



## ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ – ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

### Πρόγραμμα Μεταπτυχιακών Σπουδών «Πληροφορική»

#### Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Εφαρμογή αυτόματης υποστήριξης πολιτών σε θέματα υγείας, με χρήση Spring Boot.  Application of automatic support for citizens in health matters, using Spring Boot.
Όνοματεπώνυμο Φοιτήτριας	Παρασκευή Τσακίρη
Πατρώνυμο	Ηλίας
Αριθμός Μητρώου	ΜΠΠΛ20080
Επιβλέπων	Αλέπης Ευθύμιος, Αναπληρωτής Καθηγητής

Ημερομηνία Παράδοσης **Ιανουάριος 2023**

---

### Τριμελής Εξεταστική Επιτροπή

Ευθύμιος Αλέπης

Μαρία Βίβου

Κωνσταντίνος Πατσάκης

Αναπληρωτής Καθηγητής

Καθηγήτρια

Αναπληρωτής Καθηγητής

## Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω τον επιβλέποντα Καθηγητή μου κύριο Αλέπη Ευθύμιο, για την ευκαιρία που μου έδωσε να επιλέξω αυτή την μεταπτυχιακή διατριβή και για την καθοδήγησή του κατά τη διάρκεια της υλοποίησής της. Επιπρόσθετα, θα ήθελα να εκφράσω τις ευχαριστίες μου σε όλους τους Καθηγητές για τις γνώσεις που μου προσέφεραν κατά την φοίτησή μου στο μεταπτυχιακό πρόγραμμα της Πληροφορικής.

## Περίληψη

Στην παρούσα εργασία, δημιουργήθηκε μια εφαρμογή με χρήση της τεχνολογίας Spring Boot για την ιατρική εξυπηρέτηση των πολιτών μιας πόλης. Οι πολίτες μπορούν να εγγραφούν στο σύστημα και να επιλέξουν διάφορες ιατρικές υπηρεσίες όπως να καλέσουν ασθενοφόρο, να ενημερωθούν για το νοσοκομείο που μπορούν να αποτανθούν και να συνομιλήσουν είτε άμεσα είτε με email με τους διαθέσιμους ιατρούς.

Στο σύστημα μπορούν ελεύθερα να εγγραφούν ιατροί και χρήστες, ενώ υπάρχει και ο διαχειριστής (θεωρητικά κάποιος δημόσιος φορέας), που μπορεί να εξάγει τα στατιστικά στοιχεία της χρήσης από τους πολίτες και τους ιατρούς.

## Abstract

In this work, an application was created using Spring Boot technology for the medical service of the citizens of a city. Citizens can register in the system and choose various medical services such as calling an ambulance, being informed about the hospital they can be transferred to and chatting either directly or by email with the available doctors.

Doctors and users can freely register in the system, while there is also the administrator (theoretically some public client), who can extract the statistics of the use by citizens and doctors.

## Contents

1. Εισαγωγή.....	6
1.1 Σκοπός της εργασίας .....	7
2. Ο καιρός της Πληροφορικής.....	7
2.1 Προς μια ψηφιακή εποχή .....	7
2.1.1 Το Διαδίκτυο .....	8
2.1.2 Εξέλιξη στην αγορά εργασίας.....	8
2.2 Οι ψηφιακοί πολίτες.....	9
2.3 Η ιατρική στην ψηφιακή εποχή.....	11
3. Τεχνικά χαρακτηριστικά της διαδικτυακής εφαρμογής μας.....	12
3.1. Η γλώσσα προγραμματισμού Java .....	12
3.1.1. Τα χαρακτηριστικά της Java και η εικονική της μηχανή (Virtual Machine) .....	13
3.2 Η Java EE .....	14
3.3. Spring Framework.....	15
3.3.1 Spring vs Spring Boot .....	15
3.3.2 Τι είναι το Spring.....	16
3.3.3 Τι είναι το Spring Boot;.....	18
3.4 HTML.....	20
3.5. CSS.....	21
3.6 Javascript.....	22
3.7 jQuery .....	23
4. Ανάλυση του κώδικα της εφαρμογής μας .....	25
4.1 Εισαγωγή – Ανάλυση .....	25

4.1.1 Γενικά .....	25
4.2 SPRING BOOT – Model View Controller και η εφαρμογή μας .....	25
4.2.1 Model .....	27
4.2.2 View .....	31
4.2.3 Controller .....	33
4.3 Η ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ MySQL.....	34
4.4 Sessions.....	39
5. Η λειτουργία της εφαρμογής μας .....	40
5.1 Γενικά .....	40
5.2 Ο Χρήστης .....	41
5.3 Ο Ιατρός .....	48
5.4 Ο διαχειριστής (Administrator) .....	51
6. Γενική σύνοψη - συμπεράσματα.....	53
7. Βιβλιογραφία .....	53

## 1. Εισαγωγή

Με την ψηφιοποίηση της καθημερινότητας μας, οι ανάγκες για ιατρικές υπηρεσίες με τη χρήση υπολογιστή γίνονται όλο και πιο απαραίτητες. Είναι σημαντικό να έχει την δυνατότητα ο πολίτης σε μια μικρή ή μεγάλη κοινωνική μονάδα να μπορεί μέσω του υπολογιστή του να έρθει σε επαφή με κάποιον πάροχο ιατρικής βοήθειας . Η βοήθεια αυτή σε αυτήν την φάση θα είναι κυρίως ενημερωτική έτσι ώστε ο πολίτης να λάβει την κατάλληλη πληροφορία (όσο γίνεται πιο άμεση) και αναζητήσει έπειτα πιο λεπτομερή ιατρική υπηρεσία.

## 1.1 Σκοπός της εργασίας

Μέχρι σήμερα η ανάγκη για άμεση ιατρική συμβουλή, ικανοποιούνταν με την χρήση του τηλεφώνου κυρίως για να ενημερωθεί ο χρήστης σε ποιο σημείο θα έπρεπε να μεταβεί φυσικά για να λάβει ιατρική βοήθεια. Το σύστημα που αναπτύχθηκε τον βοηθά στο να έρθει σε επαφή άμεσα με τον ιατρό της ειδικότητας που τον ενδιαφέρει, με την ευκολία της χρήσης του υπολογιστή του και να επικοινωνήσει είτε με ηλεκτρονικό μήνυμα (email) είτε live σε μια από τις διαθέσιμες ηλεκτρονικές πλατφόρμες «ζωντανής» επικοινωνίας.

## 2. Ο καιρός της Πληροφορικής

### 2.1 Προς μια ψηφιακή εποχή

Η Εποχή της Πληροφορίας (επίσης γνωστή ως Εποχή των Υπολογιστών, Ψηφιακή Εποχή, Εποχή του Πυριτίου ή Εποχή των Νέων Μέσων) είναι μια ιστορική περίοδος που ξεκίνησε στα μέσα του 20ου αιώνα, που χαρακτηρίζεται από μια ταχεία μετατόπιση από την παραδοσιακή βιομηχανία που καθιερώθηκε από τη Βιομηχανική Επανάσταση σε μια οικονομία που βασίζεται κυρίως στην τεχνολογία της πληροφορίας.[1] Η έναρξη της Εποχής της Πληροφορίας έχει συνδεθεί με την ανάπτυξη του τρανζίστορ το 1947,[2] του βασικού δομικού στοιχείου της σύγχρονης ηλεκτρονικής, του οπτικού ενισχυτή το 1957, της βάσης των επικοινωνιών με οπτικές ίνες μεγάλων αποστάσεων, και του Unix,[3]

### 2.1.1 Το Διαδίκτυο

Το **Διαδίκτυο** (Internet) σχεδιάστηκε αρχικά ως ένα κατακεκομημένο δίκτυο που θα μπορούσε να συνδέσει υπολογιστές μεταξύ τους, να είναι ανθεκτικό και να μην έχει τρωτά σημεία. Το Διαδίκτυο δεν θα μπορούσε να καταρρεύσει ολοκληρωτικά από μια μοναδική αιτία, και αν μεμονωμένες περιοχές του δικτύου απενεργοποιηθούν τότε οι πληροφορίες θα μπορούσαν να αλλάξουν εύκολα πορεία. Δημιουργήθηκε από την APRA και η πρώτη του ονομασία ήταν APRANET. Οι πρώτες βασικές του εφαρμογές και λειτουργίες ήταν το **ηλεκτρονικό ταχυδρομείο** (e-mail) και η μεταφορά αρχείων.

Παρ' όλα αυτά, μόνο μετά την δημιουργία του **παγκόσμιου ιστού** το 1989 το **Διαδίκτυο** έγινε πραγματικά ένα παγκόσμιο δίκτυο. Σήμερα το Διαδίκτυο έχει γίνει η απόλυτη πλατφόρμα για την επιτάχυνση της ροής των πληροφοριών και είναι η ταχύτερα αναπτυσσόμενη μορφή των μέσων ενημέρωσης.

### 2.1.2 Εξέλιξη στην αγορά εργασίας

Το 1956 στις Ηνωμένες Πολιτείες οι ερευνητές παρατήρησαν ότι ο αριθμός των ατόμων που κατείχαν θέσεις εργασίας λευκού κολάρου (υπάλληλοι) , υπερέβαιναν στον αριθμό των ανθρώπων που κατείχαν θέσεις εργασίας μπλε κολάρου(εργάτες). Τότε οι ερευνητές συνειδητοποίησαν ότι η



βιομηχανική εποχή είχε φτάσει στο τέλος της. Καθώς η βιομηχανική εποχή έφτανε στο τέλος της, τα νεότερα χρόνια υιοθέτησαν τον τίτλο της εποχής της πληροφορίας.

Εκείνη την εποχή, ελάχιστες θέσεις εργασίας είχαν να κάνουν με υπολογιστές και γενικά με ότι έχει να κάνει με την τεχνολογία των ηλεκτρονικών υπολογιστών. Ο αριθμός των ανθρώπων που κρατούσαν θέσεις εργασίας ως υπάλληλοι καταστημάτων, υπάλληλοι γραφείου, νοσοκόμοι, δάσκαλοι κλπ. ολοένα και αυξανόταν. Ο **δυτικός κόσμος** μετατρέποταν σε μια οικονομία παροχής υπηρεσιών.

Τελικά η τεχνολογία της πληροφορικής και των επικοινωνιών (όπως είναι οι υπολογιστές και γενικά των ηλεκτρονικών υπολογιστικών μηχανημάτων, των οπτικών ινών, των επικοινωνιακών δορυφόρων, του Διαδικτύου) έγινε σημαντικό μέρος της σύγχρονης οικονομίας. Οι υπολογιστές εξελίσσονταν συνεχώς και σαν αποτέλεσμα πολλές επιχειρήσεις γνώρισαν δραματικές αλλαγές.

Έτσι, συχνά ο όρος **εποχή της πληροφορίας** αναφέρεται στη χρήση των κινητών τηλεφώνων της ψηφιακής μουσικής, της τηλεόρασης υψηλής ευκρίνειας, των ψηφιακών καμερών, του Διαδικτύου καθώς και σε άλλα σχετικά προϊόντα και υπηρεσίες που έχουν τεθεί σε ευρεία χρήση.

