



---

**SECD 2523 - DATABASE**  
**SEMESTER I, SESSION 2023/2024**

**Phase 3: Database Conceptual Design  
E-Clinic System**

**Lecturer: Dr. Noor Hidayah Zakaria  
Section 08**

Name	Matric Number
Ghathfan Muhamad Afifazfa	A21EC4046
Kevin Fachrezy	A21EC4048
Wang Ya Peng	A21EC4035
Yaseen Mohamed Barakat	A22EC4016
Muhammad Zafran Bin Zairul Anuar	A22EC0089

# **Table Of Content**

<b>Table Of Content.....</b>	<b>2</b>
<b>Introduction.....</b>	<b>3</b>
<b>DFD.....</b>	<b>4</b>
1.1. Context Diagram.....	4
1.2. Diagram Level 0.....	5
1.3. Diagram Level 1.....	6
1.3.1. Manage Appointment.....	6
1.3.2. Access Patient Medical Record.....	7
1.3.3. Conduct Consultation.....	8
<b>Database conceptual design.....</b>	<b>9</b>
1.4. Conceptual ERD.....	9
1.5. Enhanced ERD (EERD).....	10
1.6. Database Logical design.....	11
<b>4. Data dictionary.....</b>	<b>12</b>
<b>4.3 Normalization.....</b>	<b>17</b>
<b>5.0 Relational Database Schemas (Normalized Table).....</b>	<b>18</b>
<b>SQL Statements.....</b>	<b>20</b>
<b>Summary.....</b>	<b>24</b>

## **Introduction**

This phase 2 document will focus on the development of the logical design of the E-Clinic system. In this phase, the proposed solutions described in phase 1 will be defined using Data Flow Diagram and Entity Relationship Diagram following a proposed business rule.

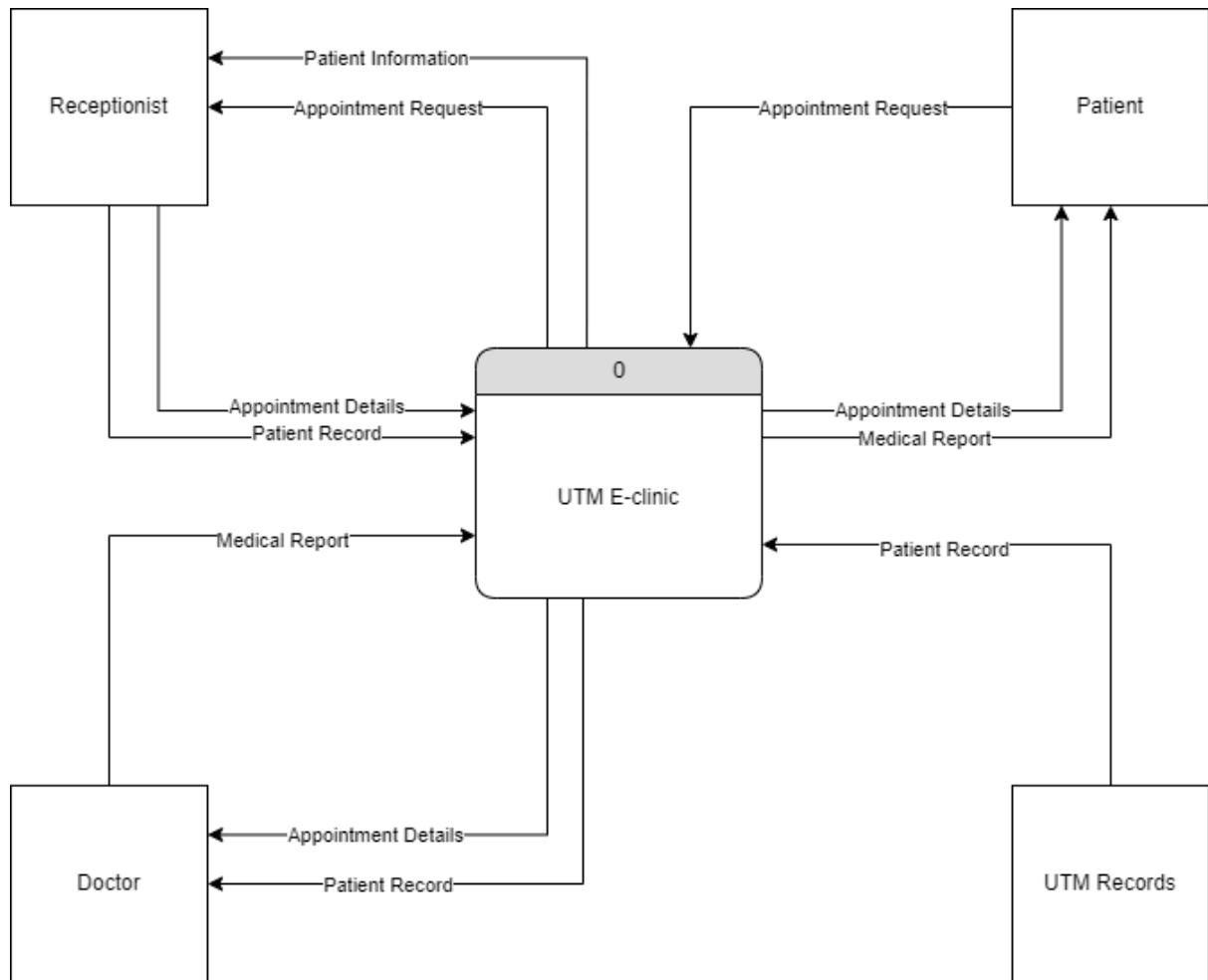
The Data Flow Diagram will be used to track which data from which actors that will flow through the system. The DFD will visualize the data from and to the external entities and the system itself. This includes primary data flow described in Context Diagram which will be expanded in the level 0 DFD where data from external entities will interact with the main system, that will be expanded into DFD level 1 where interactions with each subsystem will be defined in detail.

After making the DFD, the business rule will be defined in order to make the ERD. The business rule in this document will include business policies and constraints. After the business rule, the Entity Relationship Diagram to show the interaction of entity and system. The ERD is followed by an Enhanced ERD where the entities and its relation will be defined further.

At the end of the document, the entity and data that has been defined in the ERD will be listed in a Data Dictionary which will give the specifications such as data type, constraints, data length. This data dictionary will be used during the physical implementation.

