<u>Doctor-Patient Interaction Platform</u>

Miu Daria
Neag Dragos

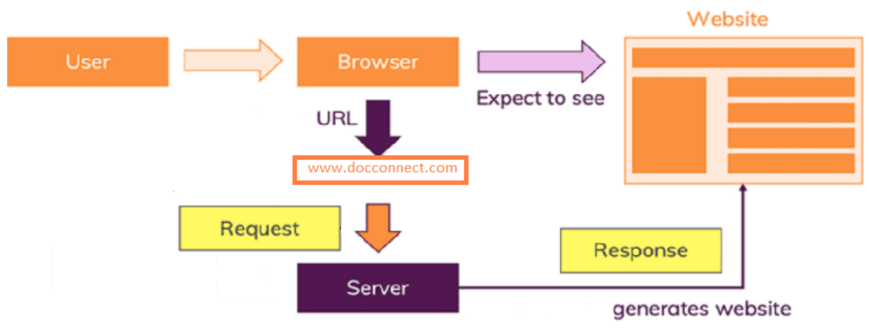
1. General Presentation

As the population of the world continues to grow, so does the need for healthcare services. This demand for healthcare, in turn, will increase the number of patients seeking care at medical facilities, hospitals. While the patient numbers increase, it builds new tasks for facility staff. This means that the processes and procedures that previously were adequate may no longer be efficient in handling new patients. This is an application that offers a modern solution to make appointments to the doctor and to keep medical records and prescriptions. The application serves as a useful tool for both patients and doctors.

2. Theoretical Fundamentals

Our application is going to be a website. Here are the theoretical fundamentals behind a website functioning.

- The user enters the browser
- The user enters the URL
- The URL gets resolved
- A Request is sent to the server of the website
- The response of the server is sent to the website
- The page is displayed



3. IT Technology

The application will be an app implemented using Java. As a framework, we will use Spring Boot + Maven. As a front-end tool we will use React. As our goal is to design an app capable of administrating a database, Java provides a convenient way of building such an app. The IDE used for developing the project will be IntelliJ IDEA. Also, for the database we'll be done in DataGrip using MySQL. This type of database covers our necessities as it is able to function smoothly with the backend of our app.

Java is a general-purpose, robust, secure, and object-oriented programming language. It is a high-level language, I.e., its syntax uses English like language. It was developed by Sun Microsystems in the year 1995. It is now maintained and distributed by Oracle. Java has its runtime environment and API; therefore, it is also called a platform.

Spring Framework is a Java platform that provides comprehensive infrastructure support for developing Java applications. Spring handles the infrastructure so you can focus on your application. Spring's focus on speed, simplicity, and productivity has made it the world's most popular Java framework.

MySQL is a relational database management system based on SQL – Structured Query Language. The application is used for a wide range of purposes, including data warehousing, e-commerce, and logging applications. MySQL, one can easily manipulate or modify the database files during run time using binaries.

React was designed to make it super easy to create interactive UIs. Its state management is efficient and only updates components when the data changes. Component logic is written in JavaScript.

4. Functionalities

The patient can:

- create an account
- make appointments with different doctors: specialty -> doctor -> date & time
- see his/her appointments in the account.
- edit account details
- see medical record and prescriptions and recommendations

The doctor can:

- create an account
- see his/her appointments
- see each patient's medical record
- write in each patient's medical record
- edit account details

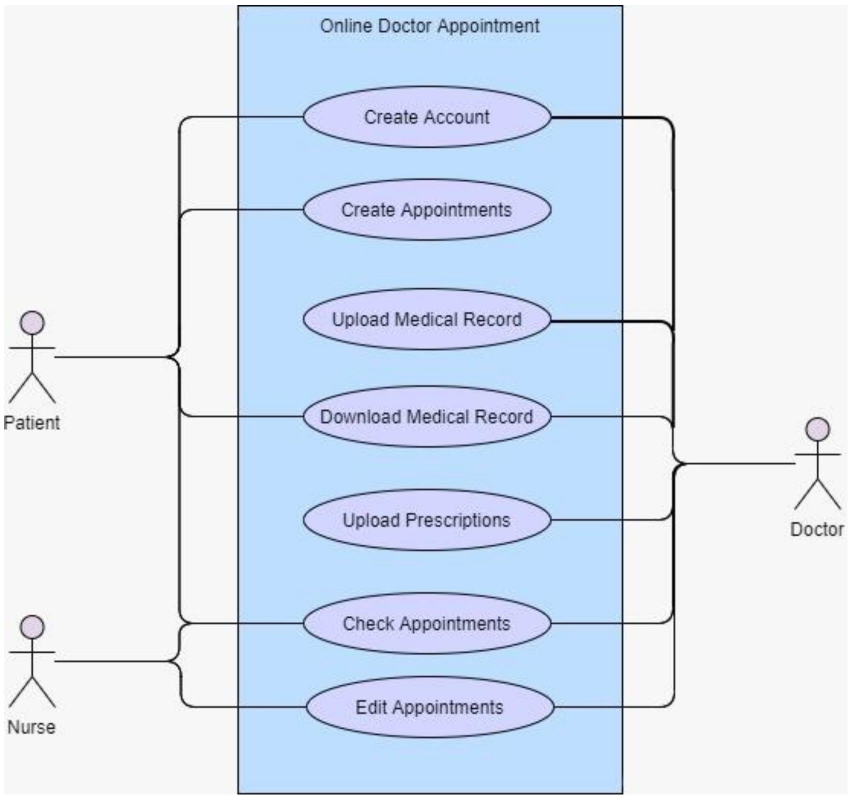
Nurse:

- validates appointments
- edit account details

5. Actors and related access rights

Actor	***Type***	***Description***
Patient	*Human*	*The client that can create an account to make appointments, upload medical investigations, and download medical records and prescriptions*
Doctor	*Human*	*Can create an account, accept appointments, see medical records, write prescriptions, and request medical investigations*
Nurse	*Human*	*The person administrating the appointments*

6. Use Case Diagrams



1.1.1 Use Case for Patient

Description: register patient

Actor: Patient
Precondition: -
Postcondition: -

Behavior:
Step1. The patient presses the “register” button
Step2. The patient inserts the fields of the client
Step3. The patient presses the “register” button
Step4. The manager inserts the client into the memory.