## 2.2 Οι ψηφιακοί πολίτες

Ο όρος ψηφιακός πολίτης χρησιμοποιείται με διαφορετικές έννοιες. Σύμφωνα με τον ορισμό που παρέχεται από την Karen Mossberger, μία από τις συγγραφείς του *Digital Citizenship: The Internet, Society, and Participation*[4], οι ψηφιακοί πολίτες είναι «αυτοί που χρησιμοποιούν το διαδίκτυο τακτικά και αποτελεσματικά». Υπό αυτή την έννοια, ένας ψηφιακός πολίτης είναι ένα

άτομο που χρησιμοποιεί την τεχνολογία της πληροφορίας (IT) προκειμένου να εμπλακεί στην κοινωνία, την πολιτική και την κυβέρνηση.

Πιο πρόσφατες επεξεργασίες της έννοιας ορίζουν την ψηφιακή ιθαγένεια ως την αυτοαναπαράσταση του ρόλου των ανθρώπων στην κοινωνία μέσω της χρήσης ψηφιακών τεχνολογιών, τονίζοντας τα χαρακτηριστικά ενδυνάμωσης και εκδημοκρατισμού της ιδέας της ιθαγένειας. Αυτές οι θεωρίες στοχεύουν να λάβουν υπόψη τη διαρκώς αυξανόμενη καταγραφή δεδομένων των σύγχρονων κοινωνιών, οι οποίες έθεσαν ριζικά υπό αμφισβήτηση την έννοια του «να είσαι (ψηφιακός) πολίτης σε μια κοινωνία με δεδομένα»,[5] επίσης. αναφέρεται ως η «αλγοριθμική κοινωνία»,[6], η οποία χαρακτηρίζεται από την αυξανόμενη συγκέντρωση δεδομένων της κοινωνικής ζωής και τη διάχυτη παρουσία πρακτικών επιτήρησης.

Το Datafication παρουσιάζει κρίσιμες προκλήσεις για την ίδια την έννοια της ιθαγένειας, έτσι ώστε η συλλογή δεδομένων να μην μπορεί πλέον να θεωρείται μόνο ζήτημα ιδιωτικότητας[5], ώστε:

Δεν μπορούμε απλώς να υποθέσουμε ότι το να είσαι πολίτης στο διαδίκτυο σημαίνει ήδη κάτι (είτε είναι η ικανότητα συμμετοχής είτε η ικανότητα να παραμείνουμε ασφαλείς) και στη συνέχεια να αναζητήσουμε εκείνους των οποίων η συμπεριφορά συμμορφώνεται με αυτό το νόημα

Αντίθετα, η ιδέα της ψηφιακής ιθαγένειας θα αντικατοπτρίζει την ιδέα ότι δεν είμαστε πλέον απλοί «χρήστες» των τεχνολογιών, καθώς αυτές διαμορφώνουν τον οργανισμό μας τόσο ως άτομα όσο και ως πολίτες.

## 2.3 Η ιατρική στην ψηφιακή εποχή

Η ψηφιακή υγεία είναι ένας κλάδος που περιλαμβάνει προγράμματα ψηφιακής περίθαλψης, για να βελτιώσει την αποτελεσματικότητα της παροχής υγειονομικής περίθαλψης και να κάνει την ιατρική πιο εξατομικευμένη και ακριβή.[7][ Χρησιμοποιεί τεχνολογίες πληροφοριών και επικοινωνιών για να διευκολύνει την κατανόηση των προβλημάτων υγείας και των προκλήσεων που αντιμετωπίζουν τα άτομα που λαμβάνουν ιατρική περίθαλψη[8] με πιο εξατομικευμένους και ακριβείς τρόπους. Οι ορισμοί της ψηφιακής υγείας και των αρμοδιοτήτων της επικαλύπτονται από πολλές απόψεις με εκείνους της υγείας και της ιατρικής πληροφορικής.

Η παγκόσμια υιοθέτηση ηλεκτρονικών ιατρικών αρχείων βρίσκεται σε άνοδο από το 1990 και συνδέεται στενά με την ύπαρξη καθολικής υγειονομικής περίθαλψης.[9] Η ψηφιακή υγεία είναι ένας πολυεπιστημονικός τομέας στον οποίο συμμετέχουν πολλοί ενδιαφερόμενοι, συμπεριλαμβανομένων κλινικών γιατρών, ερευνητών και επιστημόνων με ευρύ φάσμα εξειδίκευσης στην υγειονομική περίθαλψη, τη μηχανική, τις κοινωνικές επιστήμες, τη δημόσια υγεία, τα οικονομικά της υγείας και τη διαχείριση δεδομένων.[10]

Οι ψηφιακές τεχνολογίες υγείας περιλαμβάνουν λύσεις και υπηρεσίες τόσο υλικού όσο και λογισμικού, συμπεριλαμβανομένης της τηλεϊατρικής, των φορητών συσκευών, της επαυξημένης πραγματικότητας και της εικονικής πραγματικότητας.[11] Γενικά, η ψηφιακή υγεία διασυνδέει συστήματα υγείας για να βελτιώσει τη χρήση υπολογιστικών τεχνολογιών, έξυπνων συσκευών, τεχνικών υπολογιστικής ανάλυσης και μέσων επικοινωνίας για να βοηθήσει τους επαγγελματίες υγείας και τους ασθενείς τους να διαχειριστούν ασθένειες και κινδύνους για την υγεία, καθώς και να προωθήσουν την υγεία και την ευημερία. [12]

Αν και οι ψηφιακές πλατφόρμες υγείας επιτρέπουν γρήγορες και φθηνές επικοινωνίες, οι επικριτές προειδοποιούν για πιθανές παραβιάσεις της ιδιωτικής ζωής των προσωπικών δεδομένων υγείας και τον ρόλο που θα μπορούσε να διαδραματίσει η ψηφιακή υγεία στην αύξηση της υγείας και του ψηφιακού χάσματος μεταξύ των ομάδων κοινωνικής πλειοψηφίας και μειοψηφίας, οδηγώντας πιθανώς σε δυσπιστία και δισταγμό στη χρήση ψηφιακών συστημάτων υγείας.[13]

### 3. Τεχνικά χαρακτηριστικά της διαδικτυακής εφαρμογής μας

#### 3.1. Η γλώσσα προγραμματισμού Java

Η Sun Microsystems κυκλοφόρησε την πρώτη δημόσια υλοποίηση ως Java 1.0 το 1996. Υποσχέθηκε λειτουργικότητα εγγραφής κώδικα μία φορά, εκτέλεσης οπουδήποτε (Write Once Run All), παρέχοντας χρόνους εκτέλεσης χωρίς κόστος σε δημοφιλείς πλατφόρμες. Αρκετά

ασφαλής και με δυνατότητα ρύθμισης παραμέτρων, επέτρεπε περιορισμούς πρόσβασης σε δίκτυο και αρχεία. Τα μεγάλα προγράμματα περιήγησης ιστού ενσωμάτωσαν σύντομα τη δυνατότητα εκτέλεσης εφαρμογών Java σε ιστοσελίδες και η Java έγινε γρήγορα δημοφιλής. Με την έλευση της Java 2 (κυκλοφόρησε αρχικά ως J2SE 1.2 τον Δεκέμβριο 1998 – 1999), οι νέες εκδόσεις είχαν πολλαπλές διαμορφώσεις κατασκευασμένες για διαφορετικούς τύπους πλατφόρμων. Το J2EE περιλάμβανε τεχνολογίες και API για εταιρικές εφαρμογές που εκτελούνται συνήθως σε περιβάλλοντα διακομιστών, ενώ το J2ME παρουσίαζε API βελτιστοποιημένα για εφαρμογές για κινητές συσκευές. Η έκδοση για υπολογιστές μετονομάστηκε σε J2SE. Το 2006, για λόγους μάρκετινγκ, η Sun μετονόμασε τις νέες εκδόσεις Java version 2 σε Java EE, Java ME και Java SE, αντίστοιχα.

### 3.1.1. Τα χαρακτηριστικά της Java και η εικονική της μηχανή (Virtual Machine)

Ένας σχεδιαστικός στόχος της Java είναι η φορητότητα, που σημαίνει ότι τα προγράμματα που είναι γραμμένα για την πλατφόρμα Java πρέπει να εκτελούνται παρόμοια σε οποιονδήποτε συνδυασμό υλικού και λειτουργικού συστήματος με επαρκή υποστήριξη χρόνου εκτέλεσης. Αυτό επιτυγχάνεται με τη μεταγλώττιση του κώδικα γλώσσας Java σε μια ενδιάμεση αναπαράσταση που ονομάζεται bytecode Java, αντί για απευθείας σε κώδικα μηχανής συγκεκριμένης αρχιτεκτονικής. Οι οδηγίες bytecode Java είναι ανάλογες με τον κώδικα μηχανής, αλλά προορίζονται να εκτελεστούν από μια εικονική μηχανή (VM) που είναι γραμμένη ειδικά για το

υλικό υποδοχής. Οι τελικοί χρήστες χρησιμοποιούν συνήθως ένα Java Runtime Environment (JRE) εγκατεστημένο στη συσκευή τους για αυτόνομες εφαρμογές Java ή ένα πρόγραμμα περιήγησης Ιστού για εφαρμογές Java.

Οι τυπικές βιβλιοθήκες παρέχουν έναν γενικό τρόπο πρόσβασης σε χαρακτηριστικά ειδικά για τον κεντρικό υπολογιστή, όπως γραφικά, νήματα και δικτύωση.

Η χρήση του καθολικού bytecode καθιστά τη μεταφορά απλή. Ωστόσο, η επιβάρυνση της ερμηνείας του bytecode σε εντολές μηχανής έκανε τα ερμηνευμένα προγράμματα να εκτελούνται σχεδόν πάντα πιο αργά από τα εγγενή (native) εκτελέσιμα. Οι μεταγλωττιστές Just-in-time (JIT) που μεταγλωττίζουν κωδικούς byte σε κώδικα μηχανής κατά τη διάρκεια του χρόνου εκτέλεσης εισήχθησαν από πρώιμο στάδιο. Ο μεταγλωττιστής Hotspot της Java είναι στην πραγματικότητα δύο μεταγλωττιστές σε έναν. Η ίδια η Java είναι ανεξάρτητη από την πλατφόρμα και είναι προσαρμοσμένη στη συγκεκριμένη πλατφόρμα στην οποία πρόκειται να εκτελεστεί από μια εικονική μηχανή Java (JVM) για αυτήν, η οποία μεταφράζει τον bytecode Java στη γλώσσα μηχανής της πλατφόρμας.

## 3.2 Η Java EE

Η Java Platform, Enterprise Edition (Java EE) βασίζεται στην προδιαγραφή Java SE. Αντιπροσωπεύει μια συνεργασία μεταξύ πολλών προμηθευτών και ηγετών του κλάδου της Πληροφορικής και παρέχει συγκεκριμένη υποστήριξη υποδομής για εφαρμογές.

**Στο πλαίσιο Java EE, προστίθενται κανόνες σε δύο επίπεδα:**

- Στο επίπεδο εφαρμογής, για τη διαχείριση της δυναμικής επιχειρηματικής λογικής και της ροής εργασιών.
- Στο επίπεδο παρουσίασης, για να προσαρμοστεί η ροή σελίδας και η ροή εργασίας και να δημιουργηθούν προσαρμοσμένες σελίδες με βάση την κατάσταση περιόδου λειτουργίας.

