FACULTY OF COMPUTING
UTM Johor Bahru

Project: Phase 3

<AppointmentPro PKU>

SECD2523 - DATABASE

SEMESTER I, SESSION

2023/2024

Lecturer: Dr. Noor Hidayah

Group Name:
WELOVEDATABASE ♡

Name	Matric No
WANG YU TING	A22EC0291
LEE XIN HUI	A22EC0066
TEOH ZHI YONG	A22EC0125
ELVIS CHANG FENG JIE	A22EC0157

Table of Contents

1.0 Introduction

The healthcare system, developed for Pusat Kesihatan Universiti (PKU), is a platform that can help to manage the appointment system. In the current system, the appointment system depends on email notifications to remind patients of their appointments. However, because email is used less frequently, this approach has faced difficulties. Additionally, manual backups are still kept in physical books even though there is a computerized system in place for recording patient information. The redundant nature of this practice raises questions regarding the accuracy of the data and the effective use of technology. Therefore, our team is assigned to design a database system based on the problems found in the current system.

The healthcare system serves as more than a scheduling tool. It aims to optimize healthcare processes, focusing on patient-centricity and integration. The university's overall objective of encouraging a health-conscious campus culture is in line with its implementation. The healthcare system is a reflection of the university's commitment to prioritising the overall well-being of its staff and students by utilising technology to improve accessibility and efficiency. Beyond its practical use, the system represents a dedication to developing an environment in which everyone's health and welfare come first.

In conclusion, the PKU appointment system represents a big step towards our university's goal of having a responsive, patient-centered, and technologically advanced healthcare environment. It not only takes care of scheduling issues but also establishes the foundation for a comprehensive approach to healthcare administration. The university's dedication to providing innovative healthcare solutions to its student body is shown by this initiative.

2.0 Overview of Project

In the third phase of our project, we focus on transforming a conceptual Entity-Relationship Diagram (ERD) into a logical ERD. In order to achieve this, we remove non-relational features from the conceptual ERD in a systematic way. This involves simplifying the diagram by removing components—like relationships or complex associations—that don't belong in the relational model's framework.

The next essential step is to use this refined ERD to derive a relational schema. The information is arranged using standard database principles into relational tables during this process. After that, these derived relations are normalised to achieve the Boyce-Codd Normal Form (BCNF). Normalisation ensures that data redundancy is minimized, dependencies are appropriately managed, and the database structure remains solid and efficient.

Additionally, this phase also emphasises the importance of keeping accurate documentation. The normalised design should be used as a guide when updating the data dictionary, which acts as a comprehensive guide to the structure and content of the database. This guarantees coherence and consistency in understanding the database's elements and their relationships.

Furthermore, interface design is used to confirm that the logical ERD matches the system's operational requirements. We ensure that the suggested structure efficiently supports the intended functionalities and operations by connecting the ERD to the system's transaction requirements through interface design. This verification procedure ensures that, in the context of the system's operations, the logical ERD is not only theoretically sound but also practically achievable.

Once the final logical Entity-Relationship Diagram (ERD) has been established through the transformation and normalization process, the next step is to create the SQL statements that will drive the functionalities of the system. These SQL statements will serve as the backbone, which will make it easier to interact with the database and carry out the required tasks as stated in the functions and specifications of the system.

Lastly, thorough testing is essential before the start of the implementation phase. This includes carefully evaluating the system compared to the established requirements and specifications. By ensuring that the system meets the required requirements and completes the specified functionalities, the verification process ensures that the system functions as planned. By taking an organised approach, it is possible to make sure that the system's functionalities exactly match the requirements and specifications, which improves system reliability and user satisfaction after deployment.

3.0 Database Conceptual Design

3.1 Updated business rule

Doctor Workflow:

1. Doctor Registration and Authentication:
    a. Patients can register an account.

2. Doctor Login:
    a. Admin can log in with a registered account.

3. Doctor Dashboard:
    a. Upon successful login, direct the doctor to the admin dashboard.

4. Appointment Management:
    a. Doctors can view upcoming appointments on a color-coded calendar.
    b. Colors vary based on the number of appointments per day.
    c. Doctors can create, modify, or cancel appointments on the calendar.
    d. Doctors can manage and arrange time slots for each appointment.

5. Patient Information:
    a. Doctors can search for patient information by patient ID or name.
    b. Display patient information upon search.
    c. Doctors can update medical records for a patient.

6. Communication:
    a. Doctors can contact patients through an integrated Whatsapp chat box.
    b. Send reminders and confirmations for appointments.

7. Feedback:
    a. Doctors can submit feedback
    b. Display a feedback form for users to fill and submit.

8. Logout:
    a. Provide an option for the admin to securely log out.

Patient Workflow:

1. Patient Registration and Authentication:
    a. Patients can register an account.

2. Patient Login:
    a. Patients can log in securely to their accounts.

3. Patients Dashboard:
    a. Upon successful login, direct the patient to the patient dashboard.

4. Patient Information:
    a. Patients can access and modify personal information.
    b. Patients can access medical records.

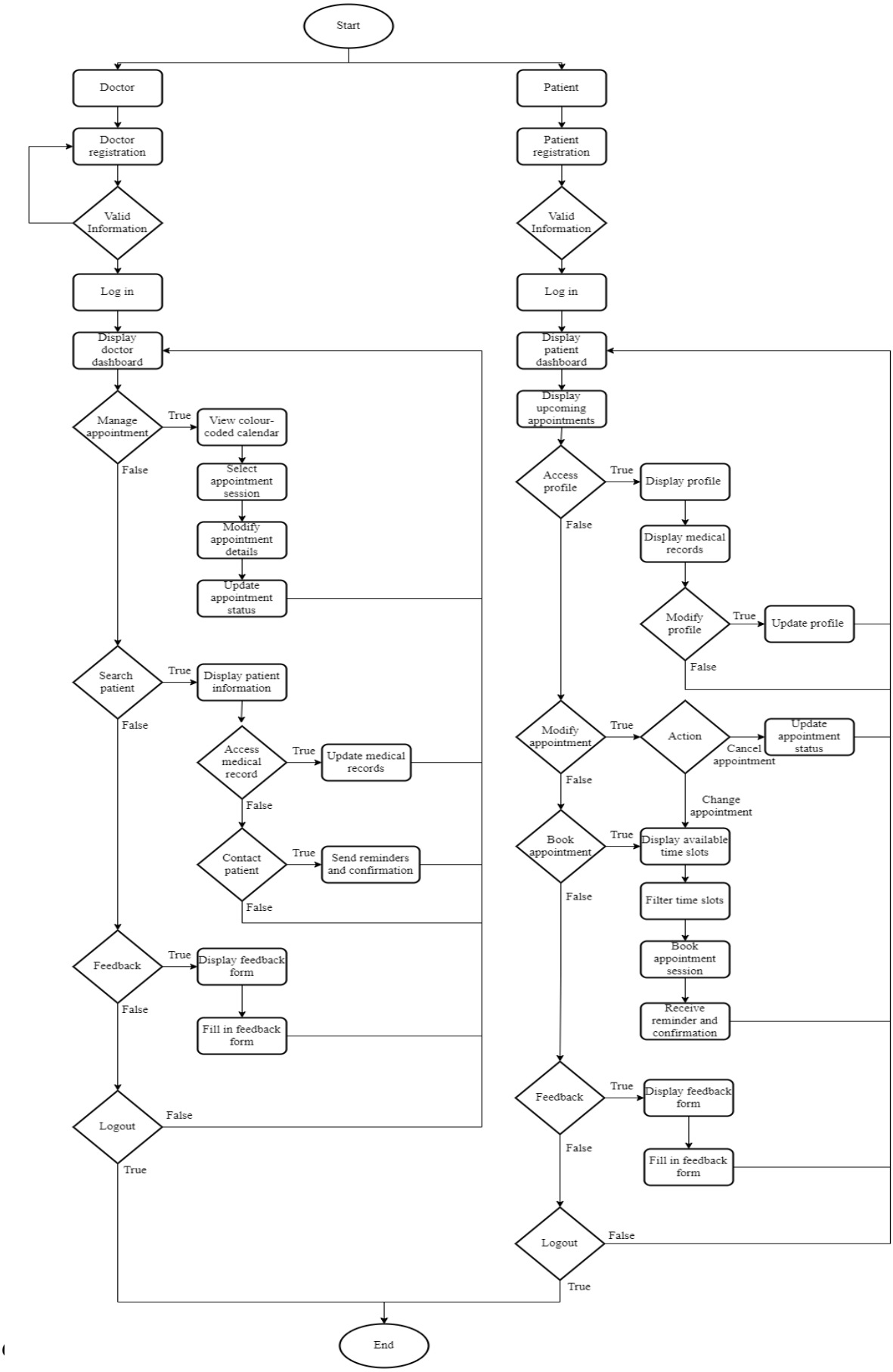
5. Appointment Booking:
    a. Patients can view available time slots for appointments.
    b. Patients can filter time slots based on date and time preferences.
    c. Patients can book appointments on selected time slots.

