Information Systems and Data Modeling – IT1090

Assignment

Title: Online Medical Portal

Batch Number: 3.02

Group Number: MLB_WD_03.02_07

Declaration:

We hold a copy of this assignment that we can produce if the original is lost or damaged.

We hereby certify that no part of this assignment has been copied from any other group's work or from any other source. No part of this assignment has been written / produced for our group by another person except where such collaboration has been authorized by the subject lecturer/tutor concerned.

Group Members:

IT22130716	MUNASINGHE M.A.M.C
IT22100320	SOMASUNDARA M.M

Submitted on: <13/10/2023>

Contents

Introduction.

In our examination of digital healthcare solutions, we considered how the internet may strengthen the relationship between patients and healthcare services. The result is our revolutionary online medical portal, a digital center designed to make access to healthcare easier. Patients can easily access their medical information, request appointments, receive medical advice, and leave insightful comments by joining our website. We offer a variety of services that make it easier to manage healthcare administration and give people the power to manage their own health through an easy-to-use internet interface.

A few technologies need to be investigated and understood in order to create an online medical portal website. They included relational databases like My SQL as well as computer languages like HTML, CSS, and JavaScript.

A database must support several types of data. Because managing a computer-based system for storing data is simpler than managing written or typed items, a database is necessary. The time and effort savings help both users and administrators. Furthermore, there is no chance of losing the data because it is kept in a database. The only ones who can affect the system are its users. Before creating a database, it is essential to gather requirements and do a requirement analysis. The data requirements, along with the functional and nonfunctional requirements, can then be defined.

Hypothetical Scenario.

An online healthcare platform called "FirstHealthChoice" offers a range of services to meet your healthcare needs. Whether it's finding a reputable general practitioner, locating a specialty healthcare specialist, or learning about top-notch healthcare facilities in their area, our users can easily find and communicate with medical professionals. Users of our website can also swiftly and securely retrieve the results of their tests. By giving all the resources necessary to locate the best medical treatment in one convenient platform, we priorities your healthcare journey. Non-registered users can browse our website and find helpful content as guests. Customers who want a more specialized experience and an easier time checking in can choose to register for an account by simply clicking the sign-up button. After successful login, our website's secure database checks user credentials and grants full access to the range of healthcare resources and services we provide. In addition to the features available to visitor users, registered users can take advantage of expanded capabilities such as appointment booking, cancellation, sharing healthcare experiences, offering feedback, and quickly updating their account details for a completely tailored healthcare journey.

Customer service is essential because it is the users' main point of contact with the system. They are in charge of answering questions and user communications as soon as possible. Additionally, customers can receive support from customer service for a few services, such as help with service requests, feedback management, appointment scheduling assistance, and real-time live chat for quick support.

On the other side, the Administrator is in charge of making sure the system is secure and runs without a hitch. They monitor frequent system upgrades to improve functionality and security in addition to system maintenance and data protection. The Administrator plays a crucial role in preserving the system's dependability and guaranteeing a modern user experience.

The Manager is in charge of numerous responsibilities and holds a major position within the platform. Along with managing all crucial platform tasks, the Manager is able to generate comprehensive reports, efficiently manage financial resources, and oversee compensation administration. The Manager, who has more authority and performs at a higher administrative level, makes sure that directives are carried out, resources are used effectively, and successes are meticulously recorded, establishing the way to excellence and ongoing improvement.

3 Requirement Analysis.

3.1 Main Requirements .

3.1.1 Functional Requirements.

The functional requirements give a description of the primary functions of the website and the interactions between users/patient and the system. Users can easily access medical information and services through an online medical portal, a digital platform created to improve and supply the healthcare services.

01.User/Patient Registration and Authentication:

User requirements –

- **User Registration:** It should be simple for the patients to register for the accounts using their contact information and personal information.
- **Profile Management:** patient must be able to manage their profiles. Access medical reports and to be able to edit and maintain their profiles.
- **Appointment Scheduling**: Patients should be able to view doctors who are available, verify their schedules, and make appointments in accordance with their schedules.
- **Medical Records:** Patients should have safe access to their electronic health records, which should include all test findings, diagnoses, and treatment plans.
- **Billing and Payment:** Patients should have access to their billing data, including insurance claims and payment history, and be able to examine and manage it.