1.1.2 Use Case for Patient

Description: appointment making

Actor: Patient
Precondition: -
Postcondition: -

Behavior:

Step1. The patient logs into the application
Step2. The patient selects a specialty by clicking on it
Step3. The patient presses the “next” button
Step4. The patient selects a doctor
Step5. The patient presses the “next” button
Step6. The patient selects a date and time
Step7. The patient presses the “finish” button
Step8. The manager inserts the appointment into the memory.

1.1.3 Use Case for Doctor

Description: medical record access

Actor: Doctor
Precondition: -
Postcondition: -

Behavior:

Step1. The doctor logs into the application
Step2. The doctor presses the “current patients” button
Step3. The doctor selects a patient
Step4. The doctor presses the “see medical record” button
Step5. The doctor presses the “download” button

Description: view appointments
Actor: Doctor
Precondition: -
Postcondition: -
Behavior:
Step1. The doctor logs into the application
Step2. The doctor presses the “view appointments” button
Step3. The doctor selects a date
Step4. The doctor presses the “view” button

1.1.4 Use Case for Nurse

Description: appointment managing

Actor: Nurse
Precondition: -
Postcondition: -

Behavior:

Step1. The nurse logs into the application
Step2. The nurse presses the “new appointments” button
Step3. The nurse selects an appointment
Step4. The nurse can press “modify” button to modify the date and time
Step5. The nurse presses the “confirm” button

Description: appointment delete
Actor: Nurse
Precondition: -
Postcondition: -
Behavior:
Step1. The nurse logs into the application
Step2. The nurse presses the “confirmed appointments” button
Step3. The nurse selects an appointment
Step4. The nurse can press “delete” button
Step5. The nurse presses the “confirm” button

7. System Architecture

The system will have a layered architecture in which each layer communicates with the layer directly below or above it (hierarchical structure).

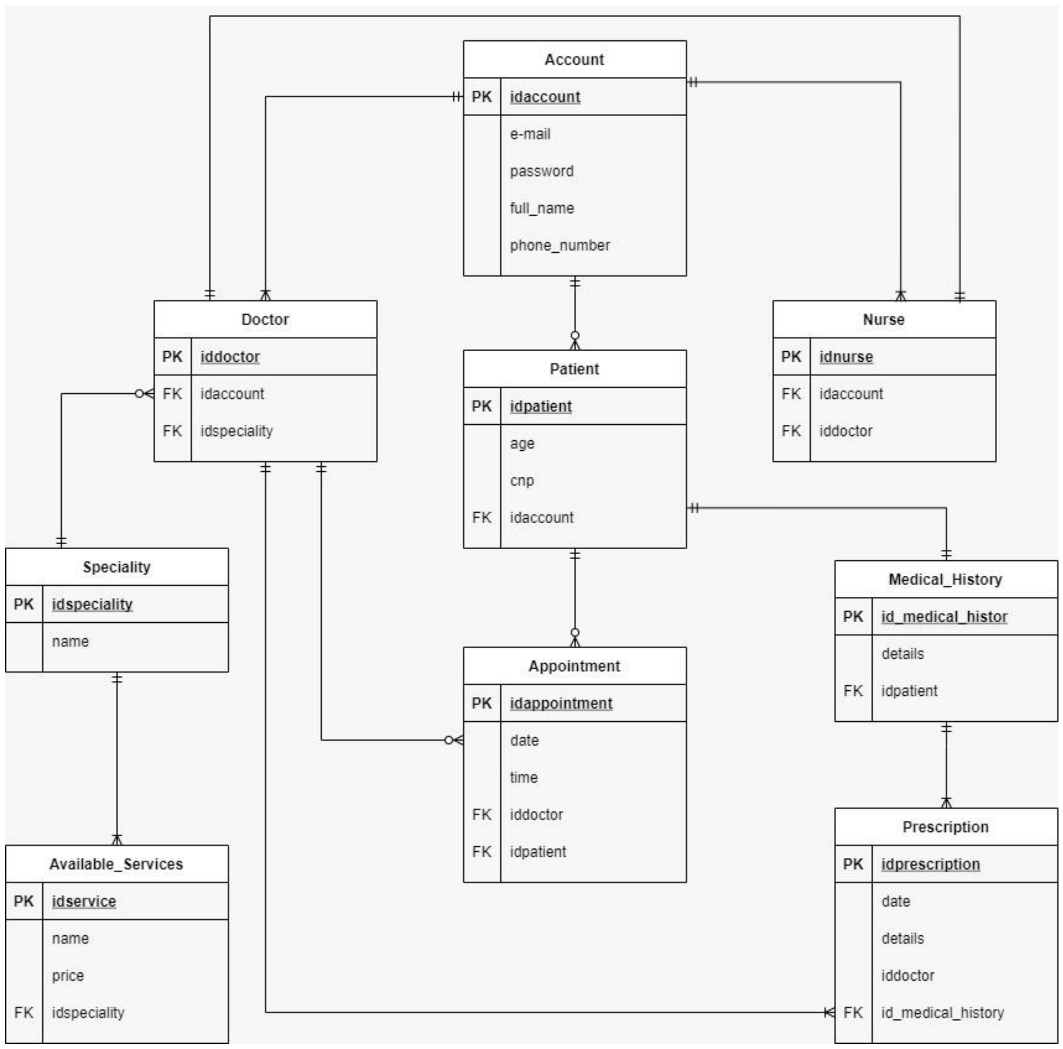
Presentation Layer: The presentation layer handles the HTTP requests, translates the JSON parameter to object, and authenticates the request and transfer it to the business layer. In short, it consists of **views**.

