία περίπτωση χρήσης (use case).

Παρακάτω φαίνεται η πρώτη έκδοση του διαγράμματος καταστάσεων για τον προγραμματισμό ενός ραντεβού.



Παρακάτω φαίνεται η πρώτη έκδοση του διαγράμματος καταστάσεων για την λειτουργία του καταλόγου.



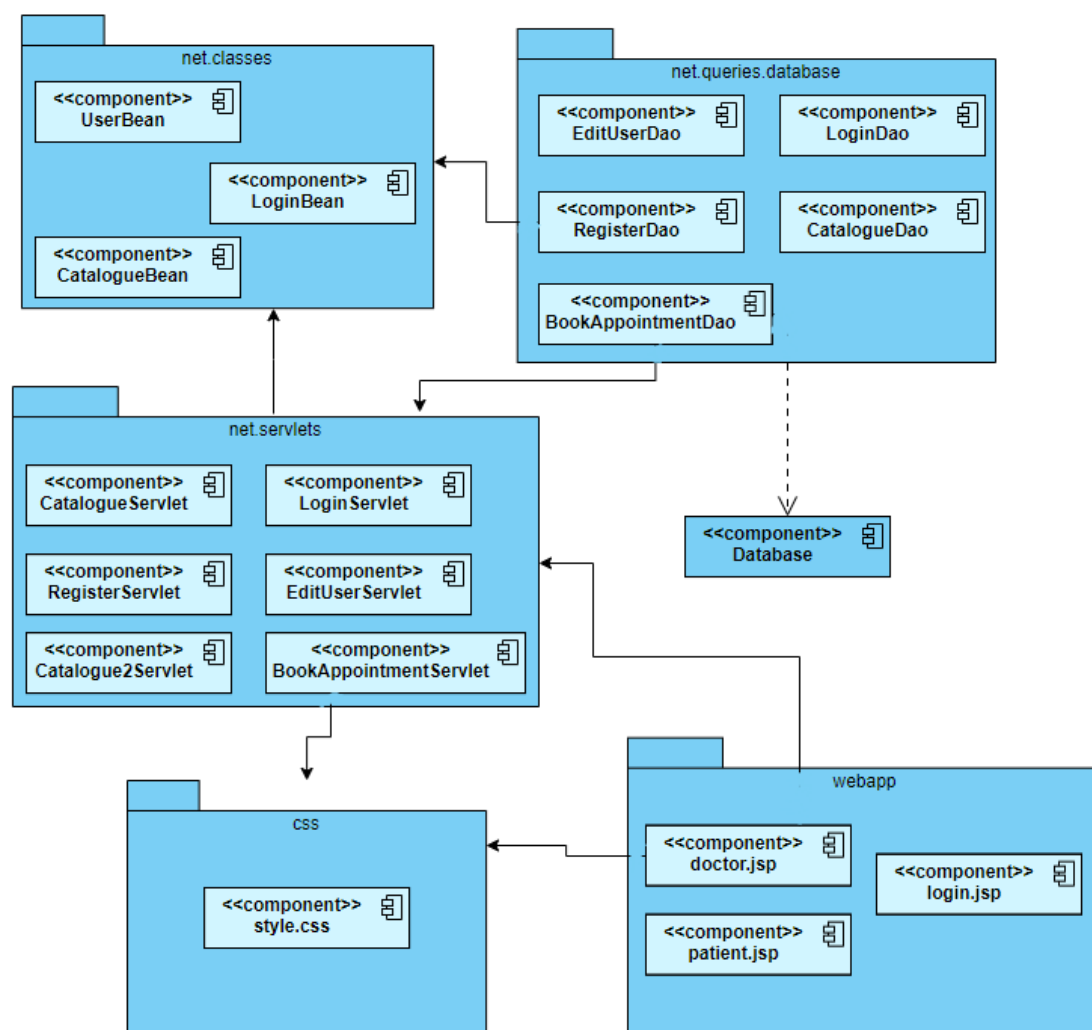
#### 4.1.8 Διάγραμμα Εξαρτημάτων –Component Diagram (1<sup>η</sup> έκδοση)

Τα εξαρτήματα αναπαριστούν όλα τα είδη των στοιχείων που αναφέρονται στην τμηματοποίηση των εφαρμογών λογισμικού. Μεταξύ άλλων, αυτά μπορεί να είναι απλά αρχεία ή βιβλιοθήκες που «διαβάζονται» και χρησιμοποιούνται δυναμικά.

Εξ' ορισμού, κάθε τάξη στο λογικό μοντέλο έχει έναν προσδιορισμό και ένα κυρίως σώμα. Ο προσδιορισμός περιέχει το σύστημα διεπαφής της τάξης, η οποία αναπαρίσταται με το συμβολισμό της UML με ένα μικρό κύκλο που συνδέεται μέσω μιας γραμμής με το σώμα της τάξης. Το σώμα περιέχει την υλοποίηση της ίδιας της τάξης και αναπαρίσταται με το ίδιο το σύμβολο του εξαρτήματος.

Οι σχέσεις εξάρτησης χρησιμοποιούνται στα διαγράμματα εξαρτημάτων για να δηλώσουν ότι ένα εξάρτημα αναφέρεται σε υπηρεσίες που προσφέρονται από άλλα εξαρτήματα. Αυτός ο τύπος εξάρτησης ανακλά επιλογές υλοποίησης. Μία σχέση εξάρτησης αναπαρίσταται με ένα βέλος με διακεκομμένη γραμμή σχεδιασμένο από τον πελάτη προς τον προμηθευτή.

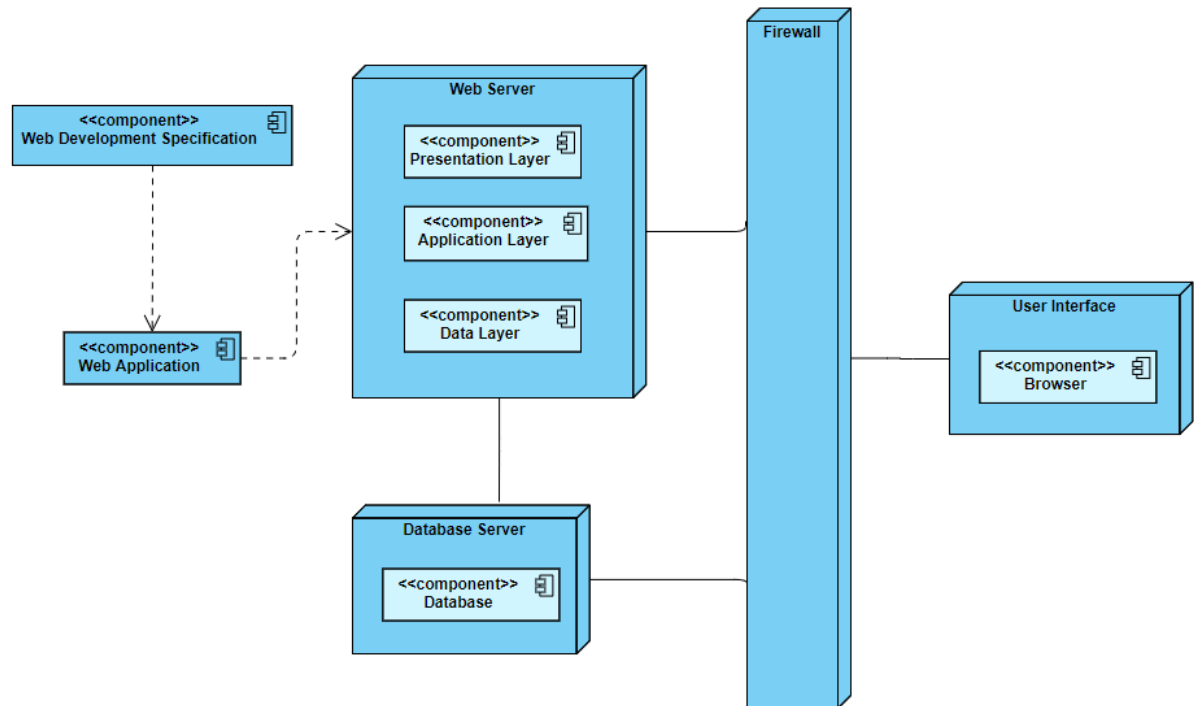
Παρακάτω φαίνεται η πρώτη έκδοση του διαγράμματος εξαρτημάτων.



#### 4.1.9 Διάγραμμα Διανομής- Deployment Diagram (1<sup>η</sup> έκδοση)

Τα διαγράμματα διανομής δείχνουν τη φυσική διάταξη των διάφορων εξαρτημάτων (κόμβων) υλικού, που αποτελούν ένα σύστημα, όπως και τη διανομή των εκτελέσιμων προγραμμάτων σε αυτό το υλικό.

Κάθε πόρος υλικού αναπαρίσταται με ένα κύβο. Κάθε σύστημα μπορεί να περιγραφεί με ένα μικρό αριθμό διαγραμμάτων διανομής. Πολλές φορές, ένα μόνο διάγραμμα είναι αρκετό.



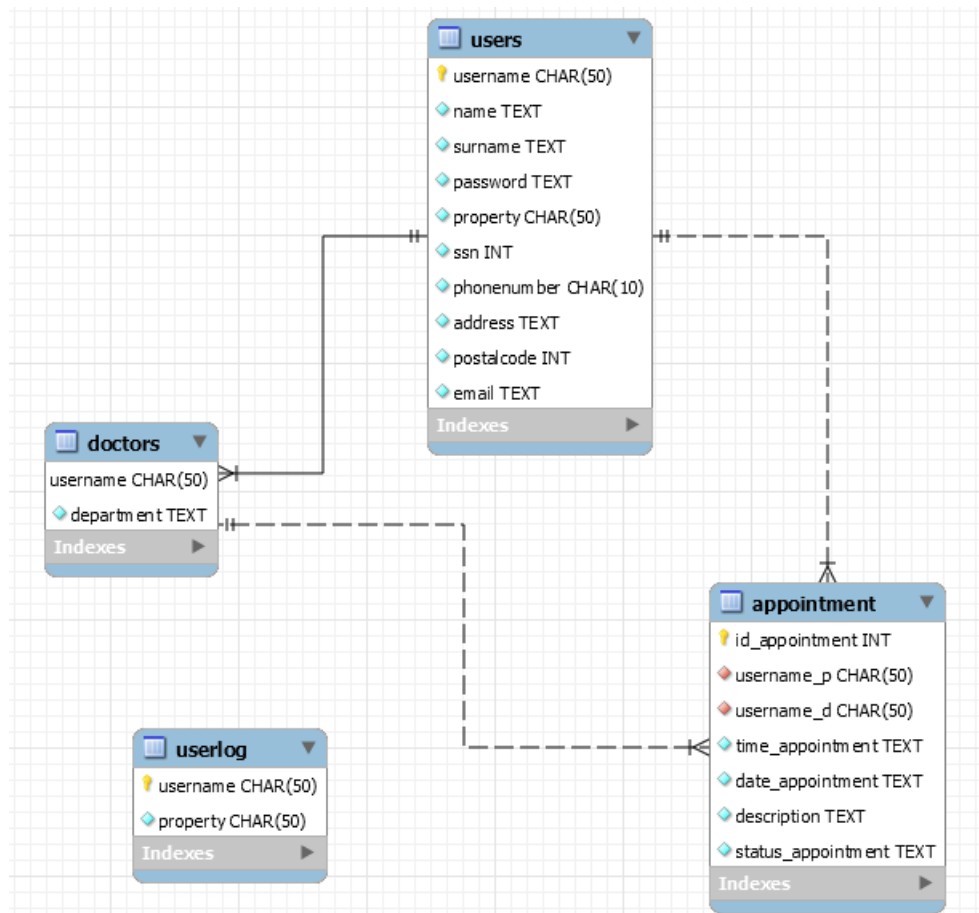
## 4.2 Υλοποίηση- Έλεγχος

### 4.2.1 Υλοποίηση: 1<sup>η</sup> εκτελέσιμη έκδοση

Με γνώμονα τα προαναφερθέντα διαγράμματα, πραγματοποιήθηκε η πρώτη εκτελέσιμη έκδοση της εφαρμογής, λαμβάνοντας υπόψιν και την ανάλυση απαιτήσεων.