- A registered user can access the system by entering the necessary user login information.
- Reservations can be made by Registered Users.
- Reserved Reservations can be cancelled by Registered Users.
- Registered users can contact the website's contact page with health-related questions.
- Registered users can communicate by sending a message outlining the problem or using the website's actual social networking capability.

System requirements-

- The patient's entered credentials should be verified by the system.
- The patient's information should be approved by the system and recorded.
- The system needs to show a reservation's availability on the webpage.
- The system needs to create a user account and approve the registration details.
- The system should allow users to upload their feedback and experiences.
- The system should ask the administrator for permission to upload feedback before storing the necessary information.
- System should notify the administrator of the cancellation of the reservation and keep the details.
- The system ought to save updated User Account information.

02. Customer Support (access to the system's back end)

User Requirements -

- By providing their login information, customer service can access the system.
- Customer service responds to user or patient messages.

System Requirements –

- System Display the messages Query list to the tour guide.
- The system should check the user credentials entered by the customer support representative.
- System enables patient and customer support to cooperation.

03. Administrator and Manager (access to the system's back end)

User Requirements-

- System enables patient and customer support cooperation.
- Administrator enters the website using the necessary login information.
- Administrator has the ability to add and remove staff accounts.
- User accounts can be enabled and disabled by the administrator.
- The administrator can arrange for refunds, approve cancelled reservations, and delete marked reservations from their calendar.
- The administrator looks at patient reviews, contacts, and comments.
- Manager can manage the employee salary.
- Manager has the ability in generating reports.

4. **Developer (can access the back end of the system).**

System Requirements –

• The system needs to check the user's login information.

• The database should be modified, and the system should update the downgraded patients' information.

• The patient reviews, contacts, and feedback are stored and displayed by the system.

• The manager's login credentials should be verified by the system.

3.1.2 Non-Functional Requirements.

Quality attributes are just non-functional requirements. It describes the aspects of the system that aren't specifically related to a given functionality. Potentially more important than functional needs are non-functional requirements. The system might not work if these goals are not met.

Speed

- The system may connect to more users concurrently without experiencing any issues.
- The system needs to operate quickly.

Reliability

- The system needs to be able to recognize incorrect user credentials.

Security

- The system should be able to safeguard user data and prevent unauthorized access, misuse, and forgeries.

- In addition, by requiring a special user ID and password, nobody else can access the system using someone else's credentials.

User Friendly

- Users with less IT knowledge should be able to navigate the system.

Availability

- System availability should be constant throughout the day and night.

