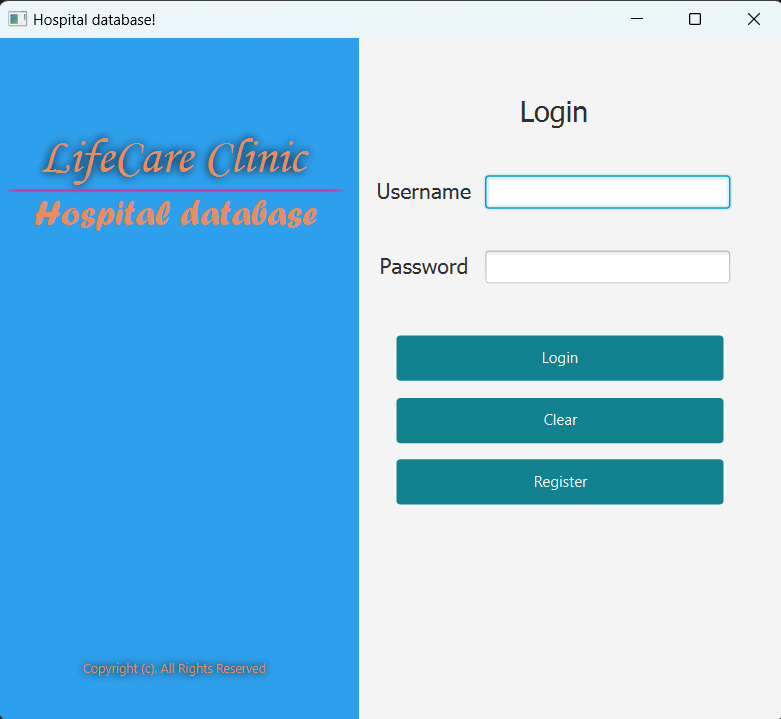
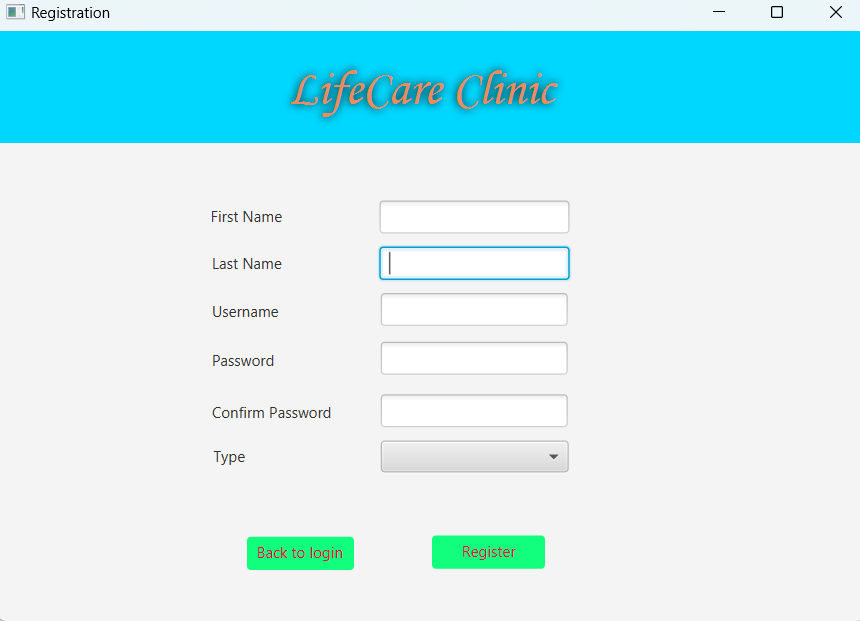
|  |  |
| --- | --- |
| **Project title** | **Hospital database management app** |
| **Author(s)** | **Cornoc Darius Andrei** |
| **Group** | **30423** |