Για την εύρυθμη λειτουργία της εφαρμογής, δημιουργήθηκε Βάση Δεδομένων MySQL με όνομα `db_medical_appointments`.

Παρακάτω φαίνεται το διάγραμμα σχέσεων της Βάσης (Relationship Diagram) το οποίο απεικονίζει τις σχέσεις μεταξύ των πινάκων. Δίνει πληροφορίες σχετικά με τα πρωτεύοντα κλειδιά, τα ξένα κλειδιά και τα πεδία του κάθε πίνακα.



Η Βάση Δεδομένων αποτελείται από τέσσερις πίνακες:

- Users: ο πίνακας αυτός περιέχει τις πληροφορίες για όλους τους χρήστες της εφαρμογής. Αποθηκεύει το username, το ονοματεπώνυμο (name, surname), το κωδικό (password), την ιδιότητα (property) ενός χρήστη, την διεύθυνσή του (address), τον ταχυδρομικό κώδικα (postalcode), το κινητό τηλέφωνο (phonenumber) και το email. Η ιδιότητα παίρνει τιμές Patient ή Doctor.
- Doctors: ο πίνακας αυτός περιέχει τις περαιτέρω πληροφορίες που αφορούν έναν χρήστη με την ιδιότητα του ιατρού (Doctor). Αποθηκεύει το username του ιατρού και την ειδικότητά του στον χώρο της ιατρικής (department).
- Appointment: σε αυτό τον πίνακα αποθηκεύονται όλες οι πληροφορίες για ένα προγραμματισμένο ραντεβού. Αυτές οι πληροφορίες περιλαμβάνουν την αποθήκευση του username του εκάστοτε ιατρού (username\_d) και ασθενή (username\_p), την ώρα (time\_appointment) και μέρα (date\_appointment) του ραντεβού, τον λόγο επίσκεψης (description) και την κατάσταση (status) του ραντεβού.
- Userlog: ο συγκεκριμένος πίνακας χρησιμοποιείται ως βοηθητικός ώστε να αποθηκεύονται το username και η ιδιότητα (property) ενός χρήστη κατά την εισαγωγή του στην εφαρμογή.

Στο πρόγραμμα προστέθηκε ένα πακέτο με όνομα <net.classes> το οποίο περιλαμβάνει όλες τις βασικές κλάσεις που δημιουργήθηκαν στο πρώτο μέρος της εργασίας. Επίσης,

δημιουργήθηκε και ένα πακέτο με όνομα <net.servlets> το οποίο περιλαμβάνει όλα τα Servlets. Σε αυτό το σημείο, αξίζει να σημειωθεί ότι ακολουθήθηκε η αρχιτεκτονική 3-tier για την υλοποίηση της εργασίας. Γι' αυτό το λόγο, όλα τα αρχεία Dao που χρειάστηκαν για την σύνδεση της Βάσης Δεδομένων με τον Application Server ομαδοποιήθηκαν σε ένα πακέτο με όνομα <net.queries.database>.

Η Αρχιτεκτονική 3-tier βασίζεται στο μοντέλο επικοινωνίας Πελάτη-Εξυπηρετητή και χαρακτηρίζει την οργάνωση του λογισμικού.

Υπάρχουν τρία λογικά επίπεδα στην εφαρμογή:

- Επίπεδο 1 (presentation): Το επίπεδο 1 ασχολείται με την παρουσίαση. Είναι το σημείο αλληλεπίδρασης της εφαρμογής με τους χρήστες. Η κύρια λειτουργία του είναι η μετάφραση των λειτουργιών και των αποτελεσμάτων της εφαρμογής σε κάτι που αναγνωρίζει ο χρήστης. Οι χρήστες της εφαρμογής χρησιμοποιούν το φυλλομετρητή για να αποκτήσουν πρόσβαση στην εφαρμογή. Ο φυλλομετρητής είναι υπεύθυνος για την μετατροπή του κώδικα HTML, ο οποίος δημιουργείται από τον Εξυπηρετητή Ιστού, σε ένα εύχρηστο περιβάλλον αλληλεπίδρασης με την εφαρμογή.
- Επίπεδο 2 (business logic): Το μεσαίο επίπεδο υλοποιεί τις λειτουργίες της εφαρμογής. Συντονίζει τις διεργασίες, λαμβάνει αποφάσεις και εκτελεί υπολογισμούς. Επιπλέον, είναι ενδιάμεσος στην επικοινωνία μεταξύ των επιπέδων 1 και 3. Εδώ βρίσκεται ο Εξυπηρετητής της Εφαρμογής (Application Server). Παρέχει την δυνατότητα δημιουργίας δυναμικών ιστοσελίδων, μέσω των οποίων ο χρήστης χρησιμοποιεί την εφαρμογή. Αναλαμβάνει την διεκπεραίωση των ενεργειών που εκτελεί ο χρήστης. Αναλαμβάνει την επικοινωνία με την βάση δεδομένων για την αποθήκευση και την ανάκτηση των δεδομένων.
- Επίπεδο 3 (data): Το επίπεδο 3 διαχειρίζεται τα δεδομένα της εφαρμογής. Η πληροφορία αποθηκεύεται και ανακτάται μέσω μιας Βάσης Δεδομένων. Η πληροφορία από την Βάση Δεδομένων επιστρέφεται στο επίπεδο 2, επεξεργάζεται και τελικά επιστρέφει στο επίπεδο 1.

Τα πλεονεκτήματα της αρχιτεκτονικής αυτής είναι:

- Επεκτασιμότητα (Scalability): Διαχωρίζοντας σε επίπεδα την εφαρμογή επιτυγχάνεται καλύτερος έλεγχος του φόρτου εργασίας. (Προσθήκη μηχανημάτων για την εξυπηρέτηση περισσότερων πελατών, προσθήκη περισσότερων εξυπηρετητών δεδομένων).
- Μεταφερισιμότητα (Portability): Καθώς το επίπεδο 1 δεν έχει μεγάλες απαιτήσεις σε υπολογιστικούς πόρους (ο κύριος όγκος της επεξεργασίας γίνεται στα επίπεδα 2,3) η εφαρμογή είναι προσβάσιμη από ένα μεγάλο εύρος υπολογιστικών συστημάτων που χρησιμοποιούν οι χρήστες (laptop, κινητά τηλέφωνα).
- Επαναχρησιμοποίηση Κώδικα.
- Καταμερισμός προγραμματιστικών εργασιών.

Τεχνολογίες που χρησιμοποιούνται για την ανάπτυξη 3-tier εφαρμογών στο Διαδίκτυο είναι:

- Application Server: Apache Tomcat, παρέχει την δυνατότητα δημιουργίας δυναμικών ιστοσελίδων μέσω Servlets.
- Βάση Δεδομένων MySQL.
- Προγραμματιστικό περιβάλλον ανάπτυξης εφαρμογών 3-tier: Eclipse, Java EE Developer edition.

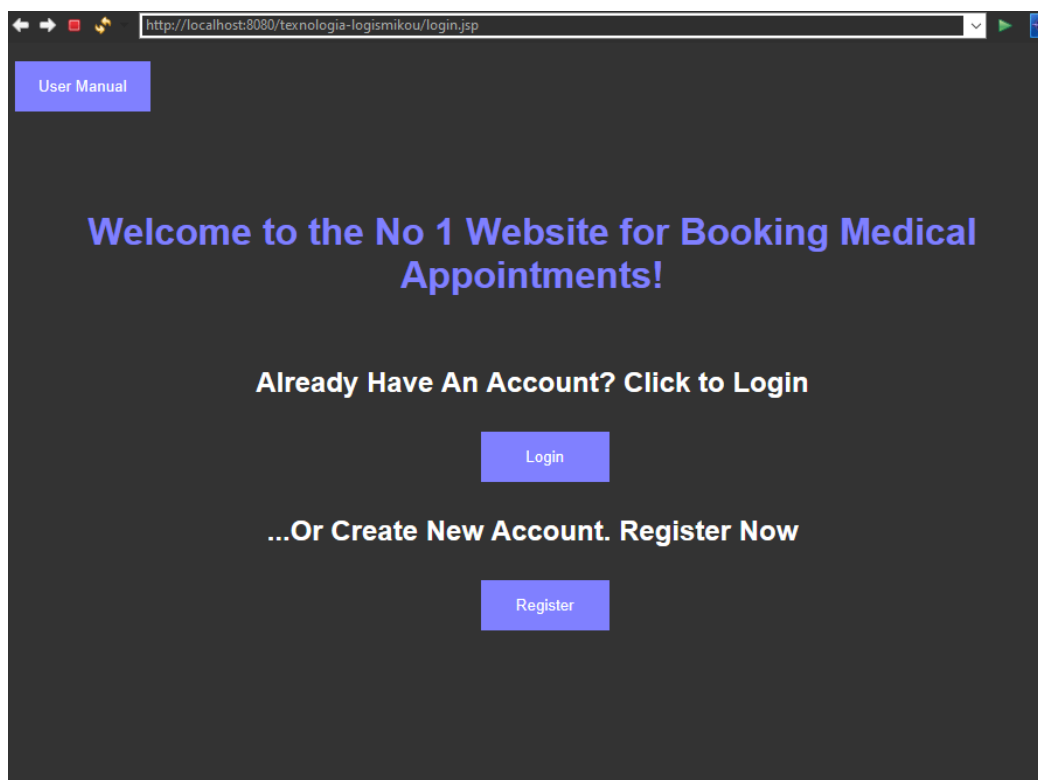
Η υλοποίηση της διαδικτυακής διεπαφής έγινε και με την δημιουργία .jsp σελίδων. Οι σελίδες αυτές βρίσκονται μέσα στον φάκελο με όνομα <Web Content> και είναι οι παρακάτω:

- Login.jsp: η αρχική σελίδα της εφαρμογής όπου γίνεται η είσοδος ή η εγγραφή ενός χρήστη.
- Doctor.jsp: η σελίδα που μεταφέρονται, ύστερα της σύνδεσης ή εγγραφής, όλοι οι χρήστες με ιδιότητα ιατρού και μπορούν να διαβάσουν τις πληροφορίες σχετικά με το προφίλ τους, να το παραμετροποιήσουν ή να το διαγράψουν.
- Patient.jsp: η σελίδα που μεταφέρονται, ύστερα της σύνδεσης ή εγγραφής, όλοι οι χρήστες με ιδιότητα ασθενούς και μπορούν να διαβάσουν τις πληροφορίες σχετικά με το προφίλ τους, να το παραμετροποιήσουν ή να το διαγράψουν.

