



## **ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ – ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

### **Πρόγραμμα Μεταπτυχιακών Σπουδών**

#### **«Πληροφορική»**

#### **Μεταπτυχιακή Διατριβή**

Τίτλος Διατριβής	<b>Σύστημα Διαχείρισης Νοσοκομείου με χρήση Spring Boot Framework</b> <b>Hospital Management System using Spring Boot Framework</b>
Όνοματεπώνυμο Φοιτητή	<b>Ιωάννης-Πρόδρομος Κούτρας</b>
Πατρώνυμο	<b>Κωνσταντίνος</b>
Αριθμός Μητρώου	<b>ΜΠΠΛ 21039</b>
Επιβλέπων	<b>Ευθύμιος Αλέπης, Αναπληρωτής Καθηγητής</b>

Ημερομηνία Παράδοσης **Νοέμβριος 2023**

---

### **Τριμελής Εξεταστική Επιτροπή**

Ευθύμιος Αλέπης  
Αναπληρωτής Καθηγητής

Μαρία Βίββου  
Καθηγήτρια

Διονύσιος Σωτηρόπουλος  
Επίκουρος Καθηγητής

## Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου κύριο Ευθύμιο Αλέπη, για την ευκαιρία που μου έδωσε να επιλέξω αυτή την μεταπτυχιακή διατριβή καθώς και για τις συμβουλές και την καθοδήγησή του κατά τη διάρκεια της υλοποίησης της. Επιπρόσθετα, θα ήθελα να εκφράσω τις ευχαριστίες μου σε όλους τους καθηγητές μου στο Πρόγραμμα Μεταπτυχιακών Σπουδών «Πληροφορική» του Πανεπιστημίου Πειραιώς, για τις γνώσεις που μου προσέφεραν κατά την διάρκεια της διετούς φοίτησης μου οι οποίες με βοήθησαν να αποκομίσω τεράστια εφόδια για την περαιτέρω σταδιοδρομία μου.

## Περιεχόμενα

<b>1. Περίληψη</b> .....	6
<b>1.1 Περίληψη</b> .....	6
<b>1.2 Abstract</b> .....	6
<b>2. Εισαγωγή</b> .....	7
<b>2.1 Στόχος της Εργασίας</b> .....	7
<b>2.2 Κίνητρο της Εργασίας</b> .....	7
<b>2.3 Τεχνολογίες</b> .....	8
<b>3. Ανάλυση Απαιτήσεων</b> .....	8
<b>4. Σχεδιασμός - Βάση Δεδομένων</b> .....	10
<b>4.1 Διάγραμμα Οντοτήτων-Συσχετίσεων</b> .....	10
<b>4.2 Πίνακες βάσης δεδομένων</b> .....	11
4.2.1 Πίνακας user .....	11
4.2.2 Πίνακας doctor .....	12
4.2.3 Πίνακας roles.....	12
4.2.4 Πίνακας users_roles.....	12
4.2.5 Πίνακας appointment .....	13
4.2.6 Πίνακας diagnosis .....	14
4.2.7 Πίνακας medicine .....	14
<b>4.3 Ασφάλεια βάσης δεδομένων</b> .....	15
<b>5. Προγραμματιστικό Περιβάλλον</b> .....	15
<b>5.1 Java</b> .....	15
<b>5.2 Framework</b> .....	16
<b>5.3 Αρχιτεκτονική Spring Boot</b> .....	16
<b>5.4 MVC Αρχιτεκτονική</b> .....	17
<b>5.5 Spring Boot</b> .....	19
5.5.1 Controlllers .....	19
5.5.2 Services .....	20
5.5.3 Repositories .....	20
5.5.4 Annotations.....	20
<b>6. Είσοδος στην Εφαρμογή</b> .....	29
<b>7. Παρουσίαση Εφαρμογής</b> .....	30
<b>7.1 Είσοδος ή Εγγραφή</b> .....	30

<b>7.2 Είσοδος Ασθενή</b> .....	34
7.2.1 Το Προφίλ μου .....	34
7.2.2 Νέο Ραντεβού.....	35
7.2.3 Τα Ραντεβού μου .....	35
7.2.4 Οι Διαγνώσεις μου.....	36
7.2.5 Εφαρμογές Υγείας & Ευεξίας.....	37
7.2.6 Ά Βοήθειες .....	38
7.2.7 Επικοινωνία .....	39
<b>7.3 Είσοδος Ιατρού</b> .....	39
7.3.1 Ραντεβού.....	40
7.3.2 Οι διαγνώσεις μου .....	41
7.3.3 Αναζήτηση Ασθενή.....	42
7.3.4 Αναζήτηση Γιατρών .....	42
<b>7.4 Είσοδος Διαχειριστή</b> .....	43
7.4.1 Όλοι οι Χρήστες .....	43
7.4.2 Έγκριση Χρηστών .....	44
7.4.3 Όλα τα Ραντεβού .....	45
7.4.4 Όλα οι Διαγνώσεις.....	45
<b>8. Συμπεράσματα - Επεκτάσεις</b> .....	46
<b>9. Βιβλιογραφία</b> .....	46
<b>Ελληνική</b> .....	46
<b>Ξενόγλωσση</b> .....	46

## 1. Περίληψη

### 1.1 Περίληψη

Η παρούσα μεταπτυχιακή διατριβή αποσκοπεί στον σχεδιασμό, την ανάπτυξη και παρουσίαση μιας web εφαρμογής, η οποία ικανοποιεί ιατρικής φύσεως ανάγκες του χρήστη. Η Clinic System πρόκειται για μια εφαρμογή η οποία χρησιμοποιείται για την άμεση αλληλεπίδραση Ιατρού και Ασθενή μίας κλινικής ή νοσοκομείου. Είναι ουσιαστικά ένα διαδικτυακό πληροφοριακό σύστημα όπου εγγεγραμμένοι χρήστες μετά τη δημιουργία προφίλ θα έχουν τη δυνατότητα να κλείσουν απομακρυσμένα ραντεβού για κάποιο ιατρείο που επιθυμούν, να παρακολουθούν το ιστορικό τους, να έχουν πρόσβαση σε μικροεφαρμογές υγείας και ευεξίας καθώς και να ενημερώνονται για ποικίλα ιατρικά νέα και θέματα.

Επίσης η εφαρμογή παρέχει τη δυνατότητα να εξάγονται τα πορίσματα των εξετάσεων από τον επιβλέποντα ιατρό και να διατηρείται ένας ιατρικός φάκελος του ασθενή.

Τέλος, η εφαρμογή αυτή περιλαμβάνει σύστημα διαχείρισης, μέσω του οποίου, οι διαχειριστές έχουν τη δυνατότητα προβολής και επεξεργασίας όλων των απαραίτητων πεδίων σε περίπτωση αδυναμίας των χρηστών ανεξαρτήτως ρόλου.

### 1.2 Abstract

This post graduate thesis aims to design, develop and present a web application, which satisfies the medical needs of the user. The Clinic System is an application which is used for direct interaction between a doctor and a patient of a clinic or hospital. It is essentially a web-based information system where registered users, after creating a profile, will be able to make remote appointments for a clinic of their choice, monitor their medical history, access health and wellness micro-applications and keep up to date with a variety of medical news and issues.

The app also provides the ability to extract test results and diagnoses from the supervising doctor and maintain a medical record of the patient.

Finally, this application includes an administration system, through which, the administrators have the possibility to view and edit all the necessary fields in case of inability of users regardless of their role.

## 2. Εισαγωγή

### 2.1 Στόχος της Εργασίας

Στόχος της παρούσας εργασίας είναι η δημιουργία ενός διαδικτυακού πληροφοριακού συστήματος διαχείρισης νοσοκομείου. Μέσα από ένα ελκυστικό και ευχάριστο περιβάλλον:

- ο ασθενής θα έχει την δυνατότητα να έχει πρόσβαση στις διαδικτυακές υπηρεσίες του νοσοκομείου, να προγραμματίζει ή να ακυρώνει ηλεκτρονικά τα ραντεβού του, να βλέπει τα αποτελέσματα των εξετάσεων του, να ενημερώνεται για ιατρικού περιεχομένου θέματα καθώς και πρόσβαση εφαρμογές σωματομετρικών μετρήσεων.
- ο ιατρός θα μπορεί να κρατάει ιστορικό για τους ασθενείς του, να ανεβάζει τα αποτελέσματα των εξετάσεων και τις διαγνώσεις του κάθε ασθενή και να βλέπει τα προγραμματισμένα του ραντεβού.
- ο διαχειριστής θα έχει την δυνατότητα να βλέπει την συνολική εικόνα των ραντεβού και των εξετάσεων, να δημιουργεί / επεξεργάζεται / διαγράφει χρήστες, επεξεργασία και καταχώρηση διάγνωσης σε περίπτωση αδυναμίας του ιατρού καθώς και να εγκρίνει την εγγραφή νέου ιατρού ώστε να μπορεί να κάνει χρήση της εφαρμογής.

### 2.2 Κίνητρο της Εργασίας

Έναυσμα για τη συγκεκριμένη διατριβή αποτέλεσε το μάθημα «Ιατρική Πληροφορική» που εντάσσεται στο Πρόγραμμα Μεταπτυχιακών Σπουδών «Πληροφορική» του Πανεπιστημίου Πειραιώς. Βασικό αντικείμενο στο συγκεκριμένο μάθημα είναι η εφαρμογή τεχνολογιών σε θέματα του κλάδου της Ιατρικής Επιστήμης.

Η υλοποίηση και η χρήση μιας εφαρμογής που σχετίζεται με θέματα Ιατρικής και Υγείας που θα καθιστήσει εύκολη την πρόσβαση του συνόλου των πολιτών χωρίς ταλαιπωρία σε συνδυασμό με το σύνολο των μαθημάτων που διδάχθηκαν τα προηγούμενα εξάμηνα συνέβαλαν στην δημιουργία μιας ολοκληρωμένης εφαρμογής που καλύπτει αυτές τις ανάγκες και την υλοποίηση της παρούσας μεταπτυχιακής διατριβής.

## 2.3 Τεχνολογίες

Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκε η γλώσσα προγραμματισμού Java, τεχνολογίες του Spring framework και για τη βάση δεδομένων το λογισμικό MySQL Workbench.

## 3. Ανάλυση Απαιτήσεων

Η εργασία απαιτεί την σχεδίαση και υλοποίηση μιας βάσης δεδομένων μιας νοσηλευτικής κλινικής. Σκοπός είναι μέσα από αυτή τη βάση δεδομένων να μπορούν ασθενής (user) και γιατρός (doctor) να βλέπουν και να καταχωρούν αναλόγως, πληροφορίες όπως τα ραντεβού (appointment), τις διαγνώσεις (diagnosis) του καθώς και τα φάρμακα (medicine) που χορηγήθηκαν.

Όλες αυτές οι πληροφορίες αποτελούν ξεχωριστές οντότητες από μόνες τους, οι οποίες όμως μπορούν να συνδεθούν με τις πληροφορίες μιας άλλης οντότητας και να μας δώσουν επιθυμητά αποτελέσματα σχετικά με την πληροφορία που αναζητούμε και η οποία δεν μπορεί να προκύψει από μόνη της σε κάποια οντότητα.

Για να πετύχουμε αυτό το σκοπό καταχωρούμε τις πληροφορίες κάθε οντότητας σε αντίστοιχους πίνακες λαμβάνοντας υπόψη τους τύπους δεδομένων κάθε πεδίου των πινάκων καθώς και τους περιορισμούς που εξάγονται .

Κάθε χρήστης αναγνωρίζεται από ένα μοναδικό κωδικό id.

Για κάθε ασθενή πρέπει να τηρούνται τα ακόλουθα στοιχεία:

**Όνομα**

**Επώνυμο**

**ΑΜΚΑ**

**Email**

**Τηλέφωνο**

**Κωδικός Πρόσβασης**

Ενώ για τους γιατρούς τηρείται επιπλέον η Ειδικότητά του.

Οι απαραίτητες πληροφορίες για κάθε ραντεβού είναι:



**Κωδικός****Ημερομηνία****Ειδικότητα Γιατρού****Αιτιολόγηση Ραντεβού****Κατάσταση Ραντεβού**

Οι απαραίτητες πληροφορίες για κάθε διάγνωση είναι:

**Κωδικός****Ημερομηνία****Διάγνωση****Θεραπεία****Κωδικός Ασθενή**

Ο ασθενής μπορεί να :

- προγραμματίζει το ραντεβού του για το ιατρείο που επιθυμεί.
- ακυρώνει το ραντεβού που έχει προγραμματίσει.
- βλέπει το ιστορικό των πραγματοποιημένων και μη ραντεβού του.
- βλέπει τις διαγνώσεις και τα αποτελέσματα από τις εξετάσεις στις οποίες έχει υποβληθεί.