6. Manage appointments:
    a. Patients can change appointment date and time
    b. Patients can cancel appointments

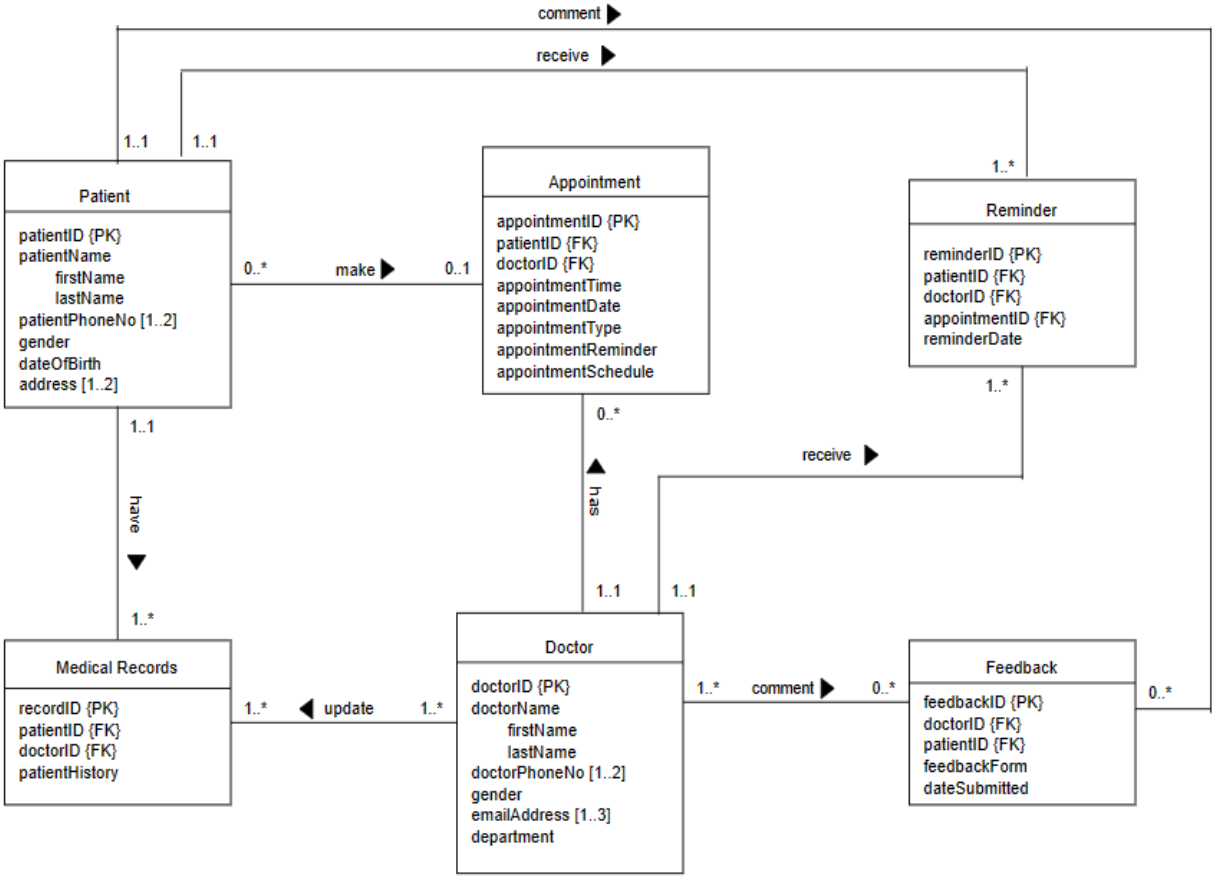
7. Reminder System:
    a. Patients receive reminders through Whatsapp a day before the scheduled appointment.

8. Feedback Submission:
    a. Patients can submit feedback.
    b. Display a feedback form for users to fill and submit.

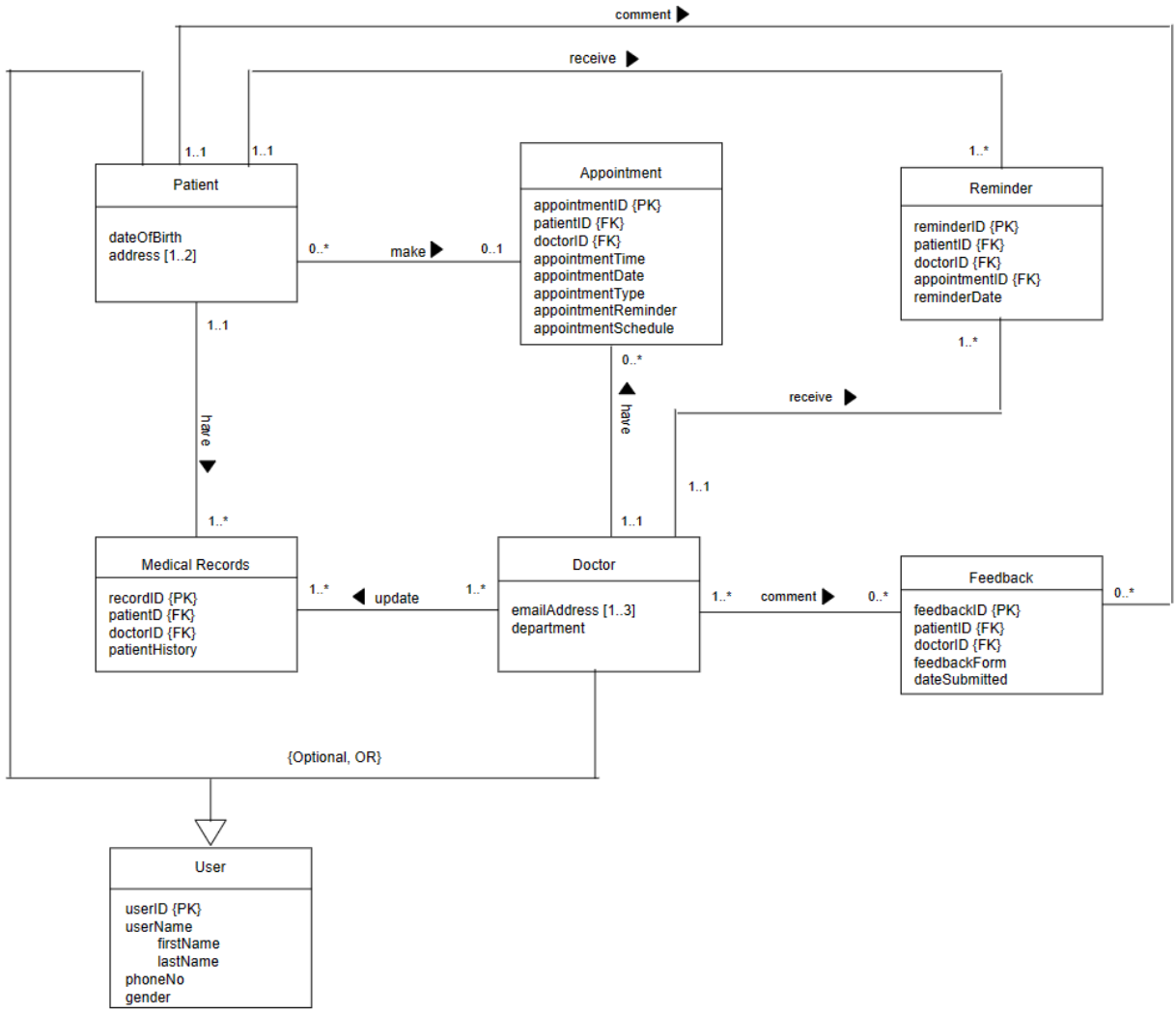
9. Logout:
    a. Provide an option for the patients to securely log out.


Start
Doctor
Doctor registration
Valid Information
Log in
Display doctor dashboard
Manage appointment
True
View colour-coded calendar
Select appointment session
Modify appointment details
Update appointment status
False
Search patient
True
Display patient information
Access medical record
True
Update medical records
False
Contact patient
True
Send reminders and confirmation
False
False
Feedback
True
Display feedback form
Fill in feedback form
False
Logout
False
True
Patient
Patient registration
Valid Information
Log in
Display patient dashboard
Display upcoming appointments
Access profile
True
Display profile
Display medical records
Modify profile
True
Update profile
False
False
Modify appointment
True
Action
Cancel appointment
Update appointment status
Change appointment
False
Book appointment
True
Display available time slots
Filter time slots
Book appointment session
Receive reminder and confirmation
False
Feedback
True
Display feedback form
Fill in feedback form
False
Logout
False
True
End

3.2 Conceptual ERD

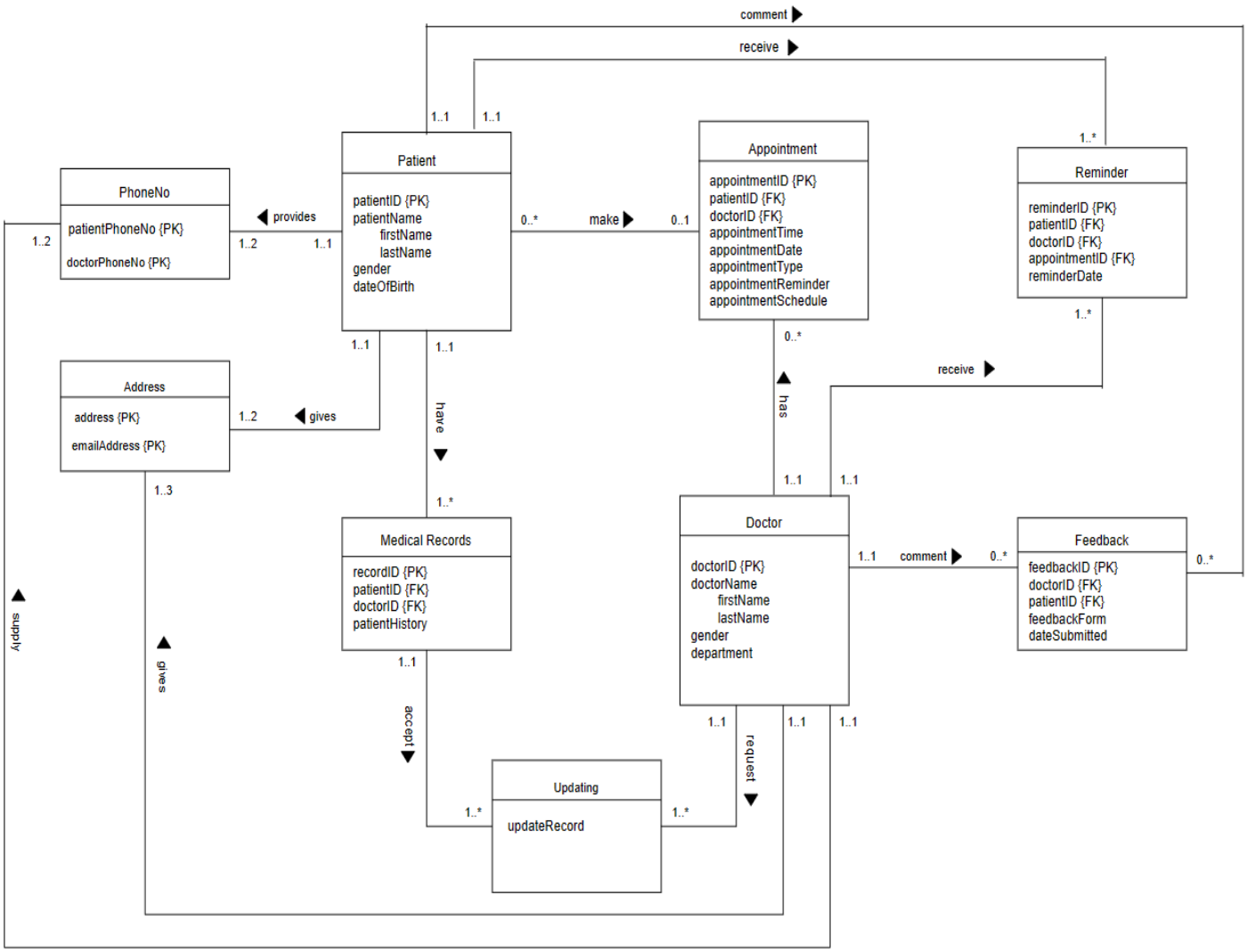


3.3 Enhanced ERD



4.0 DB logical design

4.1 Logical ERD



4.2 Updated Data Dictionary

Relation: Patient

Attributes	Data Type	Data length	Constraint	Description
patientID	NUMBER	10	PRIMARY KEY	Identification Number of the patient, automatically generated
patientName	VARCHAR2	30	PRIMARY KEY	Name of the patient
firstName	VARCHAR2	20	NOT NULL	First name of the patient
lastName	VARCHAR2	20	NOT NULL	Last name of the patient
gender	VARCHAR2	10	NOT NULL	Gender of the patient
dateOfBirth	NUMBER	10	NOT NULL	Date of birth of the patient