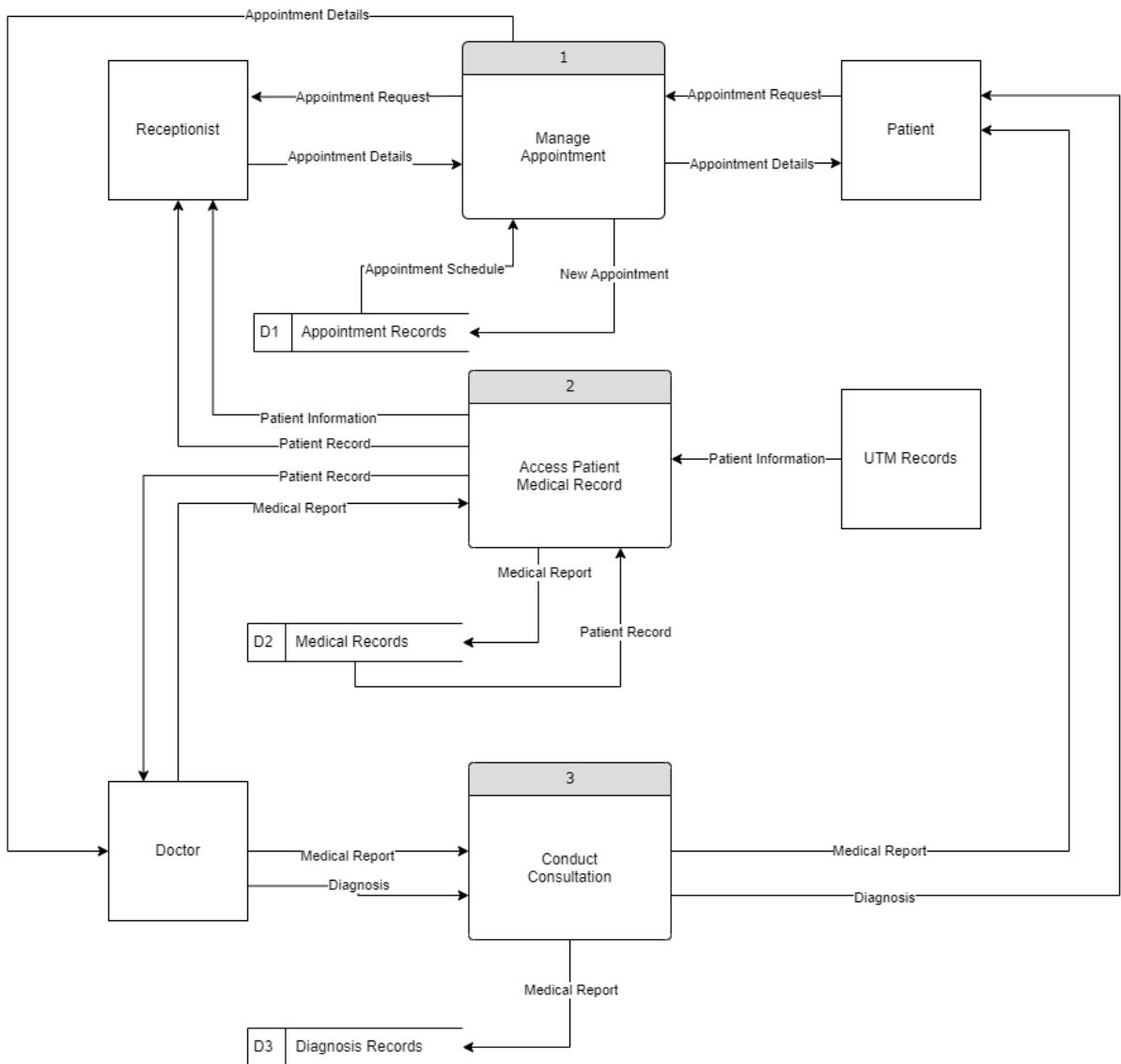
# DFD

## 1.1. Context Diagram



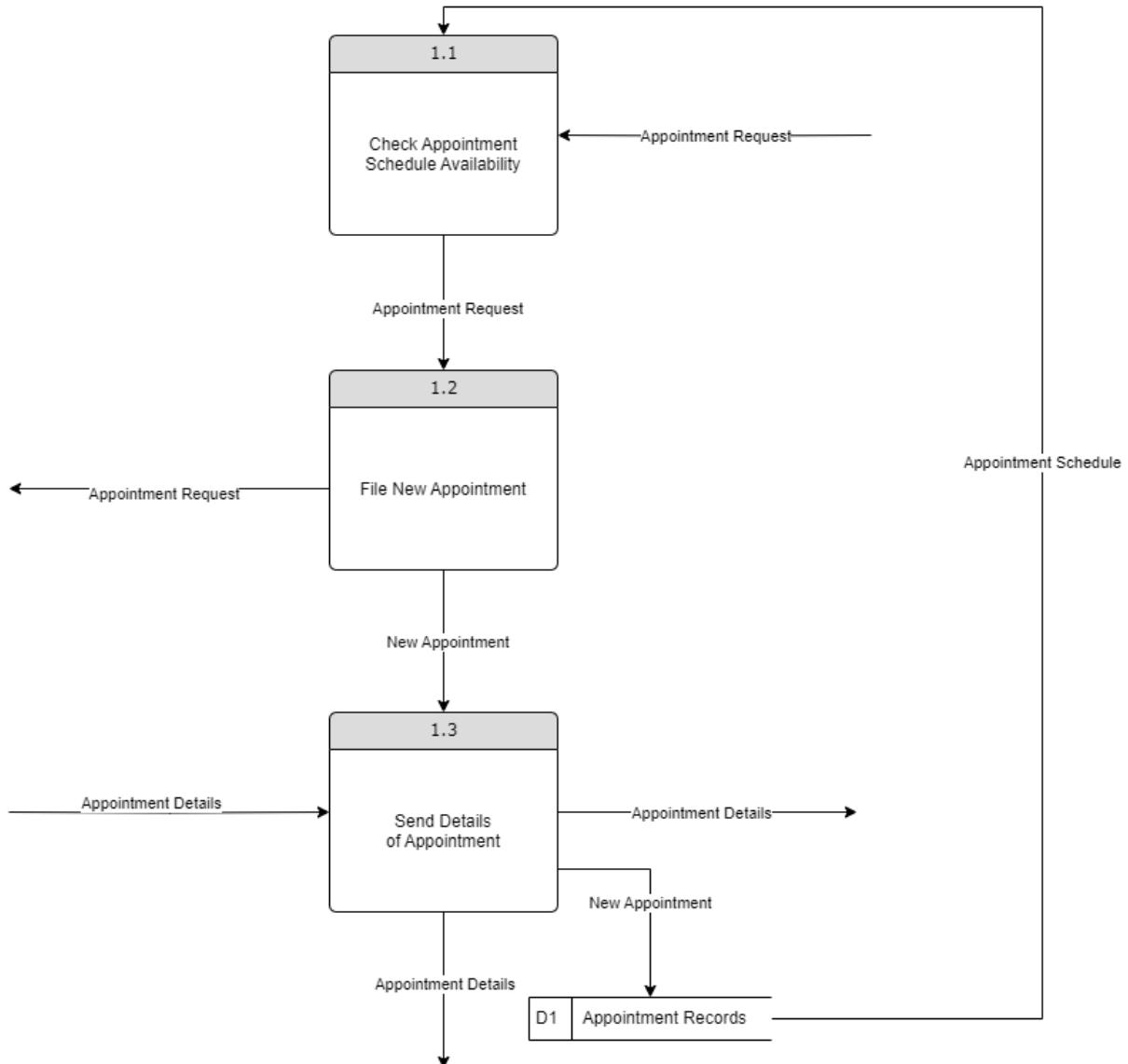
Actor: Doctor, Patient/Student , Pharmacist, Receptionist