Επίσης, υπάρχει και ένα style.css αρχείο για την εμφάνιση του user interface της εφαρμογής.

Κατά την πρώτη εκτελέσιμη μορφή της εφαρμογής, υλοποιήθηκαν οι βασικότερες λειτουργίες που αναφέρθηκαν και αναλύθηκαν στα κεφάλαια 3 και 4. Αυτές οι λειτουργίες συμπεριλαμβάνουν:

- Την σύνδεση και εγγραφή ενός χρήστη.



- Την δυνατότητα παραμετροποίησης των προσωπικών στοιχείων ενός χρήστη , όπου αυτό είναι επιτρεπτό.

The screenshot displays a web browser window at the URL `http://localhost:8080/tecnologia-logismikou/patient.jsp`. The page features a dark sidebar with navigation links: Profile, Book Appointment, Catalogue, and Log Out. The main content area shows a user profile editing form. The form is titled "Please fill only the fields you want to change" in blue text. It includes several input fields: Password (masked with dots), Phone Number (containing 6676765), Address (containing fgdfdbgfbd), Postal Code (containing 564645), and Email (containing ggfg). Below these fields are three buttons: "Save Changes" (grey), "Cancel" (red), and "Forgot password?" (blue link). The background of the page is dark, and the sidebar is visible on the left.

- Την αναζήτηση ιατρών ανά περιοχή ή ειδικότητα, με την εισαγωγή ταχυδρομικού κώδικα ή ειδικότητας αντίστοιχα, που έχει συμπληρώσει ο χρήστης- ασθενής.

The screenshot displays a web browser window at the URL `http://localhost:8080/tecnologia-logismikou/BookAppointmentServlet`. The page features a dark sidebar with navigation links: Profile, Book Appointment, Catalogue, and Log Out. The main content area shows a search form titled "Search for a Doctor:" in blue text. Below the title is a text input field with the placeholder text "Enter Postal Code or Department" and a blue "Search" button. The background of the page is dark, and the sidebar is visible on the left.

- Τον προγραμματισμό ενός ραντεβού με την εισαγωγή ώρας, μέρας και λόγου επίσκεψης στον ιατρό.

The screenshot shows a web application interface with a dark theme. At the top, there is a navigation bar with five buttons: 'Profile', 'Book Appointment' (highlighted in blue), 'Catalogue', 'Online Help', and 'Log Out'. Below the navigation bar, the form is titled 'Doctor Username' in blue. It contains a text input field with the placeholder 'Enter Doctor Username'. Below this is a section titled 'Time of Appointment' with a text input field 'Enter Time'. The next section is 'Date of Appointment' with a text input field 'Enter Date'. The final section is 'Description' with a text input field 'Enter Description'. At the bottom of the form, there are two buttons: a blue 'Submit Appointment' button and a red 'Cancel' button.

- Την επιλογή ο χρήστης να διαγράψει το προσωπικό του προφίλ.

The screenshot shows a web browser window with the URL 'http://localhost:8080/tecnologia-logismikou/patient.jsp'. The browser displays the 'Book Appointment' form from the previous screenshot. Overlaid on the form is a white confirmation dialog box with a blue title 'Are you sure you want to delete your profile?'. The dialog contains the text 'Your profile is going to be deleted permanently.' and two buttons: 'Yes, I am sure' and 'Cancel'.

- Την δυνατότητα αποσύνδεσης του χρήστη από την εφαρμογή.

The screenshot shows a close-up of the navigation bar from the web application. It contains five buttons: 'Profile', 'Book Appointment', 'Catalogue', 'Log Out', and a partially visible button on the right. The 'Log Out' button is highlighted in blue.



- Την λειτουργία του καταλόγου, δηλαδή την σελίδα στην οποία ένας χρήστης μπορεί να δει αναλυτικά όλα τα ραντεβού που έχει προγραμματίσει και να τα ακυρώσει αν επιθυμεί.

Catalogue					
All your booked appointments in one place!					
Patient	Doctor	Time of Appointment	Date of Appointment	Description	Status
oprah	toto	18:00	1/6/21	eye checkup	active
oprah	toto	7:30	30/5/21	glasses	completed
oprah	toto	17:50	7/4/21	annual checkup	cancelled

#### 4.2.2 Αναφορά ελέγχου για την 1<sup>η</sup> εκτελέσιμη έκδοση

Παρόλο που η πρώτη εκτελέσιμη έκδοση περιέχει όλα τα βασικά στοιχεία της εφαρμογής, χρειάζεται βελτίωση και προσθήκη επιπρόσθετων λειτουργιών.

Οι λειτουργίες που οφείλουν να υλοποιηθούν στην τελική εκτελέσιμη έκδοση είναι:

- Σε κάθε σελίδα της εφαρμογής πρέπει να προστεθεί το online help με οδηγίες για την σωστή περιήγηση στην εφαρμογή.
- Το User Manual πρέπει να υλοποιηθεί και να μπορεί να μεταβαίνει σε αυτό ο χρήστης με το πάτημα ενός κουμπιού.
- Ο χρήστης κατά την εγγραφή του στην εφαρμογή θα πρέπει να συμφωνεί με τους όρους χρήσης και την πολιτική απορρήτου.
- Κατά την μη εύρεση αποτελεσμάτων αναζήτησης, η εφαρμογή πρέπει να ενημερώνει με κατάλληλο μήνυμα τον χρήστη.

Για την αποτελεσματικότερη χρήση της εφαρμογής από τους χρήστες της, οφείλει να ανανεωθεί και το User Interface με περισσότερη συνέπεια (Consistency).

Διορθώσεις πρέπει να γίνουν και στην δομή της εφαρμογής. Τέτοιες αλλαγές συμπεριλαμβάνουν την συγχώνευση των Servlets.

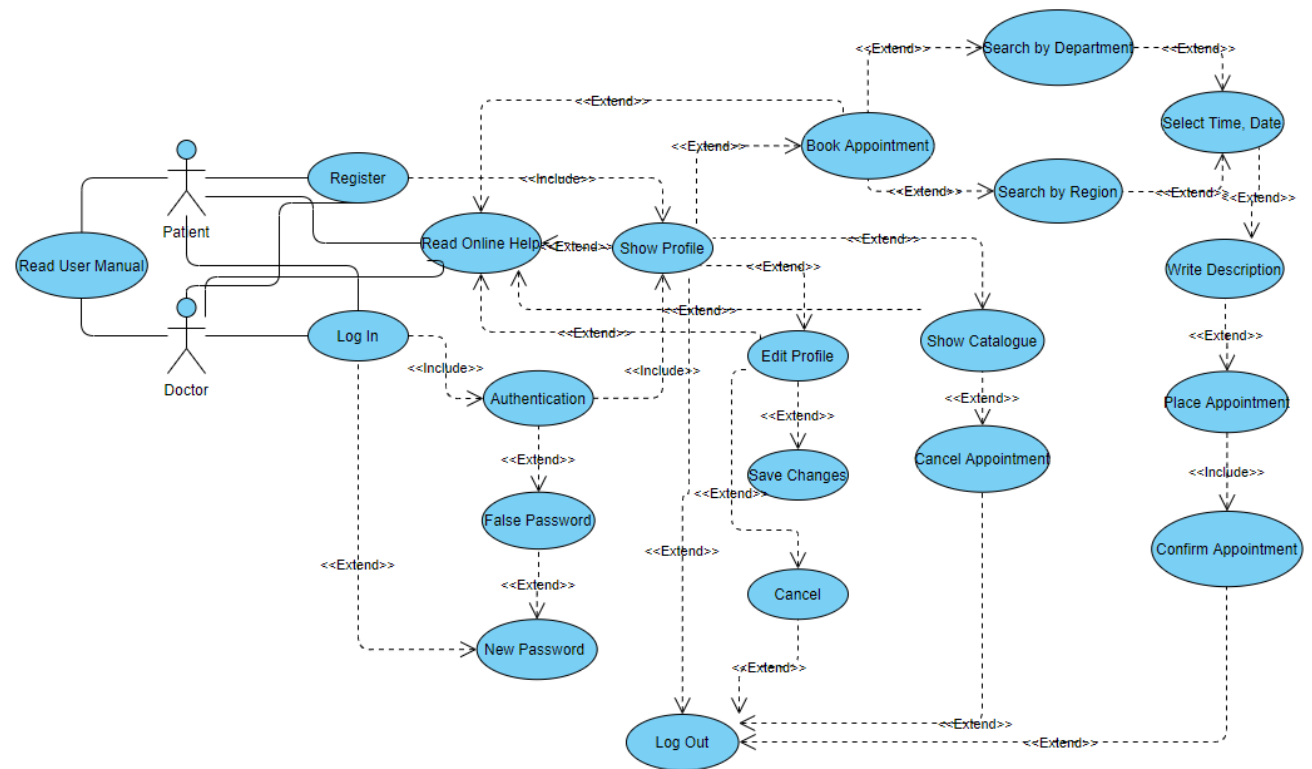
## 5. Φάση: Κατασκευή (Construction)

### 5.1 Ανάλυση-Σχεδιασμός

Σε αυτή τη φάση της εφαρμογής, στόχος είναι η καλύτερη οργάνωση του κώδικα και η προσθήκη περαιτέρω ενεργειών που δεν κατάφεραν να υλοποιηθούν στην πρώτη εκτελέσιμη έκδοση.