Relation: Doctor

Attributes	Data Type	Data length	Constraint	Description
doctorID	NUMBER	10	PRIMARY KEY	Identification Number of the doctor, automatically generated
doctorName	VARCHAR2	30	PRIMARY KEY	Name of the doctor
firstName	VARCHAR2	20	NOT NULL	First name of the doctor
lastName	VARCHAR2	20	NOT NULL	Last name of the doctor
gender	VARCHAR2	10	NOT NULL	Gender of the doctor
department	VARCHAR2	20	NOT NULL	Department of the doctor

Relation: PhoneNo

Attributes	Data Type	Data length	Constraint	Description
patientPhoneNo	NUMBER	11	NOT NULL	Contact number of the patient
doctorPhoneNo	NUMBER	11	NOT NULL	Contact number of the doctor

Relation: Address

Attributes	Data Type	Data length	Constraint	Description
address	VARCHAR2	30	NOT NULL	Email address of the patient
emailAddress	VARCHAR2	30	NOT NULL	Email address of the doctor

Relation: Appointment

Attributes	Data Type	Data length	Constraint	Description
appointmentID	NUMBER	10	PRIMARY KEY	Identification Number of the appointment, automatically generated
patientID	NUMBER	10	FOREIGN KEY	Identification Number of the patient, automatically generated
doctorID	NUMBER	10	FOREIGN KEY	Identification Number of the doctor, automatically generated
appointmentTime	NUMBER	10	NOT NULL	Appointment time
appointmentDate	NUMBER	10	NOT NULL	Appointment date
appointmentType	VARCHAR2	20	NOT NULL	Appointment type,eg. psychology
appointmentReminder	VARCHAR2	20	NOT NULL	Reminder of the appointment
appointmentSchedule	VARCHAR2	20	NOT NULL	Schedule of the appointment

Relation: Reminder

Attributes	Data Type	Data length	Constraint	Description
reminderID	NUMBER	10	PRIMARY KEY	Identification Number of the reminder, automatically generated
patientID	NUMBER	10	FOREIGN KEY	Identification Number of the patient, automatically generated
doctorID	NUMBER	10	FOREIGN KEY	Identification Number of the doctor, automatically generated
appointmentID	NUMBER	10	FOREIGN KEY	Identification Number of the appointment, automatically generated
reminderDate	NUMBER	10	NOT NULL	Reminder date

Relation: Medical Records

Attributes	Data Type	Data length	Constraint	Description
recordID	NUMBER	10	PRIMARY KEY	Identification Number of the record, automatically generated
patientID	NUMBER	10	FOREIGN KEY	Identification Number of the patient, automatically generated
doctorID	NUMBER	10	FOREIGN KEY	Identification Number of the doctor, automatically generated
patientHistory	VARCHAR2	20	NOT NULL	History treatment of the patient

Relation: Updating

Attributes	Data Type	Data length	Constraint	Description
updateRecord	NUMBER	10	PRIMARY KEY	Updates the record from doctor

Relation: Feedback

Attributes	Data Type	Data length	Constraint	Description
feedbackID	NUMBER	10	PRIMARY KEY	Identification Number of feedback, automatically generated
doctorID	NUMBER	10	FOREIGN KEY	Identification Number of the doctor, automatically generated
patientID	NUMBER	10	FOREIGN KEY	Identification Number of the patient, automatically generated
feedbackForm	VARCHAR2	20	NOT NULL	Feedback of the service
dateSubmitted	NUMBER	10	NOT NULL	Date submitted of the feedback

4.3 Normalization

1. Patient (patientID, patientName, firstName, lastName, gender, dateOfBirth)
   Fd1:
   patientID → patientName, firstName, lastName, gender, dateOfBirth
   1NF & 2NF & 3NF & BCNF:
   Patient (patientID, patientName, firstName, lastName, gender, dateOfBirth)

2. Doctor (doctorID, doctorName, firstName, lastName, gender, department)
   Fd1:
   doctorID → doctorName, firstName, lastName, gender, department
   1NF & 2NF & 3NF & BCNF:
   Doctor (doctorID, doctorName, firstName, lastName, gender, department)

3. PhoneNo (patientPhoneNo, doctorPhoneNo)
   Fd1:
   PhoneNo → patientPhoneNo, doctorPhoneNo
   1NF & 2NF & 3NF & BCNF:
   PhoneNo (patientPhoneNo, doctorPhoneNo)

4. Address (address, emailAddress)
   Fd1:
   Address → address, emailAddress
   1NF & 2NF & 3NF & BCNF:
   Address (address, emailAddress)

5. Appointment (appointmentID, patientID, doctorID, appointmentTime, appointmentDate, appointmentType, appointmentReminder, appointmentSchedule)
   Fd1:
   Appointment → appointmentID, patientID, doctorID, appointmentTime, appointmentDate, appointmentType, appointmentReminder, appointmentSchedule
   1NF & 2NF & 3NF & BCNF:
   Appointment (appointmentID, patientID, doctorID, appointmentTime, appointmentDate, appointmentType, appointmentReminder, appointmentSchedule)

6. Reminder (reminderID, patientID, doctorID, appointmentID, reminderDate)
   Fd1:
   Reminder → reminderID, patientID, doctorID, appointmentID, reminderDate
   1NF & 2NF & 3NF & BCNF:
   Reminder (reminderID, patientID, doctorID, appointmentID, reminderDate)

7. Medical Records (recordID, patientID, doctorID, patientHistory)
   Fd1:
   Medical Records → recordID, patientID, doctorID, patientHistory
   1NF & 2NF & 3NF & BCNF:
   Medical Records (<u>recordID</u>, patientID, doctorID, patientHistory)

8. Updating (updateRecord)
   Fd1:
   Updating → updateRecord
   1NF & 2NF & 3NF & BCNF:
   Updating (<u>updateRecord</u>)

9. Feedback (feedbackID, doctorID, patientID, feedbackForm, dateSubmitted)
   Fd1:
   Feedback → feedbackID, doctorID, patientID, feedbackForm, dateSubmitted
   1NF & 2NF & 3NF & BCNF:
   Feedback (<u>feedbackID</u>, doctorID, patientID, feedbackForm, dateSubmitted)

5.0 Relational DB Schemas (after normalization)

These are the set of relation schemas in relational database schema for Health System database.

Patient (patientID, patientName, firstName, lastName, gender, dateOfBirth)

Doctor (doctorID, doctorName, firstName, lastName, gender, department)

PhoneNo (patientPhoneNo, doctorPhoneNo)

Address (address, emailAddress)

Appointment (appointmentID, patientID, doctorID, appointmentTime, appointmentDate, appointmentType, appointmentReminder, appointmentSchedule)

Reminder (reminderID, patientID, doctorID, appointmentID, reminderDate)

Medical Records (recordID, patientID, doctorID, patientHistory)

Updating (updateRecord)

Feedback (feedbackID, doctorID, patientID, feedbackForm, dataSubmitted)

Patient

patientID	patientName	firstName	lastName	gender	dateOfBirth

Doctor

doctorID	doctorName	firstName	lastName	gender	department

PhoneNo

patientPhoneNo	doctorPhoneNo

Address

address	emailAddress

Appointment

appointment ID	patient ID	doctor ID	appointment Time	appointment Date	appointment Type	appointment Reminder	appointment Schedule

Reminder

reminderID	patientID	doctorID	appointmentID	reminderDate

Medical Records

recordID	patientID	doctorID	patientHistory

Updating

updateRecord

Feedback

feedbackID	doctorID	patientID	feedbackForm	dateSubmitted

6.0 SQL Statements (DDL & DML)

6.1 Data Definition Language

```
CREATE TABLE USER_def(
   userNo VARCHAR2(15) NOT NULL,
   firstName VARCHAR2(15) NOT NULL,
   lastName VARCHAR2(15) NOT NULL,
   phoneNo VARCHAR2(15) NOT NULL,
   gender VARCHAR2(10),
   CONSTRAINT userID_PK PRIMARY KEY(userNo)
);


CREATE TABLE DOCTOR(
   userNo VARCHAR2(15),
   doctorID VARCHAR2(15) PRIMARY KEY,
   emailAddress VARCHAR(100),
   department VARCHAR(50),
   CONSTRAINT userID_FK_D FOREIGN KEY(userNo) REFERENCES USER_def(userNo)
);


CREATE TABLE PATIENT(
   userNo VARCHAR2(15),
   patientID VARCHAR2(15) PRIMARY KEY,
   dateOfBirth DATE,
   address VARCHAR2(200),
   CONSTRAINT user_FK FOREIGN KEY(userNo) REFERENCES USER_def(userNo)
);


CREATE TABLE APPOINTMENT(
   appointmentID VARCHAR2(15) PRIMARY KEY,
   patientID VARCHAR2(15),
   doctorID VARCHAR2(15),
   appointmentTime TIMESTAMP,
   appointmentDate DATE,
   appointmentType VARCHAR2(50),
   appointmentReminder VARCHAR2(50),
   appointmentSchedule VARCHAR2(100),
   CONSTRAINT patientID_FK_A FOREIGN KEY(patientID) REFERENCES PATIENT(patientID),
   CONSTRAINT doctorID_FK_A FOREIGN KEY(doctorID) REFERENCES DOCTOR(doctorID)
);
```

```
CREATE TABLE REMINDER(
   reminderID VARCHAR2(15),
   patientID VARCHAR2(15),
   doctorID VARCHAR2(15),
   appointmentID VARCHAR2(15),
   reminderDate DATE,
   CONSTRAINT reminderID_PK PRIMARY KEY(reminderID),
   CONSTRAINT appointmentID_FK FOREIGN KEY(appointmentID) REFERENCES
APPOINTMENT(appointmentID),
   CONSTRAINT patientID_FK_R FOREIGN KEY(patientID) REFERENCES PATIENT(patientID),
   CONSTRAINT doctorID_FK_R FOREIGN KEY(doctorID) REFERENCES DOCTOR(doctorID)

);


CREATE TABLE MEDICAL_RECORDS(
   recordID VARCHAR2(15),
   patientID VARCHAR2(15),
   doctorID VARCHAR2(15),
   patientHistory VARCHAR2(1000),
   CONSTRAINT recordID_PK PRIMARY KEY(recordID),
   CONSTRAINT patientID_FK_M FOREIGN KEY(patientID) REFERENCES PATIENT(patientID),
   CONSTRAINT doctorID_FK_M FOREIGN KEY(doctorID) REFERENCES DOCTOR(doctorID)

);


CREATE TABLE FEEDBACK(
   feedbackID VARCHAR2(15),
   patientID VARCHAR2(15),
   doctorID VARCHAR2(15),
   feedbackFormPatient CLOB, -- Assuming feedbackForm can contain a large amount of text
   feedbackFormDoctor CLOB,
   dateSubmittedPatient DATE,
   dateSubmittedDoctor DATE,
   CONSTRAINT feedbackID_PK PRIMARY KEY(feedbackID),
   CONSTRAINT patientID_FK_F FOREIGN KEY(patientID) REFERENCES PATIENT(patientID),
   CONSTRAINT doctorID_FK_F FOREIGN KEY(doctorID) REFERENCES DOCTOR(doctorID)
);
```