## 1.2. Diagram Level 0

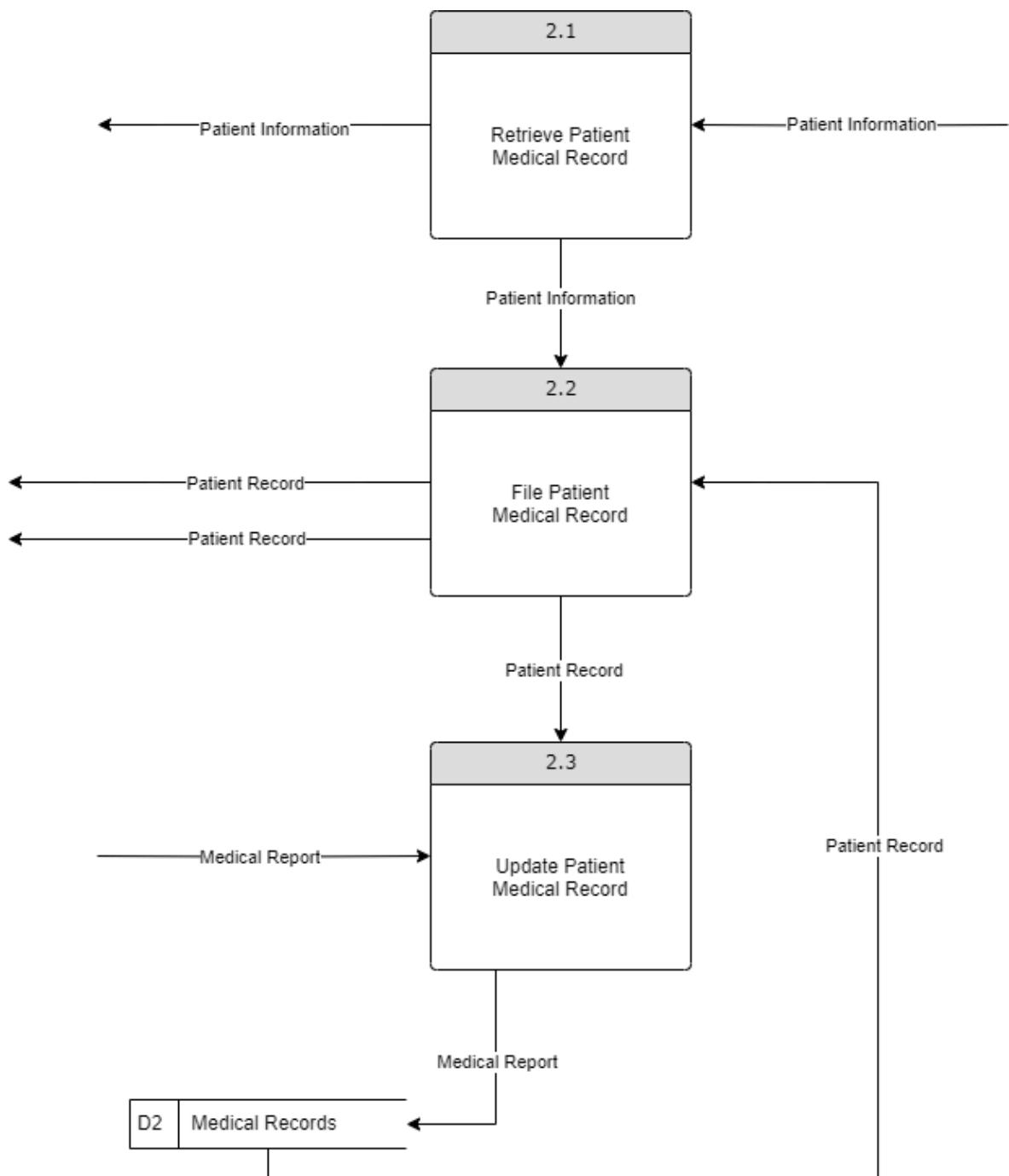


### 1.3. Diagram Level 1

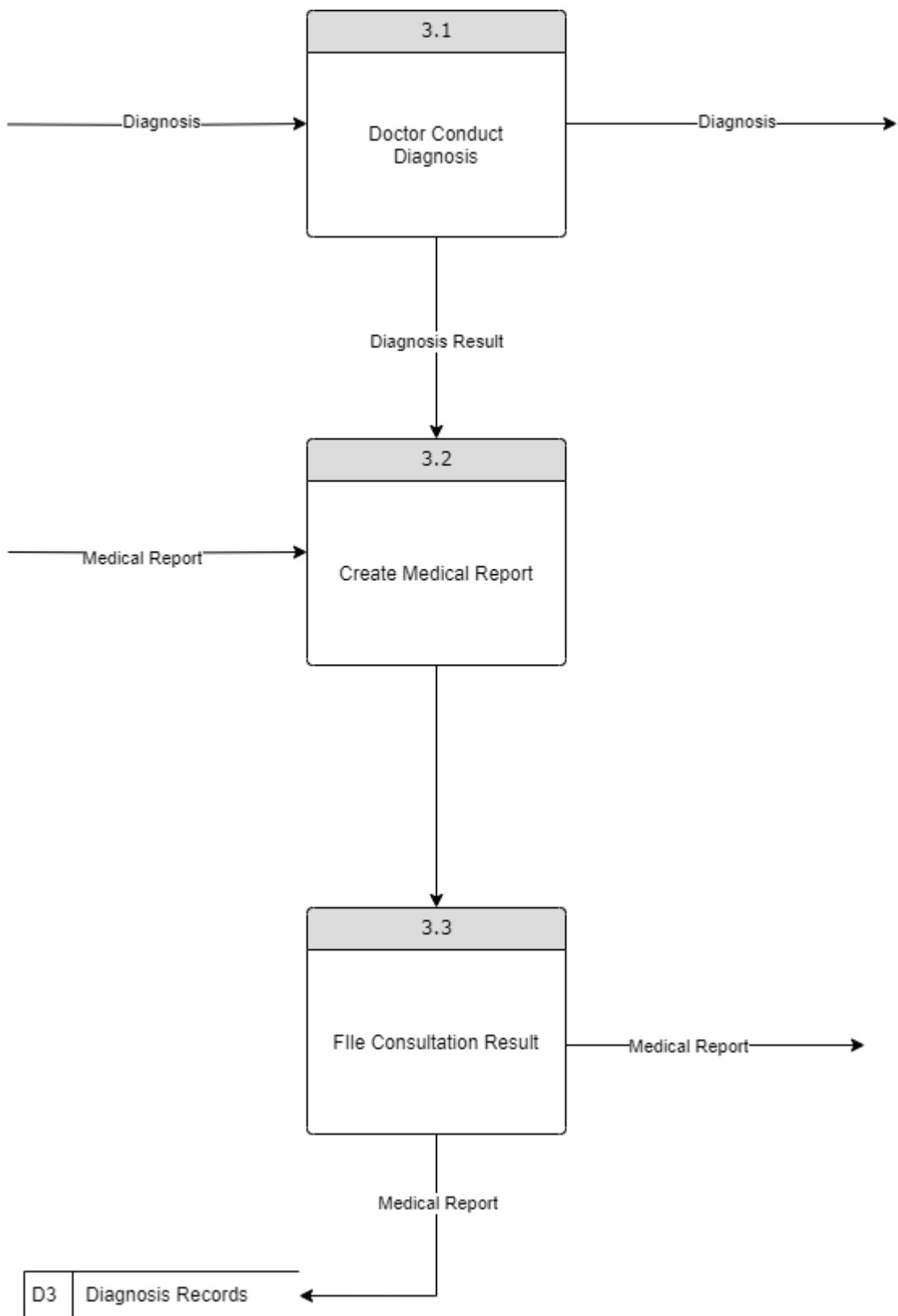
#### 1.3.1. Manage Appointment



### 1.3.2. Access Patient Medical Record

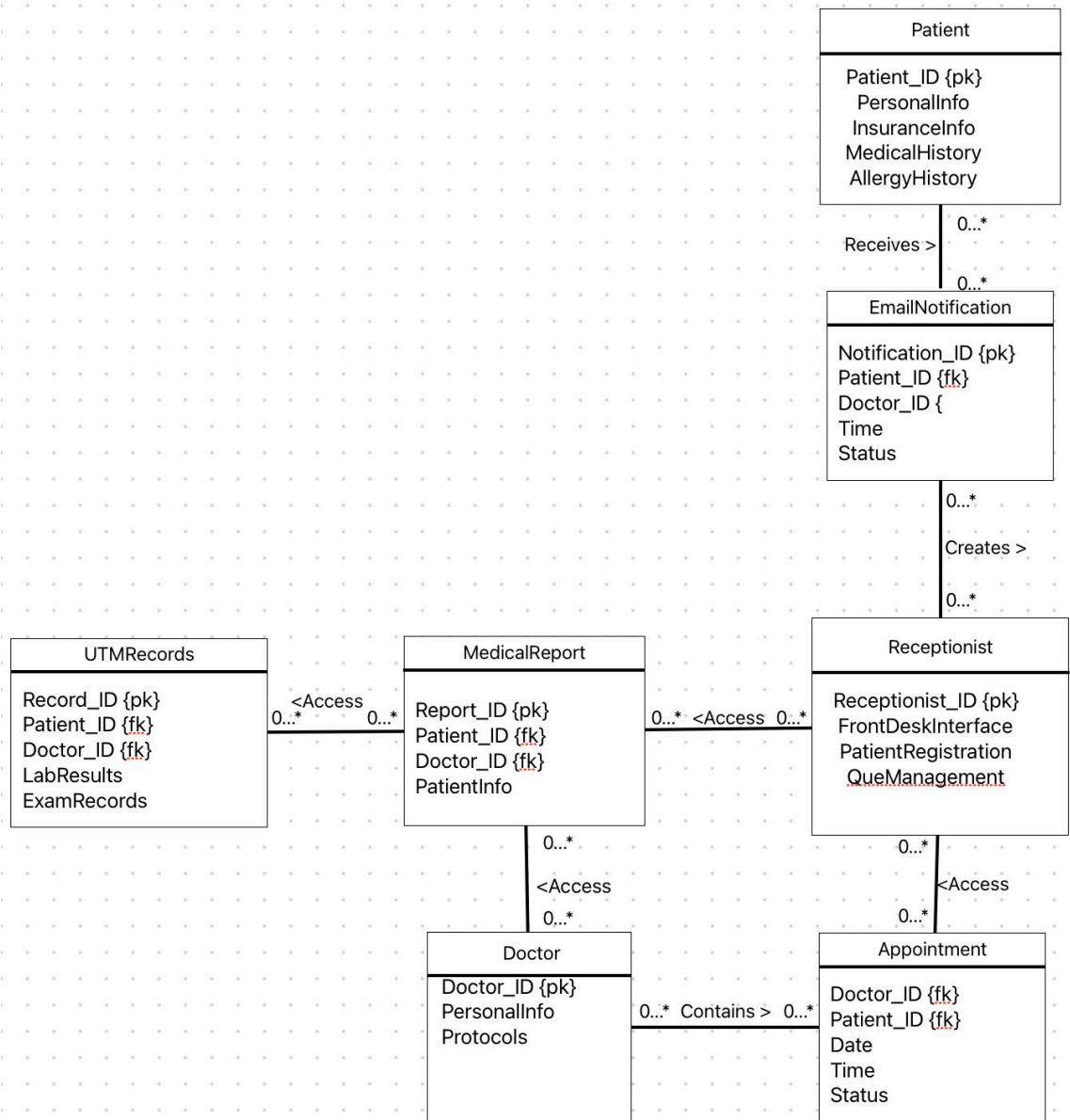


### 1.3.3. Conduct Consultation

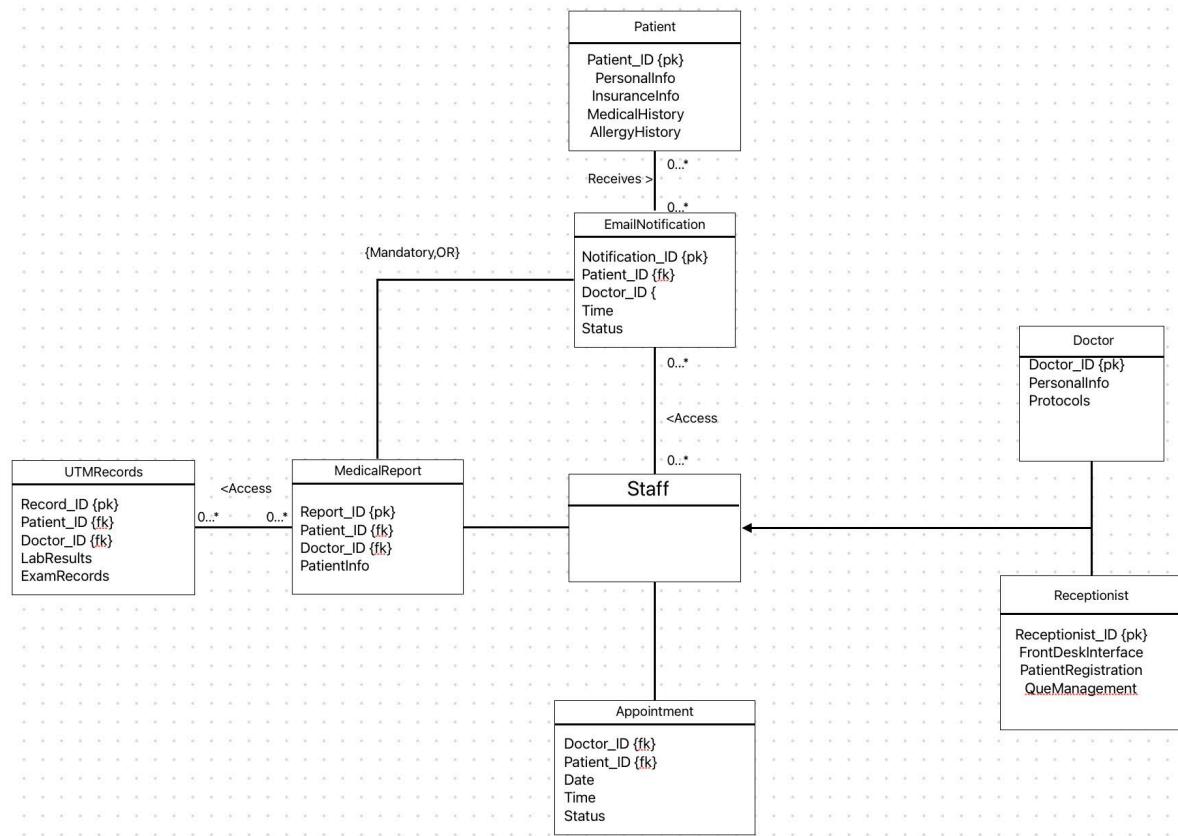


# Database conceptual design

## 1.4. Conceptual ERD



## 1.5. Enhanced ERD (EERD)



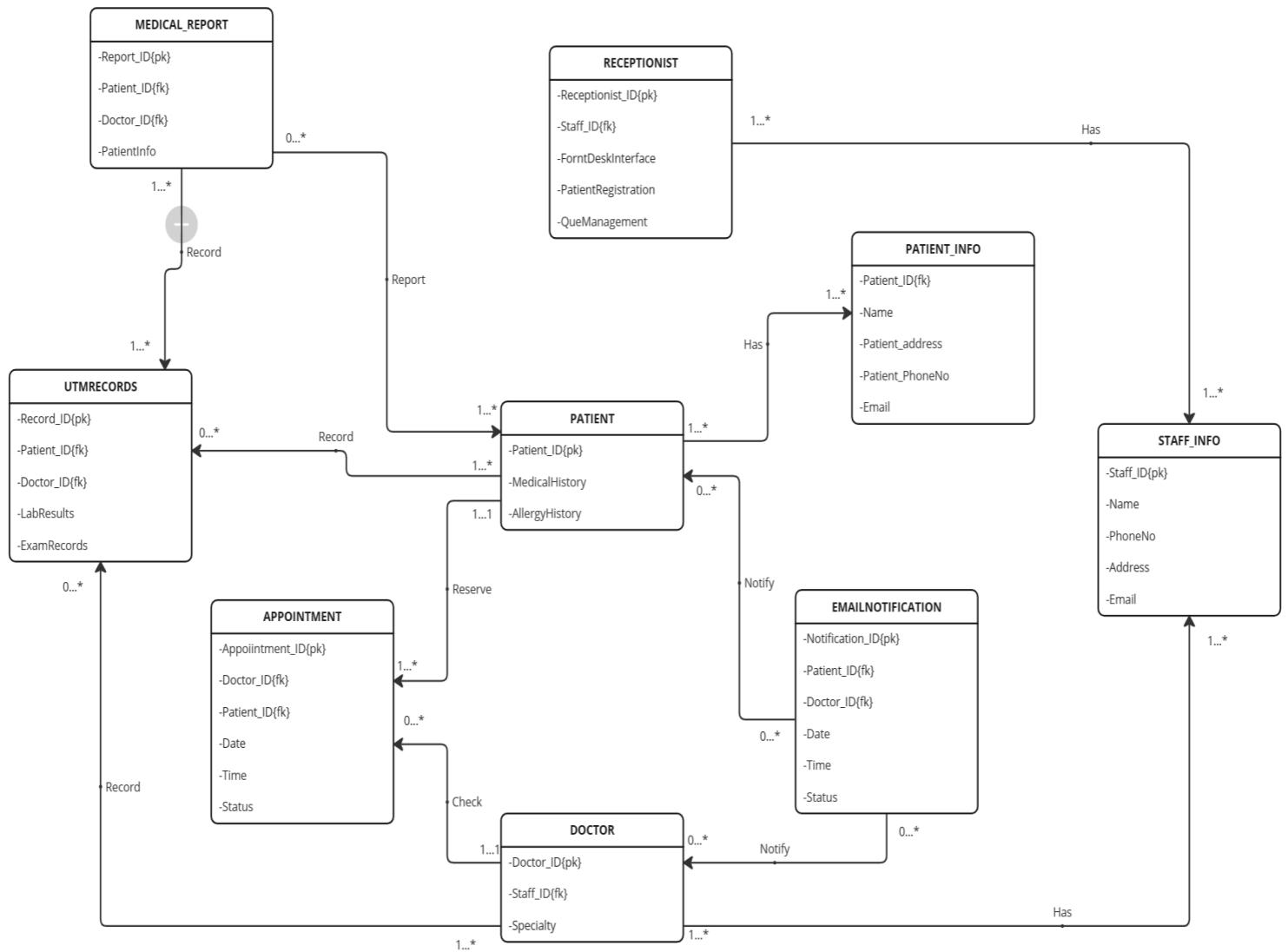
## 1.6. Database Logical design

**Strong Entities:**

- Patient
- Doctor
- Staff Info
- UTMRecords
- MedicalReport
- Appointment
- EmailNotification

**Weak Entities:**

- Patient Info





## **4. Data dictionary**

Our data dictionary provides a clear guide to the elements within your database. It details each table's attributes, including their data types, lengths, and specifications, helping users understand and manage the database efficiently.

