

DBMS Lab Mini Project Report

Submitted By:

Group Members

Rahul Mandal (20CS30039)

Anirban Haldar (20CS10008)

Abhibhu Prakash (20CS10002)

Saurabh Jaiswal (20CS30047)

Patel Yuti (20CS10043)

March 12, 2023



1 Purpose

The purpose of this project is to create a web application for a hospital management system. The system enables us to register patients, schedule appointment with doctors, maintain patient information about diagnostics tests and treatments administered, maintain information about doctors/ healthcare professionals, store admit/discharge information about the patients.

2 User Classes and their Characteristics

The system has four different kinds of users, who have different levels of access to the database and different functionalities:

- Front Desk Operators: The front desk operator registers new patients, admits them according to the rooms, and discharges patients too.
- Data Entry Operators: The data entry operator enters patient data, regarding the tests and treatments the patients have been administered.
- Doctors: The doctor makes queries to retrieve patient information.
- Database Administrator: The database administrator or admin can create or delete any user.

3 Functionalities Required

The following functionalities are expected to be in a working state in the system:

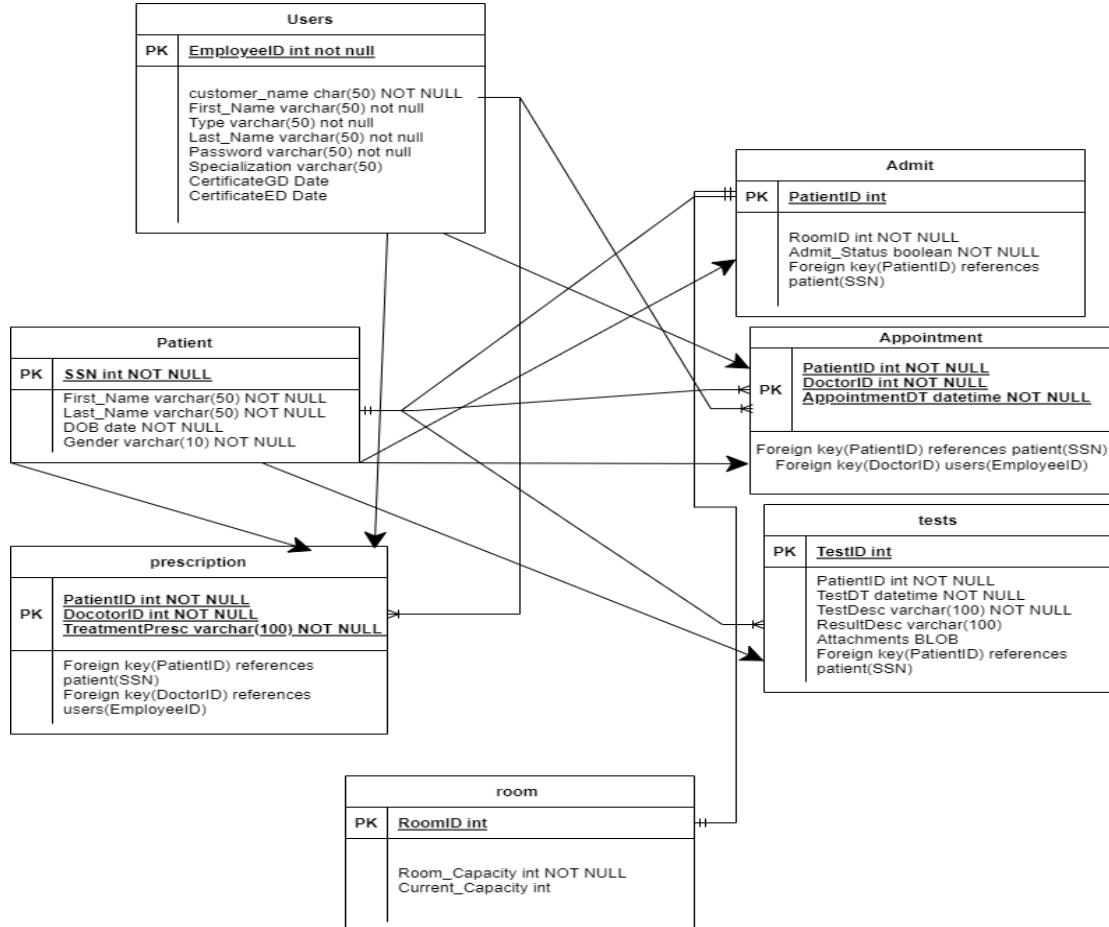
- The database administrator should be able to add and delete users.
- The doctor must see the patient details (ask for the patient details as a query) as well as the details for the medicines prescribed to the patient in the form of a dashboard.
- Information about the patient should be registered in the database. The appointment of a patient needs to be scheduled according to the availability of the doctor and the scheduled appointment needs to be displayed on the doctor's dashboard.
- The workflow should support admission and discharge of patients as well as scheduling of tests and treatment.