Η Java EE είναι φορητή και επεκτάσιμη και υποστηρίζει υπάρχοντα στοιχεία ενσωμάτωσης (intergration) που είναι βασισμένα σε αρχιτεκτονική EJB. Η Java EE απλοποιεί τις εταιρικές εφαρμογές ορίζοντας και καθορίζοντας ένα πλήρες σύνολο κοινών τυπικών υπηρεσιών, όπως η ονομασία, η διαχείριση συναλλαγών, η ταυτόχρονη χρήση, η ασφάλεια και η πρόσβαση στη βάση δεδομένων.

Η Java EE ορίζει επίσης ένα μοντέλο κοντέινερ, το οποίο φιλοξενεί και διαχειρίζεται τα διάφορα κομμάτια της εφαρμογής Java EE. Τα κοντέινερ με τη σειρά τους στεγάζονται σε διακομιστές Java EE.

### 3.3. Spring Framework

#### 3.3.1 Spring vs Spring Boot

Το Spring είναι το πιο δημοφιλές πλαίσιο(framework) ανάπτυξης εφαρμογών στην Java. Ωστόσο, τα τελευταία χρόνια, το Spring έχει γίνει όλο και πιο περίπλοκο λόγω των πρόσθετων

λειτουργιών. Η έναρξη ενός νέου έργου απαιτεί συνήθως μια μακρά διαδικασία και απαιτεί πολύ χρόνο.

Για να μην ξεκινήσετε από το μηδέν και για να εξοικονομήσετε χρόνο, παρουσιάστηκε το Spring Boot. Το Spring Boot χρησιμοποιεί το Spring ως το βασικό του στρώμα(layer) και επιτρέπει στους προγραμματιστές να δημιουργήσουν μια αυτόνομη εφαρμογή χωρίς καμία ή ελάχιστη διαμόρφωση. Στοχεύει ειδικά στη δημιουργία διαδικτυακών εφαρμογών και μικροϋπηρεσιών.

### 3.3.2 Τι είναι το Spring

Το Spring είναι ένα ελαφρύ πλαίσιο ανάπτυξης Java ανοιχτού κώδικα. Παρέχει ένα ολοκληρωμένο μοντέλο προγραμματισμού και διαμόρφωσης για την ανάπτυξη εφαρμογών Java σε EE επίπεδο. Στόχος του είναι να απλοποιήσει την ανάπτυξη της Java EE και με τον τρόπο αυτό βοηθά τους προγραμματιστές να δημιουργήσουν εφαρμογές πιο αποτελεσματικά και αποδοτικά.

Το Spring εστιάζει σε διάφορους τομείς ανάπτυξης εφαρμογών και παρέχει ένα ευρύ φάσμα δυνατοτήτων

#### **Κύρια Χαρακτηριστικά του Spring**

Ένα από τα κύρια χαρακτηριστικά του Spring είναι το dependence injection, το οποίο επιτρέπει την ανάπτυξη χαλαρά συζευγμένων εφαρμογών. Περιλαμβάνει επίσης τις ακόλουθες ενότητες:



- Spring MVC
- Spring JDBC
- Spring WEBFLOW
- Spring SECURITY
- Spring ORM
- Spring AOP

Μερικές από χρήσιμες υπηρεσίες χρησιμοποιώντας το Spring είναι:

- Εξασφάλιση κορυφαίας ασφάλειας από την πλευρά του διακομιστή
- Δημιουργία ασύγχρονων εφαρμογών
- Αρχιτεκτονική που βασίζεται σε γεγονότα

### Οφέλη της Springs

Μπορεί να χρησιμοποιηθεί για όλα τα επίπεδα ανάπτυξης εφαρμογών

- Αρκετά ελαφρύ λαμβάνοντας υπόψη το μοντέλο POJO

- Επιτρέπει χαλαρή σύζευξη και εύκολη δοκιμή
- Υποστηρίζει δηλωτικό προγραμματισμό
- Υποστηρίζει διαμορφώσεις XML και σχολιασμού
- Παρέχει υπηρεσίες ενδιάμεσου λογισμικού

### 3.3.3 Τι είναι το Spring Boot;

Το Spring Boot είναι μια επέκταση του Spring, η οποία εξαλείφει τις διαμορφώσεις που απαιτούνται για τη ρύθμιση μιας εφαρμογής Spring. Διαθέτοντας διαμόρφωση που βασίζεται σε σχολιασμούς, το Spring Boot επιτρέπει ένα ταχύτερο και πιο αποτελεσματικό οικοσύστημα ανάπτυξης.

Δεδομένου ότι το Spring Boot είναι κατασκευασμένο πάνω από το Spring, προσφέρει όλα τα χαρακτηριστικά και τα πλεονεκτήματα του Spring. Το Spring Boot στοχεύει στη μείωση του μεγέθους του κώδικα και παρέχει στους προγραμματιστές τον ευκολότερο τρόπο δημιουργίας μιας εφαρμογής.

#### **Κύρια χαρακτηριστικά του Spring Boot**

Μερικά από τα χαρακτηριστικά του Spring Boot περιλαμβάνουν τα ακόλουθα:

- Ενσωματωμένος διακομιστής
- Αρχεία WAR για Tomcat, Jetty και Undertow
- Αυτόματη διαμόρφωση για τη λειτουργία Spring
- Έλεγχος health check με μετρήσεις και άλλες αναφορές

Επομένως, το Spring Boot είναι μια επέκταση που βασίζεται στο Spring. Χρησιμοποιώντας λειτουργίες όπως η αυτόματη διαμόρφωση, εξοικονομεί τους προγραμματιστές από τη σύνταξη μακροσκελούς κώδικα. Παρόλο που αφαιρεί μέρος της ευελιξίας του Spring, συμβάλλει στη μείωση του χρόνου ανάπτυξης και στην αποφυγή περιττών διαμορφώσεων.

#### Πλεονεκτήματα της Spring Boot

- Μπορεί να χρησιμοποιηθεί για τη δημιουργία αυτόνομων εφαρμογών
- Δεν χρειάζεται να αναπτύξετε αρχεία WAR
- Δεν απαιτεί διαμόρφωση XML
- Ενσωματώνει απευθείας τα Tomcat, Jetty και Undertow
- Προσφέρει χαρακτηριστικά ετοιμότητας παραγωγής
- Πιο εύκολο να ξεκινήσει
- Ευκολότερη διαχείριση και προσαρμογή

### 3.4 HTML

Η γλώσσα σήμανσης HyperText ή HTML είναι η τυπική γλώσσα σήμανσης για έγγραφα που έχουν σχεδιαστεί για εμφάνιση σε πρόγραμμα περιήγησης Ιστού. Μπορεί να υποστηριχθεί από τεχνολογίες όπως τα Cascading Style Sheets (CSS) και γλώσσες δέσμης ενεργειών όπως η JavaScript.

Τα προγράμματα περιήγησης Ιστού λαμβάνουν έγγραφα HTML από έναν διακομιστή ιστού ή από τοπική αποθήκευση και αποδίδουν τα έγγραφα σε ιστοσελίδες πολυμέσων. Η HTML περιγράφει τη δομή μιας ιστοσελίδας σημασιολογικά και αρχικά περιλάμβανε ενδείξεις για την εμφάνιση του εγγράφου.

Τα στοιχεία HTML είναι τα δομικά στοιχεία των σελίδων HTML. Με εντολές HTML, εικόνες και άλλα αντικείμενα, όπως διαδραστικές φόρμες, μπορούν να ενσωματωθούν στη σελίδα απόδοσης. Η HTML παρέχει ένα μέσο για τη δημιουργία δομημένων εγγράφων δηλώνοντας δομική σημασιολογία για κείμενο όπως επικεφαλίδες, παραγράφους, λίστες, συνδέσμους,

εισαγωγικά και άλλα στοιχεία. Τα στοιχεία HTML οριοθετούνται με ετικέτες, γραμμένες με χρήση αγκύλων. Ετικέτες όπως `<img />` και `<input />` εισάγουν απευθείας περιεχόμενο στη σελίδα. Άλλες ετικέτες, όπως `<p>`, περιβάλλουν και παρέχουν πληροφορίες σχετικά με το κείμενο του εγγράφου και μπορεί να περιλαμβάνουν άλλες ετικέτες ως δευτερεύοντα στοιχεία. Τα προγράμματα περιήγησης δεν εμφανίζουν τις ετικέτες HTML αλλά τις χρησιμοποιούν για να ερμηνεύσουν το περιεχόμενο της σελίδας.

Η HTML μπορεί να ενσωματώσει προγράμματα γραμμένα σε μια γλώσσα δέσμης ενεργειών όπως η JavaScript, η οποία επηρεάζει τη συμπεριφορά και το περιεχόμενο των ιστοσελίδων. Η συμπερίληψη του CSS καθορίζει την εμφάνιση και τη διάταξη του περιεχομένου. Η Κοινοπραξία του Παγκόσμιου Ιστού (W3C) και σημερινός συντηρητής των προτύπων CSS, έχει ενθαρρύνει τη χρήση του CSS έναντι ρητού HTML παρουσίασης από το 1997. Μια μορφή HTML, γνωστή ως HTML5, χρησιμοποιείται για την εμφάνιση βίντεο και ήχου, χρησιμοποιώντας κυρίως το στοιχείο `<canvas>`, σε συνεργασία με τη Javascript.

### 3.5. CSS

Τα Cascading Style Sheets (CSS) είναι μια περιγραφής μορφής (style) που χρησιμοποιείται για την περιγραφή της παρουσίασης ενός εγγράφου γραμμένου σε μια γλώσσα σήμανσης όπως η HTML ή η XML (συμπεριλαμβανομένων των διαλέκτων XML όπως SVG, MathML ή XHTML). Το CSS αποτελεί μια τεχνολογία που είναι ακρογωνιαίος λίθος του Παγκόσμιου Ιστού, μαζί με HTML και JavaScript.

Το CSS έχει σχεδιαστεί για να επιτρέπει τον διαχωρισμό περιεχομένου και παρουσίασης, συμπεριλαμβανομένων της διάταξης, των χρωμάτων και των γραμματοσειρών. Αυτός ο διαχωρισμός μπορεί να βελτιώσει την προσβασιμότητα στο περιεχόμενο. Και παρέχει μεγαλύτερη ευελιξία και έλεγχο στην προδιαγραφή των χαρακτηριστικών παρουσίασης. Είναι δυνατό να εφαρμοστεί σε πολλές ιστοσελίδες ( κοινή χρήση μορφοποίησης καθορίζοντας το σχετικό CSS σε ένα ξεχωριστό αρχείο .css, το οποίο μειώνει την πολυπλοκότητα και την επανάληψη στο δομικό περιεχόμενο).

Ο διαχωρισμός μορφοποίησης και περιεχομένου καθιστά επίσης εφικτή την παρουσίαση της ίδιας σελίδας σήμανσης σε διαφορετικά στυλ για διαφορετικές μεθόδους απόδοσης. Το CSS έχει επίσης κανόνες για εναλλακτική μορφοποίηση εάν το περιεχόμενο είναι προσβάσιμο από φορητή συσκευή.

### 3.6 Javascript

Η JavaScript που συχνά συντομεύεται ως JS, είναι μια γλώσσα προγραμματισμού που αποτελεί μια από τις βασικές τεχνολογίες του Παγκόσμιου Ιστού, μαζί με την HTML και το CSS. Από το 2022, το 98% των ιστότοπων χρησιμοποιούν JavaScript στην πλευρά του client για συμπεριφορά ιστοσελίδας, συχνά ενσωματώνοντας βιβλιοθήκες τρίτων. Όλα τα μεγάλα προγράμματα περιήγησης ιστού διαθέτουν ειδική μηχανή JavaScript για την εκτέλεση του κώδικα στις συσκευές των χρηστών.