3.2 Data Requirements

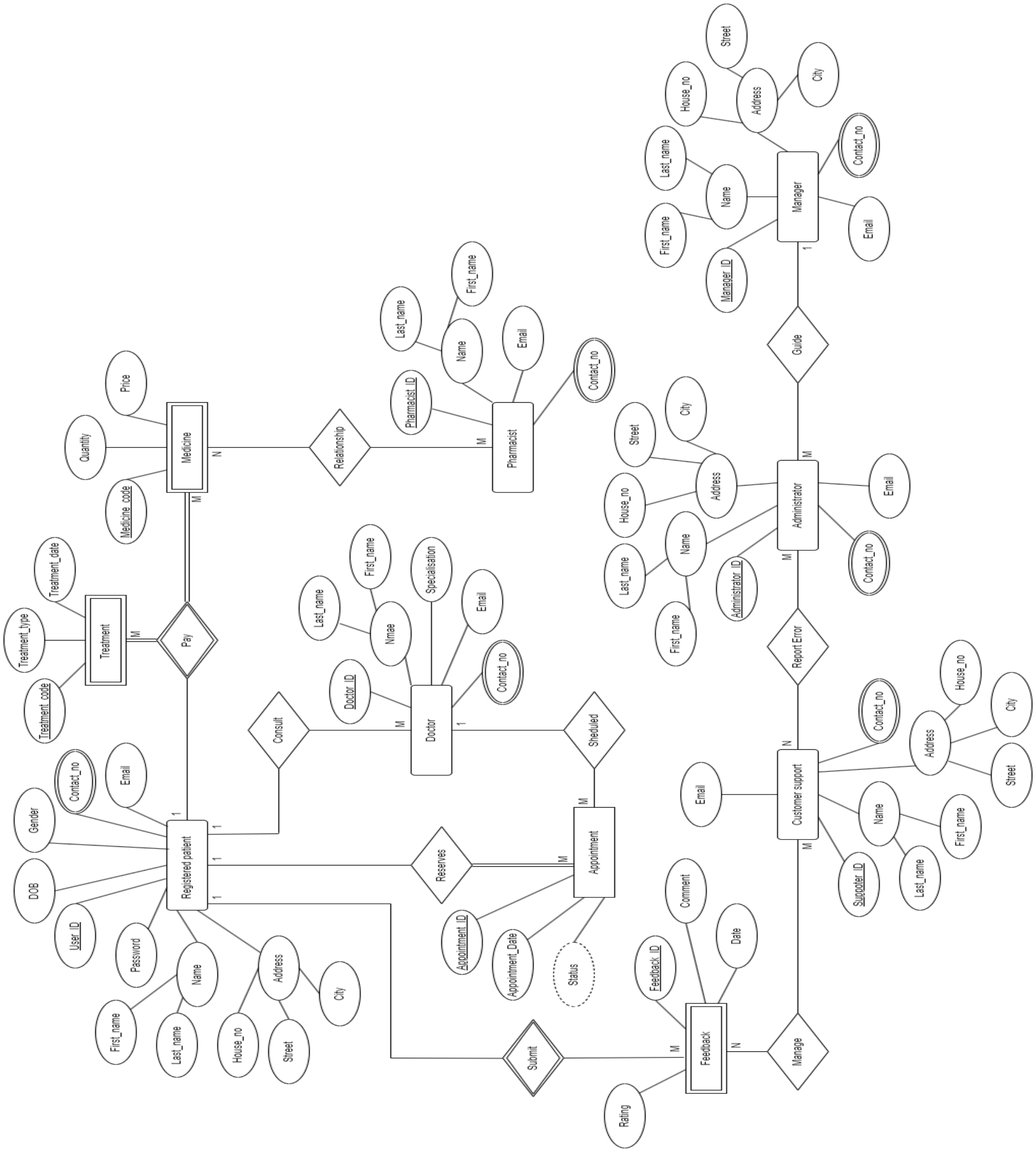
- Registered patients
    - User_ID
    - Password
    - First_name
    - Last_name
    - House_no
    - Street
    - City
    - DOB
    - Gender
    - Contact_no
    - Email
- Doctor
    - Doctor_ID
    - First_name
    - Last_name
    - Specialisation
    - Contact_no

  - Email
- Appointment
  - Appointment_ID
  - Appointment_Date
  - Status
- Feedback
  - Feedback_ID
  - Comment
  - Rating
  - Date
- Customer Support
  - Supporter_ID
  - First_name
  - Last_name
  - House_no
  - Street
  - City
  - Contact_no
  - Email

