



Project Future_Java 4 Web powered by Code.Hub

Project Specification

Περίληψη

Στα πλαίσια του Java4Web Regeneration Academy ζητείται να υλοποιηθεί το παρόν web application, το οποίο θα βοηθήσει τους συμμετέχοντες να κατανοήσουν, εφαρμόζοντας τις τεχνολογίες που παρουσιάστηκαν στα μαθήματα Databases, Java, Enterprise Development Web Development και Software Engineering.

Γενική περιγραφή project

Ο Ενιαίος Φορέας Κοινωνικής Ασφάλισης (ΕΦΚΑ) χρειάζεται ένα web application που θα δίνει τη δυνατότητα στους πολίτες να κλείνουν ραντεβού με γιατρούς διαφόρων ειδικοτήτων ώστε να προγραμματίζουν τις εξετάσεις τους.

Roles

- Admin (Διαχειριστής Συστήματος)
- Γιατρός
- Πολίτης

Functionalities

- Η **Αρχική Σελίδα** (home page) θα παρουσιάζει ένα login form στο οποίο οι διάφοροι χρήστες θα πρέπει να εισάγουν τα στοιχεία τους (username, password) . Αφου γίνει η κατάλληλη διαπίστευση τότε ανάλογα με το ρόλο του χρήστη το σύστημα θα εμφανίζει αναλογη σελίδα
- **Σελίδα Πολίτη / Ραντεβού:** Θα περιέχει τις εξής λειτουργίες: Δημιουργία Ραντεβού, Ενημέρωση και Διαγραφή και αναζήτηση ραντεβού (στο παρελθόν και στο μέλλον) .
 - **Δημιουργία:** Θα εμφανίζεται φόρμα με τα εξής πεδία: **Ειδικότητα, Γιατρός Ημερομηνία, Ώρα, Σύντομη περιγραφή ασθένειας, Λοιπές παρατηρήσεις.**
 - **Αναζήτηση:** Θα περιέχει τα εξής πεδία προς αναζήτηση: **Ειδικότητα και από-εώς ημερομηνία**

- **Ενημέρωση:** Σε ξεχωριστή σελίδα, θα μπορεί ο πολίτης να μεταβάλλει τα στοιχεία ένα προγραμματισμένου ραντεβού. Το navigation στην σελίδα μπορεί να γίνει με αντίστοιχο link στην σελίδα της αναζήτησης. Παρελθοντικά ραντεβού δε θα επιτρέπεται να μεταβάλλονται
 - **Διαγραφή:** Μέσω της σελίδας της ενημέρωσης, θα παρέχεται ειδικό κουμπί που θα επιτρέπει αυτή τη λειτουργία. Προτείνεται με το πάτημα του κουμπιού να πετάγεται Javascript παράθυρο για επιβεβαίωση της ενέργειας.
- **Σελίδα Γιατρού:** Θα περιέχει τις εξής λειτουργίες: Αναζήτηση και προβολή ραντεβού, προβολή στοιχείων πολίτη.
 - **Αναζήτηση ραντεβού:** Θα περιέχει τα εξής πεδία προς αναζήτηση: **από-εώς ημερομηνία** και ελεύθερη αναζήτηση το πεδίο ασθένειας
 - **Προβολή Ραντεβού :** Σε ξεχωριστή σελίδα, θα μπορεί ο γιατρός να βλέπει τα στοιχεία ενός προγραμματισμένου ραντεβού. Το navigation στην σελίδα μπορεί να γίνει με αντίστοιχο link στην σελίδα της αναζήτησης.
 - **Προβολή Στοιχείων Πολίτη:** Σε ξεχωριστή σελίδα, θα μπορεί ο γιατρός να βλέπει τα στοιχεία του πολίτη / ασθενούς του επιλεγμένου ραντεβού. Η προβολή θα μπορεί να γίνει με αντίστοιχο link στην σελίδα του ραντεβού είτε στην ίδια σελίδα του ραντεβού.
- **Εγγραφή Πολίτη (Sign-up form) :** Θα περιέχει φόρμα με τα εξής πεδία: **ΑΜΚΑ** (χαρακτηρίζει μοναδικά τους χρήστες), **Όνομα, Επίθετο, email, username, password, κινητό τηλέφωνο.** Όλα τα πεδία πρέπει να είναι υποχρεωτικά