Business Layer: The business layer handles all the **business logic**. It consists of service classes and uses services provided by data access layers. It also performs **authorization** and **validation**.

Persistence Layer: The persistence layer contains all the **storage logic** and translates business objects from and to database rows.

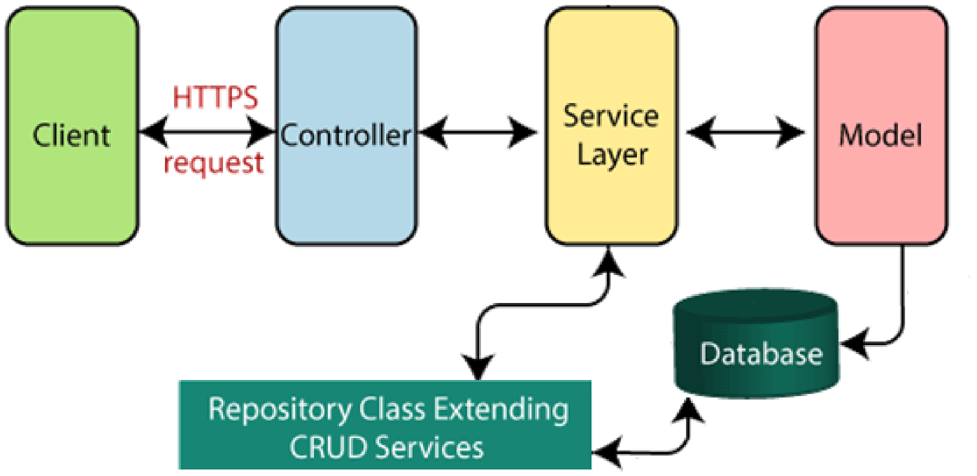
Database Layer: In the database layer, **CRUD** (create, retrieve, update, delete) operations are performed.

