



Indian Institute of Technology, Kharagpur
Department of Computer Science and Engineering

CS39202 : DATABASE MANAGEMENT SYSTEMS LAB

MINI-PROJECT : HOSPITAL MANAGEMENT SYSTEM

7th March, 2023

Abhinav Sen
20CS10003

Ashish Rekhani
20CS10010

Vishal Ravipati
20CS10076

Aman Sharma
20CS30063

Rohit Ranjan
20CS30066

Contents

1. Introduction	2
2. Interface	2
2.1 Entities	2
2.2 Explore	2
2.3 Login	2
2.4 Admin Dashboard	3
2.4.1 Add User	3
2.4.2 View User	3
2.4.3 Delete User	4
2.4.4 Add Procedure	4
2.4.5 Add Physician	4
2.4.6 Add Department	4
2.4.7 Add Medication	5
2.4.8 Add Room	5
2.4.9 Add Nurse	5
2.4.10 Add Affiliation	5
2.4.11 Add Training	6
2.5 Front-Desk Operator Dashboard	6
2.5.1 Register Patient	6
2.5.2 Admit Patient	6
2.5.3 Discharge Patient	7
2.5.4 Schedule Appointment	7
2.6 Data-Entry Operator Dashboard	7
2.6.1 Add Results and Prescription	7
2.7 Doctor Dashboard	8
2.7.1 View Patients	8
2.7.2 View Appointments	8
2.7.3 View Prescriptions	8
2.7.4 Query Patients	9
3. Framework	9
3.1 Back-End	9
3.1.1 Programming Frameworks	9
3.1.2 Integrity of Database	9
3.1.3 Database Schema	9
3.2 Front-End	10

1. Introduction

This report will explain the various interfaces and functionalities implemented as part of the mini-project on building a Hospital Management System (referred to as HMS).

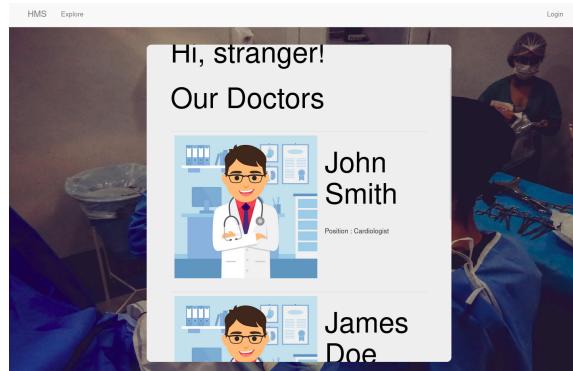
2. Interface

2.1 Entities

There are 4 different types of user entities who can use the HMS - admin (the top level with the most powers), doctors, front-desk operators and data-entry operators, each with unique powers.

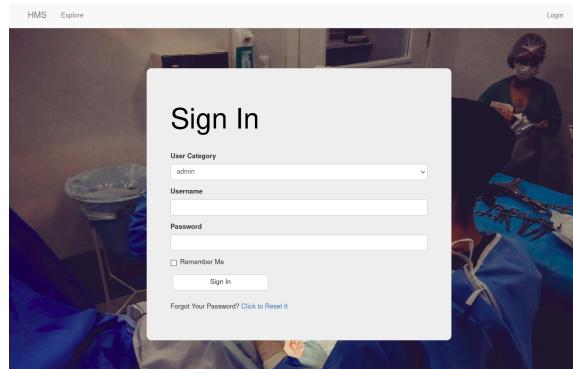
2.2 Explore

Everyone who opens the HMS web page (whether registered users or not) can view the list of doctors with names and photographs in the explore section.