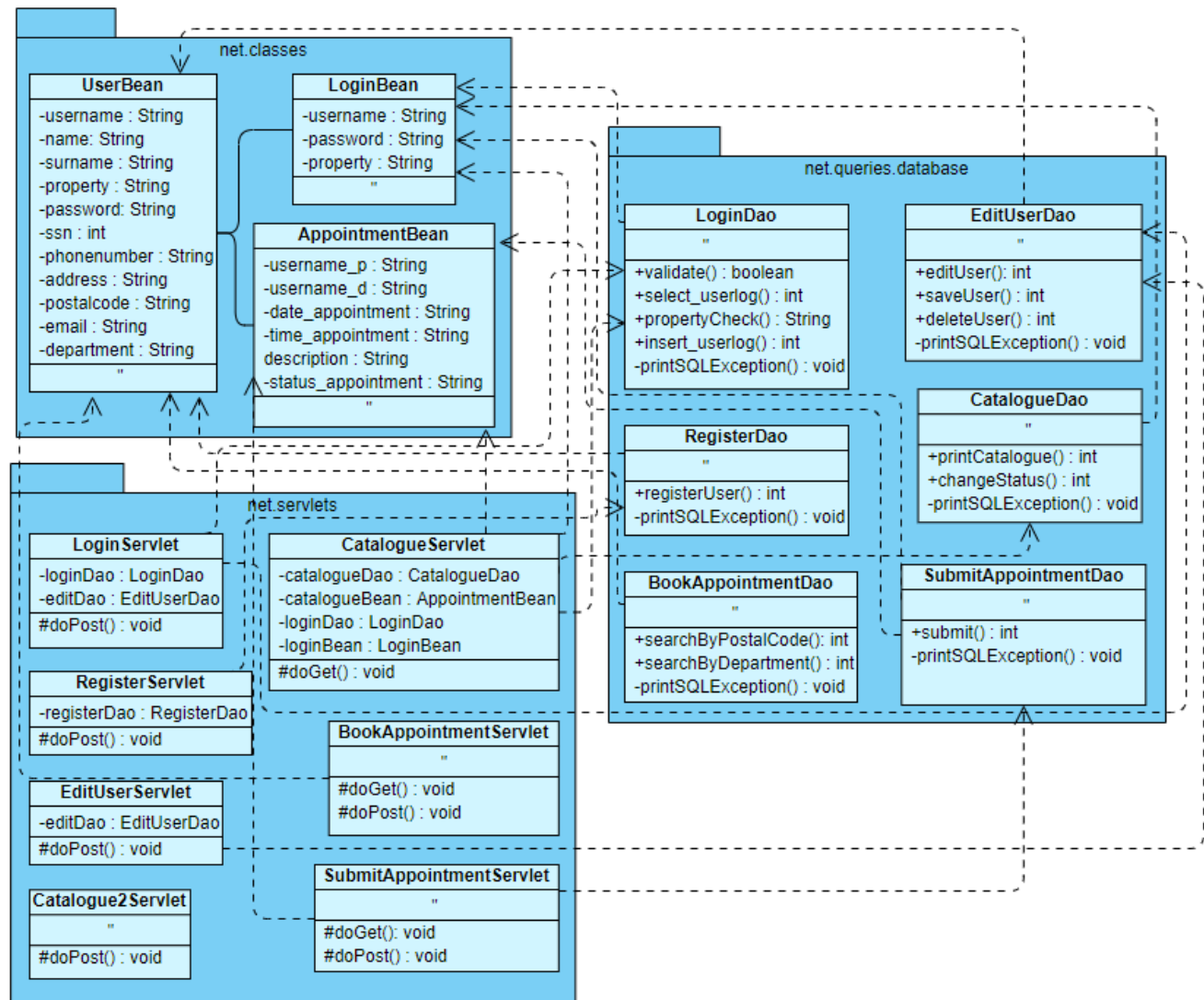
### 5.1.1 Διάγραμμα Περιπτώσεων Χρήσης- Use Case Diagram (3<sup>η</sup> έκδοση)

Παρακάτω φαίνεται η 3<sup>η</sup> και τελική έκδοση του διαγράμματος περιπτώσεων χρήσης. Περιέχει όλες τις πιθανές ενέργειες που μπορούν να τελέσουν οι χρήστες τις εφαρμογής.



### 5.1.2 Διάγραμμα Τάξεων – Class Diagram (3<sup>η</sup> έκδοση)

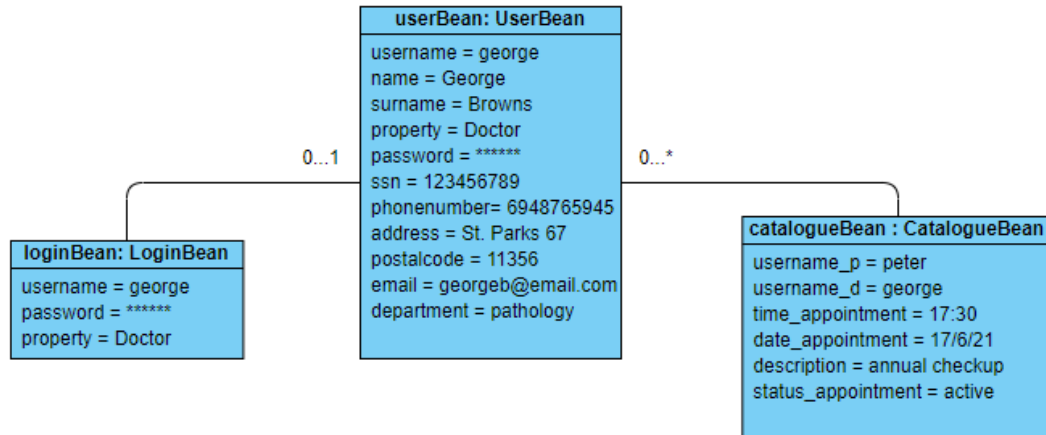
Στο τρίτο και τελικό διάγραμμα κλάσεων προστέθηκαν επιπλέον οι κλάσεις SubmitAppointmentServlet.java και η SubmitAppointmentDao.java. Επίσης, αλλαγή έγινε και στο όνομα της κλάσης CatalogueBean.java σε AppointmentBean.java διότι θεωρήθηκε ότι αυτή η ονομασία αντικατοπτρίζει καλύτερα το ρόλο της μέσα στο πρόγραμμα.



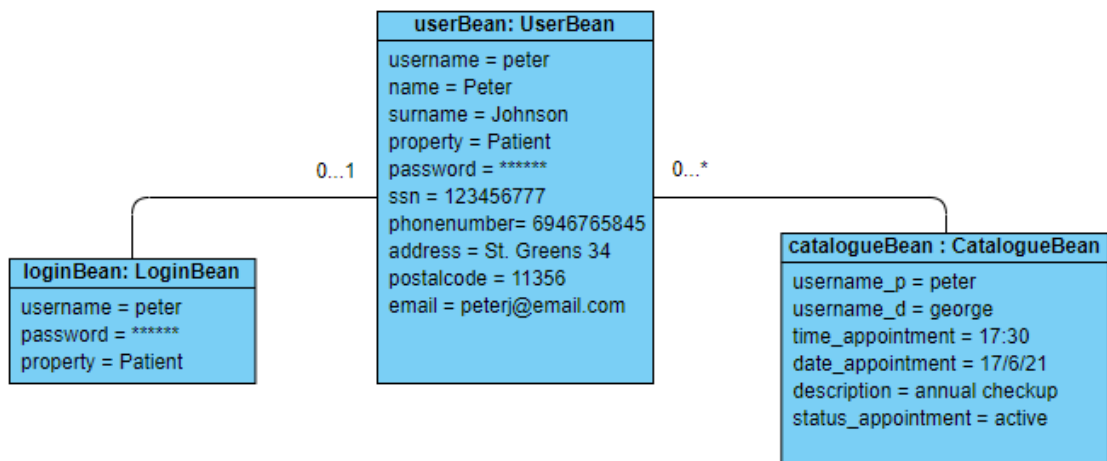
### 5.1.3 Διάγραμμα Αντικειμένων – Object Diagram (2<sup>η</sup> έκδοση)

Παρακάτω φαίνεται η δεύτερη και τελική έκδοση του διαγράμματος αντικειμένων για την περίπτωση ενός ιατρού και στη συνέχεια για την περίπτωση ενός ασθενή.

Ιατρός:

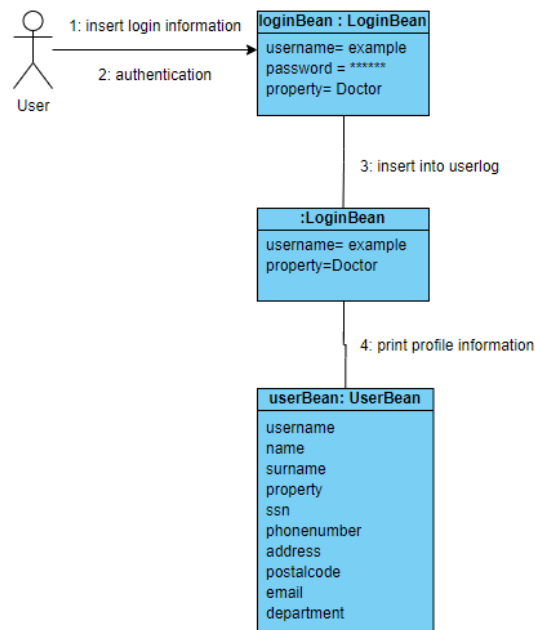


Ασθενής:

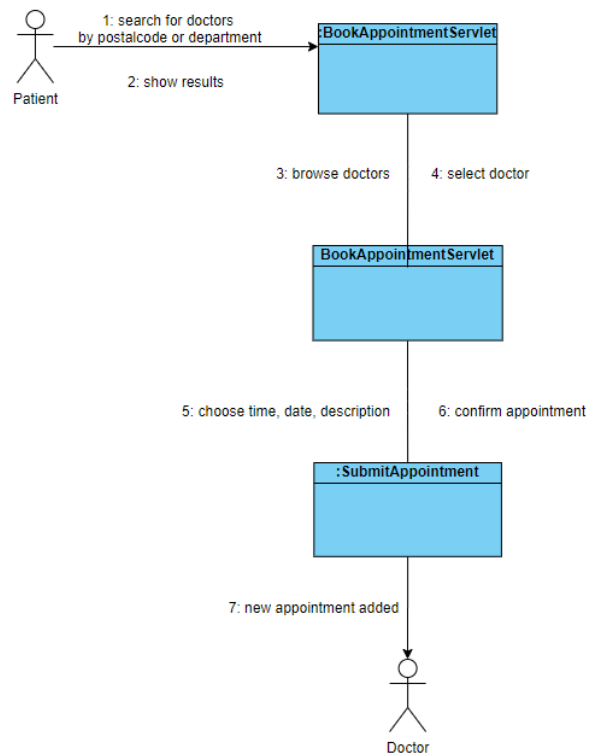


### 5.1.4 Διάγραμμα Συνεργασίας – Collaboration Diagram (2<sup>η</sup> έκδοση)

Παρακάτω φαίνεται η δεύτερη και τελική έκδοση του διαγράμματος συνεργασίας για την περίπτωση εισόδου στην εφαρμογή.

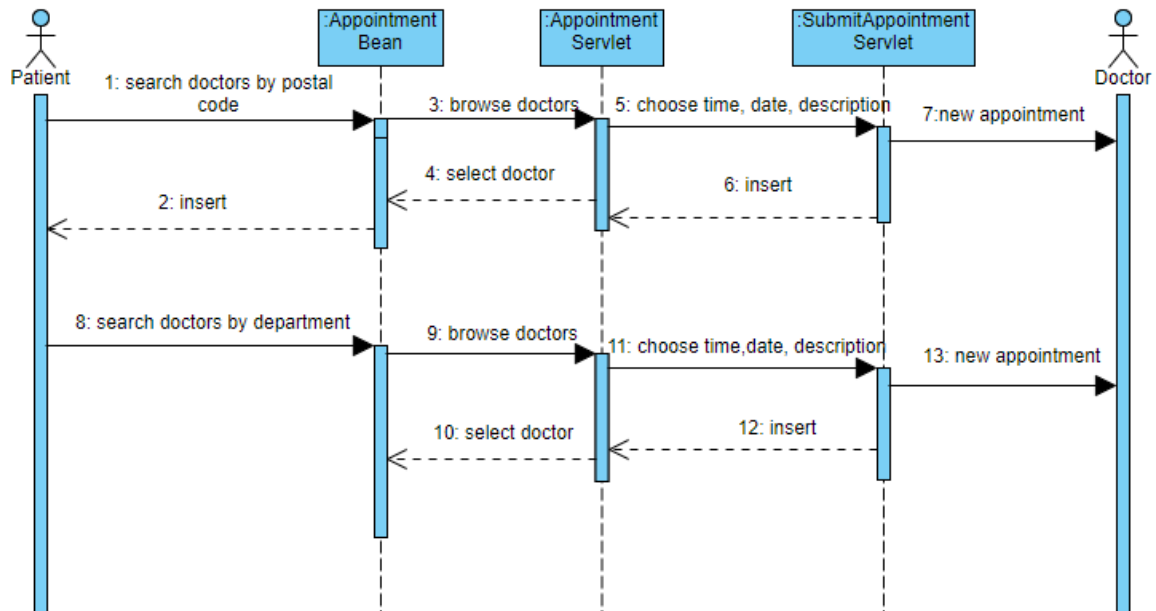


Επίσης, παρακάτω φαίνεται η δεύτερη και τελική έκδοση του διαγράμματος συνεργασίας για την περίπτωση του προγραμματισμού ενός ραντεβού.