8. Design

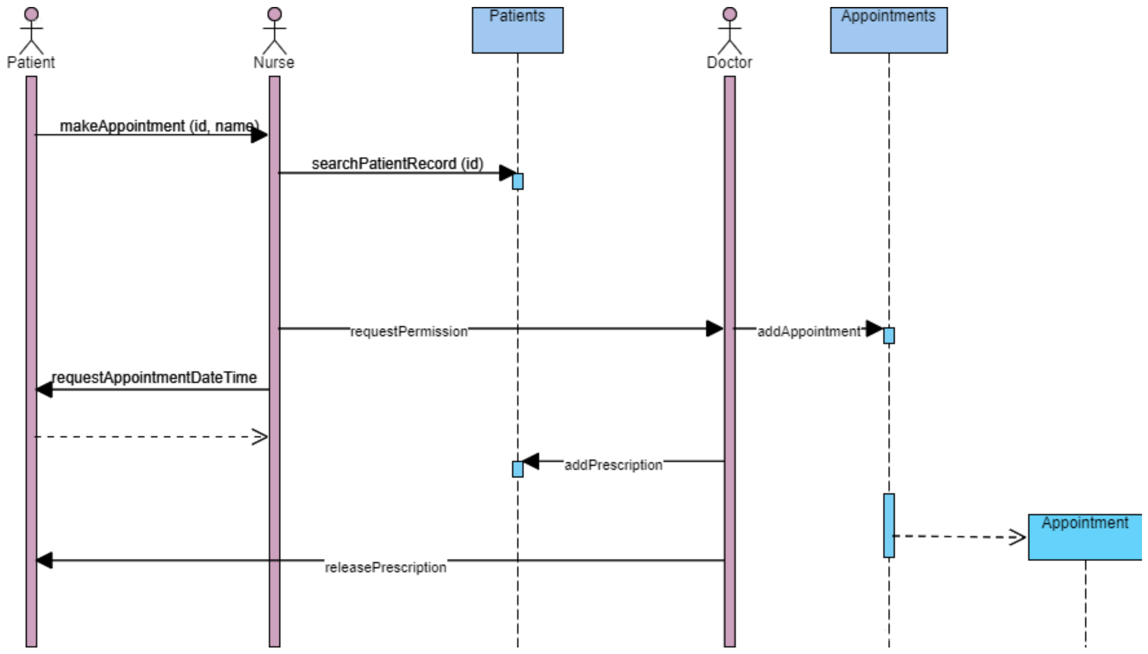


Database Diagram

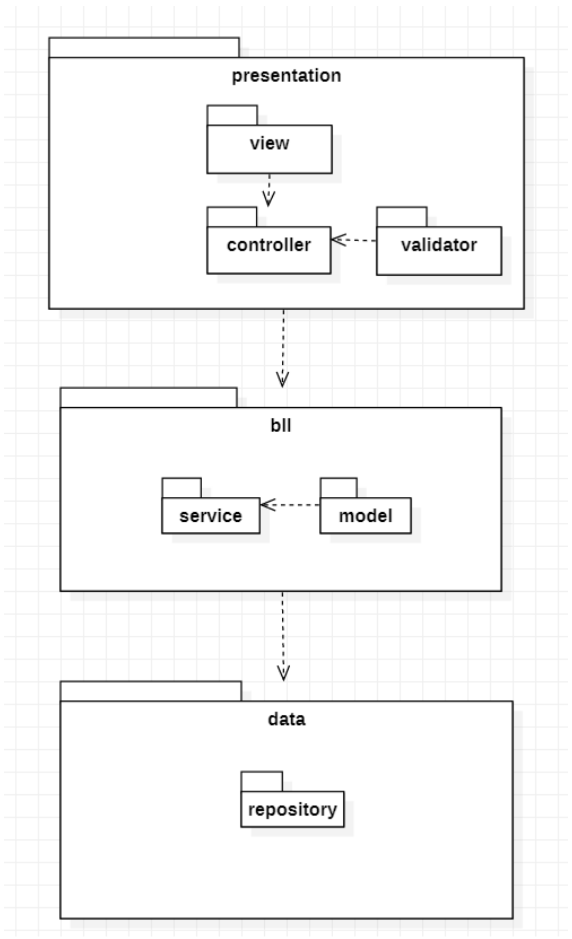
Architecture Diagram



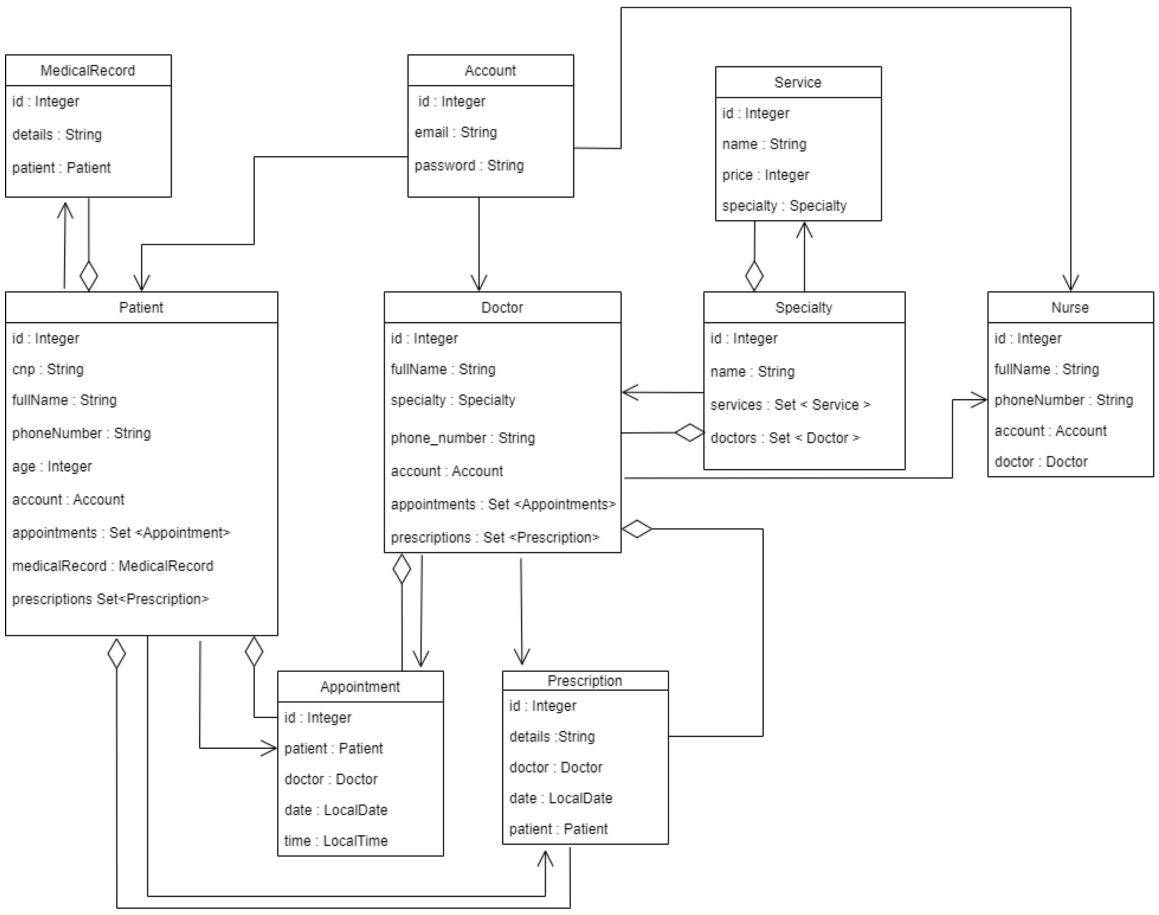
Sequence Diagram



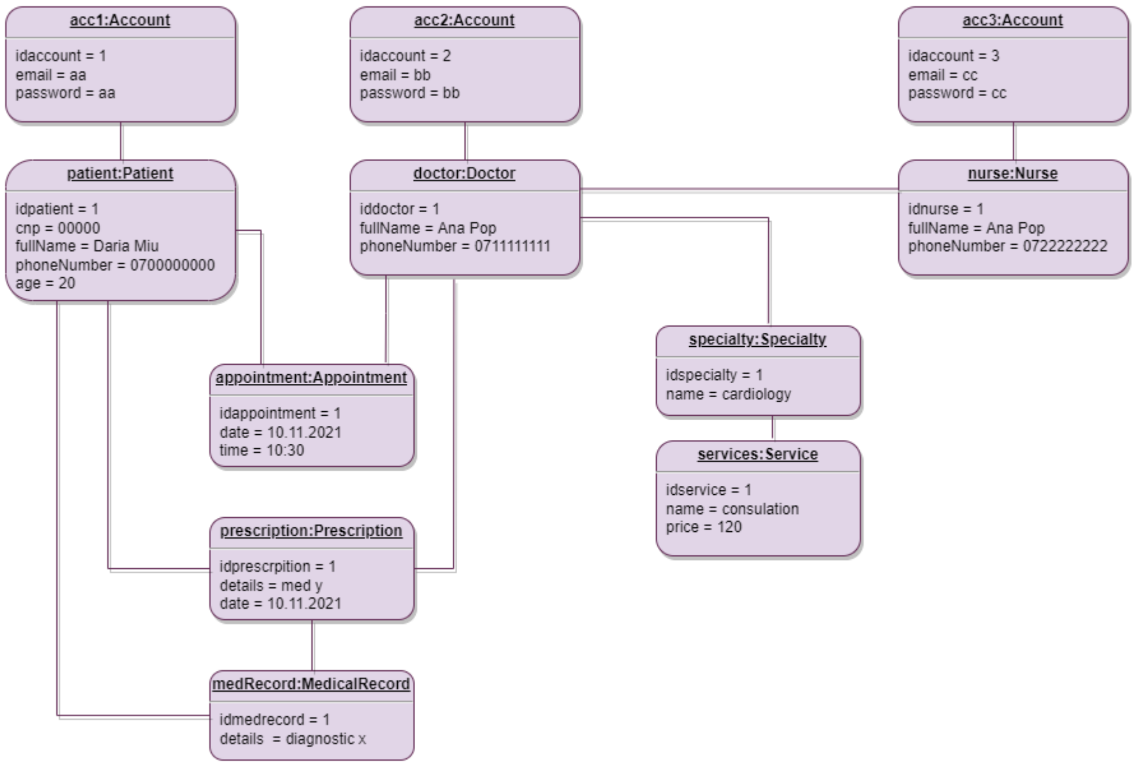
Package Diagram



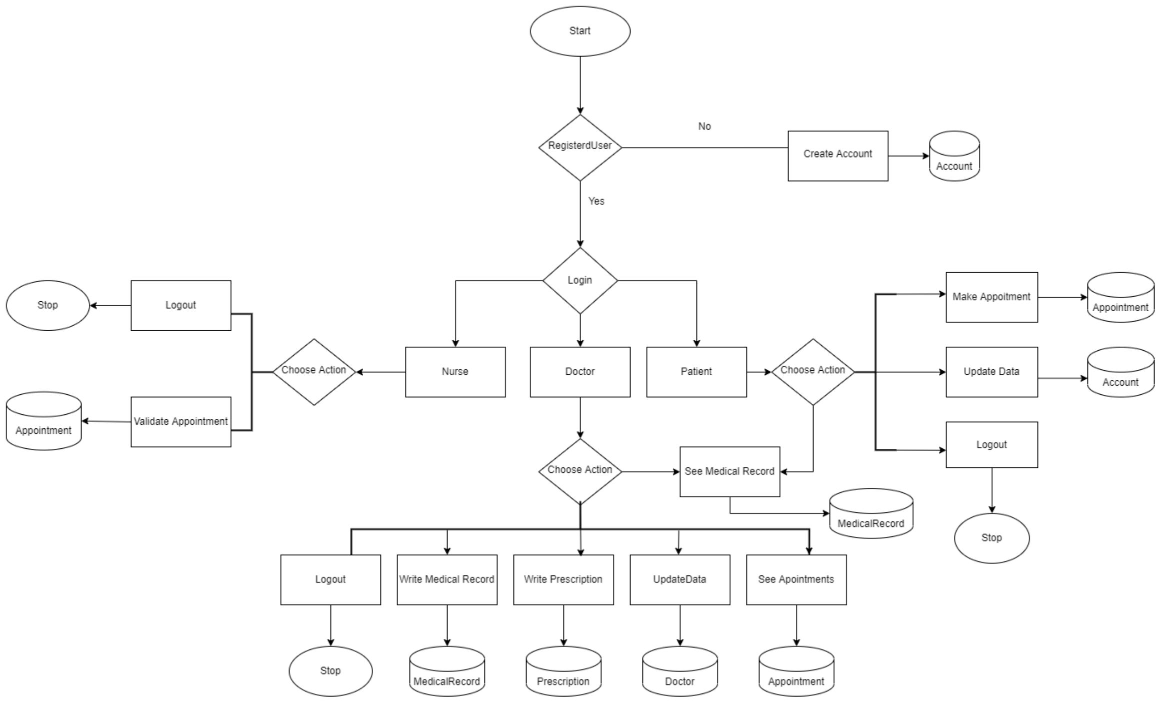
Class Diagrams

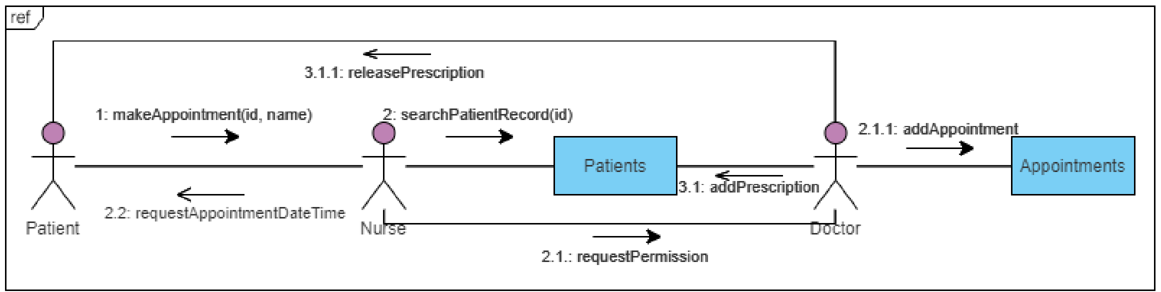