Η JavaScript είναι μια γλώσσα υψηλού επιπέδου, συχνά μεταγλωττισμένη JIT η οποία συμμορφώνεται με το πρότυπο ECMAScript. Διαθέτει αντικειμενοστραφή προσανατολισμό που βασίζεται σε πρωτότυπα και λειτουργίες first order. Είναι πολλαπλών παραδειγμάτων, υποστηρίζοντας στυλ προγραμματισμού που βασίζονται σε γεγονότα. Διαθέτει διεπαφές προγραμματισμού εφαρμογών (API) για εργασία με κείμενο, ημερομηνίες, κανονικές εκφράσεις, τυπικές δομές δεδομένων και το μοντέλο αντικειμένου εγγράφου (DOM).

Το πρότυπο ECMAScript δεν περιλαμβάνει καμία είσοδο/έξοδο (I/O), όπως εγκαταστάσεις δικτύωσης, αποθήκευσης ή γραφικών. Στην πράξη, το πρόγραμμα περιήγησης ιστού ή άλλο σύστημα χρόνου εκτέλεσης παρέχει JavaScript API για I/O.

Οι μηχανές JavaScript χρησιμοποιούνταν αρχικά μόνο σε προγράμματα περιήγησης Ιστού, αλλά τώρα αποτελούν βασικά στοιχεία ορισμένων διακομιστών και μιας ποικιλίας εφαρμογών. Το πιο δημοφιλές σύστημα με αυτήν τη χρήση είναι το Node.js.

Παρόλο που η Java και η JavaScript είναι παρόμοια ως προς το όνομα, τη σύνταξη και τις αντίστοιχες τυπικές βιβλιοθήκες, οι δύο γλώσσες είναι διαφορετικές και διαφέρουν πολύ ως προς το σχεδιασμό.

### 3.7 jQuery

Το jQuery είναι μια βιβλιοθήκη JavaScript που έχει σχεδιαστεί για να απλοποιεί τη πρόσβαση και τον χειρισμό του δέντρου HTML DOM, καθώς και τον χειρισμό συμβάντων, το CSS και την τεχνολογία Ajax. Είναι δωρεάν λογισμικό ανοιχτού κώδικα που χρησιμοποιεί την άδεια MIT. Από τον Αύγουστο του 2022, το jQuery χρησιμοποιείται από το 77% των 10 εκατομμυρίων πιο δημοφιλών ιστότοπων. Η ανάλυση Ιστού δείχνει ότι είναι η πιο ευρέως αναπτυγμένη βιβλιοθήκη JavaScript με μεγάλο περιθώριο, έχοντας τουλάχιστον 3 έως 4 φορές περισσότερη χρήση από οποιαδήποτε άλλη βιβλιοθήκη JavaScript.

Η σύνταξη του jQuery έχει σχεδιαστεί για να διευκολύνει την πλοήγηση σε ένα έγγραφο, την επιλογή στοιχείων DOM, τη δημιουργία κινούμενων εικόνων, τον χειρισμό συμβάντων και την ανάπτυξη εφαρμογών Ajax. Το jQuery παρέχει επίσης δυνατότητες στους προγραμματιστές να δημιουργούν πρόσθετα πάνω από τη βιβλιοθήκη JavaScript. Αυτό επιτρέπει στους προγραμματιστές να δημιουργούν αφαιρέσεις για αλληλεπίδραση και κινούμενα σχέδια χαμηλού επιπέδου, προηγμένα εφέ και γραφικά στοιχεία υψηλού επιπέδου, με δυνατότητα θεμάτων. Η αρθρωτή προσέγγιση στη βιβλιοθήκη jQuery επιτρέπει τη δημιουργία ισχυρών δυναμικών ιστοσελίδων και εφαρμογών Ιστού.

Το σύνολο των βασικών χαρακτηριστικών του jQuery—επιλογές στοιχείων DOM, πρόσβαση και μετατροπή— που ενεργοποιήθηκαν από τη μηχανή επιλογής του (με το όνομα "Sizzle" από την έκδοση 1.3), δημιούργησαν ένα νέο "στυλ προγραμματισμού", συνδυάζοντας αλγόριθμους και δομές δεδομένων DOM.



## 4. Ανάλυση του κώδικα της εφαρμογής μας

### 4.1 Εισαγωγή – Ανάλυση

#### 4.1.1 Γενικά

Η εφαρμογή αποτελείται από αρχεία HTML που περιέχουν Javascript και μορφοποιούνται με CSS. Τα αρχεία αυτά είναι είτε σε μορφή απλών αρχείων HTML είτε σε μορφή templates για το Spring Boot. Στο «οικοσύστημα» του Spring Boot περιλαμβάνονται αρχεία HTML (templates) και κλάσεις σε JAVA.

Η εφαρμογή είναι συνδεδεμένη με ένα MySQL server, όπου αποθηκεύονται, ενημερώνονται αλλά και ανακτώνται στοιχεία ανάλογα με το τι κάνει κάθε φορά ο χρήστης.

### 4.2 SPRING BOOT – Model View Controller και η εφαρμογή μας

Το Spring BOOT ακολουθεί την λειτουργία MVC (Model View Controller). Αυτό σημαίνει ότι αποτελείται από πλήρως διακριτά μέρη που ενώνονται κάτω από την «στέγη» του Spring BOOT.

Το **μοντέλο** (Model) είναι η αναλυτική περιγραφή των ξεχωριστών αντικειμένων (κλάσεων) για τα δεδομένα που πρόκειται να χρησιμοποιήσει η εφαρμογή.

Σε αυτή την λογική υπάρχουν οι κλάσεις: ΧΡΗΣΤΗΣ (USER) ΙΑΤΡΟΣ (DOCTOR), ΝΟΣΟΚΟΜΕΙΟ(HOSPITAL) στις οποίες περιγράφονται αναλυτικά οι ενέργειες (μέθοδοι) που μπορούν να εφαρμοστούν σε αντικείμενα των κλάσεων αυτών. Οι κλάσεις του μοντέλου,

συνδέονται με «διάφανη» μέθοδο με την βάση MySQL, έτσι οι διάφορες ενέργειες που απαιτούν «συνομιλία» με την βάση, να μην έχουν την ανάγκη ρητής χρησιμοποίησης SQL queries, αλλά απλά συνομιλίας, διάφανης όπως είπαμε, με το «εξωτερικό» μοντέλο διεπαφής των κλάσεων αυτών, μέσω των μεθόδων τους, και η μέσω του SPRING BOOT transparent στον προγραμματιστή επικοινωνία με την βάση είτε για αναζήτηση είτε για εγγραφή νέων δεδομένων στην βάση π.χ ΕΓΓΡΑΦΗ ΝΕΟΥ ΧΡΗΣΤΗ ή ΙΑΤΡΟΥ.

Το στοιχείο **View**, είναι στην ουσία το αποτέλεσμα που εμφανίζεται στον χρήστη. Μιας και πρόκειται για μια διαδικτυακή εφαρμογή, το View είναι σελίδες HTML (+CSS, JavaScript, jQuery) που εμφανίζονται στον χρήστη καθώς λειτουργεί την εφαρμογή.

Επιλέχθηκε ένα Html Template, έτσι ώστε να υπάρχει ένα ενιαίο look and feel για την εφαρμογή σε όλες τις ιστοσελίδες της. Στις διάφορες οθόνες (screens) που εμφανίζονται στον χρήστη όταν λειτουργεί την εφαρμογή, μπορούν να γίνουν διάφορες ενέργειες. Τότε είναι που το View part, μέσω του controller, θα απευθυνθεί στο model (και τελικά στην MySQL database, που είναι εκείνο αποθηκευμένο τελικά) για να πραγματοποιηθεί η οποιαδήποτε ενέργεια όπως αναζήτηση, εγγραφή, ανάκτηση κάποιων δεδομένων που είναι χρήσιμα την συγκεκριμένη χρονική στιγμή στον χρήστη.

Το στοιχείο **Controller**, όπως προαναφέρθηκε, είναι κώδικας Java που φέρνει σε επικοινωνία το View (html) με το Model (database). Σε ένα μόνο (στην δική μας περίπτωση) αρχείο Java, βρίσκεται όλη η λογική της επικοινωνίας ανάμεσα σε view-model.

Ο controller, θα ερμηνεύσει σωστά τις ενέργειες του χρήστη στις ιστοσελίδες του view (που συνήθως είναι html forms), θα απευθυνθεί στο model για αναζήτηση (ή και εγγραφή) δεδομένων, και θα επιστρέψει στην καθορισμένη ιστοσελίδα (στο view part πάλι δηλαδή) τα αποτελέσματα για να εμφανιστούν τελικά στο χρήστη.

### 4.2.1 Model

Το μοντέλο – model, είναι μια έκφραση των χαρακτηριστικών «εννοιών» ή κλάσεων που περιέχει το πρόγραμμα. Στην περίπτωση μας υπάρχουν οι κλάσεις :

- Χρήστες (user), που περιέχουν προσωπικά στοιχεία των χρηστών της εφαρμογής, αλλά και login-password στοιχεία
- Ιατροί (doctor), που περιέχονται στοιχεία για τους ιατρούς που είναι ενταγμένοι στην εφαρμογή όπως προσωπικά στοιχεία και αλλά για την επικοινωνία με τους ασθενείς
- Νοσοκομεία(hospitals), με στοιχεία για τα υπάρχοντα στην εφαρμογή νοσοκομεία (όπως π.χ. τα νοσοκομεία συγκεκριμένης πόλης)
- Ιστορικό(log), όπου καταγράφονται οι κινήσεις των χρηστών στην εφαρμογή

Ενδεικτικά από τον κώδικα της εφαρμογής παρατίθενται τα παρακάτω screenshots:

```
1 // i klasi doctor perixeia stoixeia gia toys giatroys
2
3 package com.example.demo;
4
5
6 import java.util.Date;
7
8
9
10
11
12
13
14
15
16
17 public class doctor {
18
19     private long id;
20     private String name;
21     private String emailId;
22     private String onlineid;
23     private String special;
24     private String password;
25     private String username;
26     private int onlineflag;
27     private int emailflag;
28     public long getId() {
29         return id;
30     }
31     public void setId(long id) {
32         this.id = id;
33     }
34
35     public String getName() {
36         return name;
37     }
38     public void setName(String firstName) {
39         this.name = firstName;
40     }
41     public String getUsername() {
42         return username;
43     }
44     public void setUsername(String firstName) {
45         this.username = firstName;
46     }
47 }
48
49
50
51
52
53
54
55
56
```

FIGURE 1H ΚΛΑΣΗ DOCTORS

```
3
4 package com.example.demo;
5
6
7 import java.util.Date;
24
25
26 public class userius {
27
28     private long id;
29     private String firstName;
30     private String password;
31     private String emailId;
32     private String sex;
33     private int age;
34     private String address;
35     private String telephone;
36     private String fullname;
37     private int type;
38     public long getId() {
39         return id;
40     }
41     public void setId(long id) {
42         this.id = id;
43     }
44
45     public long getType() {
46         return type;
47     }
48     public void setType(int id) {
49         this.type = id;
50     }
51
52     public String getFirstName() {
53         return firstName;
54     }
55     public void setFirstName(String firstName) {
56         this.firstName = firstName;
57     }
}
```