### **4.2.1 Description of Entity**

<b>Entity</b>	<b>Description</b>	<b>Occurrence</b>
MEDICAL_REPORT	Contains records of medical reports generated for patients	One record per patient
UTMRECORD	Stores universal medical records including exam and lab results	Multiple records per patient, as needed
APPOINTMENT	Holds information about appointments scheduled for patients	Multiple entries possible for patients, one per appointment
PATIENT	Represents the patients who receive medical services	One record per patient
DOCTOR	Information about doctors providing medical services	One record per doctor
EMAILNOTIFICATION	Description: Tracks email notifications sent to patients regarding appointments	Multiple possible, one for each notification sent
PATIENT_INFO	Detailed information about patients	Occurrence: One record per patient, containing contact details
STAFF_INFO	Details of the medical and administrative staff	One record per staff member
RECEPTIONIST	Information about receptionists at the medical facility	One record per receptionist

#### 4.2 Description of Relationship

<b>Entity</b>	<b>Multiplicity (Entity side)</b>	<b>Relationship</b>	<b>Multiplicity (Related Entity side)</b>	<b>Related Entity</b>
MEDICAL_REPORT	1...1	Is Recorded In	<u>0..*</u>	UTMRECORDS
UTMRECORDS	1...1	Records	<u>1..*</u>	MEDICAL_REPORT
APPOINTMENT	1...1	Belongs To	1...1	PATIENT
PATIENT	1...1	Has	1...1	PATIENT_INFO
PATIENT_INFO	<u>1..*</u>	Belongs To	1...1	PATIENT
DOCTOR	1...1	Is Notified By	<u>0..*</u>	EMAILNOTIFICATION
EMAILNOTIFICATION	1...1	Notifies	1...1	DOCTOR
RECEPTIONIST	1...1	Manages	<u>1..*</u>	APPOINTMENT
PATIENT	1...1	Has an Appointment	<u>1..*</u>	APPOINTMENT
DOCTOR	1...1	Has an Appointment	<u>1..*</u>	APPOINTMENT
STAFF_INFO	1...1	Is	1...1	DOCTOR
STAFF_INFO	1...1	Is	1...1	RECEPTIONIST