Ο γιατρός μπορεί να:

- να συμπληρώνει νέα διάγνωση όταν εξετάζει τον ασθενή.
- βλέπει το ιστορικό των διαγνώσεων για κάθε ασθενή.
- βλέπει το σύνολο των ραντεβού του
- αναζητήσει ιστορικό ασθενή με βάση τον ΑΜΚΑ του ασθενή.

Ο διαχειριστής μπορεί να:

- βλέπει και επεξεργάζεται όλους τους χρήστες.
- βλέπει όλα τα ραντεβού.
- βλέπει και επεξεργάζεται όλες τις διαγνώσεις.
- εγκρίνει την εγγραφή νέου ιατρού στην εφαρμογή.

## 4. Σχεδιασμός - Βάση Δεδομένων

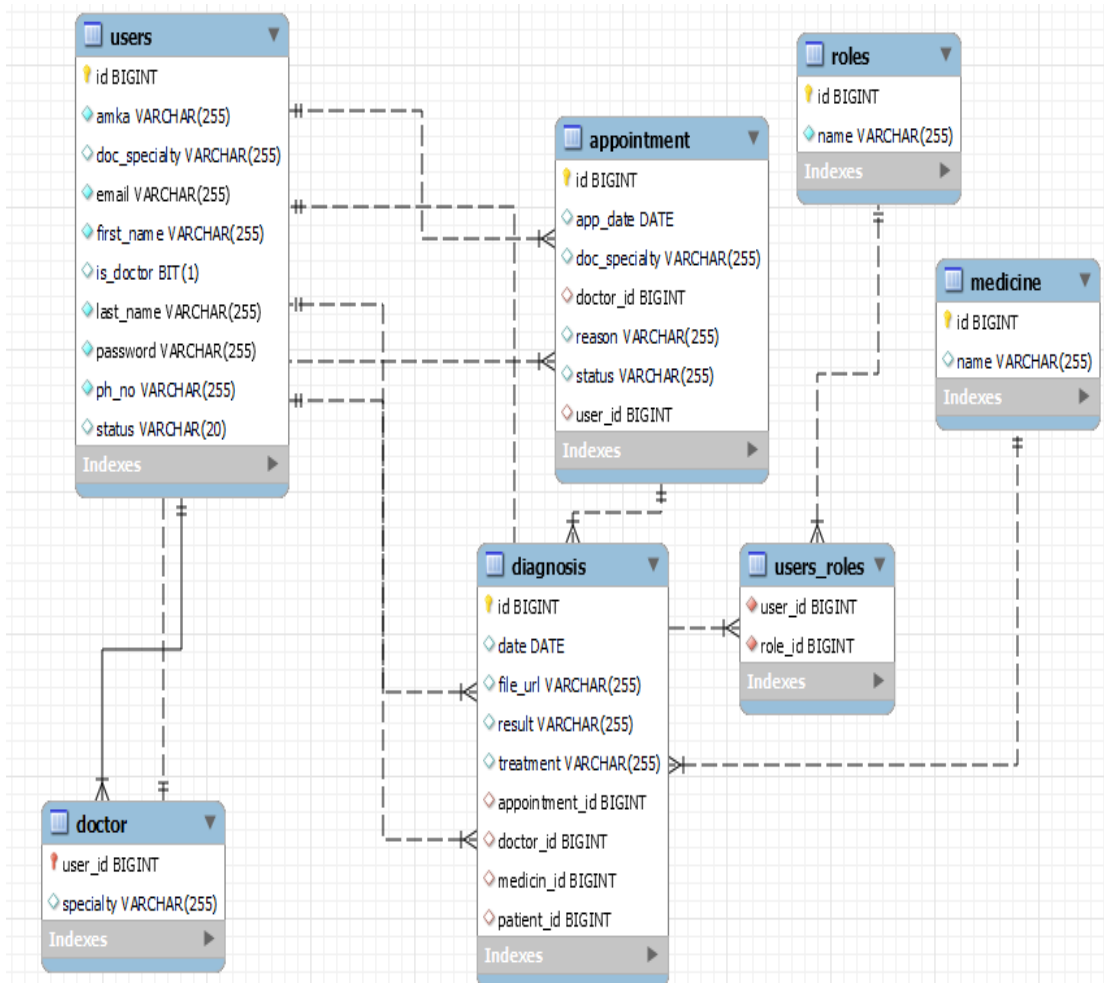
### 4.1 Διάγραμμα Οντοτήτων-Συσχετίσεων

Προκειμένου να υλοποιήσουμε τον εννοιολογικό σχεδιασμό δημιουργήσαμε ένα εννοιολογικό σχήμα για τη ΒΔ, με χρήση ενός εννοιολογικού μοντέλου δεδομένων υψηλού επιπέδου. Το εννοιολογικό σχήμα είναι μια περιεκτική περιγραφή των απαιτήσεων (ή τουλάχιστον των περισσότερων από τις απαιτήσεις) των χρηστών σχετικά με τα δεδομένα και περιλαμβάνει λεπτομερείς περιγραφές των τύπων δεδομένων, των συσχετίσεων και των περιορισμών. Χρησιμοποιήσαμε λοιπόν, το μοντέλο Οντοτήτων-Συσχετίσεων (Entity-Relationship Model).

Η Βάση δεδομένων που χρησιμοποιήθηκε έχει έξι οντότητες. Πιο συγκεκριμένα:

- **Χρήστης** (User)
- **Ιατρός** (Doctor)
- **Ρόλος** (Role)
- **Ραντεβού** (Appointment)
- **Διάγνωση** (Diagnosis)
- **Φάρμακο** (Medicine)

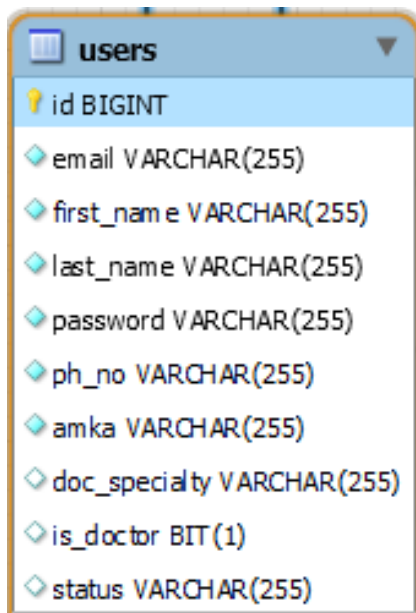
Οι συσχετίσεις μεταξύ των οντοτήτων φαίνονται στο παρακάτω σχεσιακό σχήμα.



## 4.2 Πίνακες βάσης δεδομένων

### 4.2.1 Πίνακας user

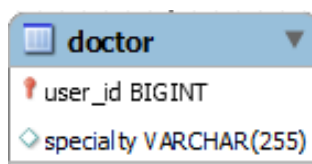
Στον πίνακα user, αποθηκεύονται οι βασικές πληροφορίες για όλους τους χρήστες ανεξαρτήτως ρόλου, είτε αυτοί είναι Ασθενείς, Ιατροί ή Διαχειριστές.



users
id BIGINT
email VARCHAR(255)
first_name VARCHAR(255)
last_name VARCHAR(255)
password VARCHAR(255)
ph_no VARCHAR(255)
amka VARCHAR(255)
doc_specialty VARCHAR(255)
is_doctor BIT(1)
status VARCHAR(255)

#### 4.2.2 Πίνακας doctor


Ο πίνακας αυτός , περιέχει τις πληροφορίες σχετικά με τους γιατρούς, που με βάση το id τους μπορούμε να αναγνωρίσουμε την ειδικότητα του συγκεκριμένου ιατρού.



doctor
user_id BIGINT
specialty VARCHAR(255)

#### 4.2.3 Πίνακας roles

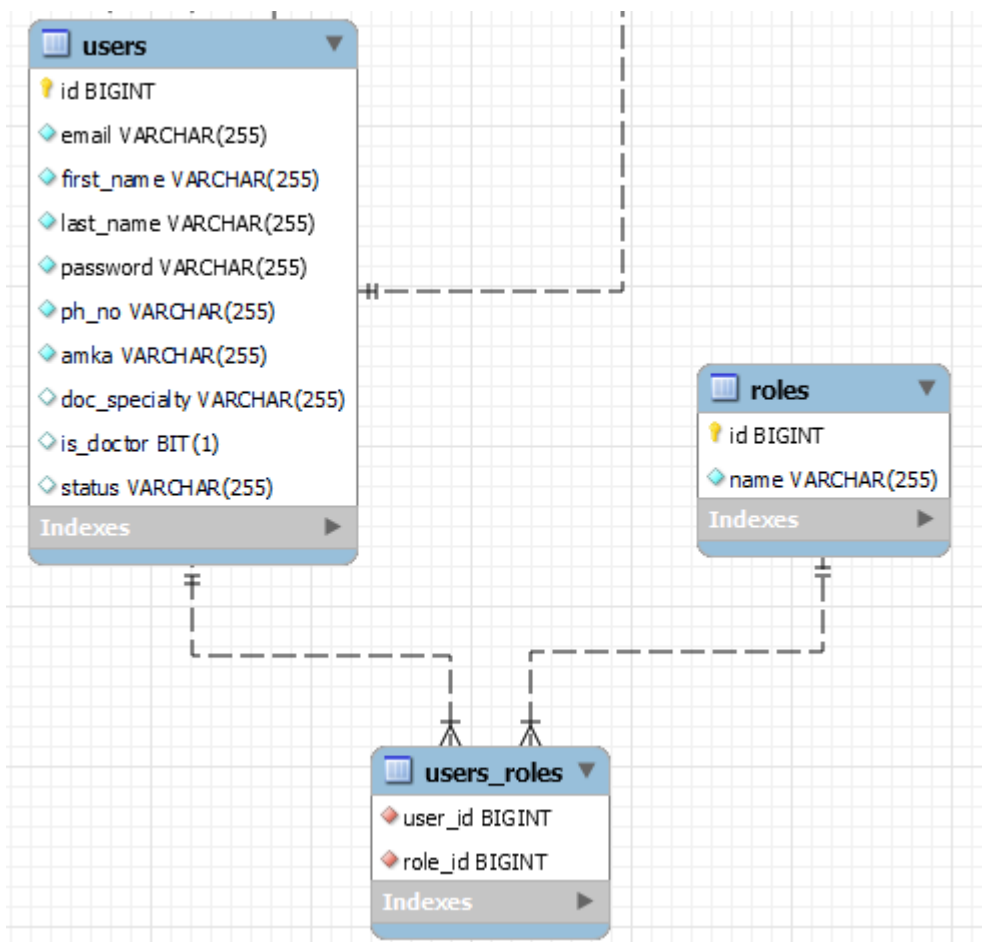
Στον πίνακα αυτό περιέχονται οι τρεις ΜΟΝΟ ρόλοι που μπορεί να έχει ένας χρήστης (PATIENT, DOCTOR, ADMIN).



roles
id BIGINT
name VARCHAR(255)

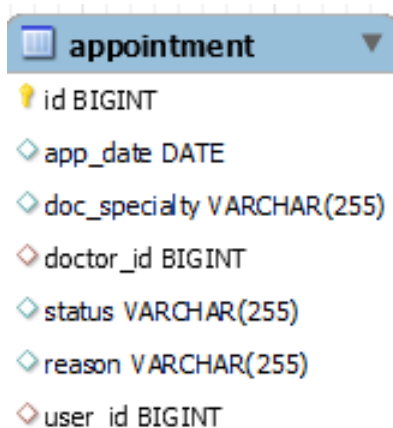
#### 4.2.4 Πίνακας users\_roles

Η σύνδεση του εκάστοτε χρήστη με κάποιον ρόλο γίνεται με τον πίνακα users\_roles γι' αυτό και βλέπουμε ότι ο πίνακας έχει τον συνδυασμό των user\_id και role\_id ως πρωτεύων κλειδί.



#### 4.2.5 Πίνακας appointment

Ο πίνακας αυτός, περιέχει τις πληροφορίες σχετικά με τα ραντεβού που προγραμματίζονται από τους ασθενείς για συγκεκριμένο ιατρείο δηλαδή ειδικότητα αλλά και για εξέταση με συγκεκριμένο γιατρό.



Τα πεδία που απαιτούνται να συμπληρωθούν είναι η **ημερ/νία** του ραντεβού, η **ειδικότητα του γιατρού**, ο **γιατρός**, ο **λόγος της εξέτασης** και τέλος η **κατάσταση** στην οποία έχει χαρακτηριστεί το ραντεβού, **ΕΚΚΡΕΜΕΙ** σε περίπτωση που δεν έχει ολοκληρωθεί ακόμα και **ΟΛΟΚΛΗΡΩΜΕΝΟ** στην αντίθετη.