6.2 Data Manipulation Language

```
INSERT INTO USER_def
VALUES ('N001', 'John', 'Doe', '123456789', 'Male');

INSERT INTO USER_def
VALUES ('N002', 'Jane', 'Smith', '987654321', 'Female');

INSERT INTO USER_def
VALUES ('N003', 'Bob', 'Johnson', '555666777', 'Male');

INSERT INTO USER_def
VALUES ('N004', 'Alice', 'Williams', '111222333', 'Female');

INSERT INTO USER_def
VALUES ('N005', 'Charlie', 'Brown', '444555666', 'Male');

INSERT INTO USER_def
VALUES ('N006', 'Eva', 'Davis', '999888777', 'Female');

INSERT INTO USER_def
VALUES ('N007', 'Frank', 'Lee', '777888999', 'Male');

INSERT INTO USER_def
VALUES ('N008', 'Grace', 'Miller', '666555444', 'Female');

INSERT INTO USER_def
VALUES ('N009', 'Harry', 'Clark', '222333444', 'Male');

INSERT INTO USER_def
VALUES ('N010', 'Ivy', 'Jones', '888999000', 'Female');

INSERT INTO USER_def
VALUES ('N011', 'Kevin', 'Moore', '333222111', 'Male');

INSERT INTO USER_def
VALUES ('N012', 'Linda', 'Wilson', '555444333', 'Female');

INSERT INTO USER_def
VALUES ('N013', 'Michael', 'Evans', '111000999', 'Male');

INSERT INTO USER_def
VALUES ('N014', 'Nancy', 'Parker', '777666555', 'Female');
```

```
INSERT INTO USER_def
VALUES ('N015', 'Oscar', 'Taylor', '444333222', 'Male');

INSERT INTO USER_def
VALUES ('N016', 'Paula', 'Roberts', '222111000', 'Female');

INSERT INTO USER_def
VALUES ('N017', 'Quincy', 'Woods', '888777666', 'Male');

INSERT INTO USER_def
VALUES ('N018', 'Rachel', 'Garcia', '666555444', 'Female');

INSERT INTO USER_def
VALUES ('N019', 'Sam', 'Fisher', '333222111', 'Male');

INSERT INTO USER_def
VALUES ('N020', 'Tina', 'Lopez', '999888777', 'Female');

INSERT INTO USER_def
VALUES ('N021', 'Lina', 'Layla', '111888777', 'Female');

INSERT INTO USER_def
VALUES ('N022', 'Paul', 'Pozz', '999555777', 'Male');

INSERT INTO USER_def
VALUES ('N023', 'Sam', 'Ropez', '666777777', 'Male');

INSERT INTO USER_def
VALUES ('N024', 'Emily', 'Joyce', '333888222', 'Female');

INSERT INTO USER_def
VALUES ('N025', 'Camily', 'Jone', '222555111', 'Female');
```

```
INSERT INTO DOCTOR
VALUES('N001','DR001','john@gmail.com','Diagnostic');

INSERT INTO DOCTOR
VALUES('N002','DR002','jane@gmail.com','Physiotherapy');

INSERT INTO DOCTOR
VALUES('N003','DR003','bob@gmail.com','Dental');

INSERT INTO DOCTOR
VALUES('N004','DR004','alice@gmail.com','Public Health');

INSERT INTO DOCTOR
VALUES('N005','DR005','charlie@gmail.com','Mental Health');



INSERT INTO PATIENT
VALUES('N006','PT001', TO_DATE('1990-01-01', 'YYYY-MM-DD'), '123 Main St');

INSERT INTO PATIENT
VALUES('N007','PT002', TO_DATE('1990-11-09', 'YYYY-MM-DD'), '111 Main St');

INSERT INTO PATIENT
VALUES('N008','PT003', TO_DATE('1999-06-23', 'YYYY-MM-DD'), '222 Main St');

INSERT INTO PATIENT
VALUES('N009','PT004', TO_DATE('2003-12-13', 'YYYY-MM-DD'), '333 Main St');

INSERT INTO PATIENT
VALUES('N010','PT005', TO_DATE('2000-07-08', 'YYYY-MM-DD'), '444 Main St');

INSERT INTO PATIENT
VALUES('N011','PT006', TO_DATE('1999-07-08', 'YYYY-MM-DD'), '555 Main St');

INSERT INTO PATIENT
VALUES('N012','PT007', TO_DATE('2001-07-08', 'YYYY-MM-DD'), '666 Main St');

INSERT INTO PATIENT
VALUES('N013','PT008', TO_DATE('2003-07-08', 'YYYY-MM-DD'), '777 Main St');

INSERT INTO PATIENT
```

VALUES('N014','PT009', TO_DATE('1996-07-08', 'YYYY-MM-DD'), '888 Main St');

INSERT INTO PATIENT
VALUES('N015','PT010', TO_DATE('2004-07-08', 'YYYY-MM-DD'), '999 Main St');

INSERT INTO PATIENT
VALUES('N016','PT011', TO_DATE('2005-07-08', 'YYYY-MM-DD'), '985 Main St');

INSERT INTO PATIENT
VALUES('N017','PT012', TO_DATE('1998-07-08', 'YYYY-MM-DD'), '753 Main St');

INSERT INTO PATIENT
VALUES('N018','PT013', TO_DATE('2000-07-08', 'YYYY-MM-DD'), '456 Main St');

INSERT INTO PATIENT
VALUES('N019','PT014', TO_DATE('1999-07-08', 'YYYY-MM-DD'), '213 Main St');

INSERT INTO PATIENT
VALUES('N020','PT015', TO_DATE('2001-07-08', 'YYYY-MM-DD'), '154 Main St');

INSERT INTO PATIENT
VALUES('N021','PT016', TO_DATE('2003-02-09', 'YYYY-MM-DD'), '184 Main St');

INSERT INTO PATIENT
VALUES('N022','PT017', TO_DATE('2002-04-06', 'YYYY-MM-DD'), '354 Main St');

INSERT INTO PATIENT
VALUES('N023','PT018', TO_DATE('2001-05-30', 'YYYY-MM-DD'), '254 Main St');

INSERT INTO PATIENT
VALUES('N024','PT019', TO_DATE('2001-06-22', 'YYYY-MM-DD'), '654 Main St');

INSERT INTO PATIENT
VALUES('N025','PT020', TO_DATE('2004-07-11', 'YYYY-MM-DD'), '934 Main St');

```
INSERT INTO APPOINTMENT
VALUES('AP001','PT001','DR001', SYSTIMESTAMP, SYSDATE, 'General Health Checkup', 'Yes',
'Routine');

INSERT INTO APPOINTMENT
VALUES ('AP002', 'PT002','DR001', SYSTIMESTAMP, SYSDATE, 'Blood Test', 'No', 'Regular');

INSERT INTO APPOINTMENT
VALUES ('AP003', 'PT003','DR001', SYSTIMESTAMP, SYSDATE, 'MRI Scan', 'Yes', 'Urgent');

INSERT INTO APPOINTMENT
VALUES ('AP004', 'PT004','DR002', SYSTIMESTAMP, SYSDATE, 'Physical Therapy Session', 'Yes',
'Routine');

INSERT INTO APPOINTMENT
VALUES ('AP005', 'PT005','DR002', SYSTIMESTAMP, SYSDATE, 'Sports Injury Rehabilitation', 'No',
'Regular');

INSERT INTO APPOINTMENT
VALUES ('AP006', 'PT006','DR002', SYSTIMESTAMP, SYSDATE, 'Joint Mobilization Session', 'Yes',
'Urgent');

INSERT INTO APPOINTMENT
VALUES ('AP007', 'PT007','DR003', SYSTIMESTAMP, SYSDATE, 'Tooth Extraction', 'Yes', 'Routine');

INSERT INTO APPOINTMENT
VALUES ('AP008', 'PT008','DR003', SYSTIMESTAMP, SYSDATE, 'Teeth Cleaning and Scaling', 'No',
'Regular');

INSERT INTO APPOINTMENT
VALUES ('AP009', 'PT009','DR003', SYSTIMESTAMP, SYSDATE, 'Dental Checkup', 'Yes', 'Urgent');

INSERT INTO APPOINTMENT
VALUES ('AP010', 'PT010','DR004', SYSTIMESTAMP, SYSDATE, 'Vaccination', 'Yes', 'Routine');

INSERT INTO APPOINTMENT
VALUES ('AP011', 'PT011','DR004', SYSTIMESTAMP, SYSDATE, 'Preventive Health Counseling', 'No',
'Urgent');

INSERT INTO APPOINTMENT
VALUES ('AP012', 'PT012','DR004', SYSTIMESTAMP, SYSDATE, 'Chronic Disease Management
Consultation', 'Yes', 'Regular');
```

```
INSERT INTO APPOINTMENT
VALUES ('AP013', 'PT013','DR005',SYSTIMESTAMP, SYSDATE, 'Individual Counseling Session',
'Yes', 'Urgent');


INSERT INTO APPOINTMENT
VALUES ('AP014', 'PT014','DR005', SYSTIMESTAMP, SYSDATE, 'Individual Counseling Session',
'No', 'Regular');

INSERT INTO APPOINTMENT
VALUES ('AP015', 'PT015','DR005', SYSTIMESTAMP, SYSDATE, 'Allergy Testing', 'Yes', 'Routine');

INSERT INTO APPOINTMENT
VALUES ('AP016', 'PT016','DR005', SYSTIMESTAMP, SYSDATE, 'Stress Management Workshop',
'Yes', 'Urgent');

INSERT INTO APPOINTMENT
VALUES ('AP017', 'PT017','DR005', SYSTIMESTAMP, SYSDATE, 'Stress Management Workshop',
'Yes', 'Routine');

INSERT INTO APPOINTMENT
VALUES ('AP018', 'PT018','DR005', SYSTIMESTAMP, SYSDATE, 'Stress Management Workshop',
'Yes', 'Urgent');

INSERT INTO APPOINTMENT
VALUES ('AP019', 'PT019','DR005', SYSTIMESTAMP, SYSDATE, 'Stress Management Workshop',
'No', 'Regular');

INSERT INTO APPOINTMENT
VALUES ('AP020', 'PT020','DR005', SYSTIMESTAMP, SYSDATE, 'Group Therapy Session', 'Yes',
'Routine');
```