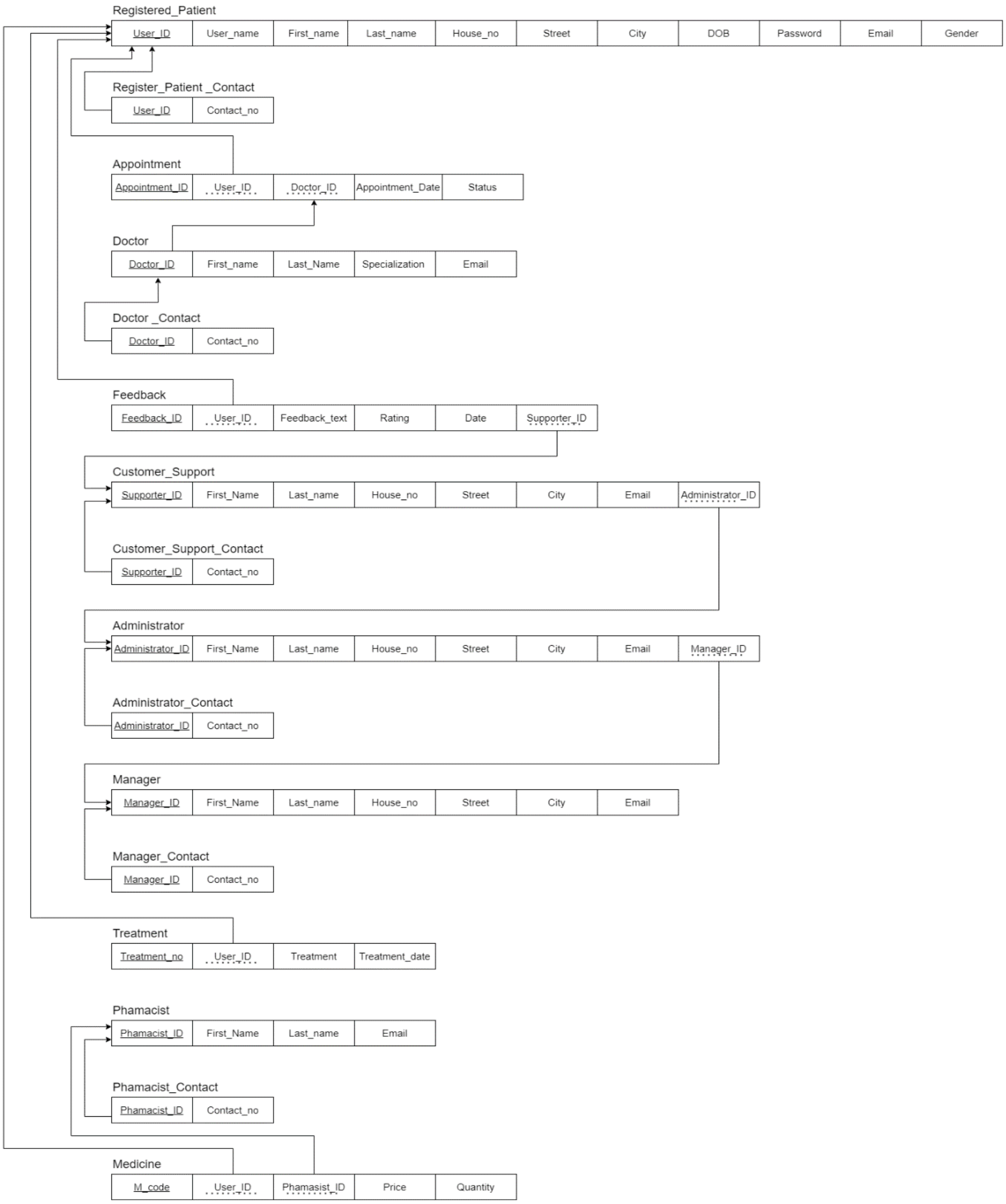
- Administrator
  - Admin_ID
  - First_name
  - Last_name
  - House_no
  - Street
  - City
  - Contact_no

    - Email
- Manager
    - Manager_ID
    - First_name
    - Last_name
    - House_no
    - Street
    - City
    - Contact_no
    - Email
- Treatment
    - Treatment_code
    - Treatment_type
    - Treatment_date
- Pharmacist
    - Phamasist_ID
    - First_name
    - Last_name
    - Contact_no
    - Email
- Medicine
    - Medicine_code
    - Quantity
    - Price

04.Entity Relationship (ER Diagram)



5 Relational Schema.



Registered_Patient

User_ID	User_name	First_name	Last_name	House_no	Street	City	DOB	Password	Email	Gender