#### 4.2.6 Πίνακας diagnosis

Στον πίνακα αυτό, περιέχονται όλες εκείνες οι πληροφορίες που αφορούν τη διάγνωση στην οποία προέβη ένας ιατρός μετά από ένα ραντεβού με έναν ασθενή. Πληροφορίες οι οποίες είναι χρήσιμες για τον ασθενή, προκειμένου να ξέρει τι εξετάσεις έχει κάνει και πότε και ποια αποτελέσματα είχαν, αλλά και για τον γιατρό για να έχει εικόνα του ιστορικού του ασθενούς και αυτό που λέμε ιατρικός φάκελος.

Field	Type
id	BIGINT
date	DATE
result	VARCHAR(255)
treatment	VARCHAR(255)
appointment_id	BIGINT
doctor_id	BIGINT
medicin_id	BIGINT
patient_id	BIGINT
file_url	VARCHAR(255)

#### 4.2.7 Πίνακας medicine

Τέλος Κάθε δραστική ουσία, που χορηγείται από τον εκάστοτε γιατρό και γράφεται στη διάγνωση του, αποθηκεύεται στον πίνακα medicine.

Field	Type
id	BIGINT
name	VARCHAR(255)

## 4.3 Ασφάλεια βάσης δεδομένων

Για την αυθεντικοποίηση και εξουσιοδότηση των συνδεδεμένων χρηστών που αποθηκεύονται στη Βάση Δεδομένων χρησιμοποιήθηκε το BCryptPasswordEncoder για την κρυπτογράφηση του κωδικού πρόσβασης.

Οι ρόλοι που μπορούν να αποδοθούν είναι τρεις: Ασθενής (**PATIENT**), Ιατρός (**DOCTOR**) και Διαχειριστής (**ADMIN**). Για τους δυο πρώτους γίνεται επιλογή κατά την εγγραφή στην εφαρμογή ενώ ο ρόλος του Διαχειριστή γίνεται μόνο μέσα από τη Βάση Δεδομένων.

Το εργαλείο που χρησιμοποιήθηκε για να απεικονιστεί η Βάση Δεδομένων είναι το MySQL Workbench όπου απεικονίζονται όλοι οι πίνακες και τα δεδομένα της εφαρμογής.

## 5. Προγραμματιστικό Περιβάλλον

### 5.1 Java

Η γλώσσα προγραμματισμού Java έχει πολλά πλεονεκτήματα που την καθιστούν δημοφιλή στις διαδικτυακές εφαρμογές:

**1. Πολυπλατφορμικότητα:** Η Java αναπτύχθηκε για να λειτουργεί σε πολλές πλατφόρμες.

**2. Ασφάλεια:** Η Java έχει ενσωματωμένα μέτρα ασφαλείας, όπως το σύστημα διαχείρισης μνήμης και τον έλεγχο πρόσβασης, που βοηθούν στην προστασία των εφαρμογών από δυνητικές απειλές ασφαλείας.

**3. Εκτελέσιμα αρχεία (JAR):** Ο κώδικας Java μπορεί να πακεταριστεί σε εκτελέσιμα JAR αρχεία, τα οποία μπορούν να εκτελεστούν απευθείας σε διάφορα περιβάλλοντα χωρίς την ανάγκη να μεταγλωττιστεί ο κώδικας σε κάθε περιβάλλον.

**4. Ευκολία στην μάθηση:** Η Java είναι γνωστή για τη σχετικά ευκολή της μάθησης και τη συγγραφή κώδικα. Οι προγραμματιστές μπορούν να επικεντρωθούν στην ανάπτυξη της λογικής της εφαρμογής αντί να ανησυχούν για λεπτομέρειες σχετικές με τη μνήμη και τη διαχείριση των πόρων.

**5. Ευρεία υποστήριξη βιβλιοθηκών:** Υπάρχει μια πληθώρα βιβλιοθηκών και πλαίσιων ανάπτυξης για την Java, που επιτρέπουν να εκτελεστούν πολλές λειτουργίες χωρίς να ξαναγραφεί ο κώδικας από την αρχή.

## 5.2 Framework

Το framework είναι ένα προκαθορισμένο περιβάλλον ανάπτυξης λογισμικού που παρέχει κοινές λειτουργίες και δομές για τη δημιουργία εφαρμογών. Σκοπός του framework είναι να διευκολύνει τον προγραμματιστή στην ανάπτυξη εφαρμογών, προσφέροντας ένα σύνολο γενικών λύσεων και βιβλιοθηκών που μπορούν να χρησιμοποιηθούν για συγκεκριμένες εργασίες ή λειτουργίες. Τα framework παρέχουν συχνά ένα σύνολο αρχιτεκτονικών και πρότυπων για την οργάνωση του κώδικα, τη διαχείριση της διάρθρωσης της εφαρμογής και τη διασύνδεση με άλλες τεχνολογίες. Ένα από τα δημοφιλέστερα πλαίσια εργασίας για την ανάπτυξη Java διαδικτυακών εφαρμογών, είναι το Spring Framework καθώς διαθέτει μία πληθώρα προγραμματιστικών εργαλείων τα οποία απλοποιούν την συγγραφή του κώδικα και δομούν την υλοποίηση με τον καλύτερο δυνατό τρόπο.

## 5.3 Αρχιτεκτονική Spring Boot

Στον πυρήνα του, το Spring Boot ακολουθεί τις αρχές του Spring Framework. Έχει σχεδιαστεί για να διευκολύνει την εγκατάσταση και την ανάπτυξη νέων εφαρμογών Spring. Το Spring Boot χρησιμοποιεί μια πολυεπίπεδη αρχιτεκτονική, η οποία διευκολύνει την δομή, τη συντηρησιμότητα και την επεκτασιμότητα των εφαρμογών. Αυτή η αρχιτεκτονική είναι ζωτικής σημασίας για την κατανόηση του τρόπου λειτουργίας του Spring Boot.

**1. Presentation Layer:** Το Presentation Layer είναι το σημείο όπου ο χρήστης αλληλεπιδρά με την εφαρμογή. Αποτελείται από ελεγκτές (controllers), όψεις (views) και πρότυπα (templates). Το Spring Boot υποστηρίζει διάφορα views. Στη προκειμένη περίπτωση χρησιμοποιήσαμε το Thymeleaf.

**2. Application Layer:** Αυτό το στρώμα περιέχει την επιχειρησιακή λογική της εφαρμογής. Περιλαμβάνει υπηρεσίες (services), οι οποίες ενθυλακώνουν τη λειτουργικότητα της εφαρμογής. Εδώ χρησιμοποιούνται ειδικές σημάνσεις (annotations:@) όπως @Service και @Controller για να ορίσουν το είδος και τη λειτουργία κάθε συστατικού (bean), και η αυτόματη διαμόρφωση (auto-configuration )

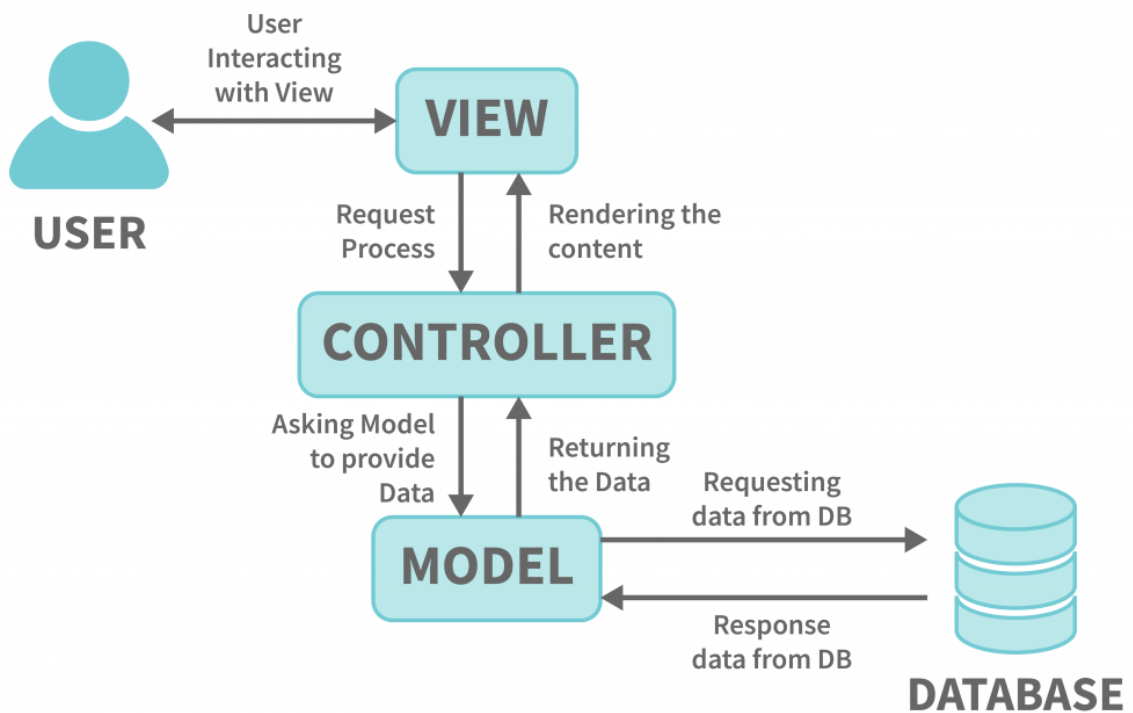


του Spring Boot θα φροντίσει για την ρύθμιση των παραμέτρων της εφαρμογής με βάση τα dependencies που είναι καταχωρημένα στο project.

**3. Data Access Layer:** Τα αντικείμενα πρόσβασης σε δεδομένα (DAOs) και τα repositories βρίσκονται σε αυτό το στρώμα. Το Spring Boot απλοποιεί την πρόσβαση στα δεδομένα παρέχοντας υποστήριξη για διάφορες πηγές δεδομένων, συμπεριλαμβανομένων των σχεσιακών βάσεων δεδομένων, των βάσεων δεδομένων NoSQL και των cloud-based υπηρεσιών αποθήκευσης. Ενσωματώνεται απρόσκοπτα με δημοφιλή frameworks όπως το Hibernate και το JPA.

**4. Configuration Layer:** Το Spring Boot βασίζεται στο convention over configuration (CoC) για την ελαχιστοποίηση της επιβάρυνσης της διαμόρφωσης. Χρησιμοποιεί λογικές προεπιλογές που εφαρμόζονται αυτόματα, εκτός αν παρακαμφθούν. Μπορούν να ρυθμίσουν λεπτομερώς οι ρυθμίσεις (configurations ) χρησιμοποιώντας αρχεία application.properties ή application.yml, μειώνοντας την ανάγκη για εκτεταμένες διαμορφώσεις που βασίζονται σε XML.

#### 5.4 MVC Αρχιτεκτονική



Η εφαρμογή στηρίζεται στην αρχιτεκτονική που απαρτίζεται από Models, Views και Controllers.

Τα **Models** είναι κλάσεις που είναι συνδεδεμένες και επικοινωνούν με τη βάση δεδομένων της εφαρμογής. Αρμοδιότητά τους είναι η ενημέρωση των Views και των Controllers.

Στην παρούσα εφαρμογή έχουμε τα εξής Models:

- **User** : Πρόκειται για το μοντέλο που διαβάζει και αποθηκεύει δεδομένα των χρηστών. Μέσω του μοντέλου αυτού ορίζονται μέθοδοι που επιστρέφουν πληροφορία σε συγκεκριμένη μορφή, καθώς και η σχέση με το μοντέλο των Ραντεβού (appointment) και Διαγνώσεων (diagnosis). Επίσης, ορίζονται validations, όπως την υποχρεωτική ύπαρξη του email και τη μοναδικότητα του A.M.K.A.

- **Doctor** : Πρόκειται για το μοντέλο που διαβάζει και αποθηκεύει τα δεδομένα των γιατρών. Μέσω του μοντέλου αυτού ορίζονται η σχέση με το μοντέλο των χρηστών, ραντεβού και διαγνώσεων.

- **Appointment** : Πρόκειται για το μοντέλο που διαβάζει και αποθηκεύει τα δεδομένα των ραντεβού. Μέσω του μοντέλου αυτού ορίζονται η σχέση με το μοντέλο των χρηστών, γιατρών και διαγνώσεων.

- **Diagnosis** : Πρόκειται για το μοντέλο που διαβάζει και αποθηκεύει δεδομένα των διαγνώσεων. Μέσω του μοντέλου ορίζονται η σχέση με το μοντέλο των χρηστών, γιατρών και διαγνώσεων.

Τα **Views** είναι αρχεία τύπου .html. Σε αυτά τα αρχεία εμπεριέχεται το markup της εκάστοτε σελίδας. Αρμοδιότητά τους είναι η προβολή των δεδομένων της εφαρμογής μέσω γραφικής αναπαράστασης, όπως ενημερώνονται από τα διάφορα μοντέλα που εμπεριέχουν τα δεδομένα αυτά.