FIGURE 2 Η ΚΛΑΣΗ USERS

```

1 package com.example.demo;
2
3 // είναι i κλάσι που αντιστοιχεί στην κλάσι doctor
6
7 import java.util.Date;
24
25 @Entity
26 @Table(name = "doctors")
27 @EntityListeners(AuditingEntityListener.class)
28 public class doctor2 {
29
30     private long id;
31     private String name;
32     private String onlineid;
33     private String emailId;
34     private String special;
35     private String password;
36     private String username;
37     private int emailflag;
38     private int onlineflag;
39 @Id
40 @GeneratedValue(strategy = GenerationType.AUTO)
41 public long getId() {
42     return id;
43 }
44 public void setId(long id) {
45     this.id = id;
46 }
47 @Column(name = "emailflag", nullable = false)
48 public int getEmailflag() {
49     return emailflag;
50 }
51 public void setEmailflag(int firstName) {
52     this.emailflag = firstName;
53 }
54
55 @Column(name = "onlineflag", nullable = false)
56 public int getOnlineflag() {
57     return onlineflag;
58 }

```

FIGURE 3 ΠΑΡΑΔΕΙΓΜΑ ΣΥΣΧΕΤΙΣΗΣ ΤΗΣ ΚΛΑΣΗΣ DOCTOR ΜΕ ΤΗΝ ΒΑΣΗ

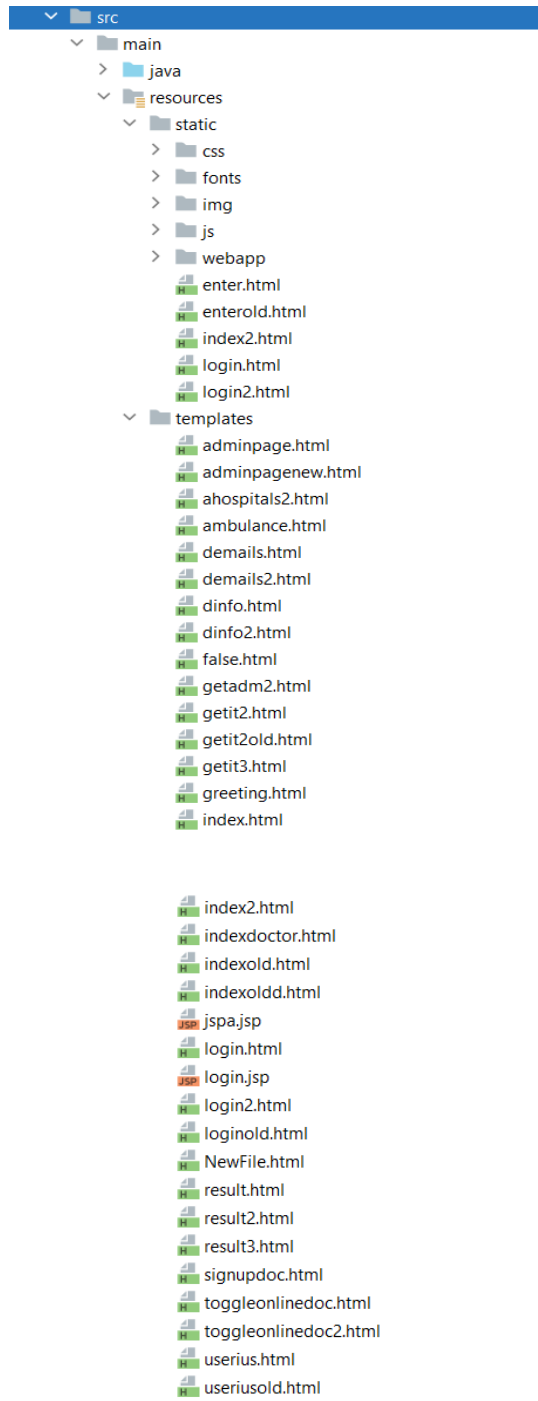
#### ΔΕΔΟΜΕΝΩΝ

## 4.2.2 View

Το στοιχείο **view**, είναι στην ουσία μια συλλογή από στατικά html αρχεία (ενισχυμένα με css, Javascript, αρχεία εικόνων) αλλά και δυναμικά html templates (που παρέχει το module Thymeleaf του Spring Boot), που «δένονται» με ένα html template (για συνολικό ομοιόμορφο feel σε όλες τις ιστοσελίδες).

Στην πραγματικότητα τα παραπάνω αρχεία, εμφανίζουν πληροφορία στον χρήστη, αλλά και μπορεί ο χρήσης να εισάγει πληροφορία (μέσω φορμών) και να επιστρέψουν πάλι σε αυτόν (μέσω ενός html template του Thymeleaf) πίσω τα όποια αποτελέσματα.

Στην παρακάτω εικόνα αποτυπώνεται το σύνολο των στατικών ή μη html files που χρησιμοποιεί η εφαρμογή

**FIGURE 4 ΣΤΑΤΙΚΑ ΚΑΙ ΔΥΝΑΜΙΚΑ HTML ΣΤΟΙΧΕΙΑ**

Εφαρμογή αυτόματης υποστήριξης πολιτών σε θέματα υγείας, με χρήση Spring Boot.



### 4.2.3 Controller

Ένα αρχείο στον κώδικά έχει το ρόλο του Controller, που όπως έχουμε αναφέρει είναι το μέρος εκείνο του προγράμματος που να αναλάβει να «πάρει» την είσοδο από το View, να έρθει σε επαφή με το Model, να κάνει κάποια ενέργεια (ανάκτηση-αποθήκευση-ανανέωση) και να επιστρέψει πάλι τα όποια αποτελέσματα στο View (και στο χρήστη).

Ο κώδικας του controller γίνεται “hooked” σε κάποια συγκεκριμένα calls του χρήστη είτε πρόκειται για html get response είτε για html post response. Λαμβάνει από το View τα δεδομένα που πιθανά δίνει ο χρήστης και επιστρέφει σε κάποιο από τα δυναμικά templates την απόκρισή του, “γεμίζοντας» παράλληλα (πιθανά) αυτά με κάποια πληροφορία που έχει σχέση με την ενέργεια που εκτέλεσε ο χρήστης.

Έτσι λοιπόν έχουμε διάφορες ενέργειες που «ενορχηστρώνονται» από τον controller όπως

- **Login Χρηστών, Ιατρών, Administrator**
- **Signup Χρηστών ,Ιατρών**
- **Ενέργειες χρηστών όπως κλήση ασθενοφόρου, λήψη διαθέσιμου email ή online id ιατρού, λήψη ονόματος νοσοκομείου**
- **Ενέργειες ιατρών όπως ενεργοποίηση/απενεργοποίηση της ηλεκτρονικής συμμετοχής του στο σύστημα**
- **Καταγραφή των κινήσεων χρηστών/ιατρών**
- **Δημιουργία και εμφάνιση στατιστικών για τον διαχειριστή**

### 4.3 Η ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ MySQL

Την διαδικτυακή μας εφαρμογή, υποστηρίζει μια σχεσιακή βάση δεδομένων (RDBMS) που βασίζεται στην MySQL . Η MySQL είναι μια σχεσιακή βάση δεδομένων, ελεύθερη για χρήση και πολύ διαδεδομένη για μικρά (η και μεγαλύτερα) projects, λόγω της ευχρηστίας της αλλά και της ωριμότητας των χαρακτηριστικών που περιλαμβάνει.

Στο MySQL server (που οφείλει να «τρέχει» κατά την διάρκεια εκτέλεσεως της εφαρμογής μας), είναι δημιουργημένοι οι διάφοροι πίνακες (tables) που φιλοξενούν τα χαρακτηριστικά (δεδομένα) που διαχειρίζεται η εφαρμογή.

Το εργαλείο (tool) που χρησιμοποιήσαμε για την δημιουργία (και γενικά την επαφή) με την βάση δεδομένων ήταν το HeidiSQL, ένα front-end που επιτρέπει σε παραθυρικό περιβάλλον την επικοινωνία με το MySQL server για την δημιουργία της βάσης και των πινάκων (tables) χωρίς την συγγραφή κώδικα SQL (Αν και κάποιος θα μπορούσε να κάνει όλες αυτές τις εργασίες και από την κονσόλα – χωρίς την χρήση ενός front-end).

Παρακάτω παρατίθενται screenshots από τους πίνακες και τα δεδομένα της βάσης δεδομένων (πάλι μέσω του HEID)

The screenshot shows the HeidiSQL interface with the 'springsession' database selected. The left sidebar shows the database structure, and the main window displays a table of tables with their properties.

Όνομα	Γραμμές	Μέγεθος	Δημιουργήθηκε	Ενημερώθηκε	Μηχανή	Σχόλιο	Τύπος
doctors	11	2,7 KiB	2022-12-15 18:30:...	2022-12-15 18:30:...	MyISAM		Table
hibernate_seq...	2	16,0 KiB	2022-12-15 18:30:...	2022-12-15 18:30:...	InnoDB		Table
hospitals	6	2,2 KiB	2022-12-15 18:30:...	2022-12-15 18:30:...	MyISAM		Table
log	5	2,2 KiB	2022-12-15 18:30:...	2022-12-15 18:30:...	MyISAM		Table
result	220	15,5 KiB	2022-12-15 18:30:...	2022-12-15 18:30:...	MyISAM		Table
spring_session	0	64,0 KiB	2022-12-15 18:30:...	2022-12-15 18:31:...	InnoDB		Table
spring_session...	5	16,0 KiB	2022-12-15 18:30:...	2022-12-15 18:31:...	InnoDB		Table
users	18	16,0 KiB	2022-12-15 18:30:...	2022-12-15 18:30:...	InnoDB		Table

At the bottom of the screenshot, the SQL command `SHOW PROCEDURE STATUS WHERE `Db`='springsession';` is visible in the command line.