Register_Patient _Contact

User_ID	Contact_no

Appointment

Appointment_ID	User_ID	Doctor_ID	Appointment_Date	Status

Doctor

Doctor_ID	First_name	Last_Name	Specialization	Email

Doctor _Contact

Doctor_ID	Contact_no

Feedback

Feedback_ID	User_ID	Feedback_text	Rating	Date	Supporter_ID

Customer_Support

Supporter_ID	First_Name	Last_name	House_no	Street	City	Email	Administrator_ID

Customer_Support_Contact

Supporter_ID	Contact_no

Administrator

Administrator_ID	First_Name	Last_name	House_no	Street	City	Email	Manager_ID

Administrator_Contact

Administrator_ID	Contact_no

Manager

Manager_ID	First_Name	Last_name	House_no	Street	City	Email

Manager_Contact

Manager_ID	Contact_no

Treatment

Treatment_no	User_ID	Treatment	Treatment_date

Phamacist

Phamacist_ID	First_Name	Last_name	Email

Phamacist_Contact

Phamacist_ID	Contact_no

Medicine

M_code	User_ID	Phamasist_ID	Price	Quantity

6 SQL Queries.

6.1 Data Base Create.

```
-- Table Registered_patient
CREATE TABLE Registered_patient
(
   User_ID varchar(15) PRIMARY KEY not null,
   User_name varchar(40) not null,
   First_name varchar(40) not null,
   Last_name varchar(40) not null,
   House_no varchar(30) not null,
   Street varchar(30) not null,
   City varchar(35) not null,
   DOB date not null,
   Password_ varchar(35) not null,
   Email varchar(50) CHECK (Email LIKE '%_@__%.__%') not null);

-- Table Registerd_Patient_Contact
CREATE TABLE Registerd_Patient_Contact
(
   User_ID varchar(15) not null,
   ContactNo int not null,
   FOREIGN KEY(User_ID) REFERENCES Registered_patient(User_ID)
);

-- Table Doctor
CREATE TABLE Doctor
(
   Doctor_ID varchar(15) PRIMARY KEY not null,
   First_name varchar(40) not null,
   Last_name varchar(40) not null,
   Specialization varchar(80) not null,
   Email varchar(50) CHECK (Email LIKE '%_@__%.__%') not null
);

-- Table Doctor_contact
CREATE TABLE Doctor_Contact
(
   Doctor_ID varchar(15) not null,
   ContactNo int not null,
   FOREIGN KEY(Doctor_ID) REFERENCES Doctor(Doctor_ID)
);
```

```
-- Table Appointments
CREATE TABLE Apointment
(
   Appointment_ID varchar(15) PRIMARY KEY not null,
   User_ID varchar(15) not null,
   Doctor_ID varchar (15) not null,
   Appointment_Date date not null,
   Status_ varchar (15) not null,
   FOREIGN KEY(User_ID) REFERENCES Registered_patient(User_ID),
   FOREIGN KEY(Doctor_ID) REFERENCES Doctor(Doctor_ID)
);

-- Table Manager
CREATE TABLE Manager
(
   Manager_ID varchar(15) PRIMARY KEY not null,
   First_name varchar(40) not null,
   Last_name varchar(40) not null,
   House_no varchar(30) not null,
   Street varchar(30) not null,
   City varchar(35) not null,
   Email varchar(50) CHECK (Email LIKE '%_@__%.__%') not null
);

-- Table Manager_Contact
CREATE TABLE Manager_Contact
(
   Manager_ID varchar(15) not null,
   ContactNo int not null,