# Task Description

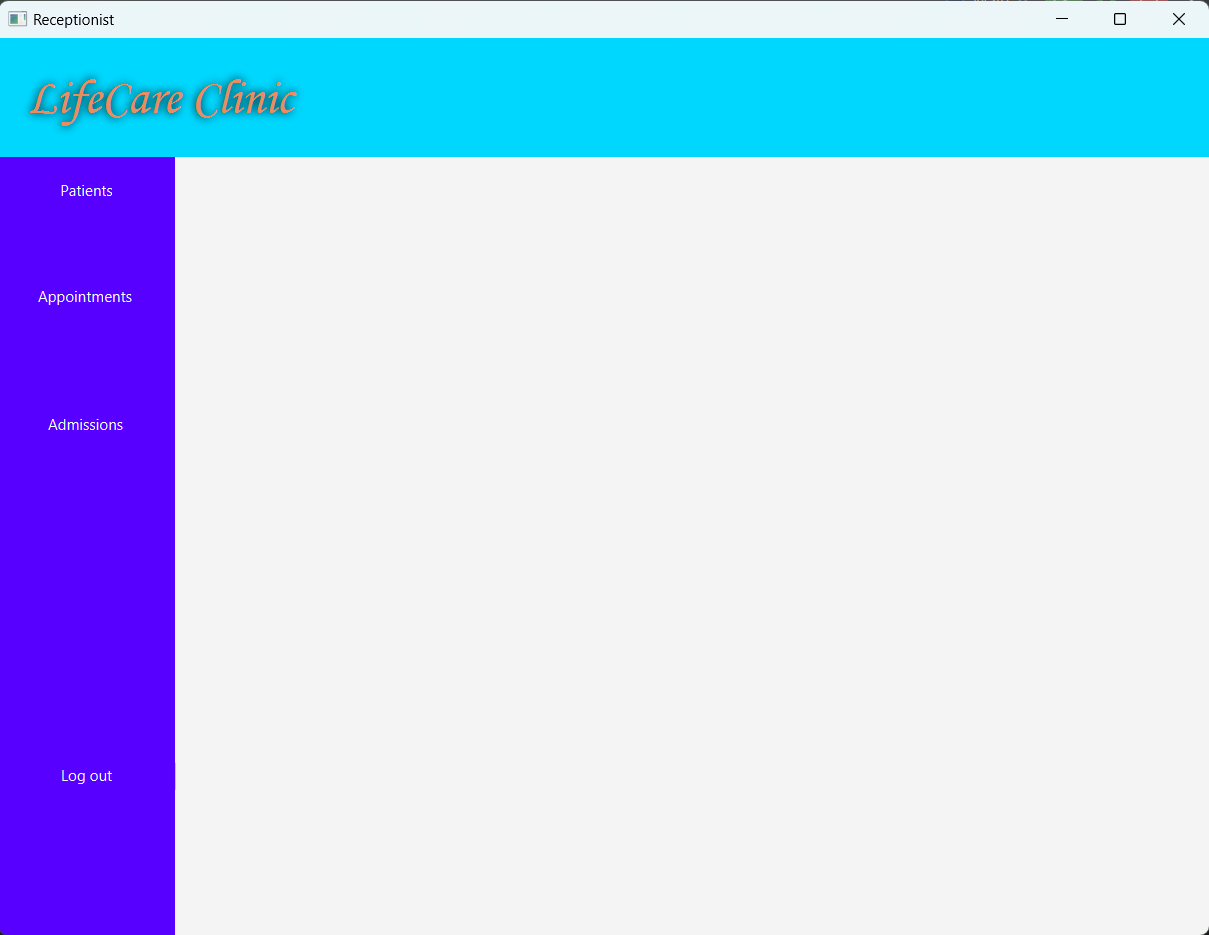
The application represents a hospital database management system. It is meant to be used by receptionists of the hospitals as well as the doctors and it starts with a login page for access.