Οι **Controllers** είναι κλάσεις που διαχειρίζονται ενέργειες (actions). Οι ενέργειες αυτές μπορεί να δίνουν εντολές στα μοντέλα ως προς τη διαχείριση των δεδομένων, αλλά και στα views ως προς τον τρόπο που θα απεικονιστούν τα δεδομένα αυτά. Οι controllers που υπάρχουν στην παρούσα εφαρμογή είναι οι ακόλουθοι:

- **AuthController**. Πρόκειται για τον controller που αφορά σε ενέργειες σχετικά με την αρχική σελίδα της εφαρμογής.
- **UserController**. Πρόκειται για τον controller που αφορά σε ενέργειες σχετικά με τους χρήστες.

- **DoctorController**. Πρόκειται για τον controller που αφορά σε ενέργειες σχετικά με τους γιατρούς.
- **AdminController**. Πρόκειται για τον controller που αφορά σε ενέργειες σχετικά με τον διαχειριστή.
- **AppointmentController**. Πρόκειται για τον controller που αφορά σε ενέργειες σχετικά με τα ραντεβού.
- **DiagnosisController**. Πρόκειται για τον controller που αφορά σε ενέργειες σχετικά με τις διαγνώσεις.

## 5.5 Spring Boot

Τα κυριότερα συστατικά για τη σωστή δόμηση του κώδικα της εφαρμογής είναι οι Controllers, τα Services και τα Repositories. Όταν υλοποιούμε τα παραπάνω συστατικά χρησιμοποιώντας το Spring Framework, χρησιμοποιούμε ειδικές σημάνσεις (annotations : @) με τις οποίες προσδιορίζουμε συγκεκριμένα το είδος του κάθε συστατικού καθώς και λειτουργίες του. Μια κλάση που φέρει το @Controller δηλώνει στο Spring ότι η συγκεκριμένη κλάση αποτελεί έναν Controller. Αναλόγως τα @Service και @Repository υποδηλώνουν στο Spring ότι οι κλάσεις που τα φέρουν αποτελούν Services ή Repositories αντιστοίχως. Έτσι όταν η Spring εφαρμογή αρχίζει να λειτουργεί και να αρχικοποιεί τα συστατικά της, μπορεί και αναγνωρίζει όλες τις κλάσεις που έχουν δηλωθεί με κάποιο από αυτά τα annotations.

### 5.5.1 Controllers

Ένας Controller είναι μια κλάση που είναι υπεύθυνη για τη διαχείριση των HTTP αιτημάτων από τους πελάτες (συνήθως έναν περιηγητή ιστού), εκτέλεση της απαραίτητης λογικής επεξεργασίας και στη συνέχεια την αποστολή των αντίστοιχων απαντήσεων. Λόγω του ότι μία εφαρμογή μπορεί να εκτελεί εκατοντάδες ή χιλιάδες διαφορετικές λειτουργίες, οι λειτουργίες αυτές ομαδοποιούνται βάσει των κοινών χαρακτηριστικών τους και κάθε ομάδα των λειτουργιών αυτών ανατίθεται σε έναν Controller, ο οποίος και θα είναι ο υπεύθυνος για την εκτέλεση τους. Όταν η εφαρμογή δεχτεί μία αίτηση, θα αναζητήσει να βρει τον Controller στον οποίο πρέπει να προωθήσει την αίτηση. Ο προσδιορισμός του Controller καθορίζεται από το URL της αιτήσεως και από το annotation @GetMapping που βρίσκεται στην δήλωση της κλάσεως του Controller. Η εντολή @GetMapping (value = "/ appointment ") στην δήλωση της κλάσεως ενός Controller σημαίνει ότι ο Controller θα παραλάβει όλες τις αιτήσεις στις οποίες το URL ξεκινάει με το <http://localhost/8080/appointment> (όπου η <http://localhost/8080/> θεωρούμε ότι είναι η βασική διεύθυνση της εφαρμογής και

παραμένει σταθερή και κοινή για όλες τις αιτήσεις που δέχεται). Μετά την παραλαβή μίας αιτήσεως ο Controller αναζητάει ανάμεσα στις μεθόδους του να βρει την κατάλληλη μέθοδο στην οποία θα αναθέσει την εκτέλεση της αιτήσεως. Κάθε μέθοδος χρησιμοποιώντας επίσης το `@GetMapping`, υποδηλώνει ποια αίτηση αναλαμβάνει για να εκτελέσει.

### 5.5.2 Services

Τα Services αναλαμβάνουν την εκτέλεση της επιχειρηματικής λογικής της εφαρμογής. Συνήθως, αποτελούν το μεσαίο στρώμα μεταξύ των Controllers και των Repositories. Ορίζουν και υλοποιούν τις λειτουργίες που απαιτούνται για την επεξεργασία των δεδομένων. Τα Services περιέχουν κλήσεις προς τα Repositories για την αποθήκευση και ανάκτηση δεδομένων τα οποία επιστρέφει στον Controller που το κάλεσε.

### 5.5.3 Repositories

Τα Repositories αναλαμβάνουν την αποθήκευση και την ανάκτηση δεδομένων από τη βάση δεδομένων. Στο Spring Boot framework, η Spring Data JPA παρέχει έναν απλό τρόπο για τη διαχείριση των δεδομένων βάσει του JPA (Java Persistence API). Τα repositories ορίζουν μια διεπαφή, και η Spring Data JPA δημιουργεί τις αναγκαίες υλοποιήσεις για την ανάκτηση και αποθήκευση δεδομένων στη βάση δεδομένων.

### 5.5.4 Annotations

Όπως είπαμε προηγουμένως τα "annotations" (ειδικές σημάνσεις) αναφέρονται σε ετικέτες ή σημάνσεις που προστίθενται στον κώδικα πηγαίου κώδικα (όπως κλάσεις, μέθοδοι, πεδία κ.λπ.) για να παρέχουν πληροφορίες ή οδηγίες στον μεταγλωττιστή (compiler) ή στο πρόγραμμα κατά την εκτέλεση. Στην Java, τα annotations είναι ειδικά σχόλια που ξεκινούν με το σύμβολο "@" και ακολουθεί το όνομα του annotation. Τα annotations παρέχουν πληροφορίες που χρησιμοποιούνται από τον μεταγλωττιστή ή την εκτελούμενη εφαρμογή.

Στο Spring Boot χρησιμοποιούνται ευρέως για την απλοποίηση του κώδικα στις κλάσεις. Ενδεικτικά από τον κώδικα της εφαρμογής παρατίθενται τα παρακάτω screenshots:

```

1 package com.example.clinicsystem.entity;
2
3 import ...
4
5
6
7
8
9
10
11
12
13 @Getter
14 @Setter
15 @NoArgsConstructor
16 @AllArgsConstructor
17 @Entity
18 @Table(name="users")
19 public class User
20 {
21     private static final long serialVersionUID = 1L;
22
23     @Id
24     @GeneratedValue(strategy = GenerationType.IDENTITY)
25     private Long id;
26
27     @Column(name = "first_name", nullable=false)
28     private String firstName;
29
30     @Column(name = "last_name", nullable=false)
31     private String lastName;
32
33     @Column(nullable = false, unique = true)

```

```

@Column(nullable=false)
private String phNo;
@Column()
private Boolean isDoctor;
@Column()
private String doc_specialty;
@Column()
private String status;

@ManyToMany(fetch = FetchType.EAGER, cascade=CascadeType.ALL)
@JoinTable(
    name="users_roles",
    joinColumns={@JoinColumn(name="USER_ID", referencedColumnName="ID")},
    inverseJoinColumns={@JoinColumn(name="ROLE_ID", referencedColumnName="ID")})
private List<Role> roles = new ArrayList<>();

@OneToMany(mappedBy = "user")
private List<Appointment> appointments;
}

```

Ξεκινώντας βλέπουμε τα `@Getter`, `@Setter`, `@NoArgsConstructor`, και `@AllArgsConstructor`. Με αυτά μειώνουμε την ποσότητα του κώδικα που πρέπει να γραφεί χειροκίνητα και διευκολύνεται έτσι η ανάπτυξη και η συντήρηση του κώδικα. Με τη βοήθειά τους, δεν χρειάζεται να γράψουμε χειροκίνητα τους getters, setters και τους constructors, εξοικονομώντας χρόνο και κάνοντας τον κώδικα πιο απλό και να διαβάζεται πιο εύκολα.

- **@Getter** : Προσθέτει τις μεθόδους "getter" (π.χ., `getFirstName()`) για τα αντίστοιχα πεδία.
- **@Setter** : Προσθέτει μεθόδους "setter" (π.χ., `setFirstName(String firstName)`).
- **@NoArgsConstructor** : Δημιουργεί ένα Constructor χωρίς παραμέτρους (default constructor).
- **@AllArgsConstructor** : Δημιουργεί έναν Constructor που δέχεται όλα τα πεδία της κλάσης ως παραμέτρους που είναι χρήσιμος για την αρχικοποίηση αντικειμένων με όλα τα πεδία.

Ανάλυση των υπολοίπων:

- **@Entity** : Υποδηλώνει ότι η κλάση `User` είναι μια JPA (Java Persistence API) οντότητα. Αυτό σημαίνει ότι η κλάση αντιπροσωπεύει ένα πίνακα στη βάση δεδομένων. Στην περίπτωσή σας, ο πίνακας ονομάζεται "users."
- **@Table(name="users")** : Το `@Table` annotation καθορίζει το όνομα του πίνακα στη βάση δεδομένων που αντιστοιχεί σε αυτήν την οντότητα. Στην περίπτωσή μας, ο πίνακας ονομάζεται "users."
- **@Id** : Το `@Id` annotation υποδηλώνει ότι το πεδίο "id" είναι το πρωτεύον κλειδί της οντότητας, που χρησιμοποιείται για τη μοναδική ταυτοποίηση κάθε εγγραφής.
- **@GeneratedValue(strategy = GenerationType.IDENTITY)** : Αυτό το annotation σε συνδυασμό με τη στρατηγική "GenerationType.IDENTITY" υποδηλώνει ότι οι τιμές του πεδίου "id" θα δημιουργηθούν αυτόματα από τη βάση δεδομένων με αυξανόμενη αρίθμηση.
- **@Column** : Το `@Column` annotation χρησιμοποιείται για τον καθορισμό των χαρακτηριστικών των στηλών του πίνακα στη βάση δεδομένων. Με αυτό το annotation καθορίζουμε το όνομα της στήλης, εάν είναι μοναδική (unique) και αν είναι υποχρεωτική (nullable).
- **@ManyToMany** και **@JoinTable** : Αυτά τα annotations χρησιμοποιούνται για την περιγραφή μιας σχέσης πολλών-προς-πολλά (Many-to-Many) μεταξύ του χρήστη και των ρόλων. Ο χρήστης μπορεί να έχει πολλούς ρόλους, και οι

ρόλοι μπορεί να ανήκουν σε πολλούς χρήστες. Το `@JoinTable` annotation καθορίζει τον πίνακα ενδιάμεσης σχέσης που συνδέει τον πίνακα "users" με τον πίνακα "roles."

- **`@OneToMany(mappedBy = "user")`** : Αυτό το annotation υποδηλώνει μια σχέση "Ένας προς Πολλά" μεταξύ του χρήστη και των ραντεβού (appointments). Κάθε χρήστης μπορεί να έχει πολλά ραντεβού, και το πεδίο "user" στην κλάση "Appointment" χρησιμοποιείται για τη σύνδεση με τον αντίστοιχο χρήστη.

Χαρακτηριστική χρήση των `@annotations` γίνεται και στις υπόλοιπες κλάσεις και για τη συσχέτιση μεταξύ των οντοτήτων.

