IMedical

Questo progetto nasce con la finalità di migliorare l’interazione tra medico e paziente, ottimizzando l’esecuzione di alcune attività normalmente svolte dagli utenti. La web application iMedical si propone di gestire la routine di uno studio medico: la gestione delle visite, quella delle prescrizioni e della cartella clinica, sia lato paziente che lato medico, oltre anche ad altre funzioni.

Il gruppo che ha sviluppato la web application è composto da tre persone: Tiziano Bari, Gianluca Capozzi e Gerardo Audino e il lavoro è stato suddiviso in base ai ruoli dell’applicazione:

1. Dottore, Paziente e Segretario curato da Tiziano Bari
2. Paziente, Proprietario e Autenticazione curato da Gianluca Capozzi
3. Segretario curato da Gerardo Audino

Per quanto riguarda la distinzione dei ruoli una volta autenticati si è scelto di utilizzare la gemma cancancan (aspetto curato da Tiziano Bari).

In allegato alla relazione si possono trovare il diagramma ER, lo schema di autorizzazione e il mockup dell’applicazione.

Il sistema prevede:

* L’utente può registrarsi (autenticarsi) come proprietario, paziente, segretario o medico
* La possibilità di visualizzare le prescrizioni mediche
* Gestione degli appuntamenti
* Lista degli studi medici e dei dottori associati a tali studi medici

In particolare, un *paziente* deve essere in grado di:

* Visualizzare la terapia a lui prescritta
* Prenotare una visita presso uno studio medico tra quelli indicati e poterla anche annullare
* Effettuare una ricerca dei medici all’interno di un determinato studio medico

Un *medico* deve essere in grado di:

* Accedere agli studi medici a cui è associato
* Accedere alle informazioni e ai dati personali del paziente
* Prescrivere terapie per i pazienti
* Visualizzare la terapia da lui prescritta
* Visualizzare la lista dei suoi pazienti di una determinata clinica precedentemente selezionata

Un *segretario*, invece, deve essere in grado di:

* Visualizzare e selezionare uno studio medico
* Cercare un medico associato allo studio precedentemente selezionato
* Gestire gli appuntamenti tra medico e paziente

Infine, il *proprietario* deve essere in grado di:

* Creare un nuovo studio ­­­medico e aggiungere i relativi dipendenti
* Gestire (ossia creare o modificare) gli studi medici da lui creati
* Associare un medico ad un determinato studio medico da lui creato
* Associare un segretario ad un determinato studio medico da lui creato

**USER STORIES iMedical**

**SEZIONE SIGNUP E LOGIN**

1. As an UNREGISTERED USER I want to LOGIN with E-MAIL so that I can BECOME A REGISTERED USER
2. As an UNREGISTERED USER I want to LOGIN with GOOGLE so that I can BECOME A REGISTERED USER
3. As a REGISTERED USER I want to CHOOSE MY ROLE so that I can BECOME A REGISTERED DOCTOR OR A REGISTERED SECRETARY OR A REGISTERED PATIENT OR A REGISTERED OWNER

**SEZIONE IMPOSTAZIONI GENERALI**

1. As a REGISTERED USER I want to HAVE SETTINGS so that I can CHANGE MY E-MAIL
2. As a REGISTERED USER I want to HAVE SETTINGS so that I can CHANGE MY PASSWORD
3. As a REGISTERED USER I want to HAVE SETTINGS so that I can CHANGE MY PHONE  NUMBER
4. As a REGISTERED USER I want to HAVE SETTINGS so that I can SET MY NAME
5. As a REGISTERED USER I want to HAVE SETTINGS so that I can SET MY SURNAME
6. As a REGISTERED USER I want to HAVE SETTINGS so that I can SET MY CF
7. As a REGISTERED USER I want to HAVE SETTINGS so that I can SET MY DATE OF BIRTH
8. As a REGISTERED USER I want to HAVE SETTINGS so that I can SET MY PLACE OF BIRTH