**FIGURE 5ΤΟ ΟΝΟΜΑ ΤΗΣ ΒΑΣΗΣ (SPRINGSESSION) ΚΑΙ ΟΙ ΠΙΝΑΚΕΣ**

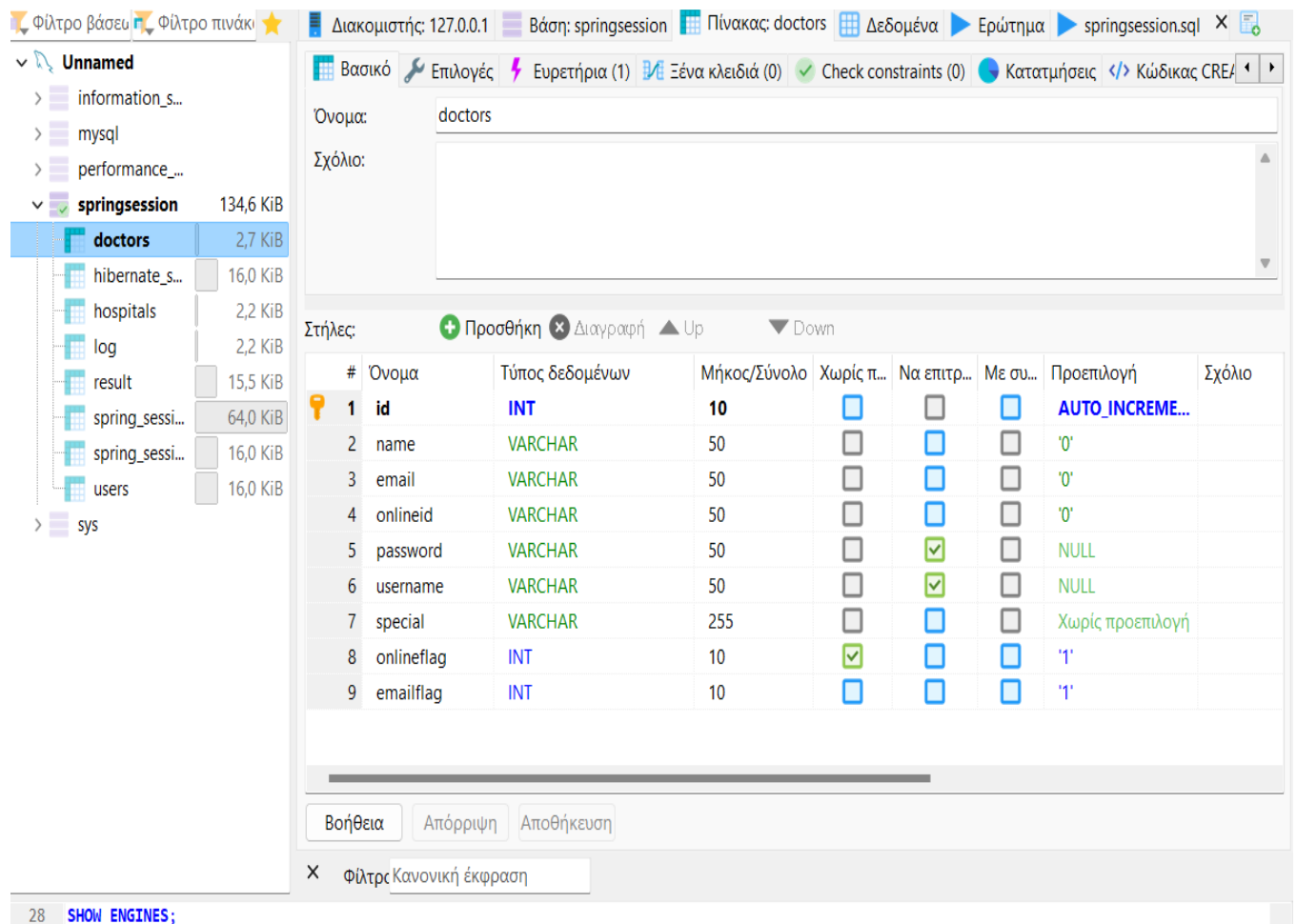


FIGURE 6 Ο ΠΙΝΑΚΑΣ DOCTOR

Εφαρμογή αυτόματης υποστήριξης πολιτών σε θέματα υγείας, με χρήση Spring Boot.

Database: springsession, Table: users

#	Όνομα	Τύπος δεδομένων	Μήκος/Σύνολο	Χωρίς π...	Να επιτρ...	Με συ...	Προεπιλογή	Σχόλιο
1	id	BIGINT	19	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Χωρίς προεπιλο...	
2	email_address	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Χωρίς προεπιλογή	
3	first_name	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Χωρίς προεπιλογή	
4	password	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Χωρίς προεπιλογή	
5	sex	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'	
6	age	INT	10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'	
7	address	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Χωρίς προεπιλογή	
8	telephone	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Χωρίς προεπιλογή	
9	fullname	VARCHAR	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL	
10	type	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL	
11	special	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL	
12	onlineid	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL	

34 SELECT \* FROM information\_schema.REFERENTIAL\_CONSTRAINTS WHERE CONSTRAINT\_SCHEMA='springsession' AND TABLE\_NAME='users' AND REFERENCED\_T/

FIGURE 7 Ο ΠΙΝΑΚΑΣ USERS

Φίλτρο βάσει Φίλτρο πινάκι  
 Διακομιστής: 127.0.0.1 Βάση: springsession Πίνακας: log Δεδομένα Ερώτημα springsession.sql  
 Βασικό Επιλογές Ευρετήρια (1) Ξένα κλειδιά (0) Check constraints (0) Κατατμήσεις Κώδικας CREA

Όνομα: log  
 Σχόλιο:

Στήλες: Προσθήκη Διαγραφή Up Down

#	Όνομα	Τύπος δεδομένων	Μήκος/Σύνολο	Χωρίς π...	Να επιτρ...	Με συ...	Προεπιλογή	Σχόλιο
1	id	INT	10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREME...	
2	type	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL	
3	doctorid	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL	
4	userid	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL	
5	timestamp	DATE		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL	
6	date	VARCHAR	255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL	

Βοήθεια Απόρριψη Αποθήκευση  
 Φίλτρο Κανονική έκφραση

```

40 SELECT * FROM information_schema.REFERENTIAL_CONSTRAINTS WHERE CONSTRAINT_SCHEMA='springsession' AND TABLE_NAME='log' AND REFERENCED_TABL
    
```

FIGURE 8 Ο ΠΙΝΑΚΑΣ LOG

Η επαφή του controller (του SPRING) με την βάση δεδομένων, γίνεται (όπως έχει προαναφερθεί) «διάφανα», χωρίς την χρήση δηλαδή κώδικα SQL. Αυτό είναι ένα (από τα πολλά) πλεονεκτήματα του SPRING BOOT. Το ότι δηλαδή, το connection με την βάση (είτε για εγγραφή-ανάγνωση-ανανέωση) γίνεται ορίζοντας κατάλληλα το μοντέλο, και συνδέοντας το (με κώδικα Java – χωρίς SQL) με την βάση δεδομένων. Το παραπάνω χαρακτηριστικό, «λύνει» κατά έναν τρόπο τα χέρια του developer, επιτρέποντας του να ασχοληθεί με την λογική του προγράμματος του και όχι με την επικοινωνία με την βάση δεδομένων άμεσα, κάνοντας επαναλαμβανόμενες ενέργειες.

Το Spring Boot, θα του παρέχει (αν έχει γίνει σωστά το connection με την Database), τις κατάλληλες μεθόδους Java, για οποιαδήποτε ενέργεια απαιτεί απόκριση από μέρους της βάσης δεδομένων χωρίς την χρήση SQL).

## 4.4 Sessions

Στην εφαρμογή γίνεται χρήση των Spring Sessions, έτσι ώστε το πρόγραμμα (ο server), να γνωρίζει ότι ο χρήστης είναι online και πραγματοποιεί κάποιες ενέργειες (ο ίδιος).

Τα Sessions (την δημιουργία και τον έλεγχο τους) τα αναλαμβάνει το σύστημα, ενώ μέσω Java μπορεί ο προγραμματιστής να ελέγχει αν ο ίδιος χρήστης εξακολουθεί να βρίσκεται σε σύνδεση με την εφαρμογή.

Σε περίπτωση αδράνειας του χρήστη (κατά την συνήθη λογική των sessions) το σύστημα θεωρεί πως ο χρήστης έχει αποσυνδεθεί.

## 5. Η λειτουργία της εφαρμογής μας

### 5.1 Γενικά

Η εφαρμογή μας, είναι ένα on line σύστημα στο οποίο χρήστες (και πιθανά ασθενείς) μπορούν να εγγραφούν και να αναζητήσουν ιατρική βοήθεια είτε μέσω email από τους ιατρούς του συστήματος, είτε με κάποια live μέθοδο, από τους ιατρούς που (εκείνη την στιγμή) είναι διαθέσιμοι.

Ο χρήστης επίσης μπορεί να καλέσει ασθενοφόρο, αλλά επίσης μπορεί να ενημερωθεί για κάποιο διαθέσιμο νοσοκομείο στην περιοχή του.


Ο ιατρός, μπορεί να εγγραφεί στο σύστημα και να ελέγχει την on-line παρουσία του. Αν δηλαδή είναι διαθέσιμος ή όχι στην επικοινωνία μέσω email ή live.

Τέλος, ο διαχειριστής μπορεί να εισέλθει στο σύστημα και να εξάγει διάφορα στατιστικά στοιχεία σχετικά με την κίνηση που έχει το σύστημα στο χρόνο λειτουργίας του.



## 5.2 Ο Χρήστης

Οι χρήστες μπορούν, αρχικά, να εγγραφούν στο σύστημα



## FORM SIGNUP

Fullname

Username

password

email

Address

Telephone

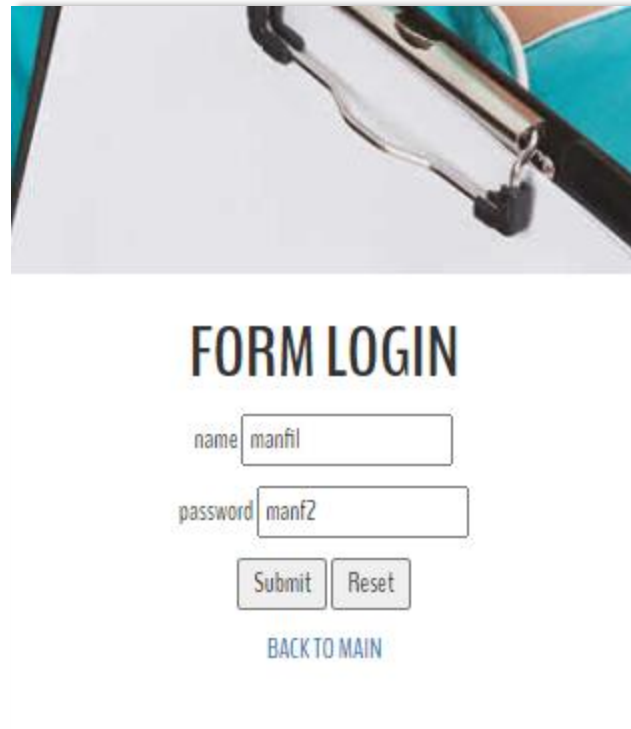
sex

age

[BACK TO MAIN](#)