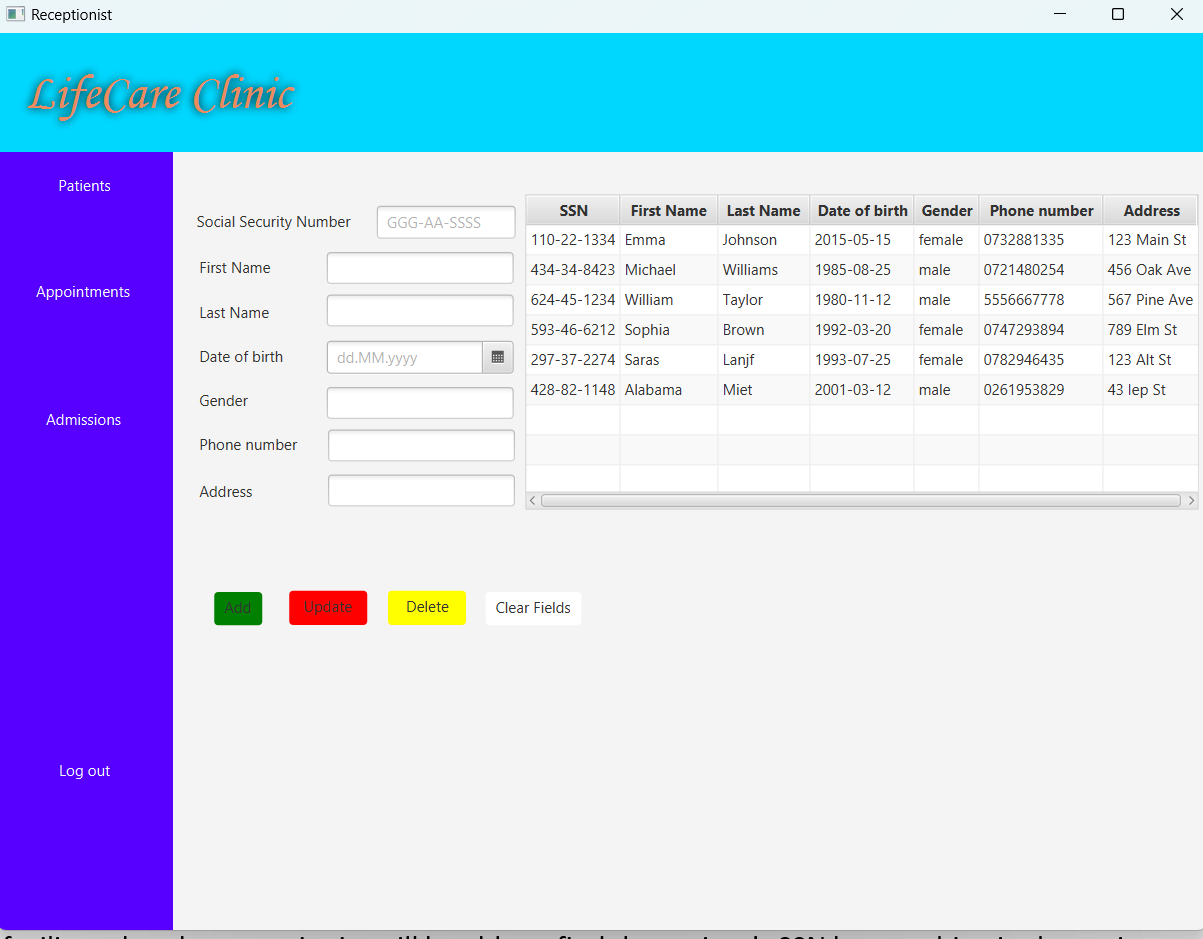
Users can create accounts via a registration pop-up by filling in fields like name, username, password, and user type (doctor or receptionist). Doctors may specify their specialization during registration. Input validation ensures correct information entry. Incorrect or missing data triggers error messages. Upon successful registration, users can log in and access different functionalities based on their role.