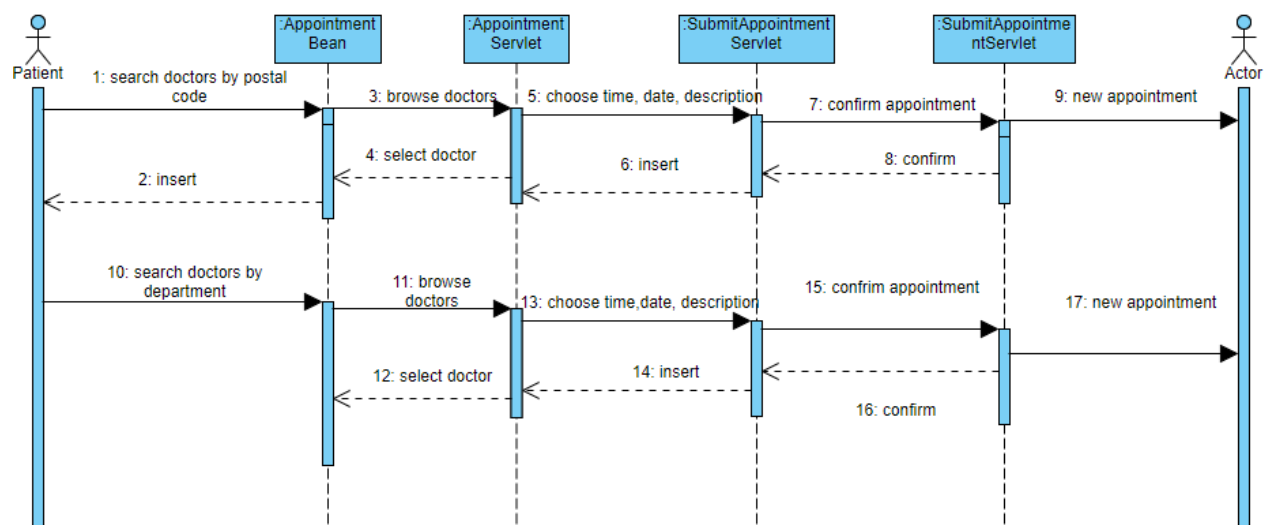


### 5.1.5 Διάγραμμα Σειράς – Sequence Diagram (2<sup>η</sup> έκδοση)

Παρακάτω φαίνεται η δεύτερη και τελική έκδοση του διαγράμματος σειράς για την αλληλεπίδραση μεταξύ των αντικειμένων, όταν ο χρήστης-ασθενής θέλει να κάνει Log In στην εφαρμογή.

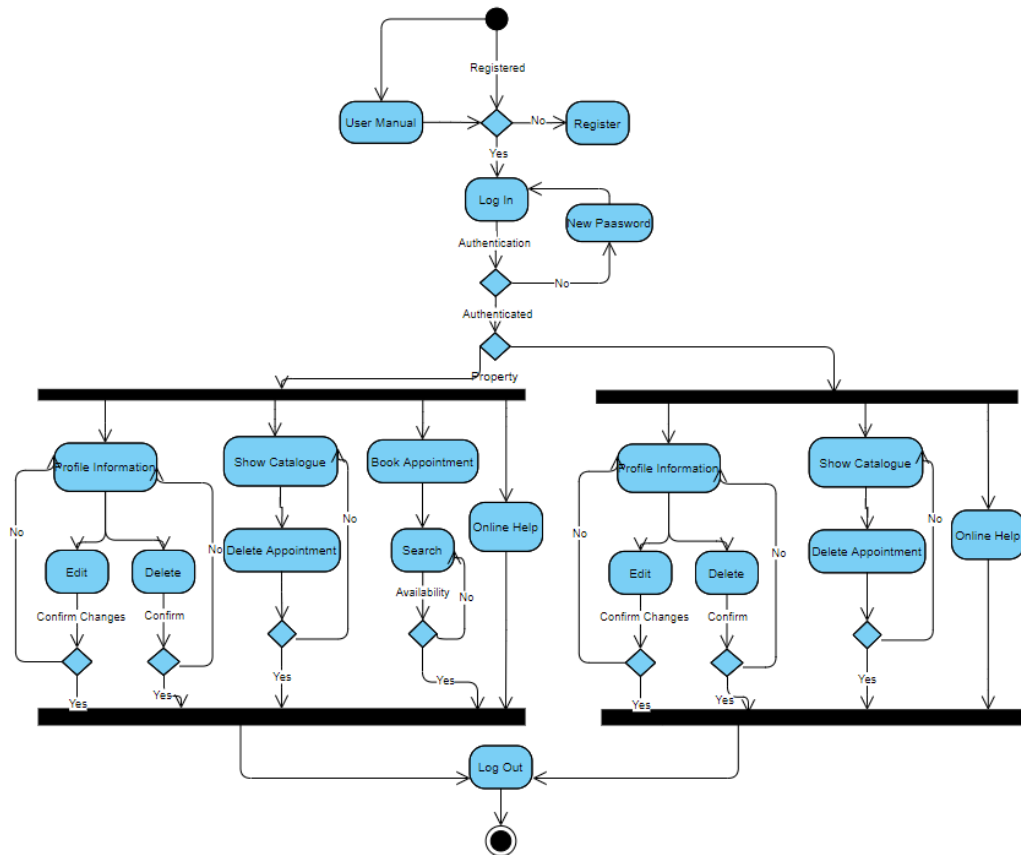


Παρακάτω φαίνεται η δεύτερη και τελική έκδοση του διαγράμματος σειράς για την αλληλεπίδραση των αντικειμένων, όταν ο χρήστης-ασθενής θέλει να προγραμματίσει ένα ραντεβού.



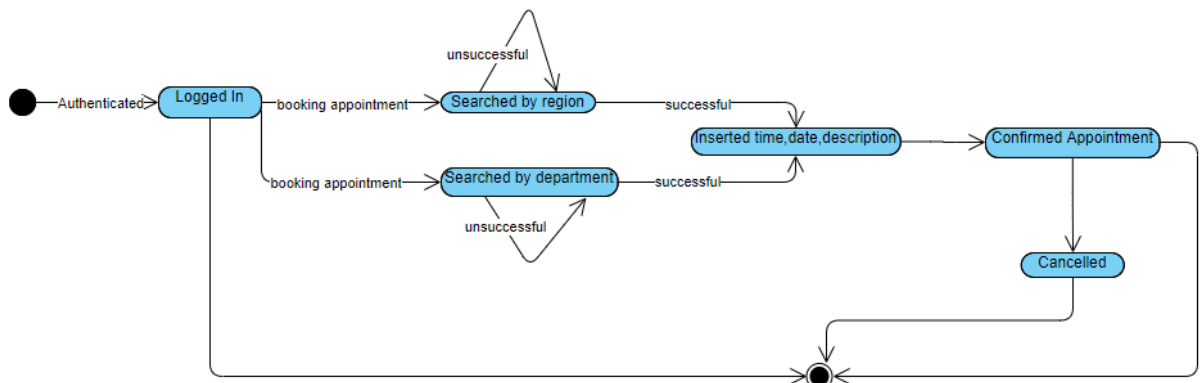
### 5.1.6 Διάγραμμα Δραστηριοτήτων – Activity Diagram (1<sup>η</sup> έκδοση)

Παρακάτω φαίνεται η 2<sup>η</sup> και τελική έκδοση του διαγράμματος δραστηριοτήτων.

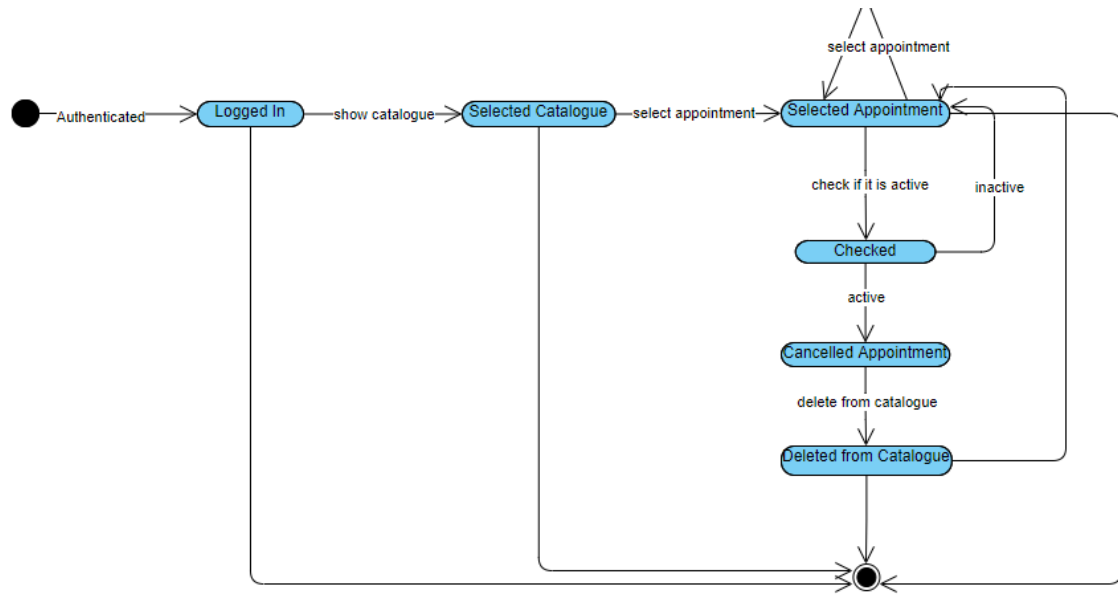


### 5.1.7 Διάγραμμα Καταστάσεων –State Chart Diagram (2<sup>η</sup> έκδοση)

Παρακάτω φαίνεται η δεύτερη και τελική έκδοση του διαγράμματος καταστάσεων για τον προγραμματισμό ενός ραντεβού.