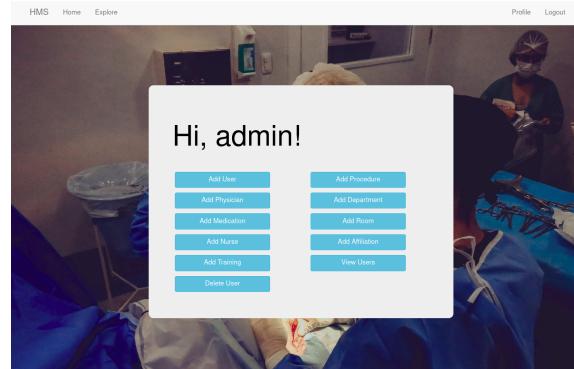
2.3 Login

We have created a single login page for all the different user entities. We have a drop-down menu to choose the user type, and each user will have to enter their credentials. Users are also given the ability to change their password if they forget it. When the system is started from scratch, only an admin entity with pre-defined username and password is created. Other users can be added to this system only by the admin after logging in.



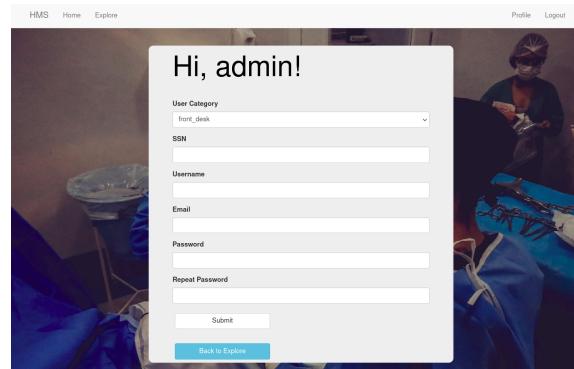
2.4 Admin Dashboard

The admin Dashboard has the greatest functionality, with the ability to add all the different entities of the HMS database schema. In particular, he can add users, departments, medications, physicians, affiliations, etc. This functionality is provided to the admin as this is critical data upon which the entire system runs. In addition, the admin can view all the users on the system with their details and can delete any user given their SSN. Each of the add buttons leads to a form, which is subject to validation checks in the back-end (discussed later) before being added to the database.

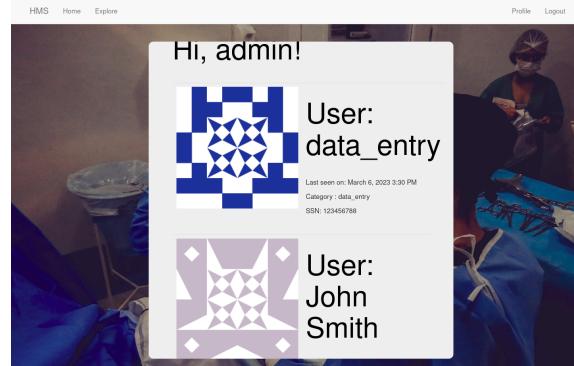


2.4.1 Add User

This form asks for the SSN, Email ID, user-type, username and password for logging into the system. The back-end checks for the uniqueness of the SSN, Email ID and username and displays an error message when duplicates are found. After validation, the user is added to the database and can be viewed by the admin and can also login into the system with the provided credentials. Note that a doctor can be added only for a physician already present in the system, as doctor is a physician with login access.