   FOREIGN KEY(Manager_ID) REFERENCES Manager(Manager_ID)
);

-- Table Administrator
CREATE TABLE Administrator
(
   Administrator_ID varchar(15) PRIMARY KEY not null,
   Manager_ID varchar(15) not null,
   First_name varchar(40) not null,
   Last_name varchar(40) not null,
   House_no varchar(30) not null,
   Street varchar(30) not null,
   City varchar(35) not null,
    Email varchar(50) CHECK (Email LIKE '%_@__%.__%') not null,
    FOREIGN KEY(Manager_ID) REFERENCES Manager(Manager_ID)
);

-- Table Administrator_Contact
CREATE TABLE Administrator_Contact
(
    Administrator_ID varchar(15) not null,
    ContactNo int not null,
    FOREIGN KEY(Administrator_ID) REFERENCES Administrator(Administrator_ID)
);

-- Table Customer_support
CREATE TABLE Customer_support
(
    Supporter_ID varchar(15) PRIMARY KEY not null,
    Administrator_ID varchar(15) not null,
    First_name varchar(40) not null,
    Last_name varchar(40) not null,
    House_no varchar(30) not null,
    Street varchar(30) not null,
    City varchar(35) not null,
    Email varchar(50) CHECK (Email LIKE '%_@__%.__%') not null,
    FOREIGN KEY(Administrator_ID) REFERENCES Administrator(Administrator_ID)
);

-- Table Customer_support_contact
CREATE TABLE Customer_support_contact
(
    Supporter_ID varchar(15) not null,
    ContactNo int not null,
    FOREIGN KEY(Supporter_ID) REFERENCES Customer_support(Supporter_ID)
);

-- Table Feedback
CREATE TABLE Feedback
(
    Feedback_ID varchar(15) PRIMARY KEY not null,
    User_ID varchar(15) not null,
    Supporter_ID varchar(15) not null,
    Feedback_text varchar(100) not null,
    Rating int not null,
    Date date not null,
    FOREIGN KEY(User_ID) REFERENCES Registered_patient(User_ID),
    FOREIGN KEY(Supporter_ID) REFERENCES Customer_support(Supporter_ID)
);

-- Table Treatment
CREATE TABLE Treatment
(
    Treatment_no varchar(15) PRIMARY KEY not null,
    User_ID varchar(15) not null,
    Treatment varchar(100) not null,
    Treatment_Date date not null,
    FOREIGN KEY(User_ID) REFERENCES Registered_patient(User_ID)
);

-- Table Phamacist
CREATE TABLE Phamacist
(
    Phamacist_ID varchar(15) PRIMARY KEY not null,
    First_name varchar(40) not null,
    Last_name varchar(40) not null,
    Email varchar(50) CHECK (Email LIKE '%_@__%.__%') not null
);

-- Table Phamacist_contact
CREATE TABLE Pharmacist_contact
(
    Pharmacist_ID varchar(15) not null,
    ContactNo int not null,
    FOREIGN KEY(Pharmacist_ID) REFERENCES Phamacist(Phamacist_ID)
);

-- Table Medicine
CREATE TABLE Medicine
(
    M_code varchar(15) PRIMARY KEY not null,
    User_ID varchar(15) not null,
    Pharmacist_ID varchar(15) not null,
    Price decimal(10, 2) not null,
    Quantity int not null,
    FOREIGN KEY(User_ID) REFERENCES Registered_patient(User_ID),
    FOREIGN KEY(Pharmacist_ID) REFERENCES Phamacist(Phamacist_ID)
);
```