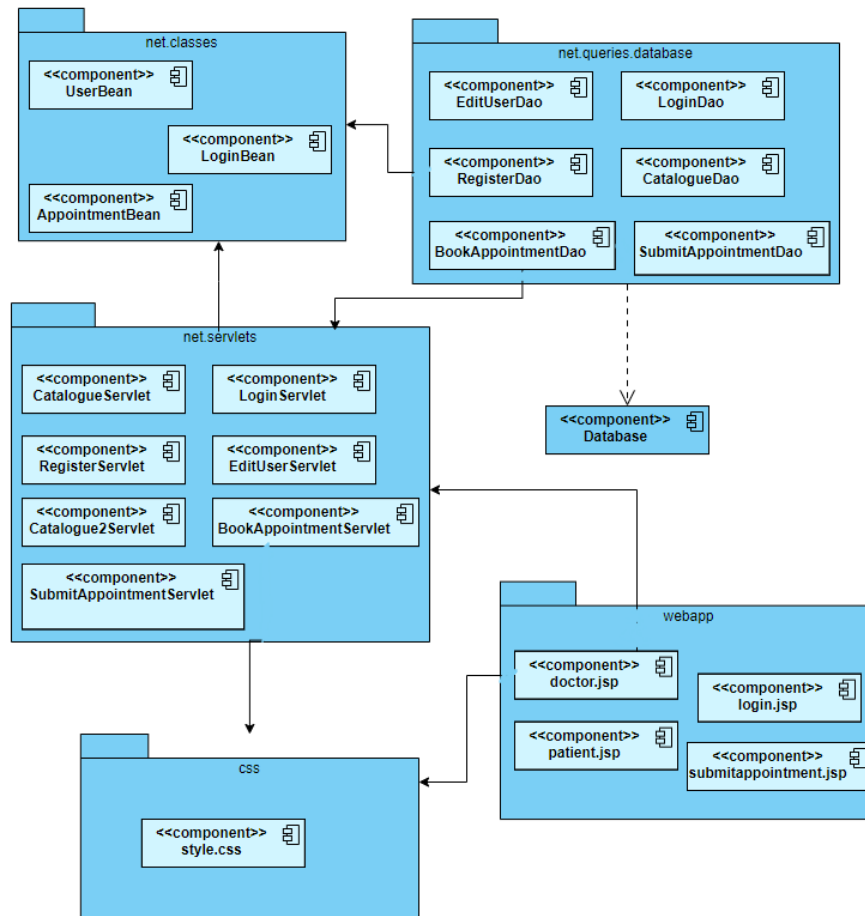


Παρακάτω φαίνεται η δεύτερη και τελική έκδοση του διαγράμματος καταστάσεων για την λειτουργία του καταλόγου.



### 5.1.8 Διάγραμμα Εξαρτημάτων –Component Diagram (2<sup>η</sup> έκδοση)

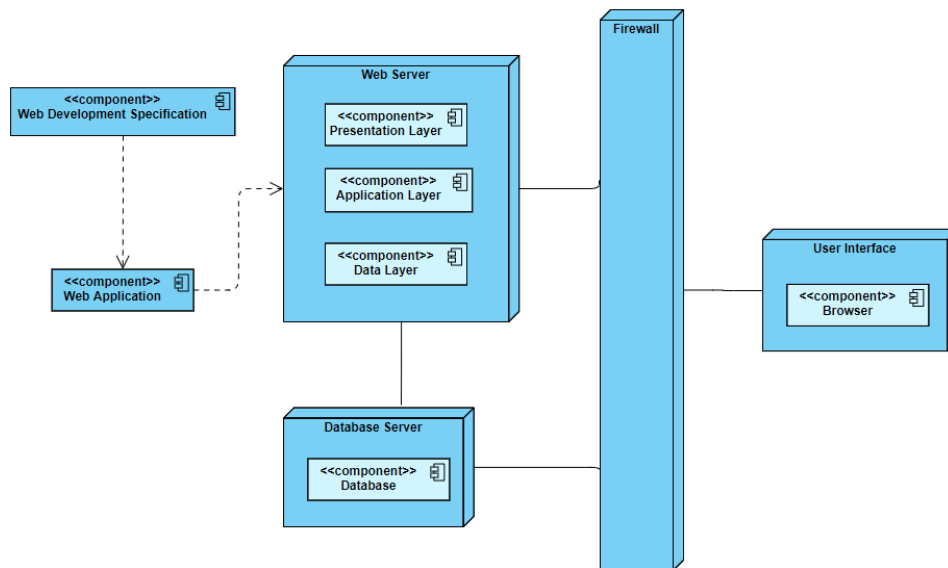
Παρακάτω φαίνεται η δεύτερη και τελική έκδοση του διαγράμματος εξαρτημάτων.





#### 4.1.9 Διάγραμμα Διανομής- Deployment Diagram (2η έκδοση)

Το διάγραμμα διανομής δεν έχει αλλάξει σε σχέση με την πρώτη έκδοση αφού τα τμήματα της φυσικής διάταξης του λογισμικού παραμένουν τα ίδια.



## 5.2 Υλοποίηση- Έλεγχος

### 5.2.1 Υλοποίηση: Τελική εκτελέσιμη έκδοση

Με γνώμονα τις τελικές εκδόσεις των διαγραμμάτων, πραγματοποιήθηκε η τελική εκτελέσιμη έκδοση της εφαρμογής, λαμβάνοντας υπόψιν και την πρώτη εκτελέσιμη έκδοση.

Αλλαγές στην Βάση Δεδομένων δεν πραγματοποιήθηκαν και έτσι η δομή της παρέμεινε ίδια με αυτήν που αναλύθηκε στην πρώτη εκτελέσιμη έκδοση.

Στην υλοποίηση της διαδικτυακής διεπαφής προστέθηκε μία καινούργια σελίδα .jsp με όνομα <<submitappointment.jsp>> η οποία παίζει καθοριστικό ρόλο για την ολοκλήρωση του προγραμματισμού ενός ραντεβού.

Κατά την τελική εκτελέσιμη μορφή της εφαρμογής, βελτιστοποιήθηκαν οι βασικότερες λειτουργίες που αναφέρθηκαν και αναλύθηκαν στα παραπάνω κεφάλαια. Αυτές οι λειτουργίες συμπεριλαμβάνουν:

- Την σύνδεση και εγγραφή ενός χρήστη.

This screenshot shows the main interface for logging in or registering. At the top, there are links for 'User Manual' and 'Online Help'. The main heading is 'Welcome to the No 1 Website for Booking Medical Appointments!'. Below this, there is a section for existing users: 'Already Have An Account? Click to Login' with a 'Login' button. Further down, there is a section for new users: '...Or Create New Account. Register Now' with a 'Register' button.

User Manual Online Help

## Welcome to the No 1 Website for Booking Medical Appointments!

Already Have An Account? Click to Login

Login

...Or Create New Account. Register Now

Register

This screenshot shows the login form. It has a 'User Manual' link and a closed 'Online Help' window. The form includes a 'Username' field with a placeholder 'Enter Username', a 'Password' field with a placeholder 'Enter Password', and a 'Login' button. There is also a 'Cancel' button and a link for 'Forgot password?'. A 'Register' button is visible at the bottom.

User Manual Online Help

## We

## nts!

Username

Enter Username

Password

Enter Password

Login

Cancel

Forgot password?

Register

This screenshot shows the registration form. It has a 'User Manual' link and a closed 'Online Help' window. The form includes fields for 'Username' (placeholder: 'Enter Username'), 'Name' (placeholder: 'Enter Name'), 'Surname' (placeholder: 'Enter Surname'), 'Password' (placeholder: 'Enter Password'), 'Property (Patient or Doctor)' (placeholder: 'Enter Property'), 'Department (for Doctors only)' (placeholder: 'Enter Department'), 'SSN' (placeholder: 'Enter SSN'), 'Phone Number' (placeholder: 'Enter Phone Number'), 'Address' (placeholder: 'Enter Address'), and 'Postal Code' (placeholder: 'Enter Postal Code'). There is a close button (X) in the top right corner of the form.

User Manual Online Help

Username

Enter Username

Name

Enter Name

Surname

Enter Surname

Password

Enter Password

Property (Patient or Doctor)

Enter Property

Department (for Doctors only)

Enter Department

SSN

Enter SSN

Phone Number

Enter Phone Number

Address

Enter Address

Postal Code

Enter Postal Code

- Την δυνατότητα παραμετροποίησης των προσωπικών στοιχείων ενός χρήστη , όπου αυτό είναι επιτρεπτό.

The screenshot shows a web application interface with a dark header containing navigation links: Profile, Catalogue, Online Help, and Log Out. Below the header, a welcome message reads "Welcome charlesleclerc,". The main section is titled "Profile Information:" and lists the following details: Username: charlesleclerc, Name: Charles, Surname: Leclerc, Password: 1234, Property: Doctor, Department: Gynecology, SSN: 56793480, Phone Number: 683546789, Address: Herilongs Beach 567, Postal Code: 11356, and Email: charles@email.com. At the bottom of the profile information, there are two buttons: "Edit Profile" and "Delete Profile".

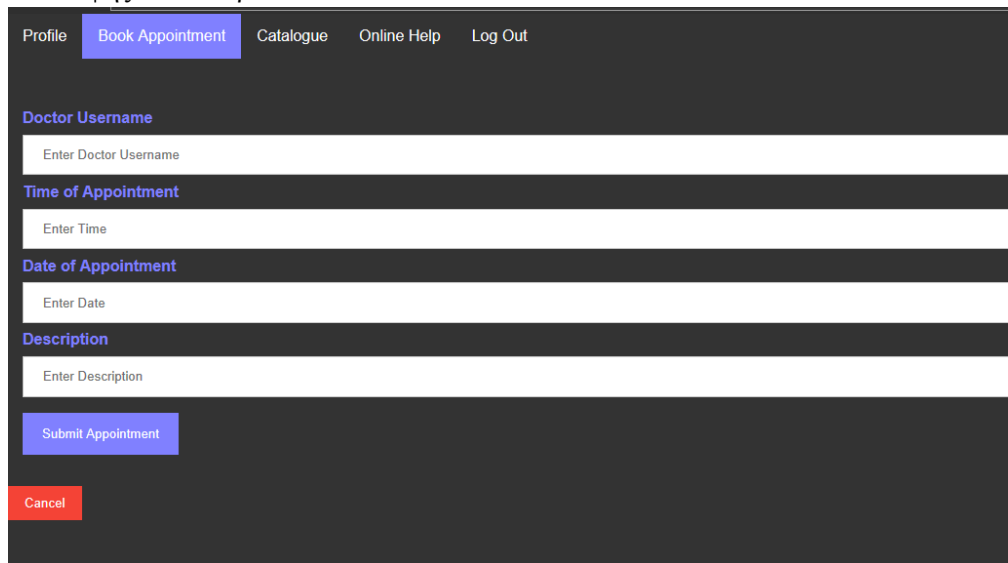
This screenshot displays the "Edit Profile" form. A modal window is open with the title "Please fill only the fields you want to change". The form contains input fields for Password, Phone Number, Address, Postal Code, and Email. The current values are: Password (masked with \*\*\*\*), Phone Number (683546789), Address (Herilongs Beach 567), Postal Code (11356), and Email (charles@email.com). Below the input fields are two buttons: "Save Changes" and "Cancel". The background shows the same profile information as the previous screenshot.

- Την αναζήτηση ιατρών ανά περιοχή ή ειδικότητα, με την εισαγωγή ταχυδρομικού κώδικα ή ειδικότητας αντίστοιχα, που έχει συμπληρώσει ο χρήστης- ασθενής.