#### 4.3 Description of Attributes

Entities	Attributes	Description	Data type	length	Null allowed?	Mult value
Patient	Patient_ID	Unique identifier for each patient	integer	10	No	No
	personalInfo	Full information for each patient	Varchar	30	Yes	Yes
	insuranceInfo	Full breakdown of policy and term	Varchar	30	Yes	Yes
	medicalHistory	Encompass complete full health details	varchar	30	Yes	Yes
	AllergyHistory	Listing allergens				
EmailNotification	Notification_ID (PK)	Unique identifier for each notification	Integer	10	No	No
	Patient_ID	Identification for the patient receiving the notification	Varchar	10	No	No
	Doctor_ID	Identification for the doctor associated with the notification	Varchar	30	Yes	No
	Time	Timestamp indicating when the notification was sent		30	No	No
	Status	Indicates the current status of the notification	varchar	30	No	No
Receptionist	Receptionist_ID (PK)	Unique identifier for each receptionist	Integer	10	No	No
	FrontDeskInterface	Capability or access level for the front desk interface	Varchar	30	Yes	No
	PatientRegistration	Capability or access level for patient registration tasks	Varchar	30	Yes	No
	QueueManagement	Capability or access level for managing queues or appointments	varchar	30	Yes	No

MedicalReport	Report_ID (PK)	Unique identifier for each medical report	Integer	10	No	No
	Patient_ID (FK)	Foreign key linking to the specific patient associated with the report	Integer	10	No	No
	Doctor_ID (FFK)	Foreign key linking to the specific doctor associated with the report	Integer	10	No	No
	PatientInfo	Field storing additional patient information related to the report	Varchar	30	Yes	No
UTMRecords	Record_ID (PK)	Unique identifier for each UTM record	Integer	10	No	No
	Patient_ID (FK)	Foreign key linking to the specific patient associated with the record	Varchar	30	No	No
	Doctor_ID (TK)	A technical key (TK) for the doctor associated with the record	Varchar	30	Yes	No
	LabResults	Field to store laboratory test results	Varchar	30	Yes	No
	ExamRecords	Field to store additional examination records or details	Varchar	30	Yes	No
Appointment	Doctor_ID (K)	Unique identifier for the doctor associated with the appointment	Integer	10	No	No
	Patient_ID (UKR)	Unique key reference for the patient associated with the appointment	Varchar	30	Yes	No
	Date	Date of the scheduled appointment	Varchar	30	Yes	No
	Time	Time of the scheduled appointment	Varchar	30	Yes	No
	Status	Indicates the current status of the appointment	varchar	30	Yes	No

Doctor	Doctor_ID (PK)	Unique identifier for each doctor	Integer	10	No	No
	Personalinfo	Field storing personal information about the doctor	Varchar	30	Yes	No
	Protocols	Field storing specific protocols or guidelines associated with the doctor's practice	varchar	30	yes	No

#### 4.3 Normalization

##### **PATIENT\_INFO**

FD: Patient\_ID → Name, Patient\_address, Patient\_PhoneNo, Email

1NF&2NF&3NF&BCNF: PATIENT\_INFO (Patient\_ID, Name, Patient\_address, (Patient\_PhoneNo, Email)

##### **DOCTOR**

FD: Doctor\_ID → Staff\_ID, Specialty

(1NF&2NF&3NF&BCNF: DOCTOR (Doctor\_ID, Staff\_ID, Specialty

\*\*

##### **STAFF\_INFO**

FD: Staff\_ID → Name, PhoneNo, Address, Email

(1NF&2NF&3NF&BCNF: STAFF\_INFO (Staff\_ID, Name, PhoneNo, Address, Email)

##### **APPOINTMENT**

FD: Appointment\_ID → Doctor\_ID, Patient\_ID, Date, Time, Status

1NF&2NF&3NF&BCNF: APPOINTMENT (Appointment\_ID, Doctor\_ID, Patient\_ID, Date, (Time, Status

## **EMAIL\_NOTIFICATION**

FD: Notification\_ID → Patient\_ID, Doctor\_ID, Date, Time, Status

1NF&2NF&3NF&BCNF: EMAIL\_NOTIFICATION (Notification\_ID, Patient\_ID, Doctor\_ID, Date, Time, Status)

## **MEDICAL\_REPORT**

FD: Report\_ID → Patient\_ID, Doctor\_ID, PatientInfo

:1NF&2NF&3NF&BCNF

(MEDICAL\_REPORT (Report\_ID, Patient\_ID, Doctor\_ID, PatientInfo

## **UTMRECORDS**

FD: Record\_ID → Patient\_ID, Doctor\_ID, LabResults, ExamRecords

NF&2NF&3NF&BCNF: UTMRECORDS (Record\_ID, Patient\_ID, Doctor\_ID, LabResults, ExamRecords)

## **PATIENT**

FD: Patient\_ID → MedicalHistory, AllergyHistory

(1NF&2NF&3NF&BCNF: PATIENT (Patient\_ID, MedicalHistory, AllergyHistory

## **5.0 Relational Database Schemas (Normalized Table)**

This is the set of relation schemas in the relational database of the E-Clinic System database.

MEDICAL\_REPORT (Report\_ID, Patient\_ID, Doctor\_ID, PatientInfo)

UTMRECORD (Record\_ID, Patient\_ID, Doctor\_ID, LabResults, ExamRecords)

APPOINTMENT (Appointment\_ID, Doctor\_ID, Patient\_ID, Date, Time, Status)

PATIENT (Patient\_ID, MedicalHistory, AllergyHistory)

DOCTOR (Doctor\_ID, Staff\_ID, Specialty)

EMAILNOTIFICATION (Notification\_ID, Patient\_ID, Doctor\_ID, Date, Time, Status)

PATIENT\_INFO (Patient\_ID, Name, Patient\_Address, Patient\_PhoneNo, Email)

STAFF\_INFO (Staff\_ID, Name, PhoneNo, Address, Email)

**RECEPTIONIST** (Receptionist\_ID, FrontDeskInterface, PatientRegistration, QueueManagement)

**MEDICAL\_REPORT**

Report_ID (PK)	Patient_ID	Doctor_ID	PatientInfo
----------------	------------	-----------	-------------

**UTMRECORD**

Record_ID (PK)	Patient_ID	Doctor_ID	LabResults	ExamRecords
----------------	------------	-----------	------------	-------------

**APPOINTMENT**

Appointment_ID (PK)	Doctor_ID	Patient_ID	Date	Time	Status
---------------------	-----------	------------	------	------	--------