-- SYSDATE - 2 is used to simulate a reminder date that is 2 days before the current date and time (SYSDATE).--

```
INSERT INTO REMINDER
VALUES('REM001', 'PT001', 'DR001', 'AP001', SYSDATE - 1);

INSERT INTO REMINDER
VALUES('REM002', 'PT002', 'DR001', 'AP002', SYSDATE - 1);

INSERT INTO REMINDER
VALUES('REM003', 'PT003', 'DR001', 'AP003', SYSDATE - 2);

INSERT INTO REMINDER
VALUES('REM004', 'PT004', 'DR002', 'AP004', SYSDATE - 2);

INSERT INTO REMINDER
VALUES('REM005', 'PT005', 'DR002', 'AP005', SYSDATE - 1);

INSERT INTO REMINDER
VALUES('REM006', 'PT006', 'DR002', 'AP006', SYSDATE - 2);

INSERT INTO REMINDER
VALUES('REM007', 'PT007', 'DR003', 'AP007', SYSDATE - 2);

INSERT INTO REMINDER
VALUES('REM008', 'PT008', 'DR003', 'AP008', SYSDATE - 1);

INSERT INTO REMINDER
VALUES('REM009', 'PT009', 'DR003', 'AP009', SYSDATE - 1);

INSERT INTO REMINDER
VALUES('REM010', 'PT010', 'DR004', 'AP010', SYSDATE - 3);

INSERT INTO REMINDER
VALUES('REM011', 'PT011', 'DR004', 'AP011', SYSDATE - 1);

INSERT INTO REMINDER
VALUES('REM012', 'PT012', 'DR004', 'AP012', SYSDATE - 2);

INSERT INTO REMINDER
VALUES('REM013', 'PT013', 'DR005', 'AP013', SYSDATE - 3);

INSERT INTO REMINDER
VALUES('REM014', 'PT014', 'DR005', 'AP014', SYSDATE - 1);
```

```
INSERT INTO REMINDER
VALUES('REM015', 'PT015', 'DR005', 'AP015', SYSDATE - 2);

INSERT INTO REMINDER
VALUES('REM016', 'PT016', 'DR005', 'AP016', SYSDATE - 3);

INSERT INTO REMINDER
VALUES('REM017', 'PT017', 'DR005', 'AP017', SYSDATE - 1);

INSERT INTO REMINDER
VALUES('REM018', 'PT018', 'DR005', 'AP018', SYSDATE - 3);

INSERT INTO REMINDER
VALUES('REM019', 'PT019', 'DR005', 'AP019', SYSDATE - 1);

INSERT INTO REMINDER
VALUES('REM020', 'PT020', 'DR005', 'AP020', SYSDATE - 2);
```

```
INSERT INTO MEDICAL_RECORDS
VALUES('REC001', 'PT001', 'DR001', 'Patient history for PT001 under the care of DR001.');

INSERT INTO MEDICAL_RECORDS
VALUES('REC002', 'PT002', 'DR001', 'Patient history for PT002 under the care of DR001.');

INSERT INTO MEDICAL_RECORDS
VALUES('REC003', 'PT003', 'DR001', 'Patient history for PT003 under the care of DR001.');

INSERT INTO MEDICAL_RECORDS
VALUES('REC004', 'PT004', 'DR002', 'Patient history for PT004 under the care of DR002.');

INSERT INTO MEDICAL_RECORDS
VALUES('REC005', 'PT005', 'DR002', 'Patient history for PT005 under the care of DR002.');

INSERT INTO MEDICAL_RECORDS
VALUES('REC006', 'PT006', 'DR002', 'Patient history for PT006 under the care of DR002.');

INSERT INTO MEDICAL_RECORDS
VALUES('REC007', 'PT007', 'DR003', 'Patient history for PT007 under the care of DR003.');
```

```
INSERT INTO MEDICAL_RECORDS
VALUES('REC008', 'PT008', 'DR003', 'Patient history for PT008 under the care of DR003.');

INSERT INTO MEDICAL_RECORDS
VALUES('REC009', 'PT009', 'DR003', 'Patient history for PT009 under the care of DR003.');

INSERT INTO MEDICAL_RECORDS
VALUES('REC010', 'PT010', 'DR004', 'Patient history for PT010 under the care of DR004.');

INSERT INTO MEDICAL_RECORDS
VALUES('REC011', 'PT011', 'DR004', 'Patient history for PT011 under the care of DR004.');

INSERT INTO MEDICAL_RECORDS
VALUES('REC012', 'PT012', 'DR004', 'Patient history for PT012 under the care of DR004.');

INSERT INTO MEDICAL_RECORDS
VALUES('REC013', 'PT013', 'DR005', 'Patient history for PT013 under the care of DR005.');

INSERT INTO MEDICAL_RECORDS
VALUES('REC014', 'PT014', 'DR005', 'Patient history for PT014 under the care of DR005.');

INSERT INTO MEDICAL_RECORDS
VALUES('REC015', 'PT015', 'DR005', 'Patient history for PT015 under the care of DR005.');

INSERT INTO MEDICAL_RECORDS
VALUES('REC016', 'PT016', 'DR005', 'Patient history for PT015 under the care of DR005.');

INSERT INTO MEDICAL_RECORDS
VALUES('REC017', 'PT017', 'DR005', 'Patient history for PT015 under the care of DR005.');

INSERT INTO MEDICAL_RECORDS
VALUES('REC018', 'PT018', 'DR005', 'Patient history for PT015 under the care of DR005.');

INSERT INTO MEDICAL_RECORDS
VALUES('REC019', 'PT019', 'DR005', 'Patient history for PT015 under the care of DR005.');

INSERT INTO MEDICAL_RECORDS
VALUES('REC020', 'PT020', 'DR005', 'Patient history for PT015 under the care of DR005.');
```

```
INSERT INTO FEEDBACK
VALUES('FDB001', 'PT001', 'DR001', 'Patient feedback for DR001. Satisfied with the treatment.', 'Doctor feedback for the appointment system: The system is user-friendly, and the appointment scheduling process is efficient. However, it would be beneficial to enhance the system with additional features for better patient management and communication.', SYSDATE, SYSDATE);

INSERT INTO FEEDBACK
VALUES('FDB002', 'PT002', 'DR001', 'Patient feedback for DR001. Good communication and care.', 'Doctor feedback for the appointment system: The system performance is satisfactory. Consider incorporating a feature to send automated reminders to patients before their appointments.', SYSDATE, SYSDATE);

INSERT INTO FEEDBACK
VALUES('FDB003', 'PT003', 'DR001', 'Positive feedback from PT003 for DR001. Excellent service.', 'Doctor feedback for the appointment system: Overall, the system meets the needs. However, improvements in the user interface and navigation could enhance the user experience.', SYSDATE, SYSDATE);

INSERT INTO FEEDBACK
VALUES('FDB004', 'PT004', 'DR002', NULL, 'Doctor feedback for PT004. Patient responded well to the treatment. Regarding the appointment system, consider adding a feature to allow doctors to view their weekly or monthly schedules at a glance.', NULL, SYSDATE);

INSERT INTO FEEDBACK
VALUES('FDB005', 'PT005', 'DR002', NULL, 'Doctor feedback for PT005. Doctor feedback for PT005. Patient recovery is progressing. Suggestion for the appointment system: An option to customize appointment time slots based on doctor preferences could improve scheduling efficiency.', NULL, SYSDATE);

INSERT INTO FEEDBACK
VALUES('FDB006', 'PT006', 'DR002', NULL, 'Doctor feedback for PT006. Further follow-up appointments recommended. Regarding the appointment system, consider implementing a feature for doctors to easily reschedule appointments if needed.', NULL, SYSDATE);

INSERT INTO FEEDBACK
VALUES('FDB007', 'PT007', 'DR003', 'Patient feedback for DR003. Pleased with the consultation.', 'Doctor feedback for the appointment system: The system is functional, but there is room for improvement in terms of data retrieval speed. Enhancing system performance would contribute to a smoother workflow.', SYSDATE, SYSDATE);

INSERT INTO FEEDBACK
VALUES('FDB008', 'PT008', 'DR003', 'Patient feedback for DR003. Timely and effective treatment.', 'Doctor feedback for the appointment system: The overall system is good. Consider implementing a
```

feature for doctors to input brief notes or reminders for each appointment to improve patient care.', SYSDATE, SYSDATE);

INSERT INTO FEEDBACK
VALUES('FDB009', 'PT009', 'DR003', 'Positive feedback from PT009 for DR003. Professionalism appreciated.', 'Doctor feedback for the appointment system: The system interface is user-friendly. Consider integrating a feature for doctors to easily access patient medical history during appointments.', SYSDATE, SYSDATE);

INSERT INTO FEEDBACK
VALUES('FDB010', 'PT010', 'DR004', NULL, 'Doctor feedback for PT010. Patient responded positively to the prescribed medication. Suggestion for the appointment system: An option for doctors to set their availability preferences for appointments could be beneficial.', NULL, SYSDATE);

INSERT INTO FEEDBACK
VALUES('FDB011', 'PT011', 'DR004', NULL, 'Doctor feedback for PT011. Follow-up tests recommended for accurate diagnosis. Regarding the appointment system, consider adding a feature for doctors to block off time slots for administrative tasks or breaks.', NULL, SYSDATE);