**SEZIONE IMPOSTAZIONI DOTTORE**

1. As a REGISTERED DOCTOR I want to HAVE SETTINGS so that I must SET MY MEDICAL IDENTIFY NUMBER

**SEZIONE OPERAZIONI PAZIENTE**

1. As a REGISTERED PATIENT I want to SEE MY MEDICAL HISTORY so that I can HAVE INFORMATION ABOUT MY HEALTH
2. As a REGISTERED PATIENT I want to SEE MY LAST MEDICAL EXAMINATION PRESCRIPTIONS so that I can HAVE INFORMATION ABOUT MY LAST MEDICAL EXAMINATION PRESCRIPTIONS
3. As a REGISTERED PATIENT I want to SEE MY LAST DRUGS PRESCRIPTIONS so that I can HAVE INFORMATION ABOUT MY LAST DRUGS PRESCRIPTIONS
4. As a REGISTERED PATIENT I want to SEARCH A DOCTOR so that I can HAVE INFORMATION ABOUT THE CHOSEN DOCTOR
5. As a REGISTERED PATIENT I want to SEARCH A DOCTOR so that I can BOOK AN APPOINTMENT
6. As a REGISTERED PATIENT I want to SEE MY APPOINTMENTS so that I can DELETE A BOOKED APPOINTMENT

**SEZIONE OPERAZIONI DOTTORE**

1. As a REGISTERED DOCTOR I want to HAVE ACCESS TO THE LIST OF THE PATIENTS so that I can CHOOSE A PATIENT
2. As a REGISTERED DOCTOR I want to SEE A PATIENT PROFILE so that I can HAVE INFORMATION ABOUT THE CHOSEN PATIENT
3. As a REGISTERED DOCTOR I want to SELECT DRUGS OR A MEDICAL EXAMINATION so that I can MAKE A PRESCRIPTION TO THE CHOSEN PATIENT

**SEZIONE OPERAZIONI PROPRIETARIO**

1. As a REGISTERED OWNER I want to MAKE OPERATIONS so that I can INSERT A NEW MEDICAL CENTER
2. As a REGISTERED OWNER I want to SELECT A MEDICAL CENTER so that I can MODIFY INFO
3. As a REGISTERED OWNER I want to SELECT A MEDICAL CENTER so that I can DELETE IT
4. As a REGISTERED OWNER I want to SELECT A MEDICAL CENTER so that I can INSERT A NEW  REGISTERED DOCTOR
5. As a REGISTERED OWNER I want to SELECT A MEDICAL CENTER so that I can DELETE AN  ASSOCIATED REGISTERED DOCTOR
6. As a REGISTERED OWNER I want to SELECT A MEDICAL CENTER so that I can INSERT A NEW  REGISTERED SECRETARY
7. As a REGISTERED OWNER I want to SELECT A MEDICAL CENTER so that I can DELETE AN  ASSOCIATED REGISTERED SECRETARY

**SEZIONE OPERAZIONI SEGRETARIO**

1. As a REGISTERED SECRETARY I want to HAVE SETTINGS so that I can CHOOSE MY CURRENT DOCTOR’S OFFICE
2. As a REGISTERED SECRETARY I want to HAVE SETTINGS so that I can CHOOSE A DOCTOR
3. As a REGISTERED SECRETARY I want to ACCESS APPOINTMENTS CALENDAR so that I can  MANAGE DOCTOR’S SCHEDULE.
4. As a REGISTERED SECRETARY I want to ACCESS APPOINTMENTS CALENDAR so that I can  MAKE AN APPOINTMENT
5. As a REGISTERED SECRETARY I want to ACCESS APPOINTMENTS CALENDAR so that I can  DELETE A SELECTED APPOINTMENT
6. As a REGISTERED SECRETARY I want to ACCESS APPOINTMENTS CALENDAR so that I can  MODIFY A SELECTED APPOINTMENT

**Realizzazione**

**Data layer**

Il livello data layer è implementato tramite sqlite3. L’interrogazione delle tabelle (attraverso le query SQL) è affidata a Rails, tramite l’uso degli Active Records, mentre la creazione delle tabelle viene effettuata tramite le migrazioni (con le migrazioni si possono creare/eliminare tabelle, gestire vincoli di tupla e di integrità referenziale, indipendentemente dal DBMS utilizzato). I vincoli tra le diverse relazioni vengono specificati nei vari model, attraverso delle direttive del tipo has\_many oppure belongs\_to. Nei model sono inseriti anche controlli sugli attributi inseriti: questo avviene grazie alle direttive di tipo validates. Le tabelle create per eseguire l’applicazione sono le seguenti:

* Users:
  + Name
  + Surname
  + Email
  + Password\_digest
  + BirthdayDate
  + BirthdayPlace
  + PhoneNumber
  + Codice Fiscale
  + DoctorID
* Clinics:
  + Owner\_id (foreign-key)
  + Name
  + Address
  + Description
  + Province
  + City
  + Latitude
  + Longitude
* Owners
* Doctors
* Patients
* Secretaries
* Manages
  + Secretary\_id (foreign-key)
  + Clinic\_id (foreign-key)
* Examinations
  + Patient\_id (foreign-key)
  + Doctor\_id (foreign-key)
  + Clinic\_id (foreign-key)
  + Start\_time
* Works
  + Doctor\_id (foreign-key)
  + Clinic\_id (foreign-key)
  + Day
  + Start\_time
  + End\_time
* Prescriptions
  + Examination\_id (foreign-key)
  + Type
  + Comment
  + drugName
* Drugs
* Prescripted Examinations

**Application Layer**

Nell’application layer si trova tutta la logica dell’applicazione, come per esempio i vincoli della specifica riguardanti il login e anche i vincoli esterni. I vincoli (di ogni tipologia) vengono implementati all’interno dei controller e dei modelli. Rails dà la possibilità di definire condizioni che vengono eseguite solo in casi specifici (come le azioni di tipo before\_save oppure before\_create).

Oltre alle operazioni descritte precedentemente, l’application layer comprende anche altre funzioni, come quelle qui di seguito descritte:

* User
  + Default user routes
* Owner
  + Default owner routes
  + #GET users/:id/newOwner => new\_owner
  + #GET /owner/:owner\_id/clinics/showClinicsForDoctor => show\_clinics\_for\_doctor
  + #GET /owner/:owner\_id/clinics/showClinicsForSecretary => show\_clinics\_for\_secretary
  + #GET /owner/:owner\_id/clinics/:id/searchDoctor => search\_doctor
  + #GET /owner/:owner\_id/clinics/:id/searchSecretary => search\_secretary
  + #GET /work/addNewDoctor/:doctor\_id&:clinic\_id => add\_new\_doctor
  + #GET /manage/addNewSecretary/:secretary\_id&:clinic\_id => add\_new\_secretary
* Doctor
  + Default doctor routes
  + #GET users/:id/setDoctorID => set\_doctorID
  + #PATCH users/:id/setDoctorID => patch\_doctorID
* Patient
  + Default patient routes
  + #GET /patient/searchClinic => search\_clinic
* Secretary
  + Default secretary routes
* Clinic
  + Default clinic routes
* Examination
  + Default clinic routes
* Manages
  + Default manage routes
* Work
  + Default work routes
* Prescription
  + Default prescription routes
  + #GET /searchDrug => search\_drug
* Drug
  + Default drug routes
* Prescripted examination
  + Default prescripted examination routes

**Presentation Layer**

Il presentation layer è scritto in erb (Embedded Ruby) in modo da poter utilizzare codice Ruby all’interno e sfruttare le variabili presenti nel controller per personalizzare le pagine. Il codice ERB permette di ottenere delle comuni pagine HTML (e, se necessario, unite a codice JavaScript e CSS). Tutte le pagine mostrate nell’applicazione si trovano nella cartella views mentre il codice JavaScript e CSS si trovano nella cartella assets. Ogni azione specificata nel controller ha una sua view dedicata.

In alcuni casi il contenuto mostrato nella pagina può variare in base al tipo di utente che accede ad essa; questo viene ottenuto tramite il settaggio della sessione (nella quale si va a memorizzare la tipologia di utente) e grazie all’uso della gemma cancancan.

**Testing**

Nelle cartelle features e spec sono implementati i test da effettuare utilizzando rispettivamente Cucumber e RSpec. I test effettuati con Cucumber effettuano la validazione del processo di prenotazione di una visita da parte di un paziente. I test effettuati con RSpec invece vanno a verificare alcuni metodi inerenti al modello user (i metodi testati si riferiscono ad un utente che ancora non ha selezionato il suo ruolo) e metodi inerenti alle visite mediche. Per lanciare questi test è necessario eseguire da terminale i due seguenti comandi:

1. bundle exec cucumber
2. rspec spec