4 Entity Relationship Diagram

The database backend includes 7 relations:

- Users
- Patient
- Prescription
- Room
- Admit
- Appointment
- Tests

The backend database is constructed using mysql, as the entity-relation diagram shows, there are 7 relations in the database. There are 4 types of users that are also defined as and when required and have their own set of passwords and usernames to allow access control. All the queries are written using mysql and django.



5 Front end Design

The front end is designed such that it leads to introductory dashboard, which is used for any user to login. It leads to separate dashboards for each user which show the data required by that user.

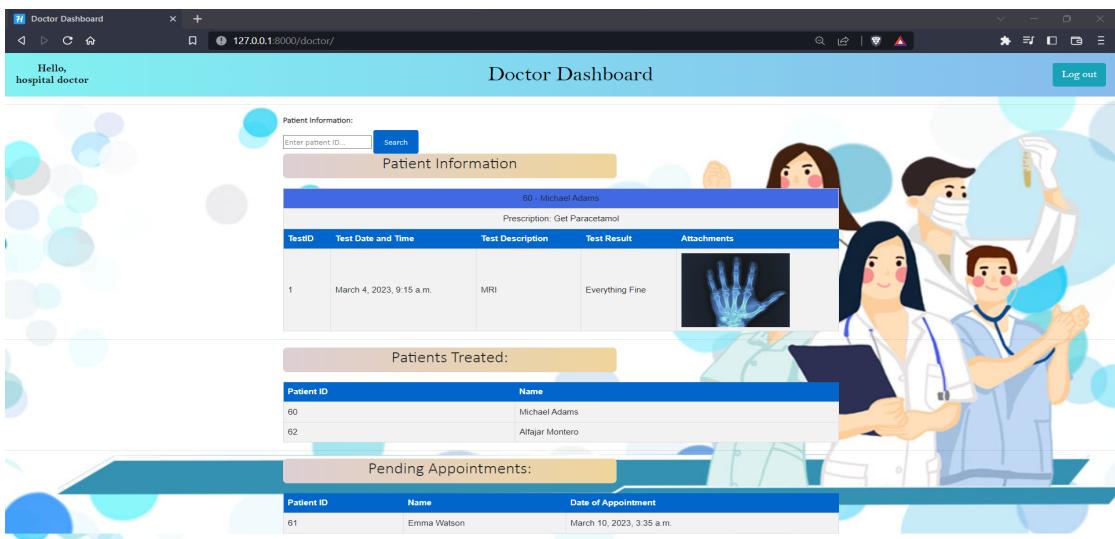
- Front desk Operator:

The screenshot shows a web browser window titled "Front Desk Operator Dashboard" at the URL 127.0.0.1:8000/frontdesk/. The dashboard has a header with "Hello, hospital frontdesk" and a "Log out" button. The main content area features a large image of a smiling man wearing a headset and a checkered shirt. Overlaid on this image are two forms: a "Register Patient" form and an "Admit Patient" form. The "Register Patient" form includes fields for First Name, Last Name, Date of Birth (dd-mm-yyyy), Gender (Male selected), and SSNID, with a "Register" button. The "Admit Patient" form shows a dropdown menu with "60- Michael Adams".

- Data Entry Operator:

The screenshot shows a web browser window titled "Data Operator Dashboard" at the URL 127.0.0.1:8000/dataoperator/. The dashboard has a header with "Hello, hospital dataop" and a "Log out" button. The main content area features three stacked forms: "Test and Result Input Form", "Patient Treatment Input Form", and "Treatment Prescription Information". The "Test and Result Input Form" includes a "Select Test" dropdown (TestID1, PatientID=60, Test Description: MRI), an "Input Test Result" text field, and an "Attach Image" section with a "Choose File" button. The "Patient Treatment Input Form" includes a "Select Patient" dropdown (61- Emma Watson) and a "Treatment Prescription Information" section with an "Input Treatment" text field.

- Doctor:



- Database Administrator:

Users			
Type	First Name	Last Name	EMPLOYEEID
doctor	hospital	doctor	10
admin	hospital	admin	1
front desk operator	hospital	frontdesk	21
data operator	hospital	dataop	31
doctor	Jay	Kumar	11

The frontend has been constructed using HTML and CSS. The basic design has been implemented using HTML and the specifications of the webpage. The frontend and backend were connected using Python and Django. The processes that are required to support the workflow have been written using Python and Django. Here is a query sample of the same.

```

def doctor(request):
    if not get_referer(request):
        return redirect('error')
    c.execute("select users.First_Name, users.Last_Name from users where users.EmployeeID=" + id)
    details = c.fetchall()

    docid = request.session['id']
    querypend = "select patient.SSN,patient.First_Name,patient.Last_Name,appointment.AppointmentDT from patient inner join appointment on patient.SSN=appointment.SSN"
    c.execute(querypend)
    patlist = c.fetchall()

    query="select patient.SSN,patient.First_Name,patient.Last_Name,prescription.TreatmentPres from patient inner join prescription on prescription.DoctorID="+docid
    c.execute(query)
    patlist1 = c.fetchall()

    patinfo = []
    ...

```

6 Technical Specifications

- Frontend and webpage: HTML and CSS
- Backend: Python using Django (a high level Python web framework)
- Database: MySQL (a relational database)