## Entity Doctor

```
package com.example.clinicsystem.entity;

import ...

@Getter
@Setter
@Entity
@Table(name = "doctor")
public class Doctor {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id", nullable = false, unique = true)
    private Long id;

    @Column(name = "specialty")
    private String specialty;

    @OneToOne(cascade = CascadeType.ALL)
    @MapsId
    @JoinColumn(name = "user_id", referencedColumnName = "id")
    private User user;

    @OneToMany(fetch = FetchType.LAZY, mappedBy = "doctor")
    private List<Appointment> appointments;
}
```

## Entity Appointment

```

package com.example.clinicsystem.entity;

import ...

@Getter
@Setter
@NoArgsConstructor
@AllArgsConstructor
@Entity
@Table(name = "appointment")
public class Appointment {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id", nullable = false)
    private Long id;

    @Temporal(TemporalType.DATE)
    @Column(name = "app_date")
    private Date app_date;

    @Column(name = "doc_specialty")
    private String doc_specialty;

    @Column(name = "reason")
    private String reason;

    @Column(name = "reason")
    private String reason;

    @Column()
    private Long doctor_id;

    @Column(name = "user_id")
    private Long userId;

    @Column(name = "status", columnDefinition = "varchar(20) default 'ΕΚΚΡΕΜΕΙ'")
    private String status;

    @ManyToOne(fetch=FetchType.LAZY)
    @JoinColumn(name = "doctor_id", insertable = false, updatable = false)
    private Doctor doctor;

    @ManyToOne
    @JoinColumn(name = "user_id", insertable = false, updatable = false)
    private User user;

    @OneToOne(mappedBy = "appointment")
    private Diagnosis diagnosis;

```



## Entity Diagnosis

```
package com.example.clinicsystem.entity;

import ...

31 usages  👤 Yannis
@Getter
@Setter
@Entity
@Table(name = "diagnosis")
public class Diagnosis {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id", nullable = false, unique = true)
    private Long id;

    @Temporal(TemporalType.DATE)
    @Column(name = "date")
    private Date date;

    @Column(name = "result")
    private String result;

    @Column(name = "treatment")
    private String treatment;

    @Column(name = "file_url")
    private String fileUrl;

    @OneToOne
    @JoinColumn(name = "appointment_id")
    private Appointment appointment;

    @ManyToOne
    @JoinColumn(name = "medicine_id")
    private Medicine medicine;

    @ManyToOne
    @JoinColumn(name = "doctor_id")
    private User doctor;

    @ManyToOne
    @JoinColumn(name = "patient_id")
    private User patient;
}
```

## 5.6 Frontend

Για το frontend, το μέρος δηλαδή της εφαρμογής που βλέπει ο χρήστης και αλληλεπιδρά μαζί του, χρησιμοποιήθηκε το Thymeleaf και το Bootstrap. Αυτό το μέρος αντιπροσωπεύει τη γραφική διεπαφή της εφαρμογής και τον τρόπο που οι χρήστες αλληλεπιδρούν με αυτήν.

Το Thymeleaf είναι ένα πρότυπο συστήματος προτεραιότητας για το HTML και το XML που χρησιμοποιείται κυρίως στην ανάπτυξη ιστοσελίδων και διαδικτυακών εφαρμογών με τη γλώσσα προγραμματισμού Java. Χρησιμοποιείται για την ενσωμάτωση δυναμικών δεδομένων στις σελίδες HTML, επιτρέποντας την απόδοση δεδομένων από την εφαρμογή στον κώδικα της ιστοσελίδας.

Το Bootstrap, από την άλλη πλευρά, είναι ένα ανοιχτού κώδικα πλαίσιο προσκαλούμενο για τον σχεδιασμό και την ανάπτυξη ενός ανταποκριτικού και επαγγελματικού frontend. Περιλαμβάνει προσαρμοσμένα CSS, JavaScript plugins, και άλλα συστατικά για να δημιουργηθούν εύκολα και γρήγορα μοντέρνες και ανταποκριτικές ιστοσελίδες.

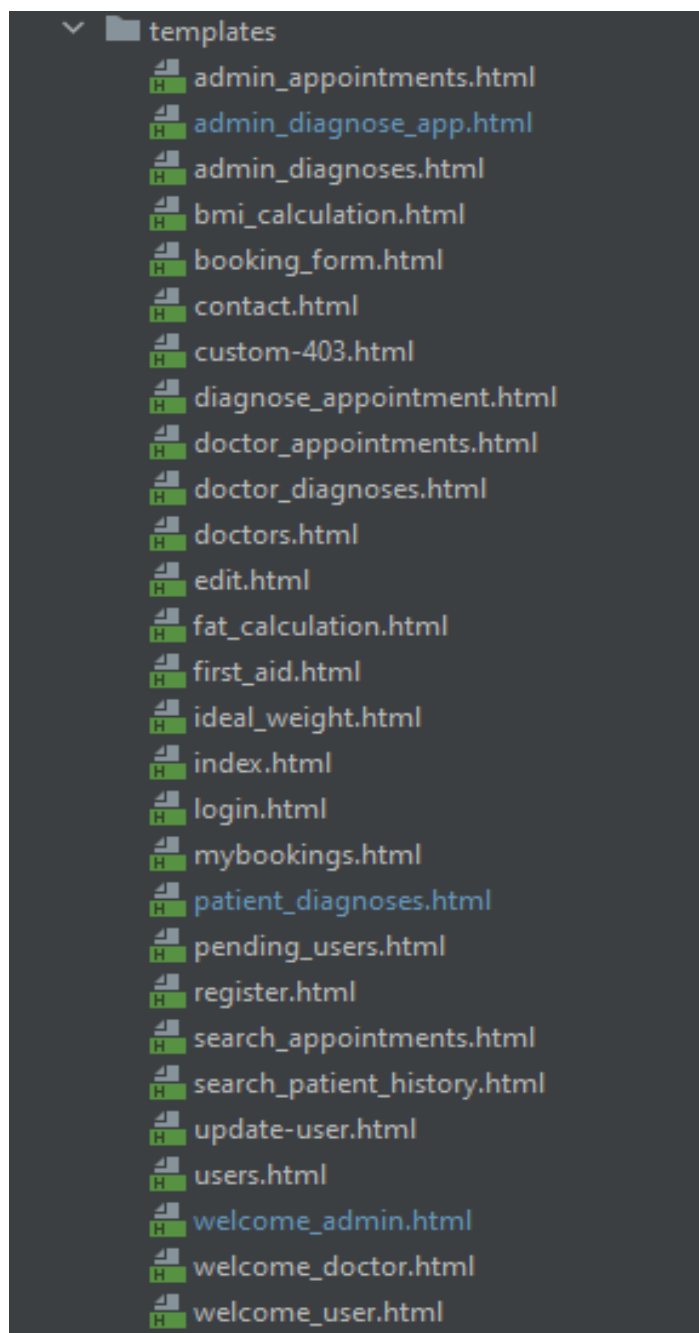
Το Thymeleaf βοηθά στην απόδοση δυναμικών δεδομένων, ενώ το Bootstrap παρέχει προσεγμένο σχεδιασμό και εύκολη χρήση για τη δημιουργία μιας όμορφης και λειτουργικής εμπειρίας χρήστη. Με τη συνδυασμένη χρήση Thymeleaf και Bootstrap μας δίνεται η δυνατότητα για γρήγορη ανάπτυξη εφαρμογών με εξαιρετική εμφάνιση και απόδοση.

Τα αρχεία που χρησιμοποιήθηκαν έχουν την κατάληξη .html και βρίσκονται στον φάκελο templates.

- login: Η αρχική σελίδα της εφαρμογής . Ο χρήστης χρειάζεται προσωπικό όνομα χρήστη και κωδικό για να εισέλθει στο σύστημα.
- register: Η φόρμα εγγραφής και καταχώρησης στη βάση δεδομένων ενός χρήστη για πρώτη φορά που θα κάνει είσοδο στην εφαρμογή.
- welcome\_user: Η κεντρική σελίδα του χρήστη με ρόλο PATIENT. Η πρόσβαση εδώ έχει πραγματοποιηθεί μετά την επιτυχή είσοδο με διαπιστευτήρια στο login.
- update\_user: Η σελίδα για την επεξεργασία του προφίλ του χρήστη.
- booking\_form: Η φόρμα καταχώρησης νέου ραντεβού προς εξέταση από τον ασθενή.

- mybookings: Η σελίδα με το σύνολο των ραντεβού του ασθενή, προγραμματισμένων και ολοκληρωμένων.
- patient\_diagnoses: Η σελίδα με το σύνολο των διαγνώσεων από τα ραντεβού του ασθενή που έχουν ολοκληρωθεί.
- bmi\_calculation: Φόρμα συμπλήρωσης σωματομετρικών στοιχείων ασθενή για εξαγωγή Δείκτη Μάζας Σώματος του.
- ideal\_weight: Φόρμα συμπλήρωσης σωματομετρικών στοιχείων ασθενή για υπολογισμό του ιδανικού του βάρους.
- fat\_calculation: Φόρμα συμπλήρωσης σωματομετρικών στοιχείων ασθενή για υπολογισμό λίπους του ασθενή.
- first\_aid: Η σελίδα με πληροφορίες για τον τρόπο παροχής Ά βοηθειών.
- contact: Η σελίδα με πληροφορίες επικοινωνίας με το νοσοκομείο.
- welcome\_doctor: Η κεντρική σελίδα του χρήστη με ρόλο DOCTOR. Η πρόσβαση εδώ έχει πραγματοποιηθεί μετά την επιτυχή είσοδο με διαπιστευτήρια στο login.
- doctor\_appointments: Η σελίδα με το σύνολο των ραντεβού του ιατρού, προγραμματισμένων και ολοκληρωμένων. Στα ραντεβού που δεν έχουν ολοκληρωθεί ο γιατρός μπορεί να κάνει διάγνωση ώστε να ολοκληρωθούν.
- doctor\_diagnoses: : Η σελίδα με το σύνολο των διαγνώσεων από τα ραντεβού του ασθενή που έχουν ολοκληρωθεί.
- diagnose\_appointment: Η σελίδα στην οποία ο γιατρός μπορεί να κάνει διάγνωση για το ραντεβού που έχει επιλέξει.
- search\_patient\_history: Η σελίδα στην οποία ο γιατρός μπορεί να κάνει αναζήτηση του ιστορικού των ραντεβού ενός ασθενή με βάση τον ΑΜΚΑ του ασθενή.
- doctors: Η σελίδα για την αναζήτηση όλων των ιατρών ανά ειδικότητα.
- welcome\_admin: Η κεντρική σελίδα του χρήστη με ρόλο ADMIN. Η πρόσβαση εδώ έχει πραγματοποιηθεί μετά την επιτυχή είσοδο με διαπιστευτήρια στο login.
- users: Η σελίδα που ο διαχειριστής της εφαρμογής μπορεί να δει και να επεξεργαστεί ή να διαγράψει όλους τους χρήστες.
- pending\_users: Η σελίδα που ο διαχειριστής της εφαρμογής μπορεί να δει και να εγκρίνει τους γιατρούς που έχουν κάνει εγγραφή στην εφαρμογή.
- admin\_appointments: Η σελίδα που ο διαχειριστής της εφαρμογής μπορεί να δει όλα τα ραντεβού που έχουν προγραμματιστεί και ολοκληρωθεί όλων των ασθενών.
- admin\_diagnoses: Η σελίδα που ο διαχειριστής της εφαρμογής μπορεί να δει όλες τις διαγνώσεις όλων των ασθενών.

- `admin_diagnose_app`: Η σελίδα που ο διαχειριστής της εφαρμογής μπορεί να επεξεργαστεί τη διάγνωση για κάποιο συγκεκριμένο ασθενή αλλά και για συγκεκριμένο γιατρό που έχει αναλάβει το ραντεβού.
- `custom-403`: Η σελίδα που εμφανίζεται σε περίπτωση που κάποιος χρήστης δεν έχει δικαίωμα να προβεί σε κάποια ενέργεια.



## 6. Είσοδος στην Εφαρμογή

Τα διαπιστευτήρια για την βάση δεδομένων βρίσκονται στο αρχείο ***application.properties***. Απαιτείται η δημιουργία βάσης με το όνομα `login_system` στο ΣΔΒΔ. Οι πίνακες δημιουργούνται αυτόματα με την 1<sup>η</sup> εκτέλεση της εφαρμογής.

Ο σύνδεσμος για την πλοήγηση είναι: <http://localhost:8080/login>.

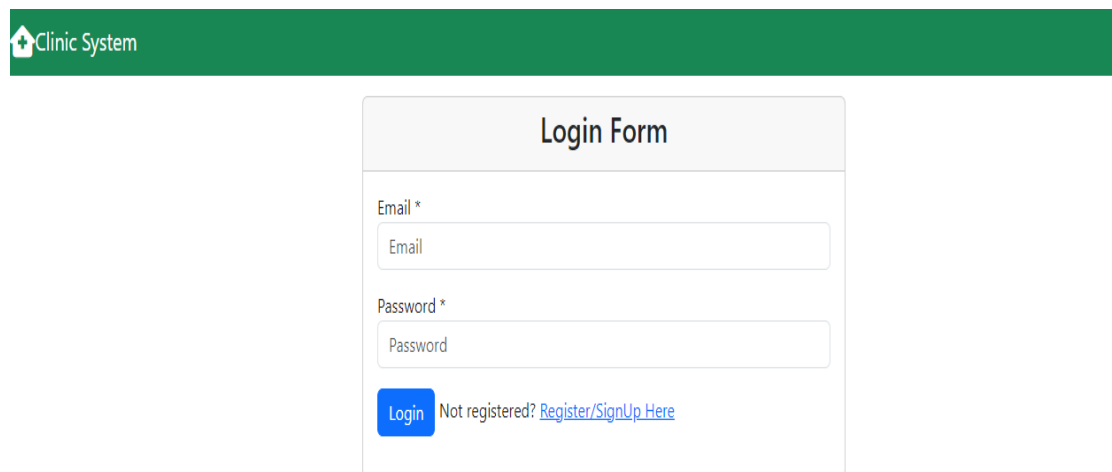
Τα διαπιστευτήρια του Διαχειριστή είναι:

**Username:** yanniskoutras@hotmail.com

**Password:** 1111

## 7. Παρουσίαση Εφαρμογής

### 7.1 Είσοδος ή Εγγραφή



The screenshot shows the 'Clinic System' header in a green bar. Below it is a 'Login Form' box with a light gray header. The form contains two input fields: 'Email \*' and 'Password \*'. Below the password field is a blue 'Login' button and a link that says 'Not registered? [Register/SignUp Here](#)'.