INSERT INTO FEEDBACK
VALUES('FDB012', 'PT012', 'DR004', NULL, 'Doctor feedback for PT012. Condition of the patient requires close monitoring. Suggestion for the appointment system: Implement a feature for doctors to easily communicate with patients through the system, such as sending messages or updates.', NULL, SYSDATE);

INSERT INTO FEEDBACK
VALUES('FDB013', 'PT013', 'DR005', 'Patient feedback for DR005. Grateful for the attentive care.', 'Doctor feedback for the appointment system: The system functionalities are good. Consider enhancing the notification system for doctors to receive timely updates on appointment changes or cancellations.', SYSDATE, SYSDATE);

INSERT INTO FEEDBACK
VALUES('FDB014', 'PT014', 'DR005', 'Patient feedback for DR005. Satisfied with the overall experience.', 'Doctor feedback for the appointment system: Overall, the system works well. A feature for doctors to easily view and manage patient feedback within the system would be useful.', SYSDATE, SYSDATE);

INSERT INTO FEEDBACK
VALUES('FDB015', 'PT015', 'DR005', 'Positive feedback from PT015 for DR005. Advice gotten from the doctor is proved helpful.', 'Doctor feedback for the appointment system: The system is efficient. Consider adding an analytics dashboard for doctors to track their appointment statistics and patient outcomes.', SYSDATE, SYSDATE);

INSERT INTO FEEDBACK

VALUES('FDB016', 'PT016', 'DR005', 'Patient feedback for DR005. Satisfied with the overall experience.', 'Doctor feedback for the appointment system: Overall, the system works well. A feature for doctors to easily view and manage patient feedback within the system would be useful.', SYSDATE, SYSDATE);

INSERT INTO FEEDBACK
VALUES('FDB017', 'PT017', 'DR005', 'Positive feedback from PT017 for DR005. Professionalism appreciated.', 'Doctor feedback for the appointment system: The system interface is user-friendly. Consider integrating a feature for doctors to easily access patient medical history during appointments.', SYSDATE, SYSDATE);

INSERT INTO FEEDBACK
VALUES('FDB018', 'PT018', 'DR005', NULL, 'Doctor feedback for PT018. Patient responded well to the treatment. Regarding the appointment system, consider adding a feature to allow doctors to view their weekly or monthly schedules at a glance.', NULL, SYSDATE);

INSERT INTO FEEDBACK
VALUES('FDB019', 'PT019', 'DR005', NULL, 'Doctor feedback for PT019. Patient recovery is progressing. Suggestion for the appointment system: An option to customize appointment time slots based on doctor preferences could improve scheduling efficiency.', NULL, SYSDATE);

INSERT INTO FEEDBACK
VALUES('FDB020', 'PT020', 'DR005', NULL, 'Doctor feedback for PT020. Further follow-up appointments recommended. Regarding the appointment system, consider implementing a feature for doctors to easily reschedule appointments if needed.', NULL, SYSDATE);

6.3 Test Query

1. View Appointment Session history

```
SELECT appointmentID as "APPOINTMENT ID",
   patientID as "PATIENT ID",
   doctorID as "DOCTOR ID",
   appointmentTime as "APPOINTMENT TIME",
   appointmentDate as "APPOINTMENT DATE",
   appointmentType as "APPOINTMENT TYPE",
   appointmentReminder as "APPOINTMENT REMINDER",
   appointmentSchedule as "APPOINTMENT SCHEDULE"
FROM APPOINTMENT
ORDER BY appointmentDate
```

APPOINTMENT ID	PATIENT ID	DOCTOR ID	APPOINTMENT TIME	APPOINTMENT DATE	APPOINTMENT TYPE	APPOINTMENT REMINDER	APPOINTMENT SCHEDULE
AP001	PT001	DR001	14-JAN-24 04.36.05.073126 PM	14-JAN-24	General Health Checkup	Yes	Routine
AP020	PT020	DR005	14-JAN-24 04.36.05.143346 PM	14-JAN-24	Group Therapy Session	Yes	Routine
AP003	PT003	DR001	14-JAN-24 04.36.05.082819 PM	14-JAN-24	MRI Scan	Yes	Urgent
AP004	PT004	DR002	14-JAN-24 04.36.05.085672 PM	14-JAN-24	Physical Therapy Session	Yes	Routine
AP005	PT005	DR002	14-JAN-24 04.36.05.089049 PM	14-JAN-24	Sports Injury Rehabilitation	No	Regular
AP006	PT006	DR002	14-JAN-24 04.36.05.093557 PM	14-JAN-24	Joint Mobilization Session	Yes	Urgent
AP007	PT007	DR003	14-JAN-24 04.36.05.096444 PM	14-JAN-24	Tooth Extraction	Yes	Routine
AP008	PT008	DR003	14-JAN-24 04.36.05.101711 PM	14-JAN-24	Teeth Cleaning and Scaling	No	Regular
AP009	PT009	DR003	14-JAN-24 04.36.05.104530 PM	14-JAN-24	Dental Checkup	Yes	Urgent
AP010	PT010	DR004	14-JAN-24 04.36.05.107785 PM	14-JAN-24	Vaccination	Yes	Routine
AP011	PT011	DR004	14-JAN-24 04.36.05.111975 PM	14-JAN-24	Preventive Health Counseling	No	Urgent
AP012	PT012	DR004	14-JAN-24 04.36.05.117442 PM	14-JAN-24	Chronic Disease Management Consultation	Yes	Regular
AP013	PT013	DR005	14-JAN-24 04.36.05.120426 PM	14-JAN-24	Individual Counseling Session	Yes	Urgent
AP014	PT014	DR005	14-JAN-24 04.36.05.126325 PM	14-JAN-24	Individual Counseling Session	No	Regular
AP015	PT015	DR005	14-JAN-24 04.36.05.129111 PM	14-JAN-24	Allergy Testing	Yes	Routine
AP016	PT016	DR005	14-JAN-24 04.36.05.132068 PM	14-JAN-24	Stress Management Workshop	Yes	Urgent
AP017	PT017	DR005	14-JAN-24 04.36.05.134946 PM	14-JAN-24	Stress Management Workshop	Yes	Routine
AP018	PT018	DR005	14-JAN-24 04.36.05.137739 PM	14-JAN-24	Stress Management Workshop	Yes	Urgent
AP019	PT019	DR005	14-JAN-24 04.36.05.140537 PM	14-JAN-24	Stress Management Workshop	No	Regular
AP002	PT002	DR001	14-JAN-24 04.36.05.079585 PM	14-JAN-24	Blood Test	No	Regular

2. Update appointment

```
UPDATE APPOINTMENT
SET appointmentDate = TO_DATE('01/15/2024', 'MM/DD/YYYY'),
        appointmentTime = TO_TIMESTAMP('12:00:00', 'HH24:MI:SS')
WHERE appointmentID = 'AP001';
```

```
SELECT *
FROM APPOINTMENT
```

APPOINTMENTID	PATIENTID	DOCTORID	APPOINTMENTTIME	APPOINTMENTDATE	APPOINTMENTTYPE	APPOINTMENTREMINDER	APPOINTME
AP001	PT001	DR001	01-JAN-24 12.00.00.000000 PM	15-JAN-24	General Health Checkup	Yes	Routine
AP002	PT002	DR001	14-JAN-24 04.36.05.079585 PM	14-JAN-24	Blood Test	No	Regular
AP003	PT003	DR001	14-JAN-24 04.36.05.082819 PM	14-JAN-24	MRI Scan	Yes	Urgent

3. Delete appointment

```
DELETE FROM REMINDER
WHERE appointmentID = 'AP001';

DELETE FROM APPOINTMENT
WHERE appointmentID = 'AP001';

SELECT  *
FROM APPOINTMENT
ORDER BY appointmentID
```

APPOINTMENTID	PATIENTID	DOCTORID	APPOINTMENTTIME	APPOINTMENTDATE	APPOINTMENTTYPE	APPOINTMENTREMINDER	APPOINTME
AP002	PT002	DR001	14-JAN-24 04.36.05.079585 PM	14-JAN-24	Blood Test	No	Regular
AP003	PT003	DR001	14-JAN-24 04.36.05.082819 PM	14-JAN-24	MRI Scan	Yes	Urgent
AP004	PT004	DR002	14-JAN-24 04.36.05.085672 PM	14-JAN-24	Physical Therapy Session	Yes	Routine

4. View Patient Information

```
SELECT
   p.patientID AS "PATIENT ID",
   p.userNo AS "USER ID",
   u.firstName||' '||u.lastName AS "NAME",
   u.gender AS "GENDER",
   u.phoneNo AS "CONTACT NO",
   p.dateOfBirth AS "BIRTH DATE",
   p.address AS "ADDRESS"
FROM PATIENT p
JOIN USER_def u
ON p.userNo = u.userNo
ORDER BY p.patientID
```

PATIENT ID	USER ID	NAME	GENDER	CONTACT NO	BIRTH DATE	ADDRESS
PT001	N006	Eva Davis	Female	999888777	01/01/1990	123 Main St
PT002	N007	Frank Lee	Male	777888999	11/09/1990	111 Main St
PT003	N008	Grace Miller	Female	666555444	06/23/1999	222 Main St
PT004	N009	Harry Clark	Male	222333444	12/13/2003	333 Main St
PT005	N010	Ivy Jones	Female	888999000	07/08/2000	444 Main St
PT006	N011	Kevin Moore	Male	333222111	07/08/1999	555 Main St
PT007	N012	Linda Wilson	Female	555444333	07/08/2001	666 Main St
PT008	N013	Michael Evans	Male	111000999	07/08/2003	777 Main St

PT009	N014	Nancy Parker	Female	777666555	07/08/1996	888 Main St
PT010	N015	Oscar Taylor	Male	444333222	07/08/2004	999 Main St
PT011	N016	Paula Roberts	Female	222111000	07/08/2005	985 Main St
PT012	N017	Quincy Woods	Male	888777666	07/08/1998	753 Main St
PT013	N018	Rachel Garcia	Female	666555444	07/08/2000	456 Main St
PT014	N019	Sam Fisher	Male	333222111	07/08/1999	213 Main St
PT015	N020	Tina Lopez	Female	999888777	07/08/2001	154 Main St
PT016	N021	Lina Layla	Female	111888777	02/09/2003	184 Main St
PT017	N022	Paul Pozz	Male	999555777	04/06/2002	354 Main St
PT018	N023	Sam Ropez	Male	666777777	05/30/2001	254 Main St
PT019	N024	Emily Joyce	Female	333888222	06/22/2001	654 Main St
PT020	N025	Camily Jone	Female	222555111	07/11/2004	934 Main St