**PATIENT**

Patient_ID (PK)	MedicalHistory	AllergyHistory
-----------------	----------------	----------------

**EMAILNOTIFICATION**

Notification_ID (PK)	Patient_ID	Doctor_ID	Date	Time	Status
----------------------	------------	-----------	------	------	--------

**PATIENT\_INFO**

Patient_ID (PK)	Name	Patient_Address	Patient_PhoneNo	Email
-----------------	------	-----------------	-----------------	-------

**STAFF\_INFO**

Staff_ID (PK)	Name	PhoneNo	Address	Email
---------------	------	---------	---------	-------

**RECEPTIONIST**

Receptionist_ID (PK)	FrontDeskInterface	PatientRegistration	QueueManagement
----------------------	--------------------	---------------------	-----------------

**DOCTOR**

Doctor_ID (PK)	Staff_ID	Specialty
----------------	----------	-----------

## SQL Statements

```
CREATE TABLE Patient (
    Patient_ID VARCHAR2(10) NOT NULL PRIMARY KEY,
    Medical_History VARCHAR2(50),
    Allergy_History VARCHAR2(50)
);

CREATE TABLE Patient_Info (
    Patient_ID VARCHAR2(10) NOT NULL,
    Patient_Name VARCHAR2(50) NOT NULL,
    Patient_address VARCHAR2(50) NOT NULL,
    CONSTRAINT Patient_ID_FK FOREIGN KEY (Patient_ID) REFERENCES Patient
(Patient_ID)
);

CREATE TABLE Staff_Info (
    Staff_ID VARCHAR2(10) NOT NULL PRIMARY KEY,
    Staff_Name VARCHAR2(50) NOT NULL,
    Staff_PhoneNo NUMBER(15) NOT NULL,
    Staff_Address VARCHAR2(50) NOT NULL,
    Staff_Email VARCHAR2(20) NOT NULL
);

CREATE TABLE Doctor (
    Doctor_ID VARCHAR2(10) NOT NULL PRIMARY KEY,
    Staff_ID VARCHAR2(10) NOT NULL,
    Speciality VARCHAR2(10) NOT NULL,
    CONSTRAINT Staff_ID_FK FOREIGN KEY (Staff_ID) REFERENCES Staff_Info
(Staff_ID)
);

CREATE TABLE MEDICAL_REPORT (
    Report_ID VARCHAR2(10) NOT NULL PRIMARY KEY,
    Patient_ID VARCHAR2(10) NOT NULL,
    Doctor_ID VARCHAR2(10) NOT NULL,
    PatientInfo VARCHAR2(30),
    CONSTRAINT Patient_ID_FK_2 FOREIGN KEY (Patient_ID) REFERENCES Patient
(Patient_ID),
    CONSTRAINT Doctor_ID_FK_2 FOREIGN KEY (Doctor_ID) REFERENCES Doctor
(Doctor_ID)
);

CREATE TABLE UTMRecords (
    Record_ID VARCHAR2(10) NOT NULL PRIMARY KEY,
```

```
Patient_ID VARCHAR2(10) NOT NULL,  
Doctor_ID VARCHAR2(10) NOT NULL,  
LabResults VARCHAR2(50),  
ConsultationRecord VARCHAR(50),  
CONSTRAINT Patient_ID_FK_3 FOREIGN KEY (Patient_ID) REFERENCES Patient  
(Patient_ID),  
CONSTRAINT Doctor_ID_FK_3 FOREIGN KEY (Doctor_ID) REFERENCES Doctor  
(Doctor_ID)  
);
```

```
CREATE TABLE Appoinment (  
    Appointment_ID VARCHAR2(10) NOT NULL PRIMARY KEY,  
    Patient_ID VARCHAR2(10) NOT NULL,  
    Doctor_ID VARCHAR2(10) NOT NULL,  
    Appointment_Date DATE NOT NULL,  
    Appointment_Time TIMESTAMP NOT NULL,  
    Appointment_Status VARCHAR2(15) NOT NULL,  
    CONSTRAINT Patient_ID_FK_4 FOREIGN KEY (Patient_ID) REFERENCES Patient  
(Patient_ID),  
    CONSTRAINT Doctor_ID_FK_4 FOREIGN KEY (Doctor_ID) REFERENCES Doctor  
(Doctor_ID)  
);
```

```
CREATE TABLE EmailNotification (  
    Notification VARCHAR2(10) NOT NULL PRIMARY KEY,  
    Patient_ID VARCHAR2(10) NOT NULL,  
    Doctor_ID VARCHAR2(10) NOT NULL,  
    EmailNotification_Date DATE NOT NULL,  
    EmailNotification_Time TIMESTAMP NOT NULL,  
    EmailNotification_Status VARCHAR2(15) NOT NULL,  
    CONSTRAINT Patient_ID_FK_5 FOREIGN KEY (Patient_ID) REFERENCES Patient  
(Patient_ID),  
    CONSTRAINT Doctor_ID_FK_5 FOREIGN KEY (Doctor_ID) REFERENCES Doctor  
(Doctor_ID)  
);
```

```
CREATE TABLE Receptionist (  
    Receptionist_ID VARCHAR2(10) NOT NULL PRIMARY KEY,  
    Staff_ID VARCHAR2(10) NOT NULL,  
    PatientRegistration VARCHAR2(10) NOT NULL,  
    QueueManagement VARCHAR2(10) NOT NULL,  
    CONSTRAINT Staff_ID_FK_5 FOREIGN KEY (Staff_ID) REFERENCES Staff_Info  
(Staff_ID)  
);
```

```
-- Patient table  
INSERT INTO Patient (Patient_ID, Medical_History, Allergy_History)
```

```
VALUES ('P001', 'Flu', 'Penicillin allergy');
```

```
INSERT INTO Patient (Patient_ID, Medical_History, Allergy_History)  
VALUES ('P002', 'Vertigo', 'None');
```

```
INSERT INTO Patient (Patient_ID, Medical_History, Allergy_History)  
VALUES ('P003', 'Fever', 'Sulfa allergy');
```

-- Patient\_Info table

```
INSERT INTO Patient_Info (Patient_ID, Patient_Name, Patient_address)  
VALUES ('P001', 'Zufar Einstein', 'KTR');
```

```
INSERT INTO Patient_Info (Patient_ID, Patient_Name, Patient_address)  
VALUES ('P002', 'Phillip Xander', 'KLG Residence');
```

```
INSERT INTO Patient_Info (Patient_ID, Patient_Name, Patient_address)  
VALUES ('P003', 'Muhammad Amirul', 'KDOJ');
```

-- Staff\_Info table

```
INSERT INTO Staff_Info (Staff_ID, Staff_Name, Staff_PhoneNo, Staff_Address, Staff_Email)  
VALUES ('S001', 'Dr. Anderson', 1234567890, 'Mutia Rini', 'anderson@example.com');
```

```
INSERT INTO Staff_Info (Staff_ID, Staff_Name, Staff_PhoneNo, Staff_Address, Staff_Email)  
VALUES ('S002', 'Dr. Davis', 0876543210, 'Taman Universiti', 'davis@example.com');
```

```
INSERT INTO Staff_Info (Staff_ID, Staff_Name, Staff_PhoneNo, Staff_Address, Staff_Email)  
VALUES ('S003', 'Dr. Johnson', 5551112222, 'KLG Residence', 'johnson@example.com');
```