6.2 Data Store in Data Base.

```
-- Registered_patient table details
INSERT INTO Registered_patient VALUES ('P001', 'JayawardanaM', 'Madura', 'Jayawardena',
'24',
'79/2', 'Maharagama', '1989-02-09', 'Madura#ja', 'madurajay@gmail.com');

INSERT INTO Registered_patient VALUES ('P002', 'DanaPerera', 'Dananjaya', 'Perera', '02',
'08/04', 'Kalaniya', '1999-08-01', 'Perera@9876', 'DananjayaPerera@gmail.com');

INSERT INTO Registered_patient VALUES ('P003', 'SamanWeerakone', 'Saman', 'Weerakone',
'56',
'44/01', 'Kandy', '1961-02-09', 'Weerakone45', 'Samanweerakone@gmail.com');

INSERT INTO Registered_patient VALUES ('P004', 'YashodaA', 'Yashoda', 'Amarasinghe', '88',
'22/36', 'Kottawa', '1998-05-04', 'yashAmarasinghe', 'yashoda98@gmail.com');

INSERT INTO Registered_patient VALUES ('P005', 'AnnSamarawickrama', 'Ann',
'Samarawickrama', '11',
'06/06', 'Malabe', '1973-03-10', 'AnnS6655', 'Ann6655@gmail.com');

-- Registerd_Patient_Contact table details
INSERT INTO Registerd_Patient_Contact VALUES ('P001','0772654235');
INSERT INTO Registerd_Patient_Contact VALUES ('P002','0724986574');
INSERT INTO Registerd_Patient_Contact VALUES ('P003','0717862198');
INSERT INTO Registerd_Patient_Contact VALUES ('P004','0702754354');
INSERT INTO Registerd_Patient_Contact VALUES ('P005','0714285425');

-- Doctor table details
INSERT INTO Doctor VALUES ('D001', 'Priyantha', 'Mendis', 'Dermatologist',
'priyanthamendis@gmail.com');
INSERT INTO Doctor VALUES ('D002', 'Jagath', 'kuamra', 'Cardiologist',
'jagathkumara@gmail.com');
INSERT INTO Doctor VALUES ('D003', 'Jamanthi', 'Athukorala', 'Neurologist',
'jayamanthiAthukorala@gmail.com');
INSERT INTO Doctor VALUES ('D004', 'Sugath', 'Sirimanne', 'Orthopedic Surgeon',
'sugathsirimanne@gmail.com');
INSERT INTO Doctor VALUES ('D005', 'Nethmi', 'Bandara', 'Psychiatrist',
'nethmibandara@gmail.com');

-- Doctor_Contact table details
```

```
INSERT INTO Doctor_Contact VALUES ('D001','0714589652');
INSERT INTO Doctor_Contact VALUES ('D002','0773254896');
INSERT INTO Doctor_Contact VALUES ('D003','0728457214');
INSERT INTO Doctor_Contact VALUES ('D004','0772168753');
INSERT INTO Doctor_Contact VALUES ('D005','0709324857');

-- Appointments table details
INSERT INTO Apointment VALUES ('A001','P001','D001','2023-11-10','scheduled');
INSERT INTO Apointment VALUES ('A002','P002','D002','2023-11-8','cancelled');
INSERT INTO Apointment VALUES ('A003','P003','D003','2023-10-28','scheduled');
INSERT INTO Apointment VALUES ('A004','P004','D004','2023-11-14','cancelled');
INSERT INTO Apointment VALUES ('A005','P005','D005','2023-10-30','scheduled');

-- Manager table details
INSERT INTO Manager VALUES
('M001','Bashitha','Nanayakkara','28','32/9','Gampaha','Bashitha0@gmail.com');
INSERT INTO Manager VALUES
('M002','Aravinda','Sirisena','68','52/16','Malabe','Aravinda32@gmail.com');

-- Manager_Contact details
INSERT INTO Manager_Contact VALUES ('M001', '0747845378');
INSERT INTO Manager_Contact VALUES ('M002', '0772356449');

-- Administrator table details
INSERT INTO Administrator VALUES ('Ad001', 'M001', 'Kalana', 'Wijesuriya', '56', '100/02',
'Pannipitiya', 'kalanawije@gmail.com');
INSERT INTO Administrator VALUES ('Ad002', 'M001', 'Nirosha', 'Ariyawansa', '06', '75/13' ,
'Kaduwela', 'niroshaA@gmail.com');
INSERT INTO Administrator VALUES ('Ad003', 'M002', 'Dulshan', 'Gamage', '11', '36/58' ,
'Thalawathugoda', 'dulshangamage@gmail.com');
INSERT INTO Administrator VALUES ('Ad004', 'M002', 'Harshana', 'Gunathilake', '24', '69/25' ,
'Battaramulla', 'harshanagunathilake@gmail.com');

-- Administrator_Contact details
INSERT INTO Administrator_Contact VALUES ('Ad001', '0771456281');
INSERT INTO Administrator_Contact VALUES ('Ad002', '0701249325');
INSERT INTO Administrator_Contact VALUES ('Ad003', '0784215432');
INSERT INTO Administrator_Contact VALUES ('Ad004', '0765894512');

-- Customer_support table details
INSERT INTO Customer_support VALUES
('S001','Ad001','Rayan','Akanayaka','16','62/02','Kandy','Rayan12@gmail.com');
```

```
INSERT INTO Customer_support VALUES
('S002','Ad002','Shehan','Disanayaka','26','32/08','Mathara','sheshan@gmail.com');
INSERT INTO Customer_support VALUES
('S003','Ad003','Kshan','Bandara','42','84/22','Ampara','kshan.k@gmail.com');
INSERT INTO Customer_support VALUES