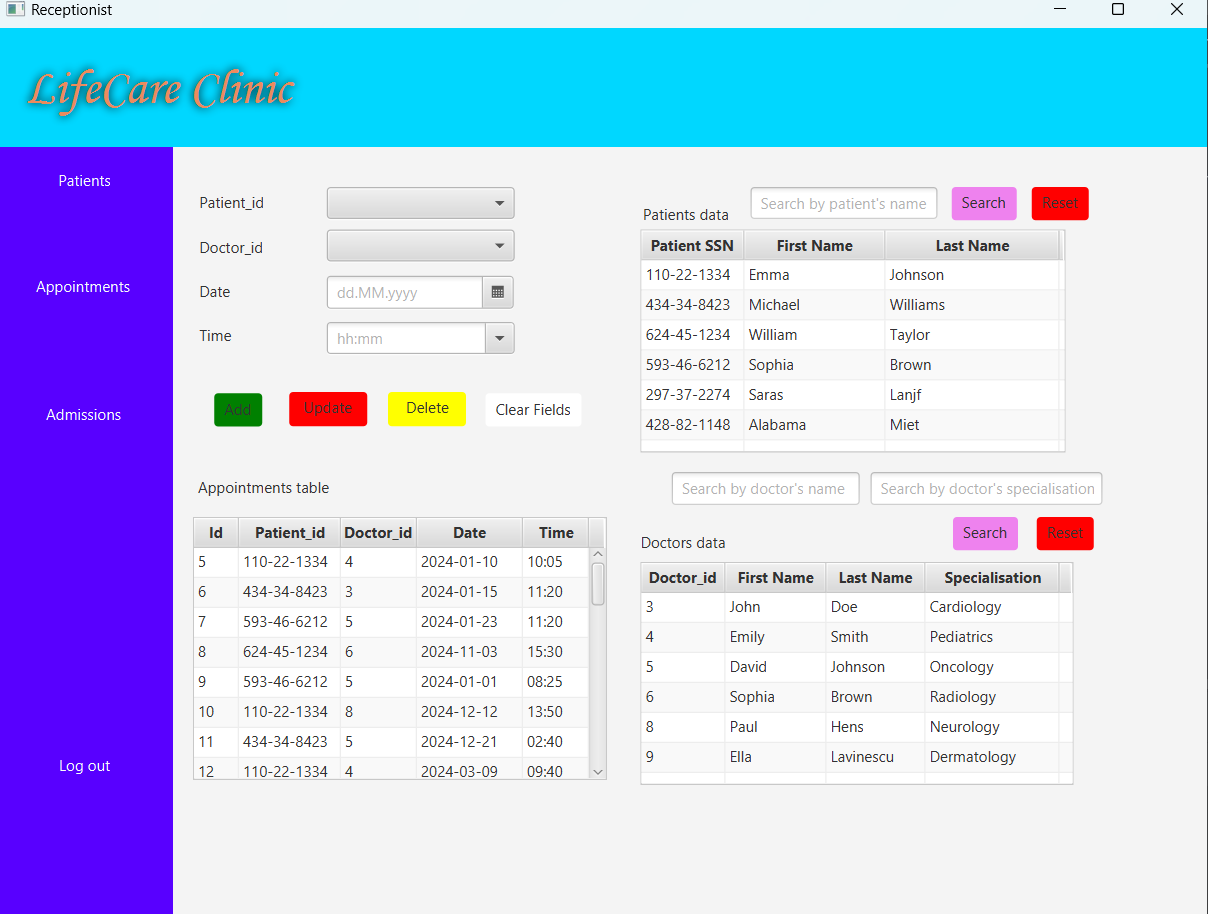
For receptionists, the system offers three main options: Patients, Appointments, and Admissions.



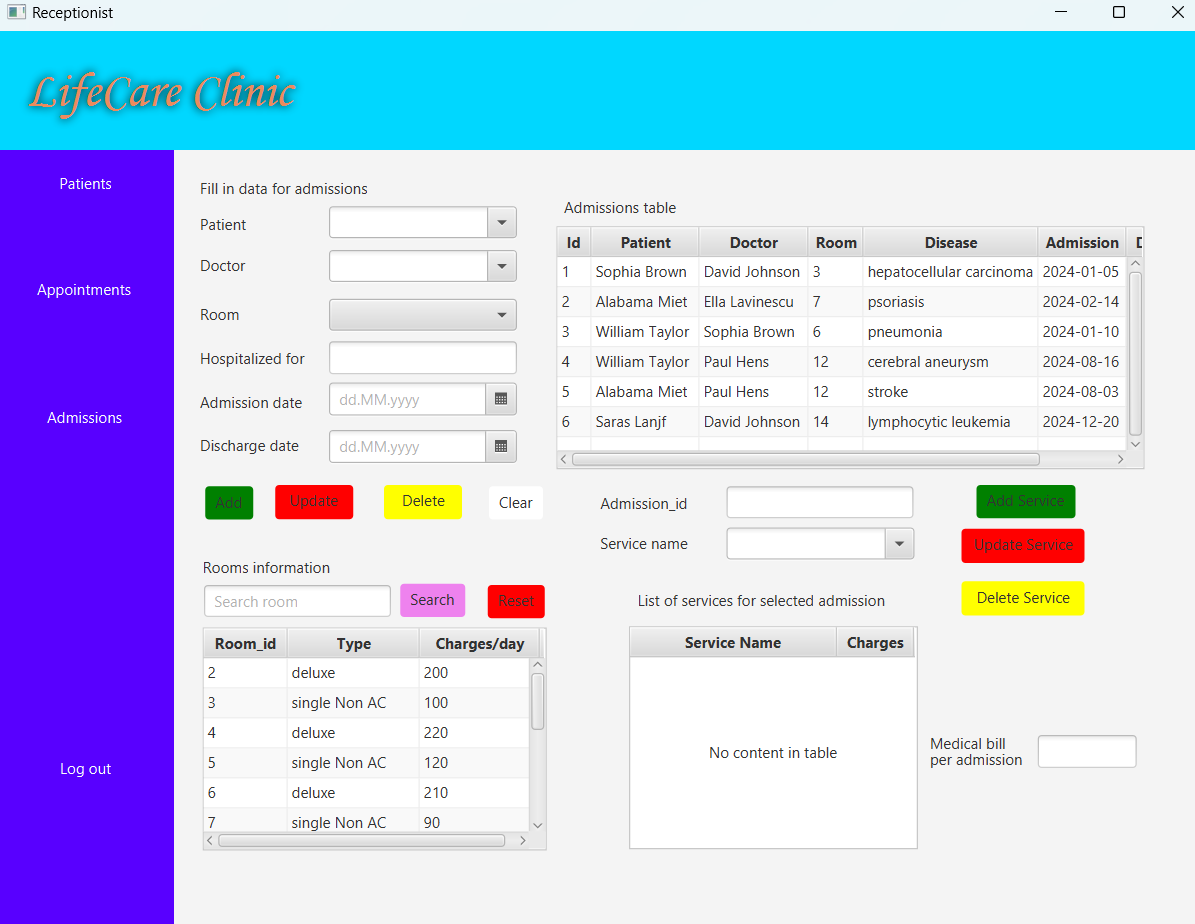
The Patients section allows adding, updating, and deleting patient information. Data validation ensures no incorrect data is introduced, and a table displays all registered patients.