Nice to have και όχι υποχρεωτικά

- Validations μέσω Javascript για φόρμες εισαγωγής στοιχείων. Για παράδειγμα, να μην μπορεί να εισαχθεί πολίτης με κενό ΑΜΚΑ.
- Validations στο BackEnd κατά την εισαγωγή στοιχείων: Για παράδειγμα, δεν μπορεί να εγγραφεί πολίτης με email ή ΑΜΚΑ που υπάρχουν ήδη
- Επιπλέον λειτουργικότητες

Διαθεσιμότητα Γιατρού:

ο Γιατρός θα μπορεί να ορίζει ποιες μέρες και ώρες δέχεται ραντεβού. Ιδανικά θα πρέπει να μπορεί να ορίζει αν μια μέρα δέχεται η όχι και τις ώρες για τη συγκεκριμένη ημέρα καθώς και να αλλάζει τις ρυθμίσεις του. Αλλαγές σε ημερομηνίες που έχουν παρέλθει δε θα πρέπει να επιτρέπονται.

Ο Πολίτης εφόσον επιλέξει γιατρό θα πρέπει να μπορεί να βλέπει ποιες ημέρες και ώρες δέχεται ραντεβού και το σύστημα δε τον αφήνει να επιλέξει ώρα/μέρα στις οποίες ο γιατρός δε δέχεται ραντεβού

Slots Ραντεβού

- ο Γιατρός θα μπορεί να ορίζει πόσα ραντεβού μπορεί να εξυπηρετήσει κάθε ώρα (1,2,3 ή 4). Το σύστημα θα πρέπει να δημιουργεί αντίστοιχα slots ραντεβού.
- Ο Πολίτης εφόσον επιλέξει γιατρό και ημερομηνία θα πρέπει να μπορεί να βλέπει ποια διαθέσιμα slots υπάρχουν. Το σύστημα δε θα πρέπει να του επιτρέπει να επιλέξει slot στο οποίο υπάρχει ήδη προγραμματισμένο ραντεβού

Technical Requirements

- Java SE 1.8
- Database: MySQL > 5.5 (Already installed in the Vagrant VM)
- Spring Boot, Spring MVC, Spring Security
- Maven
- Bootstrap
- Javascript/JQuery
- VCS: Git (Github repositories)

Hints

- Χρησιμοποιήστε Spring Initializr (<https://start.spring.io/>) για να δημιουργήσετε ένα άδειο Java Project που θα χρησιμοποιήσετε ως codebase. Για την ανάπτυξη του FE, προτείνεται η χρήση του Bootstrap Framework (<http://getbootstrap.com/>).
- Ωστόσο, δε χρειάζεται να περιοριστείτε σε αυτό μόνο, μπορείτε και για την ακρίβεια σας ενθαρρύνουμε να εμπλουτίσετε την εφαρμογή σας με δικά σας styles σε συνδυασμό με το Bootstrap ή να χρησιμοποιήσετε ένα εντελώς διαφορετικό της αρεσκείας σας. Παρ' όλα αυτά σας συμβουλεύουμε να μην αναλωθείτε στο styling των σελίδων αυτό καθ' αυτό.
- Προτείνεται breakdown των tasks στα διάφορα μέλη της ομάδας με τρόπο που θα συζητηθεί (functionality, component κτλ) και η χρησιμοποίηση Trello board για καλύτερη διαχείριση κι εποπτεία.

- Χρησιμοποιήστε github για να γίνεται σωστά το code versioning και sharing μεταξύ των μελών της κάθε ομάδας.
- Εστιάστε στα απολύτως απαραίτητα και μόνο σε αυτά που περιγράφονται.

Milestones

Τα milestones δίνονται κυρίως σαν ένας λογικός διαχωρισμός των διαφόρων tasks και θα μας βοηθήσουν να κατανείμουμε χρονικά τα διάφορα στάδια υλοποίησης. Δεν είναι υποχρεωτική η τήρησή τους, αλλά προτείνεται να ακολουθούνται, ώστε να υπάρχει μια ένδειξη της προόδου του παραδοτέου.

- 6/12: Προετοιμασία του σχήματος της βάσης (DB Entities model) και του structure του webapp και import σε κάποιο IDE (π.χ IntelliJ) καθώς και δημιουργία Github repository και accounts των μελών της ομάδας.
- 11/12: Υλοποίηση login / registration functionality
- 19/12: Υλοποίηση των αρχικών σελίδων πολίτη και γιατρού