('S004','Ad004','Himaya','Jayakodi','63','46/18','Anuradhapura','himayaJ@gmail.com');
INSERT INTO Customer_support VALUES
('S005','Ad001','Chamodi','Perera','86','74/14','Kegalle','chamodi99@gmail.com');

-- Customer_support_contact table details
INSERT INTO Customer_support_contact VALUES ('S001', '0725689110');
INSERT INTO Customer_support_contact VALUES ('S002', '0778964231');
INSERT INTO Customer_support_contact VALUES ('S003', '0763245876');
INSERT INTO Customer_support_contact VALUES ('S004', '0777246932');
INSERT INTO Customer_support_contact VALUES ('S005', '0703621870');

-- Feedback table details
INSERT INTO Feedback VALUES ('F001','P001','S001','professional online
healthcare.','4','2023-10-26');
INSERT INTO Feedback VALUES ('F002','P002','S002','Average portal, some scheduling
delays','2','2023-10-26');
INSERT INTO Feedback VALUES ('F003','P003','S003','User friendly portal.','5','2023-10-26');
INSERT INTO Feedback VALUES ('F004','P004','S004','reliable healthcare.','3','2023-10-26');
INSERT INTO Feedback VALUES ('F005','P005','S005','Average decent guidance.','2','2023-10-
26');

-- Treatment table details
INSERT INTO Treatment VALUES ('T001','P001','Radiationtherapy','2023-11-26');
INSERT INTO Treatment VALUES ('T002','P002','Diet and Nutrition Therapy','2023-10-18');
INSERT INTO Treatment VALUES ('T003','P003','Physical Therapy','2023-10-21');
INSERT INTO Treatment VALUES ('T004','P004','Chemotherapy','2023-11-06');
INSERT INTO Treatment VALUES ('T005','P005','Respiratory Therapy','2023-11-14');

-- Phamacist details
INSERT INTO Phamacist VALUES ('Ph001', 'Sanjana', 'Jayaweera',
'sanjanajayaweera@gmail.com');
INSERT INTO Phamacist VALUES ('Ph002', 'Sithmi', 'Mendis', 'sithmimendis@gmail.com');
INSERT INTO Phamacist VALUES ('Ph003', 'Isuru', 'Karunanayake',
'isurukarunanayake@gmail.com');

-- Phamacist_contact details
INSERT INTO Pharmacist_contact VALUES ('Ph001', '0776556365');
INSERT INTO Pharmacist_contact VALUES ('Ph001', '0701245798');
```

```sql
INSERT INTO Pharmacist_contact VALUES ('Ph001', '0767894567');

-- Medicine table details
INSERT INTO Medicine VALUES ('Md001','P001','Ph001','2000.00','50');
INSERT INTO Medicine VALUES ('Md002','P002','Ph002','3500.00','30');
INSERT INTO Medicine VALUES ('Md003','P003','Ph003','5000.00','100');
INSERT INTO Medicine VALUES ('Md004','P004','Ph002','4200.00','250');
INSERT INTO Medicine VALUES ('Md005','P005','Ph001','1800.00','2');
```

7 Performance Requirement.

Performance Requirements are important to the system's effectiveness. They include the following.

- For a registered user to access the system without experiencing any difficulties, it must be operational around the clock, every day of the year.
- A registered user can use their login information to access the system multiple times.
- The login process and page loading must be completed in a matter of seconds.
- The performance requirements for this system are speed and usability.
- A registered user has access to a doctor's details.
- A registered user has the ability to change or remove account information.
- System loads in a short amount of time.
- The administrator has the ability to manage user accounts, add/remove shared experiences and feedback, approve and cancel reservation requests, and edit/update appointment details.
- System enables user messages to be responded to by customer service.
- The website's developer can add new functionality.
- The system can be updated by the developer.
- In order to fix bugs and errors, developers must find them.
- Create a user interface that is easy to use.
- Users must be able to use any device or browser to access the website at any time.
- The system gives the administrator the ability to control staff accounts.

8 Security Requirements.

- Before being sent to the database, personal information from users should be encrypted.
- Restricted features should be restricted to unauthorized users.
- A database backup of all the data in the system is required.
- The user account password must be a strong password that contains uppercase, lowercase, numbers, and special characters.
- Only one user account should be created for each email address.
- The database server must be maintained with backup servers.
- Only administrators have access to and control over the system's data.