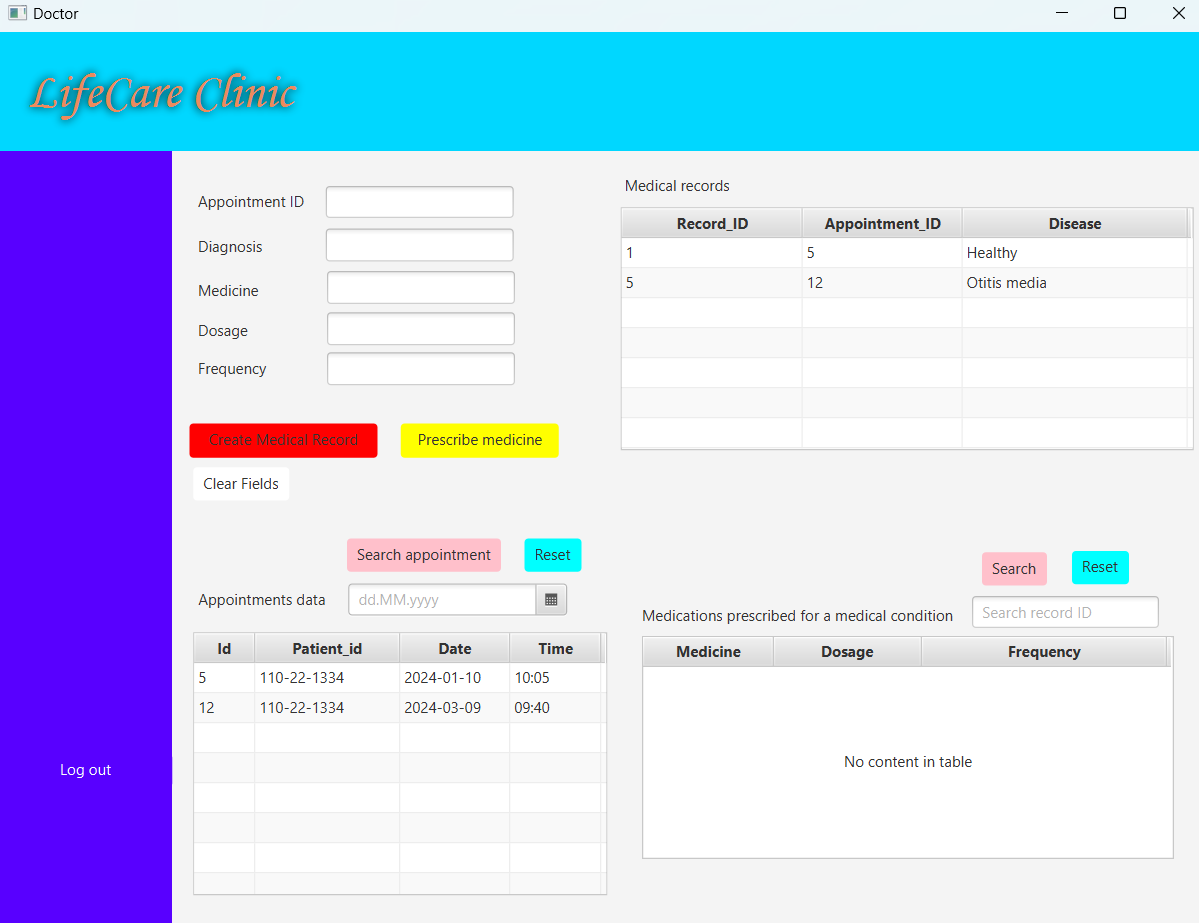
Appointments enable scheduling consultations with specialized doctors. Patient SSN, doctor IDs, and appointment details are managed here. Patients’ ID can be found by SSN or name, while doctors can be searched by name or specialization. Time slots for appointments run from 8:00 to 18:00, each with a 25-minute interval.