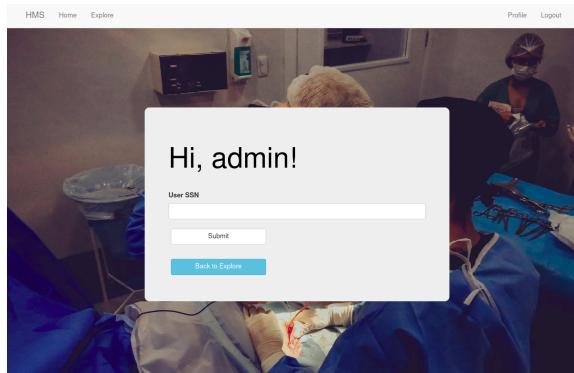


2.4.2 View User



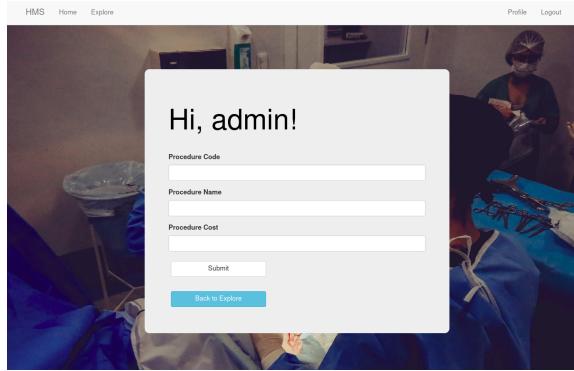
2.4.3 Delete User

This option allows the admin to delete the user from the platform given their SSN. It flashes an error on entering a SSN that doesn't exist and deletes the user if the SSN is found.



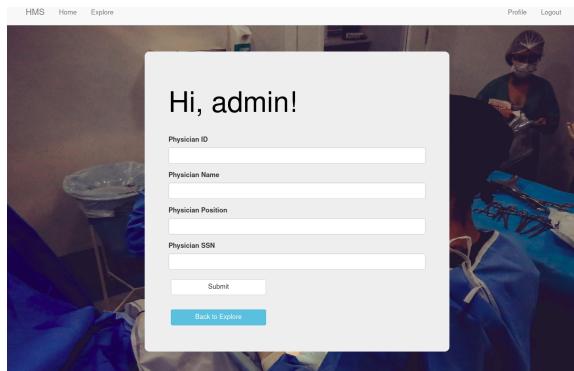
2.4.4 Add Procedure

This option allows the admin to add a medical procedure into the database, by entering the procedure code, name and cost. The system adds to the database on validation and flashes an error otherwise.



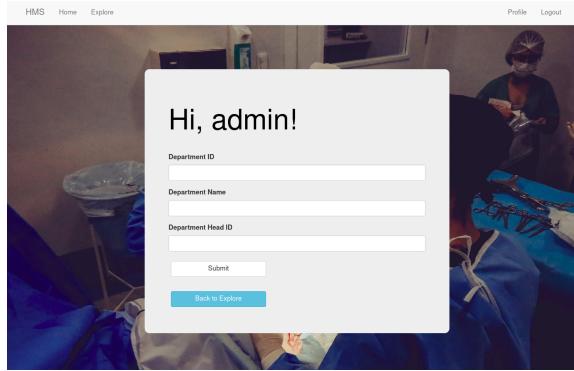
2.4.5 Add Physician

This option allows the admin to add a physician into the database given their ID, Name, Position and SSN. The system adds to the database on validation and flashes an error otherwise.



2.4.6 Add Department

This option allows the admin to add a hospital department into the database, by entering the department ID, Name and ID of the head of department. The system adds to the database on validation and flashes an error otherwise.



2.4.7 Add Medication

This option allows the admin to add a medication into the database, by entering the medication code, name, brand and description. The system adds to the database on validation and flashes an error otherwise.

The screenshot shows a modal dialog box titled "Hi, admin!" containing four input fields: "Code", "Name", "Brand", and "Description". Below these fields is a "Submit" button and a "Back to Explore" link at the bottom.

2.4.8 Add Room

This option allows the admin to add a hospital room into the database, by entering the room ID, type and availability. The system adds to the database on validation and flashes an error otherwise.

The screenshot shows a modal dialog box titled "Hi, admin!" containing three input fields: "Room ID", "Room Type", and a checkbox labeled "Room Unavailable". Below these fields is a "Submit" button and a "Back to Explore" link at the bottom.

2.4.9 Add Nurse

This option allows the admin to add a nurse into the database, by entering the nurse ID, name, position, SSN and registration status. The system adds to the database on validation and flashes an error otherwise.

The screenshot shows a modal dialog box titled "Hi, admin!" containing five input fields: "Nurse ID", "Nurse Name", "Nurse Position", "Nurse SSN", and a checkbox labeled "Nurse Registered". Below these fields is a "Submit" button and a "Back to Explore" link at the bottom.