Object Diagram


acc1:Account
idaccount = 1
email = aa
password = aa
acc2:Account
idaccount = 2
email = bb
password = bb
acc3:Account
idaccount = 3
email = cc
password = cc
patient:Patient
idpatient = 1
cnp = 00000
fullName = Daria Miu
phoneNumber = 0700000000
age = 20
doctor:Doctor
iddoctor = 1
fullName = Ana Pop
phoneNumber = 0711111111
nurse:Nurse
idnurse = 1
fullName = Ana Pop
phoneNumber = 0722222222
specialty:Specialty
idspecialty = 1
name = cardiology
services:Service
idservice = 1
name = consulation
price = 120
appointment:Appointment
idappointment = 1
date = 10.11.2021
time = 10:30
prescription:Prescription
idprescrpition = 1
details = med y
date = 10.11.2021
medRecord:MedicalRecord
idmedrecord = 1
details = diagnostic x

Flowchart Diagram

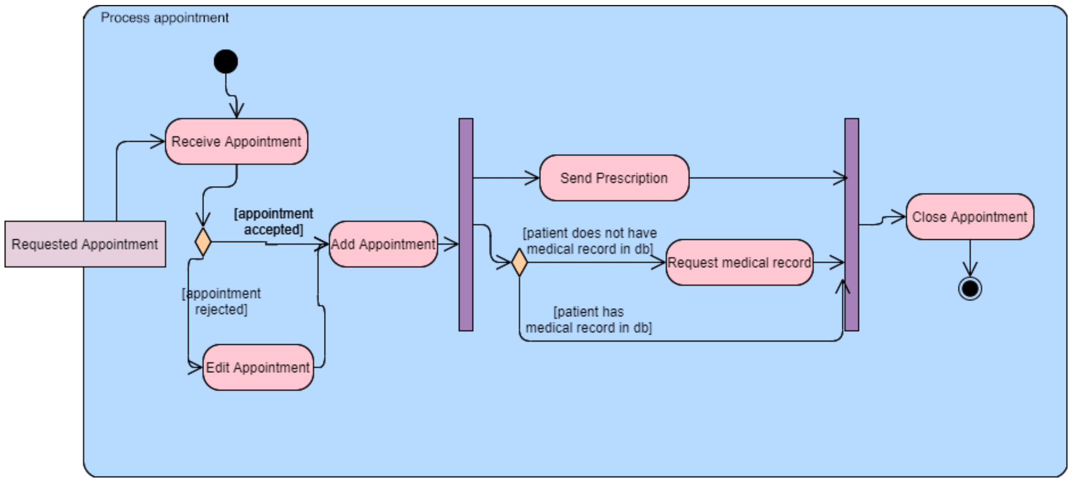




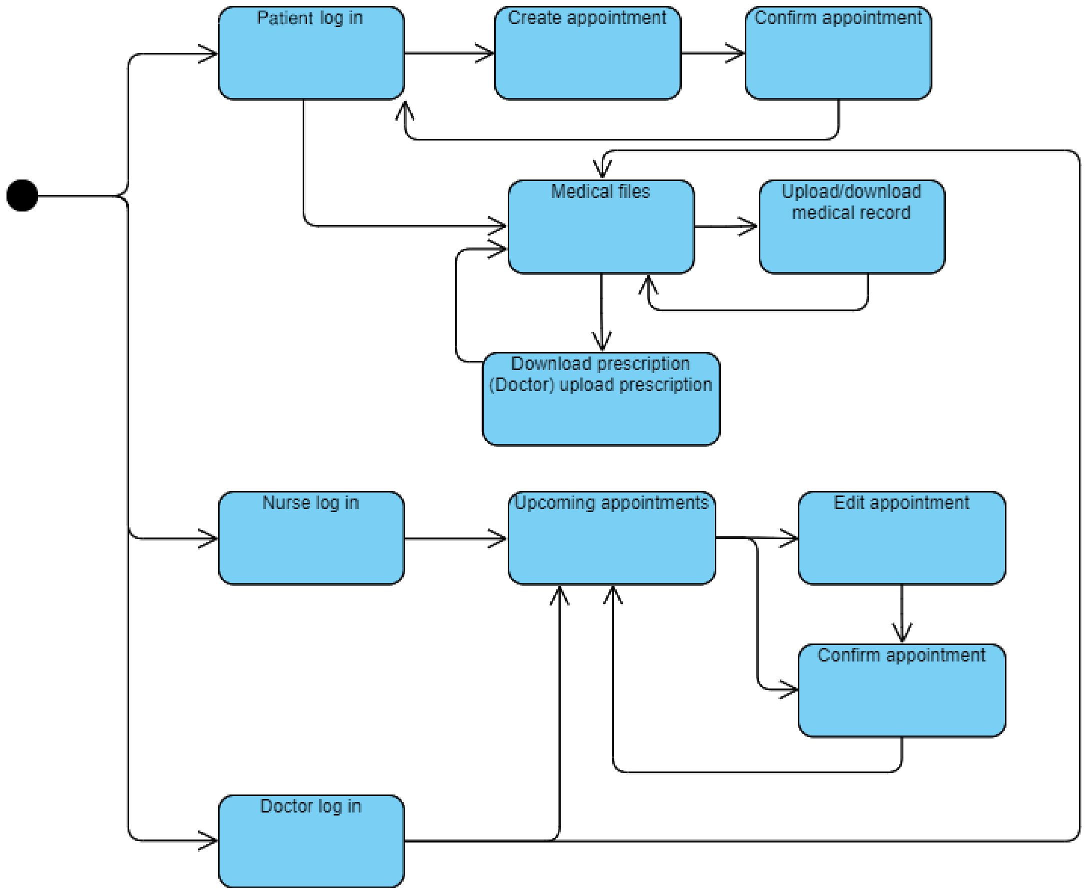
ref
3.1.1: releasePrescription
1: makeAppointment(id, name)
2: searchPatientRecord(id)
2.1.1: addAppointment
Patients
Appointments
3.1: addPrescription
2.2: requestAppointmentDateTime
Patient
Nurse
Doctor
2.1.: requestPermission