5. View list of doctor

```
SELECT d.doctorID as "DOCTOR ID",
   u.firstName ||' '|| u.lastName as "DOCTOR NAME",
   d.emailAddress as "EMAIL ADDRESS",
   d.department as "DEPARTMENT"
FROM DOCTOR d
JOIN USER_def u
ON d.userNo = u.userNo
ORDER BY doctorID
```

DOCTOR ID	DOCTOR NAME	EMAIL ADDRESS	DEPARTMENT
DR001	John Doe	john@gmail.com	Diagnostic
DR002	Jane Smith	jane@gmail.com	Physiotherapy
DR003	Bob Johnson	bob@gmail.com	Dental
DR004	Alice Williams	alice@gmail.com	Public Health
DR005	Charlie Brown	charlie@gmail.com	Mental Health

6. View patient medical record

```
SELECT m.recordID as "RECORD ID",
   m.patientID as "PATIENT ID",
   u.firstName ||' '|| u.lastName as "PATIENT NAME",
   m.patientHistory AS "MEDICAL HISTORY "
FROM MEDICAL_RECORDS m
JOIN PATIENT p
ON m.patientID = p.patientID
```

JOIN USER_def u
ON p.userNo = u.userNo
ORDER BY recordID

RECORD ID	PATIENT ID	PATIENT NAME	MEDICAL HISTORY
REC001	PT001	Eva Davis	Patient history for PT001 under the care of DR001.
REC002	PT002	Frank Lee	Patient history for PT002 under the care of DR001.
REC003	PT003	Grace Miller	Patient history for PT003 under the care of DR001.
REC004	PT004	Harry Clark	Patient history for PT004 under the care of DR002.
REC005	PT005	Ivy Jones	Patient history for PT005 under the care of DR002.
REC006	PT006	Kevin Moore	Patient history for PT006 under the care of DR002.
REC007	PT007	Linda Wilson	Patient history for PT007 under the care of DR003.
REC008	PT008	Michael Evans	Patient history for PT008 under the care of DR003.
REC009	PT009	Nancy Parker	Patient history for PT009 under the care of DR003.
REC010	PT010	Oscar Taylor	Patient history for PT010 under the care of DR004.
REC011	PT011	Paula Roberts	Patient history for PT011 under the care of DR004.
REC012	PT012	Quincy Woods	Patient history for PT012 under the care of DR004.

REC013	PT013	Rachel Garcia	Patient history for PT013 under the care of DR005.
REC014	PT014	Sam Fisher	Patient history for PT014 under the care of DR005.
REC015	PT015	Tina Lopez	Patient history for PT015 under the care of DR005.
REC016	PT016	Lina Layla	Patient history for PT015 under the care of DR005.
REC017	PT017	Paul Pozz	Patient history for PT015 under the care of DR005.
REC018	PT018	Sam Ropez	Patient history for PT015 under the care of DR005.
REC019	PT019	Emily Joyce	Patient history for PT015 under the care of DR005.
REC020	PT020	Camily Jone	Patient history for PT015 under the care of DR005.

7. Update medical record

```
UPDATE MEDICAL_RECORDS
SET patientHistory = 'Patient history for PT001 under the care of DR002'
WHERE recordID = 'REC001';

SELECT *
FROM MEDICAL_RECORDS
ORDER BY recordID;
```

RECORDID	PATIENTID	DOCTORID	PATIENTHISTORY
REC001	PT001	DR001	Patient history for PT001 under the care of DR002
REC002	PT002	DR001	Patient history for PT002 under the care of DR001.
REC003	PT003	DR001	Patient history for PT003 under the care of DR001.
REC004	PT004	DR002	Patient history for PT004 under the care of DR002.
REC005	PT005	DR002	Patient history for PT005 under the care of DR002.

8. Delete medical record

```
DELETE FROM MEDICAL_RECORDS
WHERE recordID = 'REC001';

SELECT  *
FROM MEDICAL_RECORDS
ORDER BY recordID;
```

RECORDID	PATIENTID	DOCTORID	PATIENTHISTORY
REC002	PT002	DR001	Patient history for PT002 under the care of DR001.
REC003	PT003	DR001	Patient history for PT003 under the care of DR001.
REC004	PT004	DR002	Patient history for PT004 under the care of DR002.
REC005	PT005	DR002	Patient history for PT005 under the care of DR002.
REC006	PT006	DR002	Patient history for PT006 under the care of DR002.

9. View feedback from patient

```
SELECT patientID as "PATIENT ID",
   dateSubmittedPatient as "DATE",
   feedbackFormPatient as "FEEDBACK"
FROM FEEDBACK
WHERE feedbackFormPatient IS NOT NULL
ORDER BY "DATE";
```

PATIENT ID	DATE	FEEDBACK
PT001	01/12/2024	Patient feedback for DR001. Satisfied with the treatment.
PT002	01/12/2024	Patient feedback for DR001. Good communication and care.
PT003	01/12/2024	Positive feedback from PT003 for DR001. Excellent service.
PT007	01/12/2024	Patient feedback for DR003. Pleased with the consultation.
PT008	01/12/2024	Patient feedback for DR003. Timely and effective treatment.
PT009	01/12/2024	Positive feedback from PT009 for DR003. Professionalism appreciated.
PT013	01/12/2024	Patient feedback for DR005. Grateful for the attentive care.
PT014	01/12/2024	Patient feedback for DR005. Satisfied with the overall experience.
PT015	01/12/2024	Positive feedback from PT015 for DR005. Advice gotten from doctor is proved helpful.
PT016	01/12/2024	Patient feedback for DR005. Satisfied with the overall experience.
PT017	01/12/2024	Positive feedback from PT017 for DR005. Professionalism appreciated.

10. View feedback from doctor

```
SELECT doctorID as "DOCTOR ID",
        dateSubmittedDoctor as "DATE",
        feedbackFormDoctor as "FEEDBACK"
FROM FEEDBACK
ORDER BY "DATE";
```

DOCTOR ID	DATE	FEEDBACK
DR001	01/12/2024	Doctor feedback for the appointment system: Overall, the system meets the needs. However, improvements in the user interface and navigation could enhance the user experience.
DR001	01/12/2024	Doctor feedback for the appointment system: The system is user-friendly, and the appointment scheduling process is efficient. However, it would be beneficial to enhance the system with additional features for better patient management and communication.
DR001	01/12/2024	Doctor feedback for the appointment system: The system performance is satisfactory. Consider incorporating a feature to send automated reminders to patients before their appointments.
DR002	01/12/2024	Doctor feedback for PT004. Patient responded well to the treatment. Regarding the appointment system, consider adding a feature to allow doctors to view their weekly or monthly schedules at a glance.
DR002	01/12/2024	Doctor feedback for PT006. Further follow-up appointments recommended. Regarding the appointment system, consider implementing a feature for doctors to easily reschedule appointments if needed.
DR002	01/12/2024	Doctor feedback for PT005. Doctor feedback for PT005. Patient recovery is progressing. Suggestion for the appointment system: An option to customize appointment time slots based on doctor preferences could improve scheduling efficiency.
DR003	01/12/2024	Doctor feedback for the appointment system: The system is functional, but there is room for improvement in terms of data retrieval speed. Enhancing system performance would contribute to a smoother workflow.
DR003	01/12/2024	Doctor feedback for the appointment system: The system interface is user-friendly. Consider integrating a feature for doctors to easily access patient medical history during appointments.
DR003	01/12/2024	Doctor feedback for the appointment system: The overall system is good. Consider implementing a feature for doctors to input brief notes or reminders for each appointment to improve patient care.
DR004	01/12/2024	Doctor feedback for PT011. Follow-up tests recommended for accurate diagnosis. Regarding the appointment system, consider adding a feature for doctors to block off time slots for administrative tasks or breaks.
DR004	01/12/2024	Doctor feedback for PT010. Patient responded positively to the prescribed medication. Suggestion for the appointment system: An option for doctors to set their availability preferences for appointments could be beneficial.

DR004	01/12/2024	Doctor feedback for PT012. Condition of the patient requires close monitoring. Suggestion for the appointment system: Implement a feature for doctors to easily communicate with patients through the system, such as sending messages or updates.
DR005	01/12/2024	Doctor feedback for the appointment system: The system is efficient. Consider adding an analytics dashboard for doctors to track their appointment statistics and patient outcomes.
DR005	01/12/2024	Doctor feedback for the appointment system: Overall, the system works well. A feature for doctors to easily view and manage patient feedback within the system would be useful.
DR005	01/12/2024	Doctor feedback for PT019. Patient recovery is progressing. Suggestion for the appointment system: An option to customize appointment time slots based on doctor preferences could improve scheduling efficiency.
DR005	01/12/2024	Doctor feedback for PT020. Further follow-up appointments recommended. Regarding the appointment system, consider implementing a feature for doctors to easily reschedule appointments if needed.
DR005	01/12/2024	Doctor feedback for the appointment system: The system functionalities are good. Consider enhancing the notification system for doctors to receive timely updates on appointment changes or cancellations.
DR005	01/12/2024	Doctor feedback for PT018. Patient responded well to the treatment. Regarding the appointment system, consider adding a feature to allow doctors to view their weekly or monthly schedules at a glance.
DR005	01/12/2024	Doctor feedback for the appointment system: The system interface is user-friendly. Consider integrating a feature for doctors to easily access patient medical history during appointments.
DR005	01/12/2024	Doctor feedback for the appointment system: Overall, the system works well. A feature for doctors to easily view and manage patient feedback within the system would be useful.