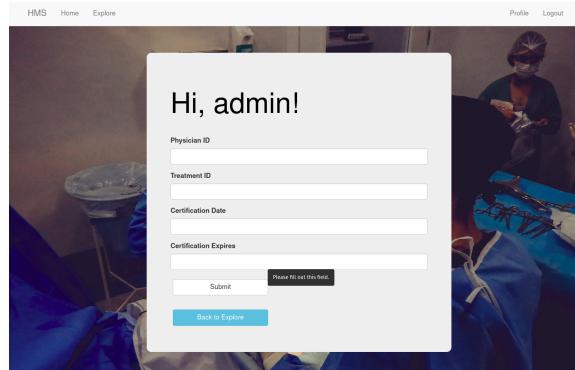
2.4.10 Add Affiliation

This option allows the admin to add a physician's affiliation with a department into the database, by entering the IDs of the physician and the department. The system adds to the database on validation and flashes an error otherwise.

The screenshot shows a modal dialog box titled "Hi, admin!" containing three input fields: "Physician ID", "Department ID", and a checkbox labeled "Primary Affiliation". Below these fields is a "Submit" button and a "Back to Explore" link at the bottom.

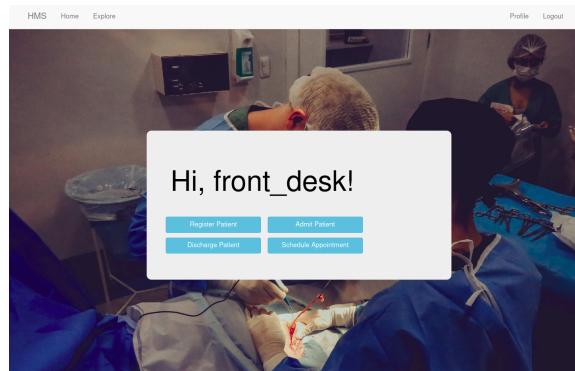
2.4.11 Add Training

This option allows the admin to add a physician's training into the database, by entering the IDs of the physician and the treatment, certification and expiry date. The system adds to the database on validation and flashes an error otherwise.



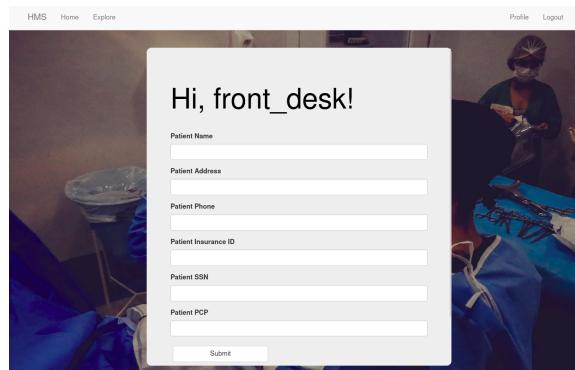
2.5 Front-Desk Operator Dashboard

The front-desk operator dashboard primarily deals with handling of patients and their appointments with the relevant doctors. They have options to register, admit and discharge patients along with the scheduling of appointments.



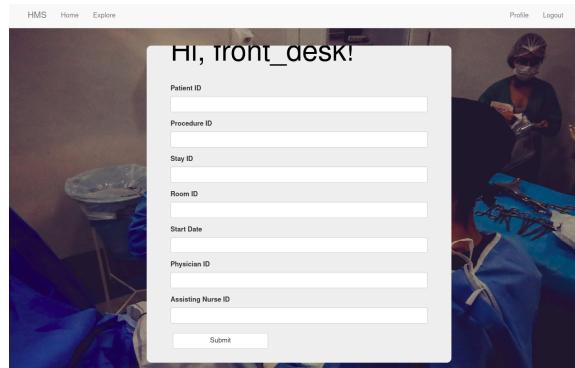
2.5.1 Register Patient

This allows the operator to add the patient into the system, by entering their name, address, phone, insurance ID, SSN and PCP. Upon validation, the patient is added into the system and an error is flashed otherwise.