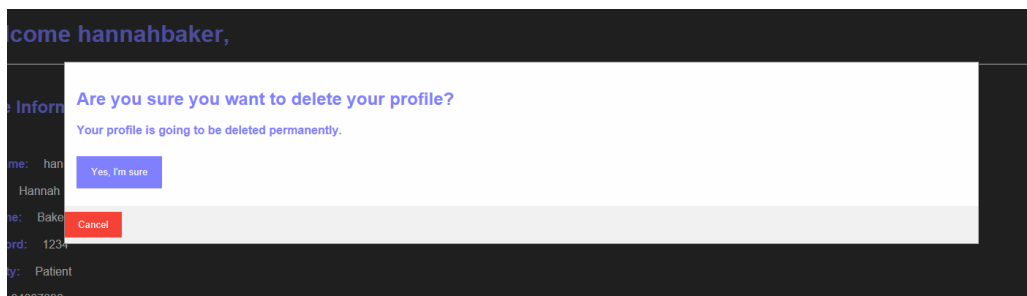
The screenshot shows the "Search for a Doctor:" page. The header includes navigation links: Profile, Book Appointment, Catalogue, Online Help, and Log Out. Below the header, there is a search bar with the placeholder text "Enter Postal Code or Department" and a "Search" button. Below the search bar, there is a table listing doctors. At the bottom left, there is a "Book Appointment" button.

Username	Name	Surname	Phone Number	Address	Postal Code	Email	Department
charlesleclerc	Charles	Leclerc	683546789	Herilongs Beach 567	11356	charles@email.com	Gynecology
jimmynewtron	Jimmy	Newtron	690457896	Rodeo Drive 456	11356	newtron@email.com	Gynecology
lewishamilton	Lewis	Hamilton	698367856	St. Jekings 130	11356	lewis@email.com	Dermatology
totoWolf	Toto	Wolf	6904839677	St. Kertings 34	11356	toto@email.com	Dermatology

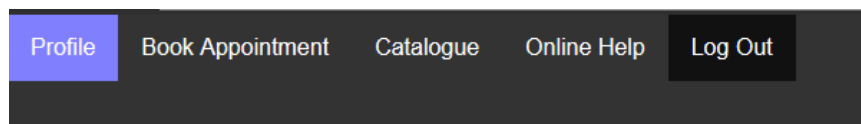
- Τον προγραμματισμό ενός ραντεβού με την εισαγωγή ώρας, μέρας και λόγου επίσκεψης στον ιατρό.



- Την επιλογή ο χρήστης να διαγράψει το προσωπικό του προφίλ.

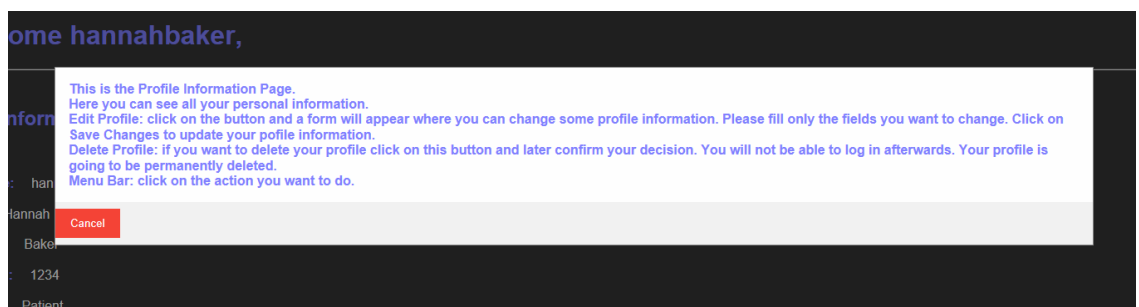


- Την δυνατότητα αποσύνδεσης του χρήστη από την εφαρμογή.

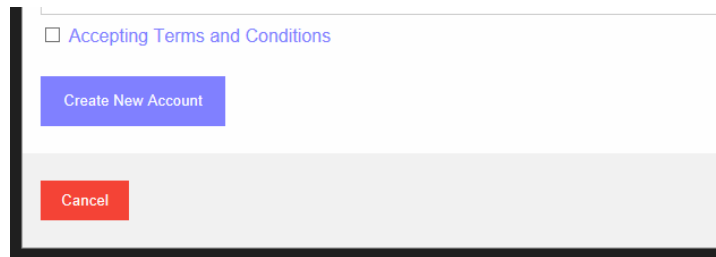


Στη συνέχεια υλοποιήθηκαν όλες εκείνες οι λειτουργίες που αναλύθηκαν στην αναφορά για την πρώτη εκτελέσιμη έκδοση της εφαρμογής και συμπεριλαμβάνει τα παρακάτω:

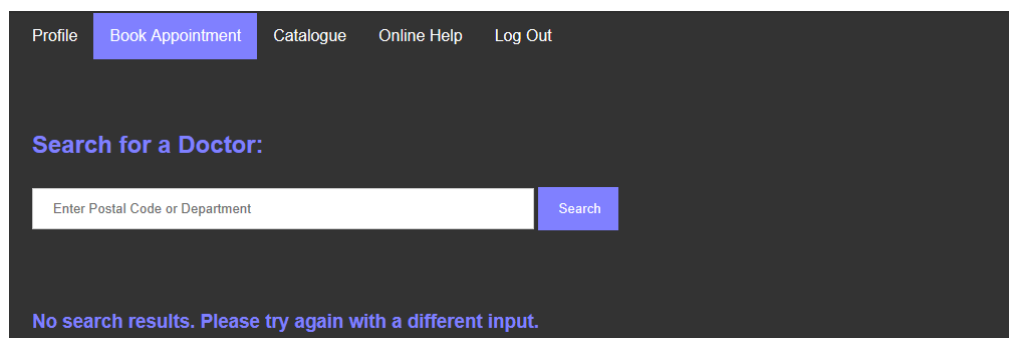
- Την παροχή Online Help.



- Την παροχή User Manual.
- Την αποδοχή των όρων χρήσης της εφαρμογής.



- Ενημέρωση του χρήστη με κατάλληλο μήνυμα όταν δεν υπάρχουν αποτελέσματα για την αναζήτηση που πραγματοποιήσε.



Επίσης, το User Interface απέκτησε μεγαλύτερη συνέπεια όπως φαίνεται και από τα ενδεικτικά στιγμιότυπα της εφαρμογής.

Αλλαγές έγιναν και στον κώδικα της εφαρμογής με την αλλαγή του ονόματος του CatalogueBean.java σε AppointmentBean.java. Η παραμετροποίηση αυτή έγινε έπειτα από περεταίρω ανάλυση του προβλήματος κατά την οποία διαπιστώθηκε ότι η αλλαγή αυτή αντικατοπτρίζει καλύτερα το ρόλο της.

Ακόμα, προστέθηκε και ο SubmitAppointmentServlet.java καθώς και το SubmitAppointmentDao.java για την ολοκλήρωση του προγραμματισμού ενός ραντεβού.

### 5.2.2 Αναφορά για την τελική εκτελέσιμη έκδοση

Μέσα από την διαδικασία της RUP, δημιουργήθηκε το τελικό αποτέλεσμα της web εφαρμογής καταχώρησης ραντεβού ασθενών- ιατρών. Όλες οι εκκρεμότητες που αναλύθηκαν παραπάνω, διευθετήθηκαν. Το πρόγραμμα είναι έτοιμο προς χρήση. Παρέχει όλα τα βασικά ζητούμενα που αναφέρθηκαν κατά την φάση της έναρξης, καθώς και επιπλέον λειτουργίες οι οποίες προέκυψαν κατά την φάση της ανάλυσης απαιτήσεων.

Η εμπορική χρήση της εφαρμογής αυτής όχι μόνο θα επισπεύσει την διαδικασία προγραμματισμού ενός ραντεβού αλλά και θα αποδεσμεύσει αρκετό προσωπικό το οποίο κύρια αρμοδιότητά του ήταν ο προγραμματισμός ιατρικών ραντεβού μέσω τηλεφώνου.

## 6. Εγχειρίδιο Χρήστη

### 6.1 Σύντομη παρουσίαση του προγράμματος

**Αρχική:** Αυτή είναι η αρχική σελίδα της εφαρμογής. Κάνοντας κλικ σε ένα από τα διαθέσιμα κουμπιά εμφανίζεται η αντίστοιχη φόρμα προς συμπλήρωση που απαιτείται για την είσοδο στην εφαρμογή.

- <<Login>> : πατώντας αυτό το κουμπί εμφανίζεται μία φόρμα στην οποία ζητείται η εισαγωγή του username και του κωδικού πρόσβασης, τα οποία είναι μοναδικά για κάθε χρήστη της εφαρμογής.
- <<Register>> : πατώντας αυτό το κουμπί εμφανίζεται μία φόρμα στην οποία ζητείται η εισαγωγή όλων των απαραίτητων προσωπικών στοιχείων που χρειάζεται η εφαρμογή για την εγγραφή ενός χρήστη. Για την ολοκλήρωση της εγγραφής, απαιτείται και η αποδοχή των όρων χρήσης της εφαρμογής.

Αν ο χρήστης αλλάξει γνώμη και δεν θελήσει να συμπληρώσει την φόρμα, μπορεί να πατήσει το αντίστοιχο κουμπί <<Cancel>> που βρίσκεται κάτω αριστερά ή να πατήσει το <<X>> που βρίσκεται πάνω δεξιά της φόρμας.

Πάνω δεξιά της σελίδας υπάρχει το κουμπί <<Online Help>> το οποίο μπορεί να πατήσει ο χρήστης αν αντιμετωπίζει κάποια δυσκολία κατά την περιήγησή του στην εφαρμογή και θέλει να αντλήσει οδηγίες.

Πάνω αριστερά της σελίδας υπάρχει το κουμπί <<User Manual>> το οποίο μπορεί να πατήσει ο χρήστης και να μεταφερθεί στο Εγχειρίδιο Χρήστη της εφαρμογής και να διαβάσει αναλυτικά όλες τις λειτουργίες της εφαρμογής.

**Profile Information:** Σε αυτή τη σελίδα ο χρήστης μπορεί να δει αναλυτικά όλα τα προσωπικά στοιχεία τα οποία είχε εισάγει κατά την εγγραφή του στην σελίδα. Στο κάτω μέρος υπάρχουν δύο κουμπιά:

- <<Edit Profile>> : με το πάτημα αυτού του κουμπιού ο χρήστης μπορεί να παραμετροποιήσει ορισμένες προσωπικές του πληροφορίες. Εμφανίζεται μία φόρμα με τα πεδία που μπορεί να αλλάξει και συμπληρώνει μόνο αυτά που επιθυμεί. Ύστερα επικυρώνει τις αλλαγές με το πάτημα του κουμπιού <<Save Changes>>.
- <<Delete Profile>> : με το πάτημα αυτού του κουμπιού ο χρήστης μπορεί να διαγράψει οριστικά τον λογαριασμό του. Πριν την διαγραφή εμφανίζεται προειδοποιητικό μήνυμα και ο χρήστης αν όντως επιθυμεί την διαγραφή θα πρέπει να πατήσει το κουμπί <<Yes, I'm sure>>.

Στο πάνω μέρος της σελίδας υπάρχει το μενού της εφαρμογής από το οποίο μπορεί να μεταφερθεί σε όλες τις λειτουργίες της εφαρμογής.