Σε περίπτωση που πρόκειται για πρώτη είσοδο και χρειάζεται εγγραφή ο χρήστης επιλέγει τον σύνδεσμο [Register/SignUp Here](#).

Εγγραφή Χρήστη

Όνομα  
Εισάγετε Όνομα

Επώνυμο  
Εισάγετε Επώνυμο

Email  
Εισάγετε address

Password  
Εισάγετε password

AMKA  
Εισάγετε τον AMKA

Αρ. Τηλεφώνου  
Εισάγετε Αρ. Τηλεφώνου

\*Τα παρακάτω πεδία αφορούν ΜΟΝΟ αν κάποιος είναι γιατρός  
 Γιατρός  
 Όχι Γιατρός

Επιλέξτε Ειδικότητα

Εγγραφή Είσαι ήδη εγγεγραμμένος? [Είσοδος εδώ](#)

Στην περίπτωση που ο χρήστης εισάγει ένα email το οποίο είναι ήδη καταχωρημένο στη Βάση Δεδομένων, θα εμφανιστεί το κατάλληλο μήνυμα όταν πάει να ολοκληρώσει τη διαδικασία της εγγραφής του.

Εγγραφή Χρήστη

Όνομα  
ΙΩΑΝΝΗΣ

Επώνυμο  
ΚΟΥΤΡΑΣ

Email  
yannis@gmail.com  
Υπάρχει ήδη λογαριασμός με αυτό το email

Password  
Εισάγετε password

AMKA  
999999

Αρ. Τηλεφώνου  
1234567

\*Τα παρακάτω πεδία αφορούν ΜΟΝΟ αν κάποιος είναι γιατρός  
 Γιατρός  
 Όχι Γιατρός

Επιλέξτε Ειδικότητα

Εγγραφή Είσαι ήδη εγγεγραμμένος? [Είσοδος εδώ](#)

Το ίδιο θα συμβεί και όταν ο χρήστης πάει να καταχωρήσει έναν ΑΜΚΑ που υπάρχει ήδη στη Βάση Δεδομένων της εφαρμογής.

### Εγγραφή Χρήστη

Όνομα

Επώνυμο

Email

Password

ΑΜΚΑ  
  
Υπάρχει ήδη χρήστης με το ίδιο ΑΜΚΑ

Αρ. Τηλεφώνου

\*Τα παρακάτω πεδία αφορούν ΜΟΝΟ αν κάποιος είναι γιατρός

Γιατρός  
 Όχι Γιατρός

Είσαι ήδη εγγεγραμμένος? [Είσοδος εδώ](#)

Με την επιλογή του πεδίου «Γιατρός» ένα μήνυμα υποχρεωτικότητας επιλογής **Ειδικότητας** εμφανίζεται πάνω από το πεδίο. Σε περίπτωση που δεν συμπληρωθεί δεν μπορεί να ολοκληρωθεί η διαδικασία της εγγραφής.



### Εγγραφή Χρήστη

Όνομα  
ΙΩΑΝΝΗΣ

Επώνυμο  
ΚΟΥΤΡΑΣ

Email  
yanniskoutras@hotmail.com

Password  
\*\*\*\*


ΑΜΚΑ  
999999  
Υπάρχει ήδη χρήστης με το ίδιο ΑΜΚΑ

Αρ. Τηλεφώνου  
2345678

\*Τα παρακάτω πεδία αφορούν ΜΟΝΟ αν κάποιος είναι γιατρός

Γιατρός  
 Όχι Γιατρός

Επιλέξτε Ειδικότητα

**Εγγραφή** Είσαι ήδη εγγεγραμμένος?  Επιλέξτε ένα στοιχείο από τη λίστα.

Αντίστοιχα μηνύματα λάθους εμφανίζονται και στην οθόνη Εισόδου του χρήστη σε περίπτωση συμπλήρωσης λανθασμένων διαπιστευτηρίων.

Invalid Email and Password.

### Login Form

Email \*

Email

Password \*

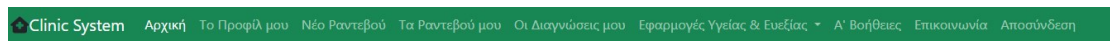
Password

**Login**

Not registered? [Register/Sign Up Here](#)

## 7.2 Είσοδος Ασθενή

Με την επιτυχή είσοδό του στην Εφαρμογή ο ασθενής μπορεί να δει στην αρχική του σελίδα ένα ονομαστικό μήνυμα καλωσορίσματος καθώς και το μενού και τις επιλογές που έχει. Πιο αναλυτικά μπορεί να δει και να επεξεργαστεί τα στοιχεία του προφίλ του, να προγραμματίσει νέο ραντεβού, να δει όλα τα ραντεβού του (αυτά που είναι προγραμματισμένα αλλά και αυτά που έχουν ολοκληρωθεί), να δει τις διαγνώσεις του, να δοκιμάσει μετρήσεις του σε εφαρμογές υγείας και ευεξίας, να ενημερωθεί για τρόπο παροχής Ά βοηθειών και τέλος να δει τα στοιχεία επικοινωνίας του νοσοκομείου.

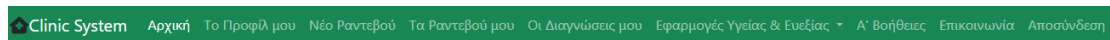


Καλωσήρθες και πάλι ΦΙΛΙΠΠΟΣ ΝΑΚΑΣ

Copyright © Koutras Ioannis-Prodromos

### 7.2.1 Το Προφίλ μου

Επιλέγοντας «**Το Προφίλ μου**» ο ασθενής μπορεί να δει και να επεξεργαστεί τα στοιχεία που είχε καταχωρήσει κατά την εγγραφή του.



### Επεξεργασία Προφίλ

Όνομα:

Επώνυμο:

Email:

ΑΜΚΑ:

Αρ. Τηλεφώνου:

[Αποθήκευση](#)

### 7.2.2 Νέο Ραντεβού

Επιλέγοντας «**Νέο Ραντεβού**» ο ασθενής μπορεί να προγραμματίσει εξέταση στο ιατρείο που θα επιλέξει. Ανάλογα με την ειδικότητα ιατρού, που ενδιαφέρεται να προγραμματίσει ραντεβού και θα επιλέξει, εμφανίζονται στο επόμενο πεδίο μόνο οι γιατροί αυτής της ειδικότητας ώστε να επιλέξει συγκεκριμένο γιατρό που θέλει να τον εξετάσει. Αφού συμπληρώσει την ημερομηνία και την αιτία που θέλει να προγραμματίσει το συγκεκριμένο ραντεβού, με το κουμπί **Κράτηση** ολοκληρώνεται το ραντεβού με αντίστοιχο μήνυμα επιτυχίας κράτησης.

The screenshot shows the 'Φόρμα Κράτησης' (Appointment Form) interface. At the top, there is a navigation bar with the following items: Clinic System, Αρχική, Το Προφίλ μου, Νέο Ραντεβού, Τα Ραντεβού μου, Οι Διαγνώσεις μου, Εφαρμογές Υγείας & Ευεξίας, Α' Βοήθειες, Επικοινωνία, Αποσύνδεση. The form itself has a title 'Φόρμα Κράτησης' and contains the following fields:

- Ημερομηνία Κράτησης:** A date picker field with the format 'ηη/μμ/εεεε'.
- Ειδικότητα Γιατρού:** A dropdown menu with the label 'Επιλέξτε ειδικότητα'.
- Επιλέξτε γιατρό:** A dropdown menu with the label 'Επιλέξτε γιατρό'.
- Αιτία Κράτησης:** A text area for entering the reason for the appointment.
- Κράτηση:** A blue button to submit the appointment.

### 7.2.3 Τα Ραντεβού μου

Στην καρτέλα «**Τα Ραντεβού μου**» ο ασθενής μπορεί να δει όλα τα ραντεβού που έχει προγραμματίσει είτε αυτά που έχει ολοκληρώσει είτε αυτά που βρίσκονται σε εξέλιξη. Μπορεί επίσης να ακυρώσει ένα προγραμματισμένο ραντεβού.

## Τα Ραντεβού μου

Όνομ/νυμο:  
ΝΑΚΑΣ ΦΙΛΙΠΠΟΣ

Ραντεβού Νο	Ημερ/νία	Ειδικότητα Ιατρού	Όνομα Ιατρού	Λόγος Κράτησης	Κατάσταση	Ενέργειες
39	2023-10-05	ΠΑΘΟΛΟΓΟΣ	ΚΑΤΕΡΙΝΑ ΑΛΙΣΑΒΑΚΗ	ΞΗΡΟΛΑΙΜΙΑ	ΕΚΚΡΕΜΕΙ	ΑΚΥΡΩΣΗ
40	2023-10-05	ΟΔΟΝΤΙΑΤΡΟΣ	ΜΑΡΙΑ ΜΑΚΡΙΔΟΥ	ΣΙΔΕΡΑΚΙΑ	ΕΚΚΡΕΜΕΙ	ΑΚΥΡΩΣΗ
43	2023-09-30	ΟΦΘΑΛΜΙΑΤΡΟΣ	ΖΩΗ ΓΙΤΣΑ	ΣΥΝΤΑΓΗ ΓΥΑΛΙΩΝ	ΟΛΟΚΛΗΡΩΜΕΝΟ	

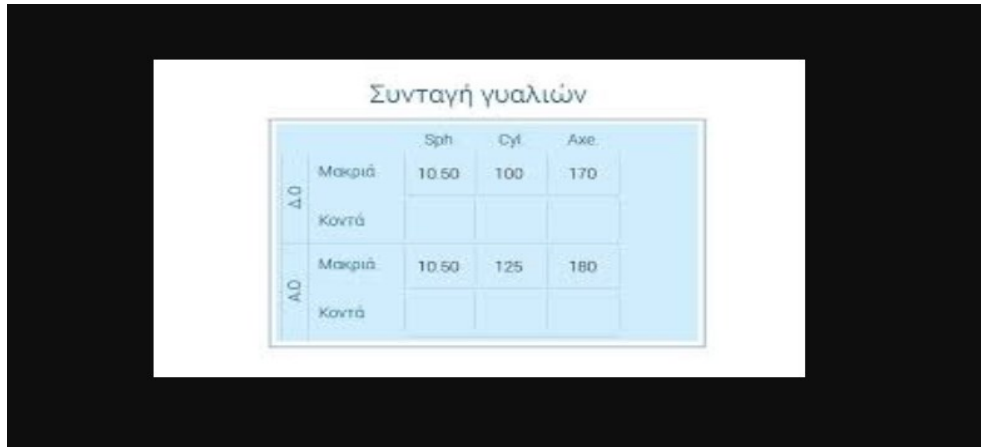
### 7.2.4 Οι Διαγνώσεις μου

Στην καρτέλα «**Οι Διαγνώσεις μου**» ο ασθενής μπορεί να δει τις διαγνώσεις από τα ραντεβού που έχει πραγματοποιήσει. Στις διαγνώσεις μπορεί να δει πληροφορίες όπως την ημερ/νία της εξέτασης τον γιατρό, το πόρισμα της εξέτασης, τα φάρμακα που χορηγήθηκαν καθώς και οδηγίες-κατευθύνσεις. Σε διαγνώσεις που υπάρχει και κάποιο αποτέλεσμα εξετάσεων (πχ μια ακτινογραφία, ένα καρδιογράφημα, οι αιματολογικές εξετάσεις κ.ά.) εμφανίζεται η επιλογή «**Άνοιγμα Αρχείου**» όπου εμφανίζει το συνημμένο αρχείο.