Communication Diagram

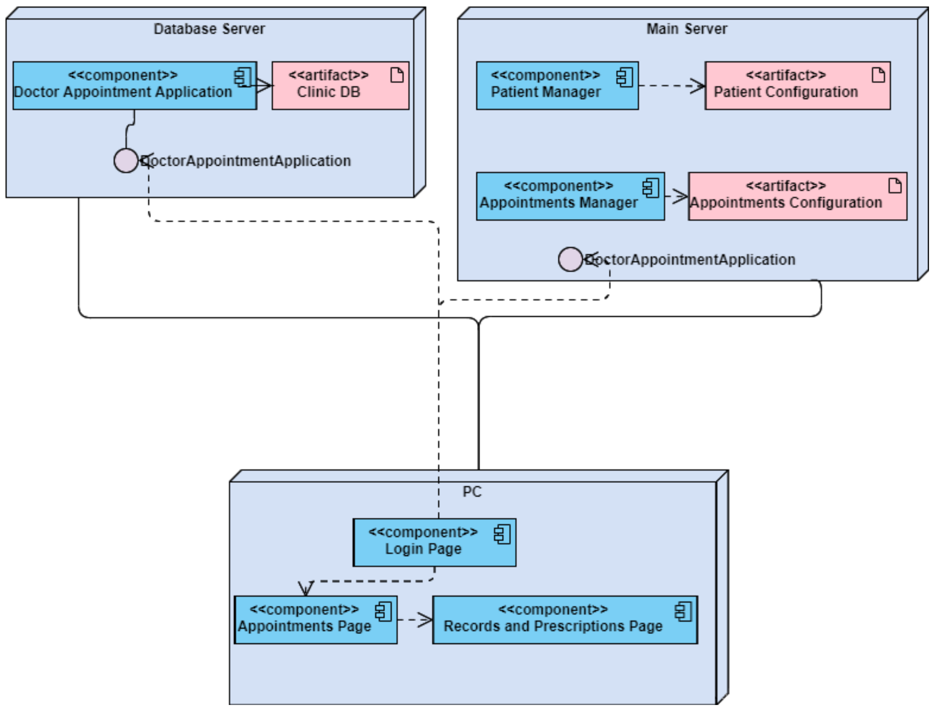
Activity Diagram



State Transition Diagram

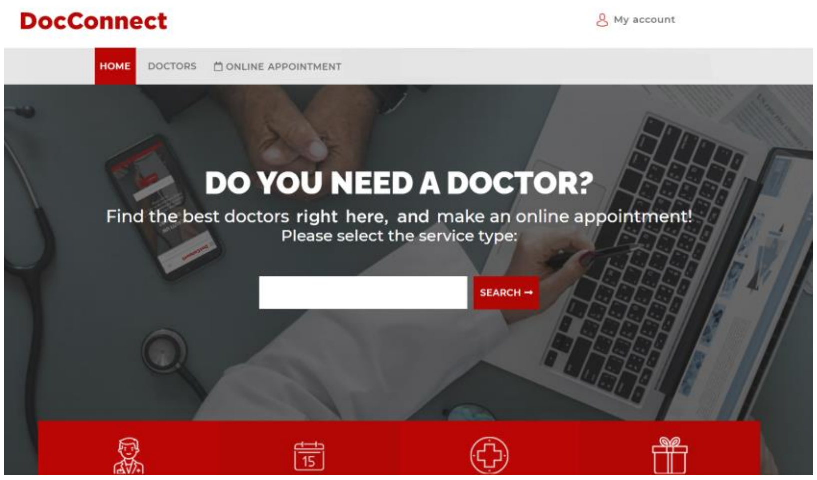


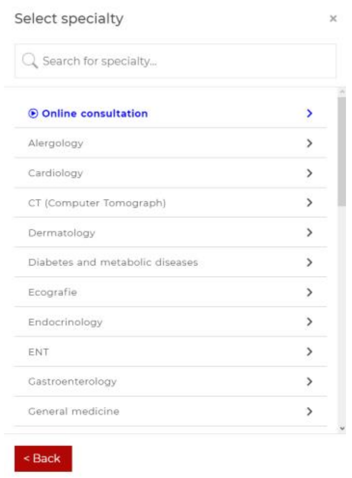
Deployment diagram

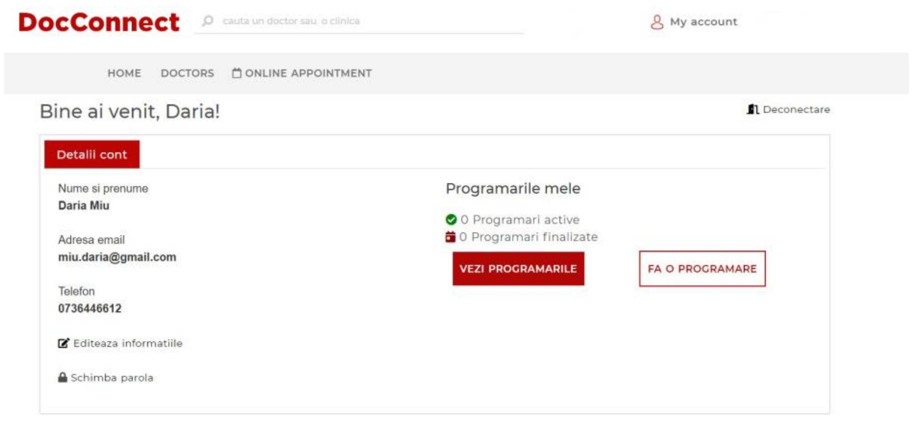


9. **Operation Mode (Interactivity presentation) + screen shots (UI design)**

Graphical User Interface Design


DocConnect
My account
HOME
DOCTORS
ONLINE APPOINTMENT
DO YOU NEED A DOCTOR?
Find the best doctors right here, and make an online appointment!
Please select the service type:
SEARCH ➞




DocConnect cauta un doctor sau o clinica My account

HOME DOCTORS ONLINE APPOINTMENT

Bine ai venit, Daria! Deconectare

Detalii cont

Nume si prenume
Daria Miu

Adresa email
miu.daria@gmail.com

Telefon
0736446612

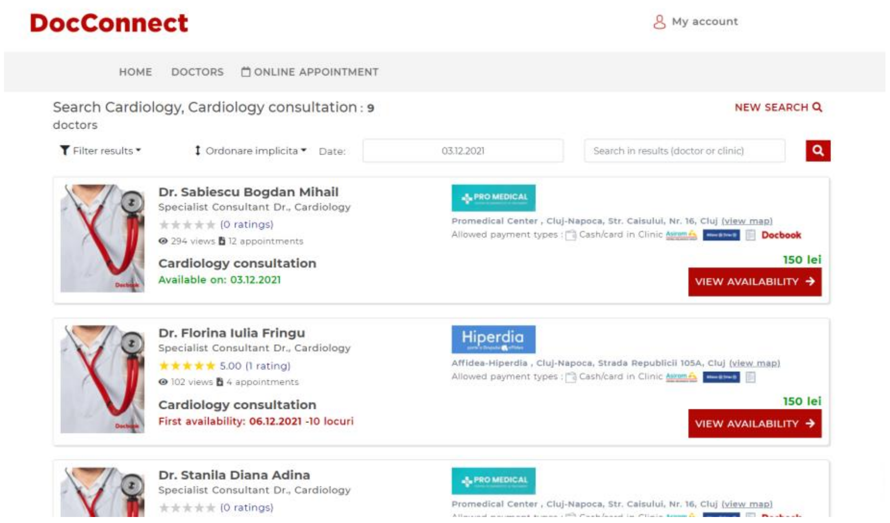
Editeaza informatiile

Schimba parola

Programarile mele

0 Programari active
0 Programari finalizate

VEZI PROGRAMARILE FA O PROGRAMARE


DocConnect My account

HOME DOCTORS ONLINE APPOINTMENT

Search Cardiology, Cardiology consultation : 9 doctors NEW SEARCH

Filter results Ordonare implicita Date: 03.12.2021 Search in results (doctor or clinic)

Dr. Sabiescu Bogdan Mihail
Specialist Consultant Dr., Cardiology
(0 ratings)
294 views 12 appointments
Cardiology consultation
Available on: 03.12.2021
Promedical Center , Cluj-Napoca, Str. Caisului, Nr. 16, Cluj (view map)
Allowed payment types : Cash/card in Clinic Docbook
150 lei
VIEW AVAILABILITY

Dr. Florina Iulia Fringu
Specialist Consultant Dr., Cardiology
5.00 (1 rating)
102 views 4 appointments
Cardiology consultation
First availability: 06.12.2021 -10 locuri
Affidea-Hiperdia , Cluj-Napoca, Strada Republicii 105A, Cluj (view map)
Allowed payment types : Cash/card in Clinic
150 lei
VIEW AVAILABILITY

Dr. Stanila Diana Adina
Specialist Consultant Dr., Cardiology
(0 ratings)
Promedical Center , Cluj-Napoca, Str. Caisului, Nr. 16, Cluj (view map)

10.Portability

Portability is one of an application’s non-functional requirements.

Spring Boot lets the developer focus on the application’s development first, and removes the need to be overly concerned with every other aspect of its lifecycle, including deployment, portability and management. Web apps developed with Spring are supported on all browsers.

11. Competing software

Right now, there are several other applications created for doctor appointments. However, most of the times, hospitals have separate software for appointments and database for example. Our goal was creating an app that can support the full needs of a clinic, so that it can be the only software such an institution requires.

Some similar websites :

- https://en.doctena.de/
- https://www.md.com/