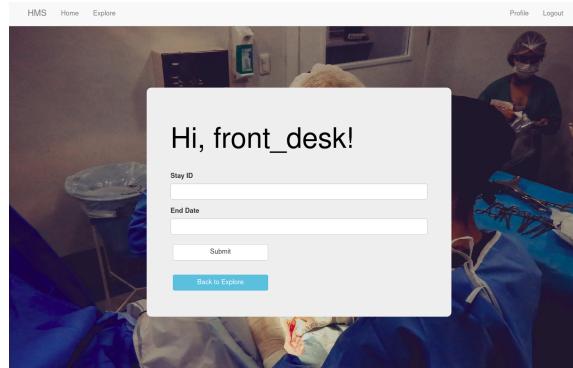
2.5.2 Admit Patient

This allows the operator to admit the patient into the hospital, by entering the patient ID, procedure to be performed, stay details (ID, room number), start date of admission and ID of the performing physician and the assisting nurse. Upon validation, the patient is added into the system and an error is flashed otherwise.



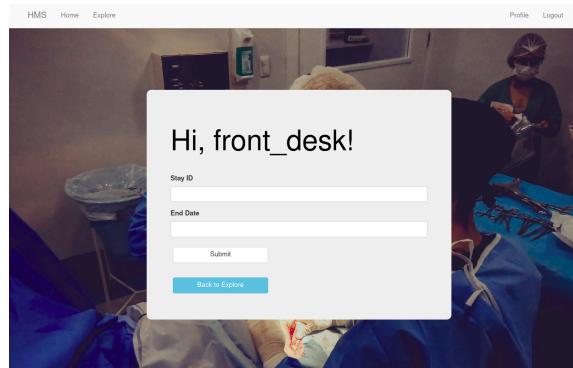
2.5.3 Discharge Patient

This allows the operator to discharge the patient from the hospital, by entering the stay ID and date for end of stay. Upon validation, the patient is added into the system and an error is flashed otherwise.



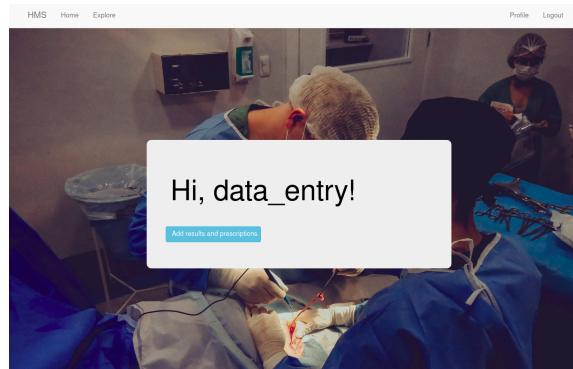
2.5.4 Schedule Appointment

This allows the operator to schedule an appointment for a patient with the doctor, by providing details about the patient, doctor, preparing nurse, the start and end date of the appointment. It is ensured that the maximum length of the appointment is an hour and that the appointment is scheduled in the working hours of 8 AM to 5 PM. Upon validation, the appointment is added to the database and an error is flashed otherwise.



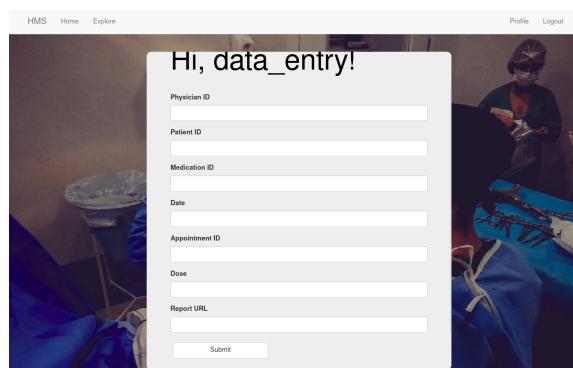
2.6 Data-Entry Operator Dashboard

The data-entry operator has limited functionality, which is to enter information about the various prescriptions, results of any X-ray tests, etc. into the system for future reference of the doctors.