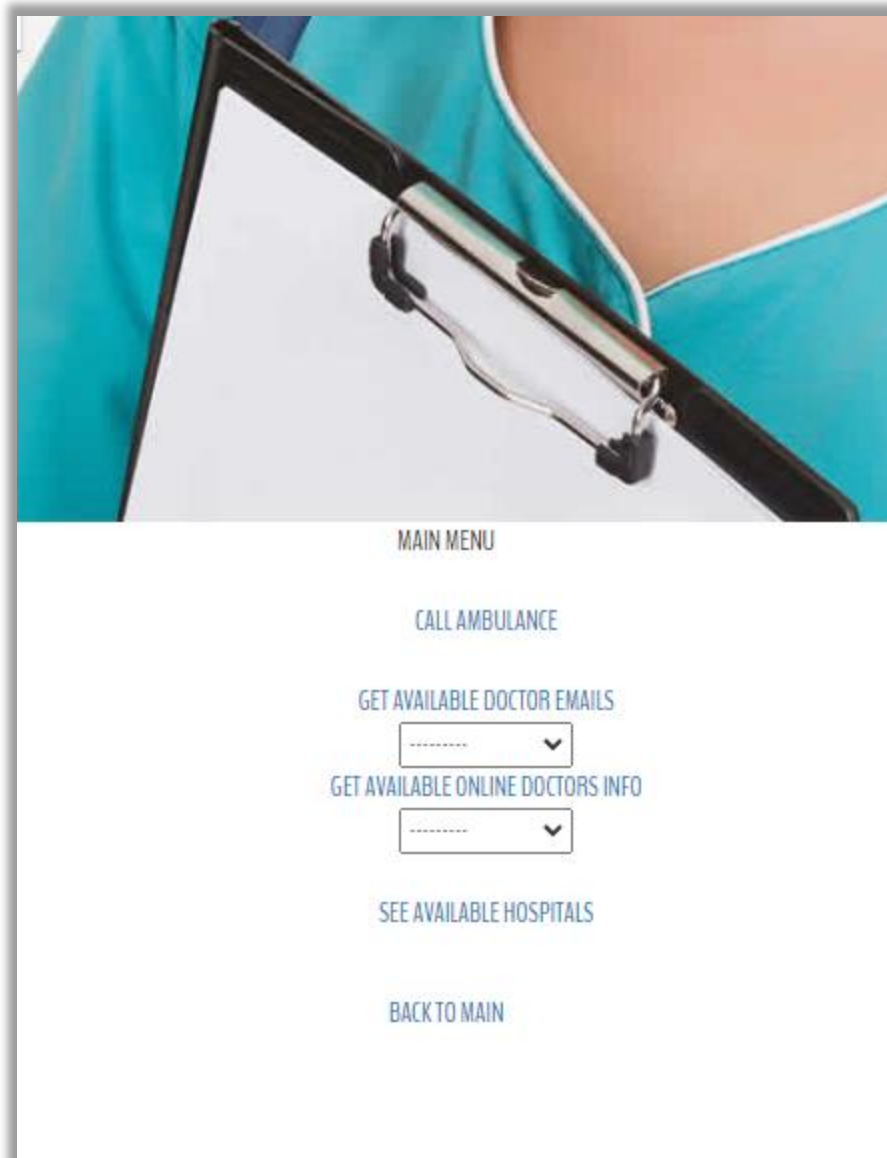
**FIGURE 9 ΕΓΓΡΑΦΗ ΧΡΗΣΤΗ**

Έπειτα ο χρήστης μπορεί να εισέλθει στο σύστημα



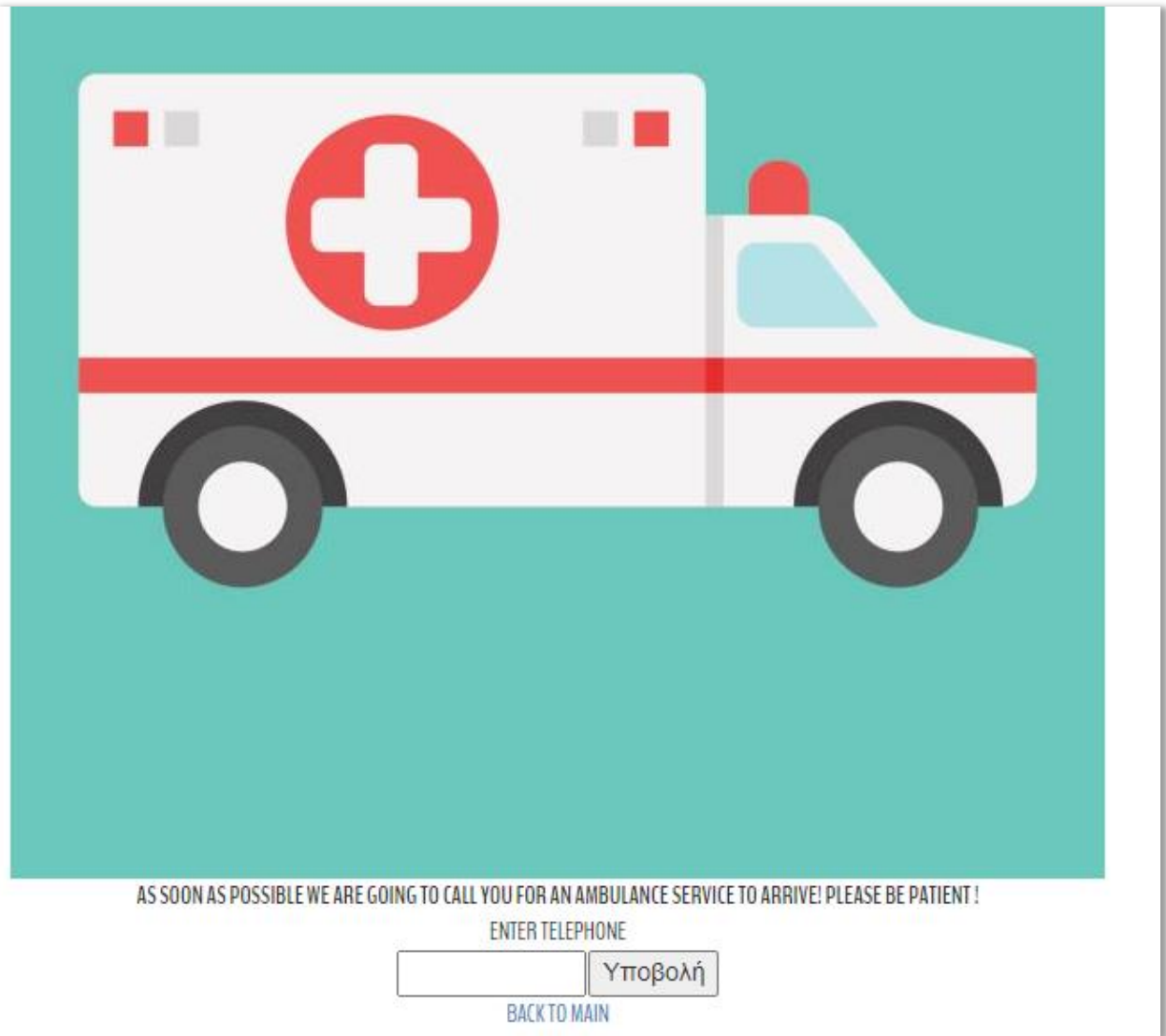
**FIGURE 10 USER LOG IN**

Και του παρουσιάζεται το κεντρικό μενού επιλογών:



**FIGURE 11 ΧΡΗΣΤΗΣ - ΜΕΝΟΥ ΕΠΙΛΟΓΩΝ**

Επιλέγοντας το “CALL AMBULANCE” ο χρήστης, θα οδηγηθεί σε μια σελίδα όπου θα κληθεί να επιβεβαιώσει την θέλησή του για κλήση ασθενοφόρου, μέσω της προτροπής να εισάγει το τηλέφωνό του.



**FIGURE 12 ΧΡΗΣΤΗΣ - ΚΛΗΣΗ ΑΣΘΕΝΟΦΟΡΟΥ**

Ο χρήστης μπορεί από το κεντρικό μενού να επιλέξει είτε το να πάρει τα στοιχεία email ενός ιατρού (που τον ενδιαφέρει) είτε το online id του (όσων ιατρών είναι διαθέσιμων).

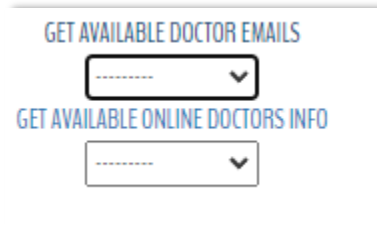
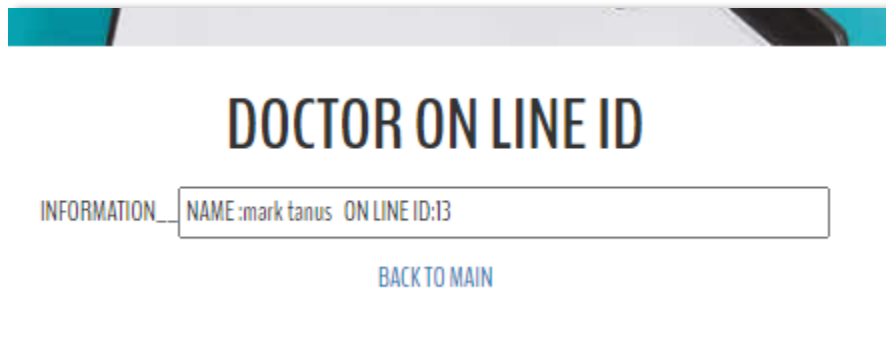


FIGURE 13 ΧΡΗΣΤΗΣ - ΕΠΙΚΟΙΝΩΝΙΑ ΜΕ ΙΑΤΡΟ



FIGURE 14 ΧΡΗΣΤΗΣ - ΛΗΨΗ EMAIL ΙΑΤΡΟΥ



**FIGURE 15 ΧΡΗΣΤΗΣ - ΛΗΨΗ ON LINE ID ΙΑΤΡΟΥ**

Τέλος, ο χρήστης μπορεί να ενημερωθεί για διαθέσιμο νοσοκομείο.

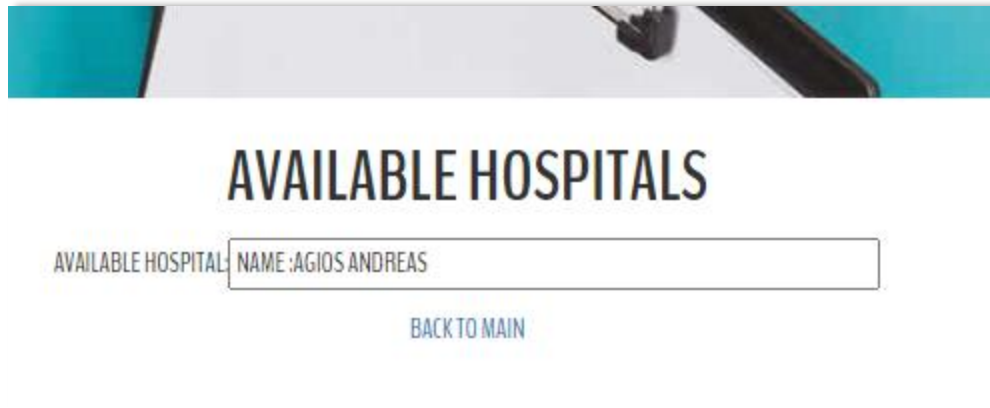
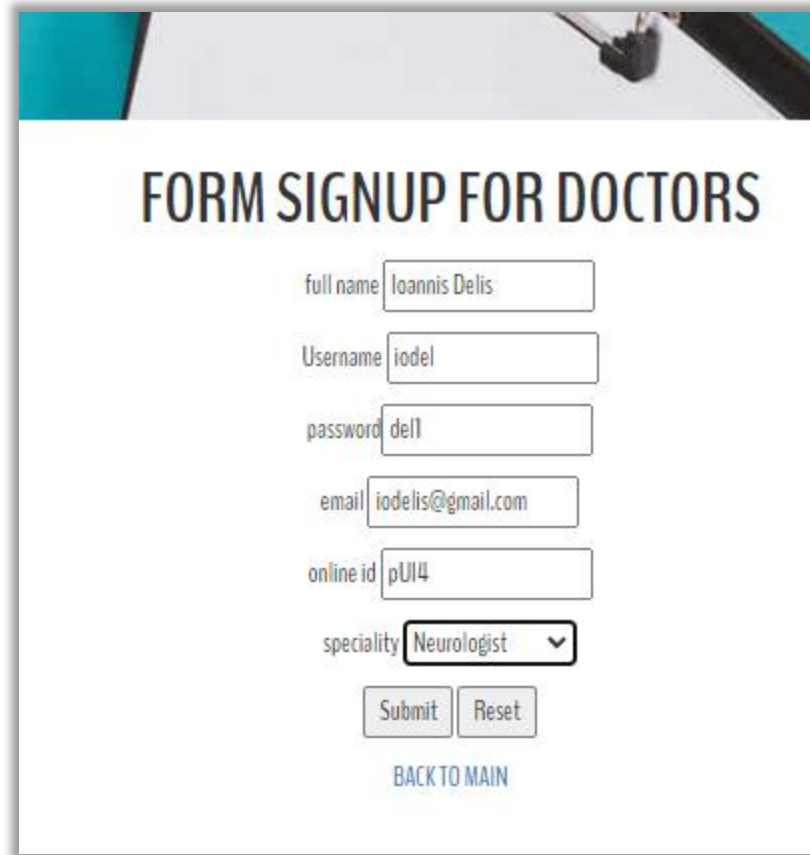


FIGURE 16 ΧΡΗΣΤΗΣ - ΔΙΑΘΕΣΙΜΟ ΝΟΣΟΚΟΜΕΙΟ

### 5.3 Ο Ιατρός

Ο ιατρός αρχικά εγγράφεται στο σύστημα, παρέχοντας στοιχεία όπως η ειδικότητά του, το email του και το on line id του.