11. Check reminder status

```
SELECT appointmentID as "APPOINTMENT ID",
    patientID as "PATIENT ID",
    appointmentTime as "APPOINTMENT TIME",
    appointmentReminder as "Reminder Sent?"

FROM APPOINTMENT
ORDER BY appointmentReminder;
```

APPOINTMENT ID	PATIENT ID	APPOINTMENT TIME	Reminder Sent?
AP002	PT002	14-JAN-24 04.36.05.079585 PM	No
AP011	PT011	14-JAN-24 04.36.05.111975 PM	No
AP019	PT019	14-JAN-24 04.36.05.140537 PM	No
AP014	PT014	14-JAN-24 04.36.05.126325 PM	No
AP008	PT008	14-JAN-24 04.36.05.101711 PM	No
AP005	PT005	14-JAN-24 04.36.05.089049 PM	No
AP017	PT017	14-JAN-24 04.36.05.134946 PM	Yes
AP009	PT009	14-JAN-24 04.36.05.104530 PM	Yes
AP010	PT010	14-JAN-24 04.36.05.107785 PM	Yes
AP020	PT020	14-JAN-24 04.36.05.143346 PM	Yes
AP012	PT012	14-JAN-24 04.36.05.117442 PM	Yes
AP013	PT013	14-JAN-24 04.36.05.120426 PM	Yes

AP003	PT003	14-JAN-24 04.36.05.082819 PM	Yes
AP015	PT015	14-JAN-24 04.36.05.129111 PM	Yes
AP007	PT007	14-JAN-24 04.36.05.096444 PM	Yes
AP006	PT006	14-JAN-24 04.36.05.093557 PM	Yes
AP016	PT016	14-JAN-24 04.36.05.132068 PM	Yes
AP004	PT004	14-JAN-24 04.36.05.085672 PM	Yes
AP018	PT018	14-JAN-24 04.36.05.137739 PM	Yes

12. Update user information

```
UPDATE USER_def
SET phoneNo = '011123456'
WHERE userNo = 'N001';

SELECT userNo, phoneNo
FROM USER_def
WHERE userNo = 'N001';
```

Before

USERNO	PHONENO
N001	123456789

After

USERNO	PHONENO
N001	011123456

13. View Patient Under Dr004, Alice Williams

```
SELECT
   m.patientID AS "PATIENT ID",
   u.firstName || ' ' || u.lastName AS "PATIENT NAME",
   m.patientHistory AS "Medical History",
   m.doctorID AS "DOCTOR ID",
   a.firstName || ' ' || a.lastName AS "DOCTOR IN-CHARGE"
FROM
   MEDICAL_RECORDS m
JOIN
   PATIENT p ON m.patientID = p.patientID
JOIN
   USER_def u ON p.userNo = u.userNo
LEFT JOIN
   DOCTOR d ON m.doctorID = d.doctorID
```

```
LEFT JOIN
   USER_def a ON d.userNo = a.userNo
WHERE
   a.firstName || ' ' || a.lastName = 'Alice Williams'
```

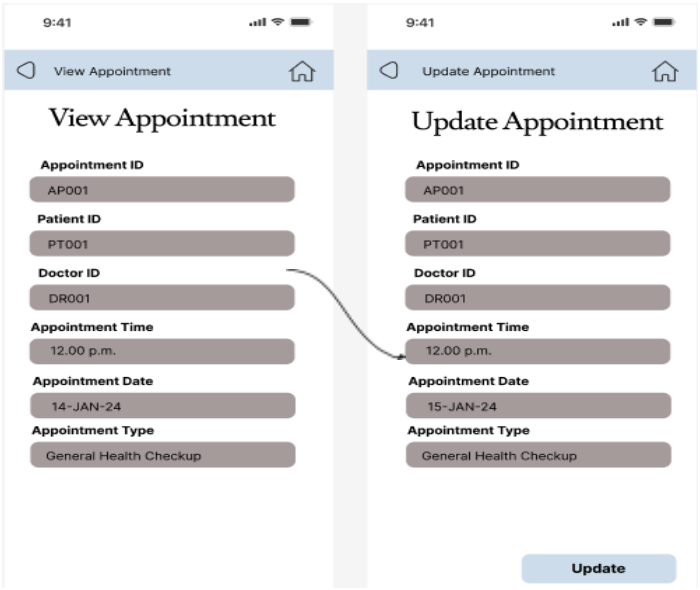
PATIENT ID	PATIENT NAME	Medical History	DOCTOR ID	DOCTOR IN-CHARGE
PT010	Oscar Taylor	Patient history for PT010 under the care of DR004.	DR004	Alice Williams
PT011	Paula Roberts	Patient history for PT011 under the care of DR004.	DR004	Alice Williams
PT012	Quincy Woods	Patient history for PT012 under the care of DR004.	DR004	Alice Williams

7.0 Interface

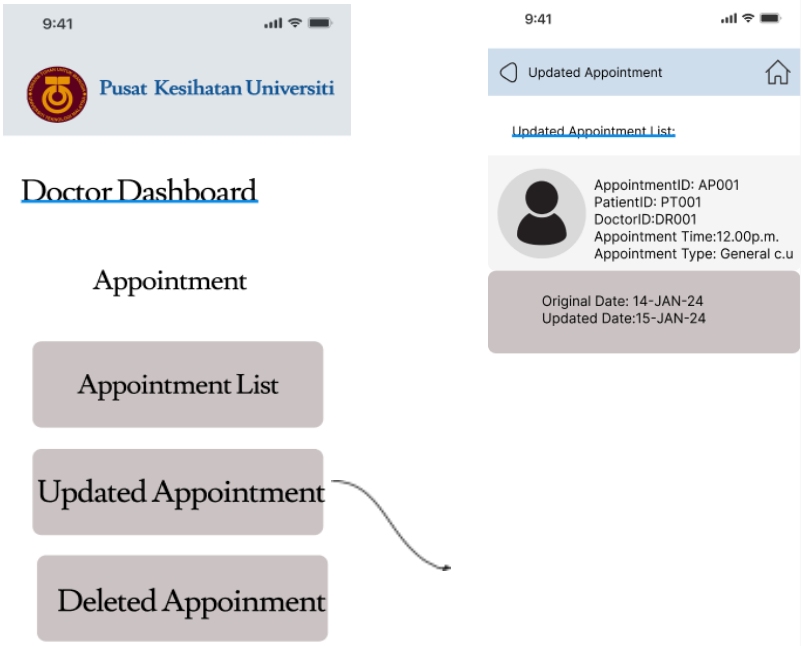
1. From test query

2. Update Appointment

Patient View:



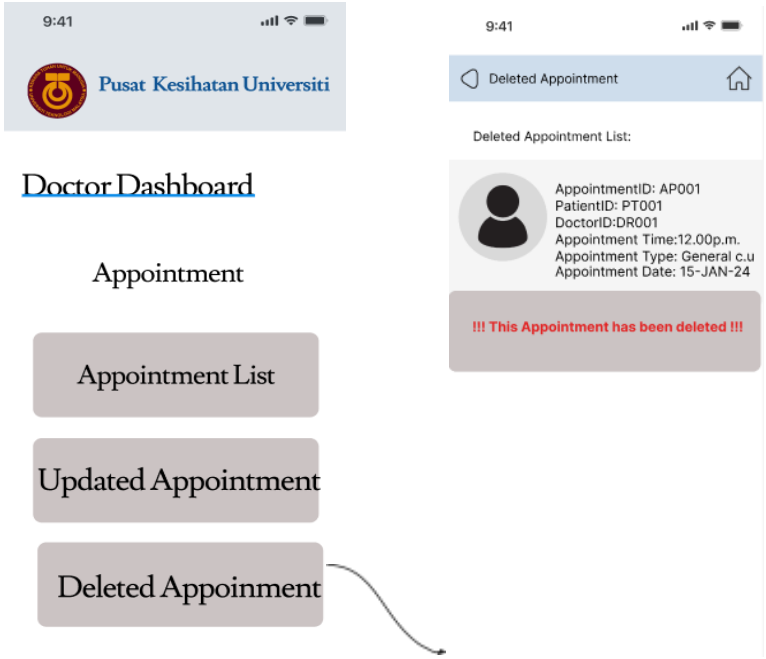
Doctor View:



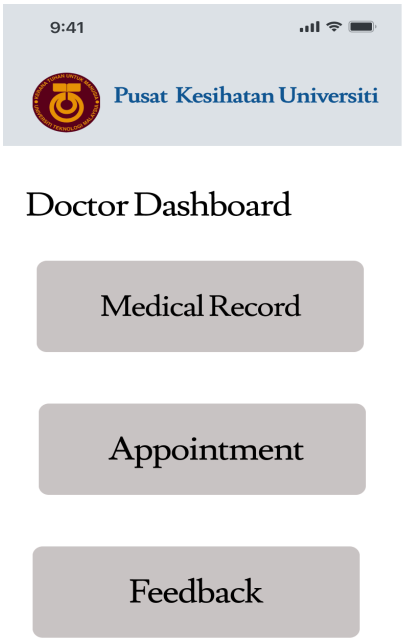
2. From test query
   3. Delete Appointment

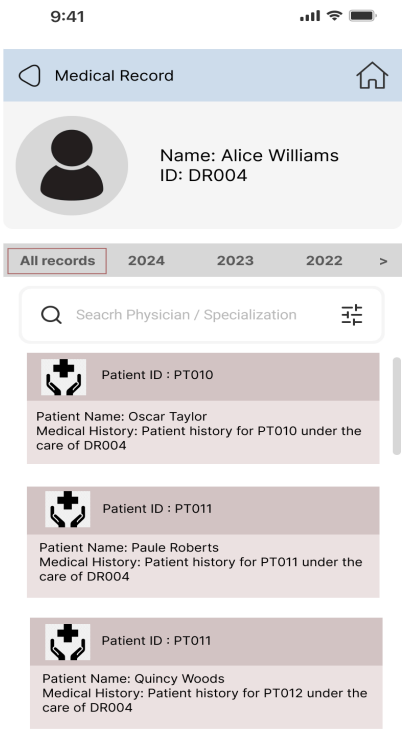
Patient view:

Doctor view:



3. From test query
   13. View Patient Under Dr004, Alice Williams





8.0 Summary

In conclusion, we have focused on converting a conceptual ERD into a logical one and developing relational schemas while maintaining normalisation in Phase 3 of our project for the PKU Appointment System. We have been working on converting the ERD into relational tables that follow BCNF, reduce redundant data, and ensure a strong database structure.

The updated business rules cover the workflows for doctors and patients, including registration, login, dashboard features, appointment scheduling, information access, communication, sending feedback, and safe logout choices. These guidelines are used in the system's design and implementation.

Entities with associated attributes and relationships, such as Patient, Doctor, PhoneNo, Address, Appointment, Reminder, Medical Records, Updating, and Feedback, are shown in the logical ERD. Database management is facilitated by this representation, which makes data structures simpler. To ensure a clear understanding of the database structure, the data dictionary offers extensive information on each entity's attributes, data types, lengths, and constraints.

The normalisation process ensures data integrity by removing any anomalies and confirming that every relation is in BCNF. As a result, relationships become well-organized and effectively manage and store information.

To operationalize the system, the last step is creating SQL statements (DDL & DML). The basis of the database's functionalities is these statements, which allow for task execution and interaction by system requirements.

Overall, the project has systematically progressed from conceptualization to a refined logical design, giving data integrity, efficiency, and user experience the highest priority while guaranteeing that the system's functionalities meet the intended operational requirements.