2.6.1 Add Results and Prescription

The data-entry enters the IDs of the physician, patient, medication prescribed and the relevant appointment along with the date of the appointment, dose of the medication and the URL of the report for the results of tests (X-ray scan, for example). The data is saved into the system upon validation, otherwise an error is flashed.

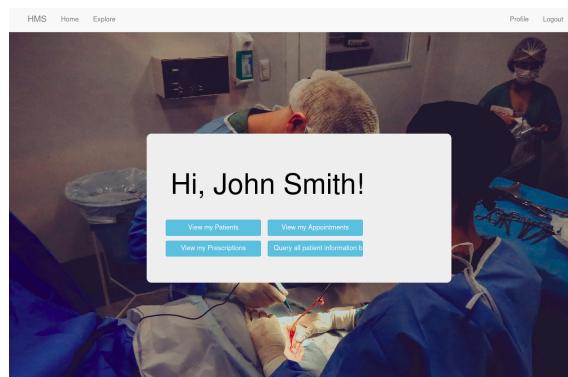


2.7 Doctor Dashboard

The doctor dashboard has functionality for viewing the entire history of appointments, patients and prescriptions related to the doctor. It also allows the doctor to query patients in the database who have taken treatment under the doctor using SSN.

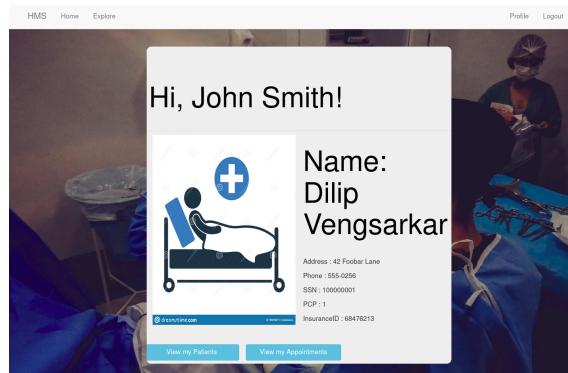
2.7.1 View Patients

This allows the doctor to view all the patients related to the doctor and information relating to them.



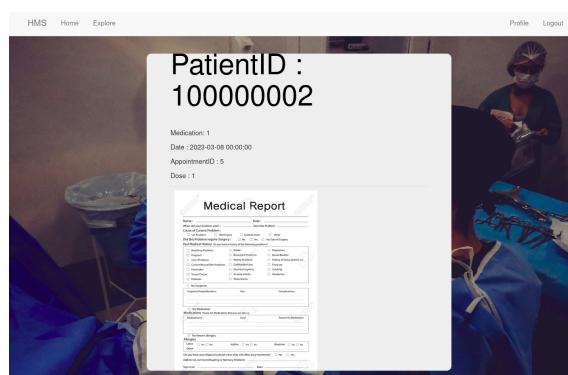
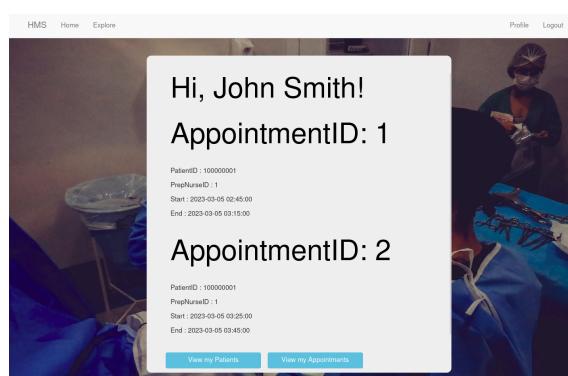
2.7.2 View Appointments

This allows the doctor to view all the past and upcoming appointments of the doctor and information relating to them.