Admissions assist in creating admission documents and calculating medical bills for hospitalized patients. Room selection, admission/discharge dates, and associated services are managed. The patient can be searched by entering part of their first or last name directly into the corresponding field. The system automatically filters the list of patients. Similar to patients, doctors can also be selected based on their name. In addition, the hospital exclusively offers private rooms. Rooms are filtered by availability and type. By filling in the fields corresponding to the admission date and discharge date and, optionally, writing a room type and then clicking on the search button from the Rooms information section, the receptionist can see rooms that are available in that time interval. As discharge dates may not always be known, the field can be left empty. Search functionalities remain operational in this scenario. Also, deleting an admission also removes all associated services. The receptionist can also manage the services that were used by the patient during its hospitalization. The services specific to an admission can be seen in the services table by clicking on a specific admission. Also, a service can be searched in the same way as patient’s name.



Doctor functionalities include adding prescriptions and medical records. To create a medical record, they select the appointment, auto-filling the appointment ID field, and add a diagnosis. They can optionally prescribe medication by filling related fields and clicking "create medical record." Adding medicines is done by filling again the medicine-related fields and clicking on "prescribe medicine" and it updates the prescription table. They can create multiple records for various diseases and add medicines accordingly. The system prevents duplicate medical records if they do not have any medicines prescribed. A specific prescription’s medicines can be seen by filling the record\_id field.



The system maintains tables for appointments, patients, medical records, and prescriptions, from which you can select if the data to sorted in ascending or descending order. Also, by clicking on the tables’ entries, the data will automatically fill the corresponding fields. Also, the data introduced in the datepickers is not checked if it is valid, so it should be introduced in ‘yyyy-MM-dd’.

# Class Discovery

|  |  |
| --- | --- |
| **Admission** | |
| Responsibilities: | Collaborator classes: |
| Manage admission details such as admission date, discharge date, disease, medical bill  Store patient, doctor and room id | Patient, Doctor, Room |

|  |  |
| --- | --- |
| **User** | |
| Responsibilities: | Collaborator classes: |
| Manage user information: user ID, username, password, type, first name, last name |  |

|  |  |
| --- | --- |
| **Room** | |
| Responsibilities: | Collaborator classes: |
| Handle room information: room ID, type and charges per day |  |

|  |  |
| --- | --- |
| **Doctor** | |
| Responsibilities: | Collaborator classes: |
| Manage doctor details: doctor ID, specialization, but also user-related data such as first name, last name and password  Add medical record and prescriptions | User, MedicalRecord, Prescription |

|  |  |
| --- | --- |
| **Patient** | |
| Responsibilities: | Collaborator classes: |
| Manage patient information such as ID, name, date of birth, gender, contact details, address  Check references to admissions and appointments that belong to a patient | Appointment, Admission |

|  |  |
| --- | --- |
| **Service** | |
| Responsibilities: | Collaborator classes: |
| Manage service information like service ID, name, charges per service  Selecting and updating services that belong to a specific admission | Admission |

|  |  |
| --- | --- |
| **Prescription** | |
| Responsibilities: | Collaborator classes: |
| Handle prescription details such as ID, record ID, medicine, dosage, frequency  Search the medications related to a specific medical record | MedicalRecord |

|  |  |
| --- | --- |
| **MedicalRecord** | |
| Responsibilities: | Collaborator classes: |
| Manage medical record information: record ID, appointment ID, diagnosis | Prescription |

|  |  |
| --- | --- |
| **Appointments** | |
| Responsibilities: | Collaborator classes: |
| Manage appointment details: appointment ID, doctor ID, patient ID, appointment date, time  Check references to medical records | Doctor, Patient, MedicalRecord |

# Class Diagram