## Οι Διαγνώσεις μου

Όνομ/νυμο:  
ΝΑΚΑΣ ΦΙΛΙΠΠΟΣ

Ημερ/νία	Ιατρός	Ειδικότητα	Διάγνωση	Θεραπεία	Φάρμακα	Αρχείο
2023-09-30	ΖΩΗ ΓΙΤΣΑ	ΟΦΘΑΛΜΙΑΤΡΟΣ	ΜΥΩΠΙΑ ΔΕ 1,25 ΑΡ 1,00	ΣΥΝΤΑΓΟΓΡΑΦΗΣΗ ΓΥΑΛΙΩΝ	-	Άνοιγμα Αρχείου



### 7.2.5 Εφαρμογές Υγείας & Ευεξίας

- **Υπολογισμός Δείκτη Μάζας Σώματος (ΔΜΣ)**

Στην καρτέλα «Υπολογισμός ΔΜΣ» ο ασθενής μπορεί να εισάγει τα σωματομετρικά του στοιχεία και να υπολογίσει τον ΔΜΣ του το οποίο θα εμφανιστεί στο κάτω μέρος της οθόνης του.

Υπολογισμός ΔΜΣ

Φύλο:

Βάρος (kg):

Ύψος (cm):

Ηλικία:

Ο ΔΜΣ σας είναι: 24.84 (Κανονικό Βάρος)

- **Υπολογισμός Ιδανικού Βάρους**

Στην καρτέλα «Υπολογισμός Ιδανικού Βάρους» ο ασθενής με τον ίδιο τρόπο μπορεί να ενημερωθεί για το πού κυμαίνεται το φυσιολογικό για τον ίδιο βάρος.

Clinic System Αρχική Το Προφίλ μου Νέο Ραντεβού Τα Ραντεβού μου Οι Διαγνώσεις μου Εφαρμογές Υγείας & Ευεξίας Α' Βοήθειες Επικοινωνία Αποσύνδεση

### Υπολογισμός Ιδανικού Βάρους

Υψος (cm):  
185

Φύλο:  
Ανδρας

Υπολογισμός

Το ιδανικό βάρος για εσάς είναι: 76.25 (± 10%) kg

*Φροντίστε να εξισορροπήσετε την διατροφή σας και την σωματική σας άσκηση ώστε να έλθετε σε αυτά τα κλά.*

Copyright © Koutras Ioannis-Prodrimos

- **Υπολογισμός Λίπους Σώματος**

Στην καρτέλα «Υπολογισμός Λίπους Σώματος» ο ασθενής μπορεί να ενημερωθεί για το ποσοστό λίπους του και για το ποια κατηγορία ανήκει (πχ Παχύσαρκος, Κανονικός κλπ).

Clinic System Αρχική Το Προφίλ μου Νέο Ραντεβού Τα Ραντεβού μου Οι Διαγνώσεις μου Εφαρμογές Υγείας & Ευεξίας Α' Βοήθειες Επικοινωνία Αποσύνδεση

### Υπολογισμός Λίπους Σώματος

Φύλο:  
Ανδρας

Υψος (cm):  
185

Περιφέρεια μέσης (cm):  
100

Περιφέρεια ισχίου (cm):  
70

Περιφέρεια λαγυμού (cm):  
30

Υπολογισμός

Το ποσοστό λίπους σας είναι: 36.66%  
Κατηγορία Παχυσαρκία (>25%)

## 7.2.6 Α Βοήθειες


Στην καρτέλα «Α Βοήθειες» ο χρήστης μπορεί να ενημερωθεί μέσω μιας παρουσίασης για τον τρόπο παροχής Α βοηθειών σε περίπτωση που κληθεί να αντιμετωπίσει ένα τέτοιο περιστατικό.

Clinic System Αρχική Το Προφίλ μου Νέο Ραντεβού Τα Ραντεβού μου Οι Διαγνώσεις μου Εφαρμογές Υγείας & Ευεξίας Α' Βοήθειες Επικοινωνία Αποσύνδεση

**Πρώτες βοήθειες**

Παρακαλούμε καλέστε στο 166 αν έχετε πόνους στον θώρακα, δυσκολία στην αναπνοή, ακατάσχετη αιμορραγία, ξαφνική αδυναμία, μούδιασμα ή νομίζετε ότι η κατάσταση σας χρήζει άμεσης ιατρικής βοήθειας

**ΣΥΝΕΧΙΣΕ ΚΑΑ**



The illustration shows two steps of CPR. On the left, a person is performing chest compressions on another person lying on their back. The number '30' is written below this step. On the right, a person is performing rescue breaths on another person lying on their back. The number '2' is written below this step. The text 'ΣΥΝΕΧΙΣΕ ΚΑΑ' is written above the illustrations.

### 7.2.7 Επικοινωνία

Στην καρτέλα «Επικοινωνία» μπορεί να βρει τα στοιχεία επικοινωνίας του Νοσοκομείου.

Clinic System Αρχική Το Προφίλ μου Νέο Ραντεβού Τα Ραντεβού μου Οι Διαγνώσεις μου Εφαρμογές Υγείας & Ευεξίας Α' Βοήθειες Επικοινωνία Αποσύνδεση

**Επικοινωνήστε μαζί μας**

Τηλεφωνικό Κέντρο: +2101111111  
 Τηλεφωνικό Ραντεβού: +2102222222  
 Ηλεκτρονικό Ταχυδρομείο: [clinic@gmail.com](mailto:clinic@gmail.com)  
 Διεύθυνση: Πλατεία Κολιάτσου 85, 15345, Αθήνα

### 7.3 Είσοδος Ιατρού

Στην περίπτωση που ένας χρήστης εγγραφεί σαν Ιατρός, η διαδικασία της εγγραφής θα ολοκληρωθεί μόνο αν εγκριθεί από τον Διαχειριστή της εφαρμογής. Σε διαφορετική περίπτωση όταν πάει να κάνει είσοδο με τα διαπιστευτήρια που δήλωσε, θα εμφανίζεται σε αυτόν μήνυμα απαγόρευσης εισόδου.

Συγγνώμη δεν έχετε δικαίωμα πρόσβασης σε αυτή τη σελίδα... :(  
Παρακαλώ απευθυνθείτε στον Διαχειριστή για επίλυση του προβλήματος.

Αφού περάσει από το στάδιο της έγκρισης ο γιατρός μπορεί να εισέλθει πλέον στην εφαρμογή. Στην Αρχική Σελίδα του θα εμφανιστεί ένα ονομαστικό μήνυμα καλωσορίσματος καθώς και το μενού και τις επιλογές που έχει. Πιο αναλυτικά μπορεί να δει τα ραντεβού του χρονολογικά και αυτά που έχουν ολοκληρωθεί και τα νέα που έχουν προγραμματιστεί, να δει τις διαγνώσεις των ασθενών του, να προσθέσει διάγνωση και αποτελέσματα εξετάσεων στους ασθενείς που έχει εξετάσει και τέλος να αναζητήσει ανά ειδικότητα άλλους γιατρούς του νοσοκομείου.



Καλωσήρθες και πάλι DOCTOR MARIA MAKRIDOU

Copyright © Koutras Ioannis-Prodrimos

### 7.3.1 Ραντεβού

Επιλέγοντας «**Ραντεβού**» ο γιατρός μπορεί να δει τα προγραμματισμένα του ραντεβού και τα ραντεβού που έχουν ολοκληρωθεί. Επιλέγοντας «**Διάγνωση**» μπορεί να προσθέσει την διάγνωση από την επίσκεψη καθώς και τα αποτελέσματα από τυχόν

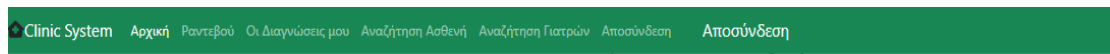


εξετάσεις του ασθενούς ώστε να ενημερωθεί ο ασθενής και να διατηρεί ιατρικό αρχείο ο γιατρός.



## Τα Ραντεβού μου

A/A	Ημερ/νία	Επώνυμο	Όνομα	Λόγος Επίσκεψης	Κατάσταση	Ενέργειες
34	2023-09-30	ΠΑΠΑΔΟΠΟΥΛΟΣ	ΚΩΝΣΤΑΝΤΙΝΟΣ	ΣΦΡΑΓΙΣΜΑ	ΟΛΟΚΛΗΡΩΜΕΝΟ	<a href="#">Διάγνωση</a>
36	2023-09-30	ΠΑΠΑΔΟΠΟΥΛΟΣ	ΚΩΝΣΤΑΝΤΙΝΟΣ	ΚΑΘΑΡΙΣΜΟΣ	ΟΛΟΚΛΗΡΩΜΕΝΟ	<a href="#">Διάγνωση</a>
40	2023-10-05	ΝΑΚΑΣ	ΦΙΛΙΠΠΟΣ	ΣΙΔΕΡΑΚΙΑ	ΕΚΚΡΕΜΕΙ	<a href="#">Διάγνωση</a>



## Διάγνωση Ραντεβού

Ημερ/νία: 2023-10-05

Ασθενής: ΦΙΛΙΠΠΟΣ ΝΑΚΑΣ

Result:

Treatment:

Επιλέξτε Φάρμακο:

Επιλέξτε Φάρμακο

Ανέβασμα Αρχείου:  Δεν επιλέχθηκε κανένα αρχείο.

[Αποθήκευση Διάγνωσης](#)

### 7.3.2 Οι διαγνώσεις μου

Επιλέγοντας «**Οι Διαγνώσεις μου**» ο γιατρός μπορεί να δει τις διαγνώσεις που έχει κάνει για τα ραντεβού που έχουν ολοκληρωθεί καθώς και συνημμένα αρχεία αποτελεσμάτων των εξετάσεων ή να τα προσθέσει μετέπειτα.

## Οι Διαγνώσεις μου

ID	Ημερ/νία	Ασθενής	Διάγνωση	Θεραπεία	Φάρμακο	Αρχείο
42	2023-09-30	ΚΩΝΣΤΑΝΤΙΝΟΣ ΠΑΠΑΔΟΠΟΥΛΟΣ	ΜΥΩΠΙΑ	ΣΥΝΤΑΓΗ ΟΠΤΙΚΩΝ ΟΡΑΣΕΩΣ	SAGILIA	<a href="#">Άνοιγμα Αρχείου</a>
43	2023-09-30	ΚΩΝΣΤΑΝΤΙΝΟΣ ΠΑΠΑΔΟΠΟΥΛΟΣ	ΕΜΦΑΝΙΣΗ ΠΕΤΡΑΣ	ΚΑΘΑΡΙΣΜΟΣ	ALGOFREN	

### 7.3.3 Αναζήτηση Ασθενή

Επιλέγοντας «**Αναζήτηση Ασθενή**» ο γιατρός μπορεί να εισάγει τον ΑΜΚΑ ενός ασθενή και να αναζητήσει το ιστορικό του από προηγούμενες επισκέψεις που είχε και τί φαρμακευτική αγωγή του είχε χορηγηθεί.

### Αναζήτηση Ιστορικού Ασθενή

Εισάγετε ΑΜΚΑ για αναζήτηση: ΑΜΚΑ:

[Αναζήτηση](#)

Ραντεβού Νο	Ημερ/νία	Όνομα Ασθενή	Διάγνωση	Θεραπεία	Φάρμακα
34	2023-09-30	ΚΩΝΣΤΑΝΤΙΝΟΣ ΠΑΠΑΔΟΠΟΥΛΟΣ	ΜΥΩΠΙΑ	ΣΥΝΤΑΓΗ ΟΠΤΙΚΩΝ ΟΡΑΣΕΩΣ	SAGILIA
36	2023-09-30	ΚΩΝΣΤΑΝΤΙΝΟΣ ΠΑΠΑΔΟΠΟΥΛΟΣ	ΕΜΦΑΝΙΣΗ ΠΕΤΡΑΣ	ΚΑΘΑΡΙΣΜΟΣ	ALGOFREN

### 7.3.4 Αναζήτηση Γιατρών

Επιλέγοντας «**Αναζήτηση Γιατρών**» ο γιατρός μπορεί να ανατρέξει και να αναζητήσει κάποιον άλλο ιατρό του νοσοκομείου αναλόγως της ειδικότητας ώστε να παραπέμψει τον ασθενή του είτε για να ενημερωθεί για το σύνολο των γιατρών.

### Αναζήτηση Ιατρών

Αναζήτηση ανά Ειδικότητα:

[Αναζήτηση](#)

**Όλοι οι Ιατροί**

ID Ιατρού	Όνομα	Ειδικότητα
68	ΚΑΤΕΡΙΝΑ ΑΛΙΣΑΒΑΚΗ	ΠΑΘΟΛΟΓΟΣ
74	ΠΑΝΑΓΙΩΤΗΣ ΧΩΜΑΤΕΝΟΣ	ΠΑΘΟΛΟΓΟΣ
87	ΙΟΡΔΑΝΗΣ ΠΑΠΑΣΩΤΗΡΙΟΥ	ΠΑΘΟΛΟΓΟΣ

Copyright © Koutras Ioannis-Prodrmos

## 7.4 Είσοδος Διαχειριστή

Με την επιτυχή είσοδό του στην εφαρμογή ο διαχειριστής μπορεί να δει στην αρχική του σελίδα ένα ονομαστικό μήνυμα καλωσορίσματος καθώς και το μενού και τις επιλογές που έχει. Πιο αναλυτικά μπορεί να βλέπει-επεξεργάζεται και διαγράφει όλους τους χρήστες, να εγκρίνει τους ιατρούς μετά την εγγραφή τους, να βλέπει όλα τα ραντεβού, όλες τις διαγνώσεις και να μπορεί ο ίδιος να επεξεργαστεί κάποια διάγνωση.