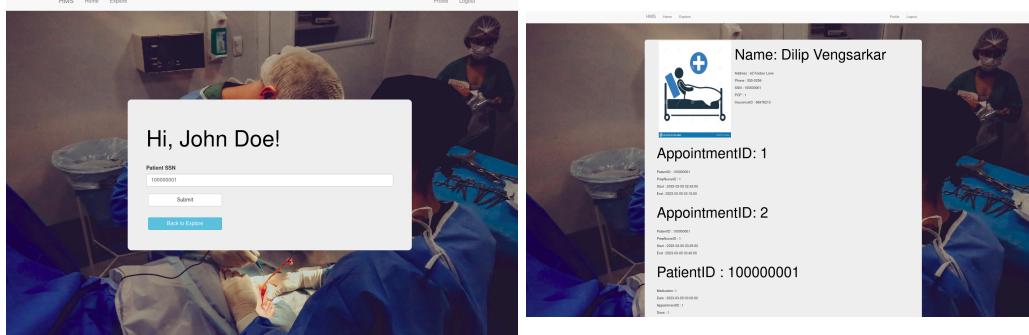
2.7.3 View Prescriptions

This allows the doctor to view all the prescriptions made by the doctor and information relating to them.



2.7.4 Query Patients

This allows the doctor to search for specific patients by entering their SSN, and then obtain the information about the patient, or an error if the patient is not present in the database.



3. Framework

3.1 Back-End

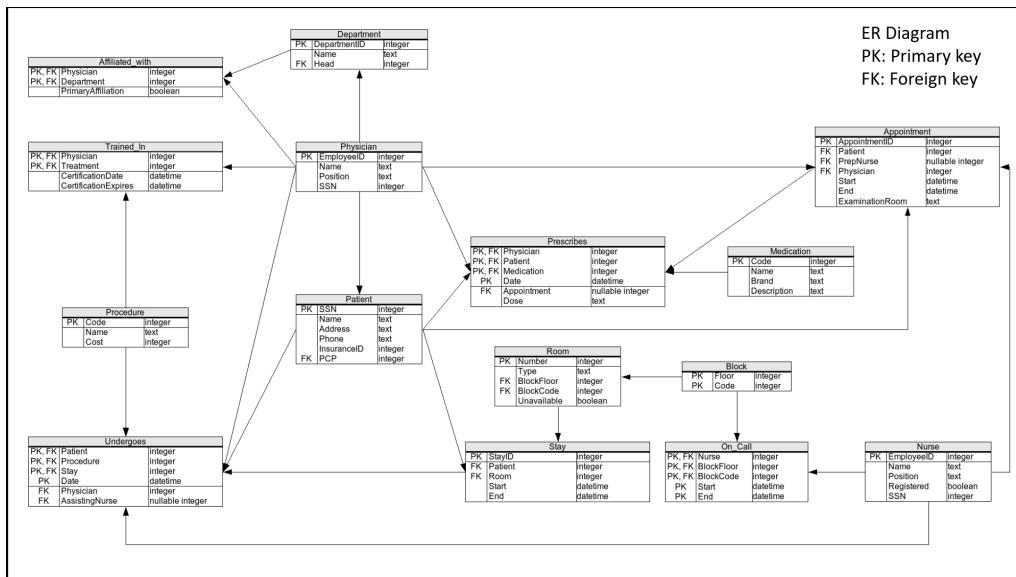
3.1.1 Programming Frameworks

Python has been used as the primary programming language in the back-end. This is because of its ease of use and integration with several front-end and database frameworks. MySQL has been used as the language for maintaining the database and is well-integrated with Python.

3.1.2 Integrity of Database

Whenever new data is submitted through the form, the back-end has functions that retrieve the data and check for its validity. This is done primarily by the MySQL framework by enforcing the primary and foreign key constraints to prevent the entering of duplicate data. Checks are also implemented to verify that the correct data type has been entered in the form. When an error is caused by attempting to add the data to the database, a rollback is performed and the error message is flashed onto the page.

3.1.3 Database Schema



3.2 Front-End

Flask has been used as the web-app framework. Flask and its integration with Python provides for the use of dynamic HTML templates, removing the need for hard-coding separate templates for minor changes. These templates are rendered using Flask variables and routing functions that call the appropriate template.

Bootstrap has been used to provide a simple, clean interface for all the templates, and CSS has been used to provide a better look to the website.