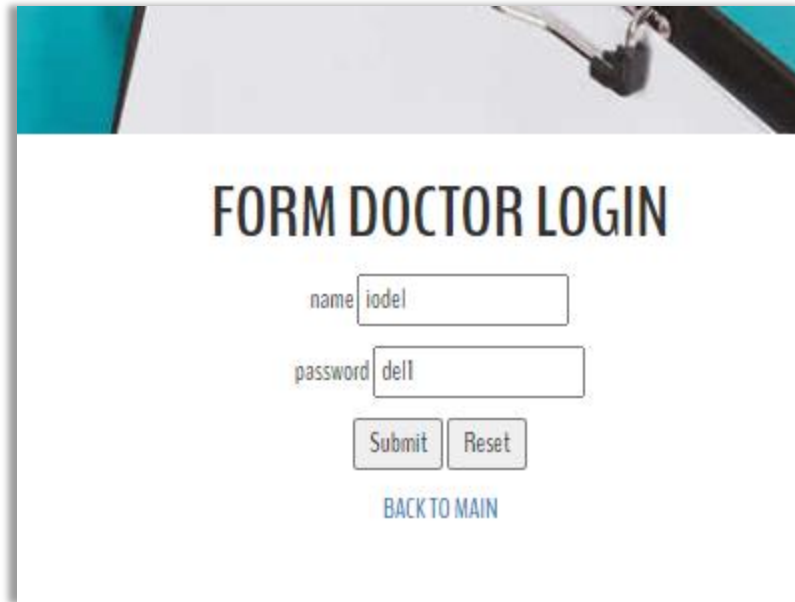




The image shows a web form for doctor registration. The form is titled "FORM SIGNUP FOR DOCTORS" in a large, bold, black font. Below the title, there are several input fields and a dropdown menu, each with a label to its left. The fields are: "full name" with the value "Ioannis Delis", "Username" with "iodel", "password" with "del1", "email" with "iodelis@gmail.com", and "online id" with "pU14". Below these is a dropdown menu for "speciality" with "Neurologist" selected. At the bottom of the form, there are two buttons: "Submit" and "Reset", and a link labeled "BACK TO MAIN" in blue text.

**FIGURE 17 ΙΑΤΡΟΣ – ΕΓΓΡΑΦΗ**

Έπειτα μπορεί να κάνει login:



FORM DOCTOR LOGIN

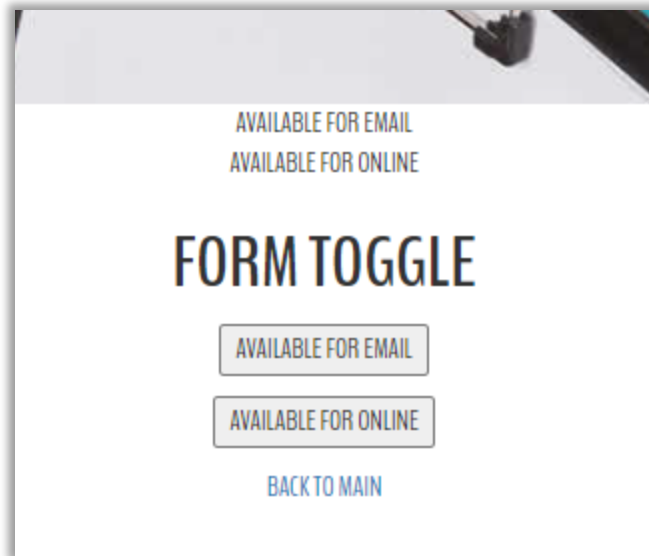
name

password

[BACK TO MAIN](#)

FIGURE 18 ΙΑΤΡΟΣ - LOG IN

Έπειτα εμφανίζεται το menu του ιατρού, στο οποίο φαίνεται η διαθεσιμότητά του σχετικά με το πως μπορεί να εξυπηρετήσει τους ασθενείς. Αν είναι, δηλαδή, διαθέσιμος για επικοινωνία με email ή live (on line) . Ο ιατρός μπορεί να κάνει ενεργοποίηση/απενεργοποίηση των παραπάνω πατώντας τα κουμπιά.



**FIGURE 19 ΙΑΤΡΟΣ - ΔΙΑΘΕΣΙΜΟΤΗΤΑ**

#### 5.4 Ο διαχειριστής (Administrator)

Ο διαχειριστής, μπορεί να εισέλθει στο σύστημα σαν χρήστης (εισάγοντας τα username/pass, τα οποία είναι ενσωματωμένα στην βάση)

Ο διαχειριστής μπορεί να εξάγει στατιστικά στοιχεία σχετικά με την κίνηση του συστήματος από τους χρήστες και τους ιατρούς. Αυτό είναι δυνατόν καθώς, όλες οι κινήσεις των χρηστών του συστήματος (και ποιος τις έκανε) καταγράφονται σε log.

ADMINISTRATOR PAGE		
DESCRIPTION	INFO	VALUE
GENERAL INFO	TOTAL PEOPLE	9
GENERAL INFO	TOTAL MEN	5
GENERAL INFO	TOTAL WOMMEN	4
GENERAL INFO	AT AGE until 10	0
GENERAL INFO	AT AGE until 20	0
GENERAL INFO	AT AGE until 30	0
GENERAL INFO	AT AGE until 40	1
GENERAL INFO	AT AGE until 50	4
GENERAL INFO	AT AGE until 60	4
GENERAL INFO	AT AGE until 70	0
GENERAL INFO	AT AGE until 80	0
GENERAL INFO	AT AGE until 90	0
GENERAL INFO	AT AGE until 100	0
AMBULANCE INFO	TOTAL AMBULANCE CALLERS	1
SEARCHED DOCTORS FOR ONLINE	Cardiologist	1
SEARCHED DOCTORS FOR EMAIL	Cardiologist	5
SEARCHED DOCTORS FOR EMAIL	Psychiatrist	2
SEARCHED FOR HOSPITALS	TOTAL PEOPLE	1

**FIGURE 20 ADMINISTRATOR - STATISTICS**

## 6. Γενική σύνοψη - συμπεράσματα

Στην εργασία αυτή, δημιουργήθηκε μια on line Java εφαρμογή, με χρήση του Spring Boot, για την ηλεκτρονική διαχείριση περιστατικών προβλημάτων υγείας σε μια κοινότητα. Σε αυτήν την εφαρμογή, συμμετέχουν ιατροί και πολίτες, έτσι ώστε οι πρώτοι να δώσουν κατευθύνσεις σε διάφορα περιστατικά υγείας στους δεύτερους. Ο διαχειριστής (που είναι μοναδικός) μπορεί να εξάγει διάφορα στατιστικά στοιχεία για την χρήση του συστήματος (από το site ή από την άμεση επαφή με την MySQL database που υποστηρίζει το πρόγραμμα). Η χρήση τέτοιων συστημάτων στο μέλλον, είτε με την παρούσα απλή μορφή είτε με περισσότερο σύνθετες εφαρμογές, φαίνεται επιβεβλημένη σε έναν κόσμο που ολοένα και πιο πολύ στρέφεται σε διαδικτυακές εφαρμογές για όλες τις πτυχές της καθημερινότητάς του.

## 7. Βιβλιογραφία

1. *Zimmerman, Kathy Ann (September 7, 2017). "History of Computers: A Brief Timeline". [livescience.com](http://livescience.com).*
2. *"General Concepts - Seconds Since the Epoch". [pubs.opengroup.org](http://pubs.opengroup.org).*

3. *Kluver, Randy. "Globalization, Informatization, and Intercultural Communication". un.org.*
4. *Mossberger, Karen (2008). Digital citizenship : the internet, society, and participation. Caroline J. Tolbert, Ramona S. McNeal. Cambridge, Mass.: MIT Press. ISBN 978-0-262-28028-0. OCLC 181030871.*
5. *Hintz, Arne (2019). Digital citizenship in a datafied society. Lina Dencik, Karin Wahl-Jorgensen. Cambridge, UK. p. 43.*
6. *Balkin, Jack (2017-01-01). "The Three Laws of Robotics in the Age of Big Data". Faculty Scholarship Series.*
7. *Fadahunsi KP, O'Connor S, Akinlua JT, Wark PA, Gallagher J, Carroll C, et al. (May 2021). "Information Quality Frameworks for Digital Health Technologies: Systematic Review". Journal of Medical Internet Research. 23 (5): e23479. doi:10.2196/23479. PMC 8167621. PMID 33835034.*
8. *Bhavnani SP, Narula J, Sengupta PP (May 2016). "Mobile technology and the digitization of healthcare". European Heart Journal. 37 (18): 1428–38. doi:10.1093/eurheartj/ehv770. PMC 4914890. PMID 26873093.*
9. *"WHO | Global diffusion of eHealth: Making universal health coverage achievable". WHO.*
10. *O'Donoghue J, Herbert J (1 October 2012). "Data Management within mHealth Environments: Patient Sensors, Mobile Devices, and Databases". Journal of Data and Information Quality. 4 (1): 1–20. doi:10.1145/2378016.2378021. S2CID 2318649.*

11. *Widmer RJ, Collins NM, Collins CS, West CP, Lerman LO, Lerman A (April 2015). "Digital health interventions for the prevention of cardiovascular disease: a systematic review and meta-analysis". Mayo Clinic Proceedings. 90 (4): 469–80. doi:10.1016/j.mayocp.2014.12.026. PMC 4551455. PMID 25841251.*
12. *"Digital health". US Food and Drug Administration. 19 July 2019. Retrieved 23 September 2019.*
13. *Mclaughlin M, Delaney T, Hall A, Byaruhanga J, Mackie P, Grady A, et al. (February 2021). "Associations Between Digital Health Intervention Engagement, Physical Activity, and Sedentary Behavior: Systematic Review and Meta-analysis". Journal of Medical Internet Research. 23 (2): e23180. doi:10.2196/23180. PMC 8011420. PMID 33605897.*