**Book Appointment:** Σε αυτή τη σελίδα ο χρήστης μπορεί να αναζητήσει ανά περιοχή ή ανά ειδικότητα ιατρούς.

- Η αναζήτηση ανά περιοχή γίνεται με την εισαγωγή του ταχυδρομικού κώδικα που εξυπηρετεί τον χρήστη-ασθενή. Στην συνέχεια, εμφανίζεται μία λίστα με όλους τους διαθέσιμους ιατρούς που έχουν τον αντίστοιχο ταχυδρομικό κώδικα. Στην λίστα αυτή αναγράφεται εκτός από τον όνομα του ιατρού και η διεύθυνσή του, ο αριθμός τηλεφώνου, το email, και η ειδικότητά του.
- Η αναζήτηση ανά ειδικότητα γίνεται με την εισαγωγή της ειδικότητας που επιθυμεί ο χρήστης- ασθενής. Στην συνέχεια, εμφανίζεται μία λίστα με όλους τους διαθέσιμους

ιατρούς που έχουν την αντίστοιχη ειδικότητα. Στην λίστα αυτή αναγράφεται εκτός από τον όνομα του ιατρού και η διεύθυνσή του, ο αριθμός τηλεφώνου, το email, και η ειδικότητά του.

Στο πάνω μέρος της σελίδας υπάρχει το μενού της εφαρμογής από το οποίο μπορεί να μεταφερθεί σε όλες τις λειτουργίες της εφαρμογής.

**Catalogue:** Σε αυτή τη σελίδα ο χρήστης μπορεί να δει όλα τα προγραμματισμένα του ραντεβού σε έναν κατάλογο. Επίσης, μπορεί να δει και όλα τα ακυρωμένα και περασμένα-ολοκληρωμένα ραντεβού του. Για κάθε ραντεβού υπάρχει ένδειξη active, cancelled ή completed για να γνωρίζει εύκολα και γρήγορα ο ασθενής ή ο ιατρός την κατάσταση των ραντεβού. Ακόμα, αναγράφεται το όνομα του ιατρού, η ώρα, ημερομηνία και λόγο επίσκεψης στον ιατρό.

Εφόσον ένα ραντεβού είναι ακόμα ενεργό, μπορεί ο χρήστης να το ακυρώσει πατώντας το αντίστοιχο κουμπί που βρίσκεται δίπλα. Αμέσως, η κατάσταση του ραντεβού αλλάζει και από ενεργό γίνεται ακυρωμένο.

Στο πάνω μέρος της σελίδας υπάρχει το μενού της εφαρμογής από το οποίο μπορεί να μεταφερθεί σε όλες τις λειτουργίες της εφαρμογής.

**Log Out:** Πατώντας αυτό το κουμπί, ο χρήστης αποσυνδέεται από τον λογαριασμό του και μεταφέρεται αυτόματα στην αρχική σελίδα της εφαρμογής, εκεί δηλαδή που μπορεί να κάνει εκ νέου σύνδεση ή εγγραφή.

## 6.2 Παρουσίαση σεναρίων λειτουργίας

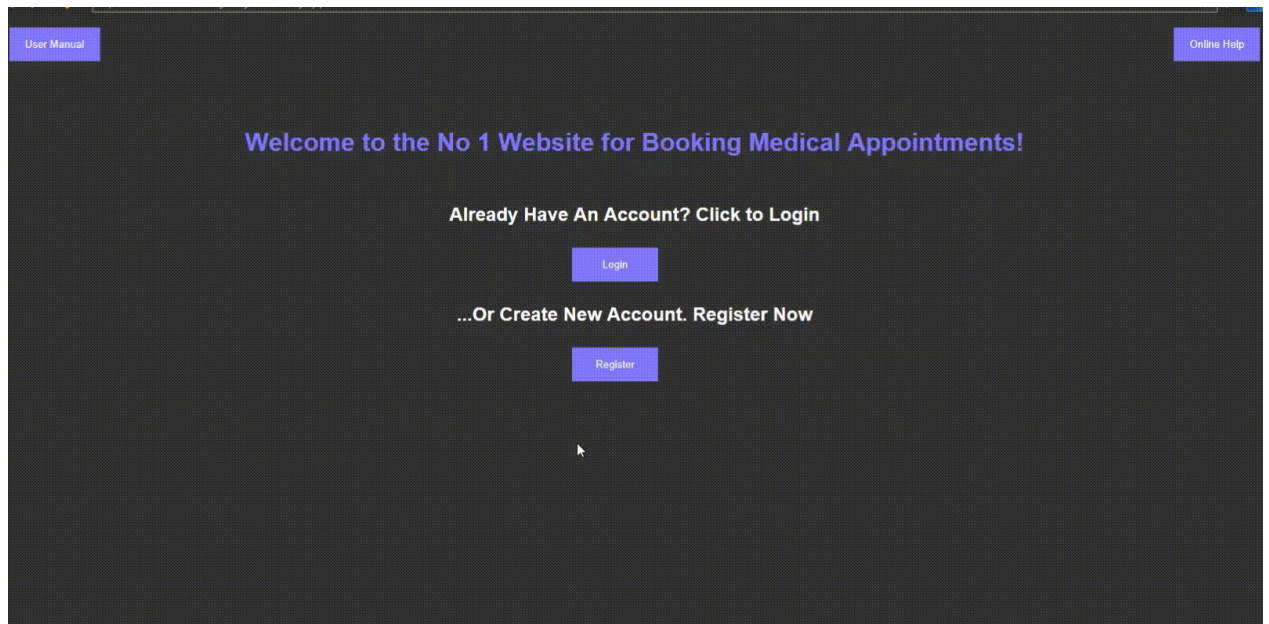
Παρακάτω παρουσιάζονται διαφορετικά σενάρια χρήσης της εφαρμογής.

### 6.2.1 Σενάριο 1 : Εγγραφή ενός νέου χρήστη στην εφαρμογή

Για την εγγραφή ενός νέου χρήστη στην εφαρμογή, είτε με την ιδιότητα του ιατρού είτε με αυτήν του ασθενή, απαιτείται η εισαγωγή όλων των απαραίτητων στοιχείων που χρειάζεται η εφαρμογή για την καταχώρησή του.

Ο χρήστης διαλέγει το username και το password που επιθυμεί. Ύστερα εισάγει την ιδιότητά του, δηλαδή ασθενής (patient) ή ιατρός (doctor). Εφόσον είναι ιατρός συμπληρώνει και την ειδικότητά του (department). Επίσης, προσθέτει την διεύθυνσή του, το email του και το κινητό τηλέφωνό του. Στη συνέχεια, αποδέχεται τους όρους χρήσης της εφαρμογής και πατά το κουμπί <<Create Account>>. Με το που πατήσει το κουμπί μεταφέρεται αυτόματα στην σελίδα που βλέπει τις προσωπικές του πληροφορίες.

Παρακάτω παρουσιάζεται η διαδικασία αυτή σε μορφή gif.



## 6.2.2 Σενάριο 2 : Παραμετροποίηση των προσωπικών στοιχείων ενός χρήστη

Η παραμετροποίηση των στοιχείων του χρήστη γίνεται μέσα από την σελίδα <<Profile Information>>. Εκεί μπορεί να διαβάσει τις πληροφορίες του και πατώντας το κουμπί <<Edit Profile>> να τις παραμετροποιήσει. Μπορεί να παραμετροποιήσει μόνο ορισμένα πεδία, όπως είναι αυτά του κωδικού πρόσβασης (password), της διεύθυνσης, του κινητού τηλεφώνου και του email.

Πατώντας το κουμπί, του εμφανίζεται μία φόρμα με τα πεδία εκείνα που επιτρέπεται να αλλάξει και ο χρήστης συμπληρώνει μόνο εκείνα που επιθυμεί να παραμετροποιήσει. Ύστερα, επικυρώνει τις αλλαγές πατώντας το κουμπί <<Save Changes>>. Εφόσον το κουμπί αυτό πατηθεί, ο χρήστης θα δει κατευθείαν τις αλλαγές που έκανε στις προσωπικές του πληροφορίες.



### 6.2.3 Σενάριο 3 : Προγραμματισμός ενός ραντεβού

Για το προγραμματισμό ενός ραντεβού, ο χρήστης ασθενής πρέπει να μεταβεί στη σελίδα με όνομα <<Book Appointment>>. Εκεί του δίνεται η δυνατότητα να αναζητήσει ιατρούς με βάση τη περιοχή του ή την ειδικότητα που επιθυμεί. Για την περιοχή αρκεί να εισάγει τον ταχυδρομικό κώδικα που επιθυμεί. Στη συνέχεια, πατά το κουμπί <<Search>> και τα διαθέσιμα αποτελέσματα του εμφανίζονται σε ένα πίνακα. Αν δεν υπάρχουν αποτελέσματα, εμφανίζεται κατάλληλο μήνυμα.

Εφόσον επιλέξει τον ιατρό με τον οποίο επιθυμεί τον προγραμματισμό του ραντεβού, πληκτρολογεί την ώρα, μέρα και λόγο επίσκεψης και πατά το κουμπί <<Book Appointment>>. Αυτόματα μεταφέρεται στην σελίδα του καταλόγου όπου εκεί μπορεί να δει όλα τα προγραμματισμένα του ραντεβού.

[Profile](#) [Book Appointment](#) [Catalogue](#) [Online Help](#) [Log Out](#)

**Search for a Doctor:**

Username	Name	Surname	Phone Number	Address	Postal Code	Email	Department
charlesleclerc	Charles	Leclerc	683546789	Herlongs Beach 567	11356	charles@email.com	Gynecology
jimmynewtron	Jimmy	Newtron	690457896	Rodeo Drive 456	11356	newtron@email.com	Gynecology
lewishamilton	Lewis	Hamilton	698367856	St. Jekings 130	11356	lewis@email.com	Dermatology
toto	Toto	Wolf	6904839677	St. Kerlings 34	11356	toto@email.com	Dermatology

[Profile](#) [Book Appointment](#) [Catalogue](#) [Online Help](#) [Log Out](#)

**Doctor Username**

**Time of Appointment**

**Date of Appointment**

**Description**

#### 6.2.4 Σενάριο 4 : Ακύρωση προγραμματισμένου ραντεβού

Εφόσον ένα προγραμματισμένο ραντεβού είναι ενεργό, μπορεί ο χρήστης ασθενής μέσα από την σελίδα <<Catalogue>> να το ακυρώσει πατώντας το κουμπί <<Cancel Appointment>> που βρίσκεται δίπλα από το αντίστοιχο ενεργό ραντεβού. Αμέσως, το status του ραντεβού αλλάζει και από ενεργό γίνεται ακυρωμένο.

[Profile](#) [Book Appointment](#) [Catalogue](#) [Online Help](#) [Log Out](#)

## Catalogue

All your booked appointments in one place!

Patient	Doctor	Time of Appointment	Date of Appointment	Description	Status	
hannahbaker	totowolf	12:00	12/7/21	annual checkup	cancelled	
hannahbaker	lewishamilton	12:00	30/10/21	prescription medicine	active	<a href="#">Cancel Appointment</a>

[Profile](#) [Book Appointment](#) [Catalogue](#) [Online Help](#) [Log Out](#)

## Catalogue

All your booked appointments in one place!

Patient	Doctor	Time of Appointment	Date of Appointment	Description	Status	
hannahbaker	totowolf	12:00	12/7/21	annual checkup	cancelled	
hannahbaker	lewishamilton	12:00	30/10/21	prescription medicine	cancelled	