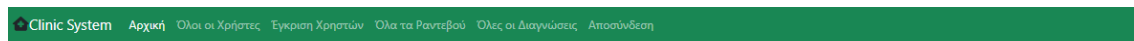
**Καλωσήρθες και πάλι ADMIN ΓΙΑΝΝΗΣ ΚΟΥΤΡΑΣ**

Copyright © Koutras Ioannis-Prodrmos

### 7.4.1 Όλοι οι Χρήστες

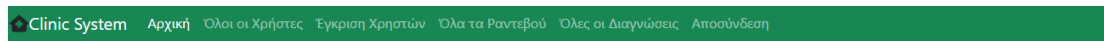
Σε αυτή την καρτέλα ο διαχειριστής μπορεί να δει το σύνολο των εγγεγραμμένων χρηστών είτε είναι ασθενείς είτε ιατροί με πλήθος προσωπικών πληροφοριών τους. Από

εδώ μπορεί να επεξεργαστεί τα στοιχεία που έχουν δηλώσει κατά την εγγραφή τους ή και να διαγράψει τελείως ένα χρήστη.



### Λίστα Εγγεγραμμένων Χρηστών

id	Επώνυμο	Όνομα	Email	ΑΜΚΑ	Αρ.Τηλεφώνου	Γιατρός	Ειδικότητα	Κατάσταση	Ενέργειες
1	ΚΟΥΤΡΑΣ	ΠΑΝΝΗΣ	yanniskoutras@hotmail.com	111222	6947584827				<a href="#">Επεξεργασία</a> <a href="#">Διαγραφή</a>
2	ΓΙΟΒΑΝΗΣ	ΑΛΕΞΑΝΔΡΟΣ	a@a	222111	12321				<a href="#">Επεξεργασία</a> <a href="#">Διαγραφή</a>
94	ΒΑΣΔΕΚΗΣ	ΜΙΧΑΛΗΣ	vasdekis@gmail.com	120306	6987546321	false		ΕΝΕΡΓΟΣ	<a href="#">Επεξεργασία</a> <a href="#">Διαγραφή</a>
95	ΑΝΑΣΤΑΣΙΟΣ	ΛΕΛΑΚΗΣ	lelakis@gmail.com	134679	693258753951	true	ΟΔΟΝΤΙΑΤΡΟΣ	ΕΝΕΡΓΟΣ	<a href="#">Επεξεργασία</a> <a href="#">Διαγραφή</a>
96	ΚΟΨΙΑΛΗΣ	ΑΛΕΞΑΝΔΡΟΣ	kopsialis@yahoo.gr	111111	6984532158	false		ΕΝΕΡΓΟΣ	<a href="#">Επεξεργασία</a> <a href="#">Διαγραφή</a>
97	ΦΥΡΟΥ	ΣΟΦΙΑ	fyou@gmail.com	222222	6980258963	false		ΕΝΕΡΓΟΣ	<a href="#">Επεξεργασία</a> <a href="#">Διαγραφή</a>
98	ΜΑΚΡΙΔΟΥ	ΜΑΡΙΑ	makridou@gmail.com	123456	6932058741	true	ΠΑΘΟΛΟΓΟΣ	ΕΝΕΡΓΟΣ	<a href="#">Επεξεργασία</a> <a href="#">Διαγραφή</a>
99	ΜΟΥΣΤΑΚΙΔΗΣ	ΗΛΙΑΣ	ilmoust@gmail.com	654321	6983578954	true	ΟΦΘΑΛΜΙΑΤΡΟΣ	ΕΝΕΡΓΟΣ	<a href="#">Επεξεργασία</a> <a href="#">Διαγραφή</a>
100	ΝΤΑΣΙΟΥ	ΣΟΦΙΑ	ntasiou@gmail.com	987654	6937539514	true	ΟΔΟΝΤΙΑΤΡΟΣ	ΕΝΕΡΓΟΣ	<a href="#">Επεξεργασία</a> <a href="#">Διαγραφή</a>
101	ΣΦΥΡΗΣ	ΔΗΜΗΤΡΙΟΣ	sfyris@gmail.com	654789	6971536978	true	ΠΑΘΟΛΟΓΟΣ	ΕΝΕΡΓΟΣ	<a href="#">Επεξεργασία</a> <a href="#">Διαγραφή</a>



### Επεξεργασία Χρήστη

Εισάγετε Όνομα

Εισάγετε Επώνυμο

Εισάγετε Email

Εισάγετε ΑΜΚΑ

Εισάγετε Αρ. Τηλεφώνου

[Αποθήκευση](#)

### 7.4.2 Έγκριση Χρηστών

Σε αυτή την καρτέλα ο διαχειριστής μπορεί να δει τους γιατρούς που έχουν κάνει εγγραφή στην εφαρμογή αλλά δεν έχουν πάρει ακόμα δικαιώματα χρήσης της. Έτσι αν ο γιατρός έχοντας προσκομίσει πχ και τα κατάλληλα δικαιολογητικά μπορεί να αποδεχτεί την αίτησή τους για χρήση της εφαρμογής.

### Εκκρεμείς Χρήστες

ID	Επώνυμο	Όνομα	ΑΜΚΑ	Email	Ειδικότητα	Τηλέφωνο	Κατάσταση	Ενέργειες
93	ΚΟΛΕΤΗΣ	ΧΡΗΣΤΟΣ	7391456	ch@ch	ΟΡΘΟΠΕΔΙΚΟΣ	3719654	ΕΚΚΡΕΜΕΙ	Αποδοχή

### 7.4.3 Όλα τα Ραντεβού

Από αυτή την καρτέλα ο διαχειριστής μπορεί να δει όλα τα ραντεβού τα οποία έχουν κλειστεί μέσω της εφαρμογής ολοκληρωμένα και μη.

### Ραντεβού-Διαχειριστής

Ιατρός	Ειδικότητα	Ημερ/νία	Επώνυμο	Όνομα	Λόγος Επίσκεψης	Κατάσταση	Actions
ΚΑΤΕΡΙΝΑ ΑΛΙΣΑΒΑΚΗ	ΠΑΘΟΛΟΓΟΣ	2023-10-05	ΝΑΚΑΣ	ΦΙΛΙΠΠΟΣ	ΞΗΡΟΛΑΙΜΙΑ	ΕΚΚΡΕΜΕΙ	Διάγνωση
ΜΑΡΙΑ ΜΑΚΡΙΔΟΥ	ΟΔΟΝΤΙΑΤΡΟΣ	2023-10-05	ΝΑΚΑΣ	ΦΙΛΙΠΠΟΣ	ΣΙΔΕΡΑΚΙΑ	ΕΚΚΡΕΜΕΙ	Διάγνωση
ΕΛΕΝΗ ΚΑΔΟΓΛΟΥ	ΟΔΟΝΤΙΑΤΡΟΣ	2023-10-06	ΠΕΛΚΑΣ	ΣΠΥΡΙΔΩΝ	ΠΑΝΟΡΑΜΙΚΗ	ΟΛΟΚΛΗΡΩΜΕΝΟ	Διάγνωση
ΑΓΓΕΛΙΚΗ ΗΛΙΑΔΗ	ΟΦΘΑΛΜΙΑΤΡΟΣ	2023-11-18	ΜΕΛΕΤΗ	ΕΛΕΟΝΩΡΑ	ΠΡΟΛΗΠΤΙΚΑ	ΟΛΟΚΛΗΡΩΜΕΝΟ	Διάγνωση
ΖΩΗ ΓΙΤΣΑ	ΟΦΘΑΛΜΙΑΤΡΟΣ	2023-09-30	ΝΑΚΑΣ	ΦΙΛΙΠΠΟΣ	ΣΥΝΤΑΓΗ ΓΥΑΛΙΩΝ	ΟΛΟΚΛΗΡΩΜΕΝΟ	Διάγνωση

Μπορεί επίσης να προβεί σε καταχώρηση διάγνωσης ενός ραντεβού σε περίπτωση που κάποιος ιατρός αδυνατεί, έτσι ώστε να εμφανιστεί η διάγνωση στο προφίλ του γιατρού.

### 7.4.4 Όλα οι Διαγνώσεις

Σε αυτή την καρτέλα ο διαχειριστής μπορεί να δει όλες τις διαγνώσεις τις οποίες έχουν κάνει όλοι οι γιατροί ανεξαρτήτως ειδικότητας καθώς και να δει τα συνημμένα αρχεία των εξετάσεων.

### Διαγνώσεις Ιατρού

ID	Ημερ/νία	Ιατρός	Ειδικότητα Ιατρού	Ασθενής	Διάγνωση	Θεραπεία	Φάρμακο	Αρχείο
46	2023-10-06	ΕΛΕΝΗ ΚΑΔΟΓΛΟΥ	ΟΔΟΝΤΙΑΤΡΟΣ	ΣΠΥΡΙΔΩΝ ΠΕΛΚΑΣ	ΕΜΦΑΝΙΣΗ ΑΝΩ ΔΕ ΤΡΑΠΕΖΙΤΗ	ΕΠΑΝΕΞΕΤΑΣΗ ΣΕ 2 ΒΔΟΜΑΔΕΣ	ALGOFREN	Άνοιγμα Αρχείου
47	2023-11-18	ΑΓΓΕΛΙΚΗ ΗΛΙΑΔΗ	ΟΦΘΑΛΜΙΑΤΡΟΣ	ΕΛΕΟΝΩΡΑ ΜΕΛΕΤΗ	ΟΛΑ ΚΑΛΩΣ	ΕΠΑΝΕΞΕΤΑΣΗ ΣΕ ΕΝΑ ΧΡΟΝΟ	ALGOFREN	
48	2023-09-30	ΖΩΗ ΓΙΤΣΑ	ΟΦΘΑΛΜΙΑΤΡΟΣ	ΦΙΛΙΠΠΟΣ ΝΑΚΑΣ	ΜΥΩΠΙΑ ΔΕ 1,25 ΑΡ 1,00	ΣΥΝΤΑΓΟΓΡΑΦΗΣΗ ΓΥΑΛΙΩΝ	-	Άνοιγμα Αρχείου

## 8. Συμπεράσματα - Επεκτάσεις

Η εφαρμογή δημιουργήθηκε με σκοπό να διευκολύνει τους πολίτες στη διαδικασία κλεισίματος ενός ραντεβού για εξέταση με αποφυγή ταλαιπωρίας αναμονής με φυσική παρουσία ή αδιέξοδων τηλεφωνικών επικοινωνιών. Επίσης δίνοντας τη δυνατότητα στους γιατρούς να ελέγξουν με τη χρήση της εφαρμογής τα ραντεβού τους και έχοντας εικόνα του ιστορικού του ασθενή να βγάλουν πιο ασφαλείς και ορθές διαγνώσεις και παρέχοντας την ίδια εικόνα και στους ασθενείς τους ώστε να έχουν γνώση οποιαδήποτε στιγμή το επιθυμούν.

Όσο η τεχνολογία εξελίσσεται πολλά πράγματα και καθημερινές δραστηριότητες μας μπορούν να απλοποιηθούν με την ανάπτυξη και χρήση τέτοιου είδους εφαρμογών.

Ως μελλοντική επέκταση της εφαρμογής θα μπορούσε να προστεθεί η εξαγωγή στατιστικών στοιχείων ως προς τον αριθμό των κλινικών με τη περισσότερη κίνηση, ποσοστά ανδρών-γυναικών καθώς και ηλικιακά στατιστικά. Αυτό απαιτεί προσθήκη νέων πεδίων στη βάση δεδομένων.

## 9. Βιβλιογραφία

### Ελληνική

Γιάννης Θεοδωρίδης, Σημειώσεις μαθήματος «*Βάσεις & Αποθήκες Δεδομένων*», Παν. Πειραιά

### Ξενόγλωσση

Spring Boot Baeldung <https://www.baeldung.com/spring-boot>

Spring Boot web applications <https://spring.io/web-applications>

Tutorialspoint <https://www.tutorialspoint.com>

W3Schools SQL Tutorial <https://www.w3schools.com/sql/>

MySQL Official Developers page <https://dev.mysql.com>

Thymeleaf Official Page <https://www.thymeleaf.org/>

Bootstrap 5.1 Official Page <https://getbootstrap.com/docs/5.1/getting-started/>

Craig Walls, 2016, Spring Boot IN ACTION, MANNING