-- Doctor table

```
INSERT INTO Doctor (Doctor_ID, Staff_ID, Speciality)  
VALUES ('D001', 'S001', 'General Practitioner');
```

```
INSERT INTO Doctor (Doctor_ID, Staff_ID, Speciality)  
VALUES ('D002', 'S002', 'General Practitioner');
```

```
INSERT INTO Doctor (Doctor_ID, Staff_ID, Speciality)  
VALUES ('D003', 'S003', 'General Practitioner');
```

-- MEDICAL\_REPORT table

```
INSERT INTO MEDICAL_REPORT (Report_ID, Patient_ID, Doctor_ID, PatientInfo)  
VALUES ('R001', 'P001', 'D001', 'Medical checkup');
```

```
INSERT INTO MEDICAL_REPORT (Report_ID, Patient_ID, Doctor_ID, PatientInfo)  
VALUES ('R002', 'P002', 'D002', 'Medical checkup');
```

```
INSERT INTO MEDICAL_REPORT (Report_ID, Patient_ID, Doctor_ID, PatientInfo)  
VALUES ('R003', 'P003', 'D003', 'Medical checkup');
```

```
-- UTMRecords table
INSERT INTO UTMRecords (Record_ID, Patient_ID, Doctor_ID, LabResults,
ConsultationRecord)
VALUES ('U001', 'P001', 'D001', 'N/A', 'Regular checkup');
```

```
INSERT INTO UTMRecords (Record_ID, Patient_ID, Doctor_ID, LabResults,
ConsultationRecord)
VALUES ('U002', 'P002', 'D002', 'Low Blood Pressure', 'Regular checkup');
```

```
INSERT INTO UTMRecords (Record_ID, Patient_ID, Doctor_ID, LabResults,
ConsultationRecord)
VALUES ('U003', 'P003', 'D003', 'N/A', 'Regular checkup');
```

-- Appointment table

```
INSERT INTO Appointment (Appointment_ID, Patient_ID, Doctor_ID, Appointment_Date,
Appointment_Time, Appointment_Status)
VALUES ('A001', 'P001', 'D001', TO_DATE('2024-01-15', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-15 10:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'Scheduled');
```

```
INSERT INTO Appointment (Appointment_ID, Patient_ID, Doctor_ID, Appointment_Date,
Appointment_Time, Appointment_Status)
VALUES ('A002', 'P002', 'D002', TO_DATE('2024-01-16', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-16 11:30:00', 'YYYY-MM-DD HH24:MI:SS'), 'Completed');
```

```
INSERT INTO Appointment (Appointment_ID, Patient_ID, Doctor_ID, Appointment_Date,
Appointment_Time, Appointment_Status)
VALUES ('A003', 'P003', 'D003', TO_DATE('2024-01-17', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-17 14:45:00', 'YYYY-MM-DD HH24:MI:SS'), 'Canceled');
```

-- EmailNotification table

```
INSERT INTO EmailNotification (Notification, Patient_ID, Doctor_ID, EmailNotification_Date,
EmailNotification_Time, EmailNotification_Status)
VALUES ('N001', 'P001', 'D001', TO_DATE('2024-01-15', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-15 09:30:00', 'YYYY-MM-DD HH24:MI:SS'), 'Sent');
```

```
INSERT INTO EmailNotification (Notification, Patient_ID, Doctor_ID, EmailNotification_Date,
EmailNotification_Time, EmailNotification_Status)
VALUES ('N002', 'P002', 'D002', TO_DATE('2024-01-16', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-16 12:15:00', 'YYYY-MM-DD HH24:MI:SS'), 'Delivered');
```

```
INSERT INTO EmailNotification (Notification, Patient_ID, Doctor_ID, EmailNotification_Date,
EmailNotification_Time, EmailNotification_Status)
VALUES ('N003', 'P003', 'D003', TO_DATE('2024-01-17', 'YYYY-MM-DD'),
TO_TIMESTAMP('2024-01-17 15:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'Failed');
```

select

```
"PATIENT_ID",
"MEDICAL_HISTORY",
```

```
"ALLERGY_HISTORY"  
from "PATIENT";
```

```
select  
    "APPOINTMENT_ID",  
    "PATIENT_ID",  
    "DOCTOR_ID",  
    "APPOINTMENT_DATE",  
    "APPOINTMENT_TIME",  
    "APPOINTMENT_STATUS"  
from "APPOINTMENT";
```

```
select  
    "NOTIFICATION",  
    "PATIENT_ID",  
    "DOCTOR_ID",  
    "EMAILNOTIFICATION_DATE",  
    "EMAILNOTIFICATION_TIME",  
    "EMAILNOTIFICATION_STATUS"  
from "EMAILNOTIFICATION";
```

```
select * from Doctor where Doctor_ID LIKE '%D002%';
```

```
SELECT *  
FROM UTMRECORDS  
INNER JOIN APPOINTMENT ON UTMRECORDS.DOCTOR_ID =  
APPOINTMENT.DOCTOR_ID  
where UTMRECORDS.Doctor_ID LIKE '%D002%'
```

# User Interface Design

The home page features the UTM e-clinic logo at the top left. A search bar and a yellow circular icon are on the right. The main content area has a banner with a building and mountains, followed by two tables for operating hours.

DAY	TIME
Sunday – Wednesday	8.00 am – 10.00 pm
Thursday	8.00 am – 10.00 pm
Friday – Saturday	8.30 am – 12.30 pm
Public Holiday	CLOSE

DAY	TIME
Sunday – Wednesday	8.00 am – 5.00 pm
Thursday	8.00 am – 3.30 pm
Friday – Saturday	8.30 am – 12.30 pm
Public Holiday	CLOSE

The appointment request page includes the UTM e-clinic logo and navigation menu. The main section has two forms: Patient Information and Appointment Information, both with dropdown menus and availability indicators.

Patient Name Ghathfan Muhamad Afifazfa	Matrik Number A21EC4046
---	----------------------------

Patient Address Room 717 ,Block C, KLG Residence - UTM Johor Bahru
---

Contact Number +60 1234567890	Email Address ghathfan@graduate.utm.my
----------------------------------	---

Date of Appointment DD/MM/YY	Time of Appointment 10:00 AM	Consultation Type Online
---------------------------------	---------------------------------	-----------------------------

Doctor Name - Speciality Dr. Mohd Zaki Bin Yunos - General Practitioners
---

Availability AVAILABLE
---------------------------

**Patient Information**

**Appointment Information**

**Submit**



# UTM E-Clinic System

## Patient Profile



Patient Name

Ghathfan Muhamad Afifazfa

Matric Number

A21EC4046

Contact Number

+60 1234567890

Year - Semester

Year 2 - 4th Semester

Faculty

Computing

Email Address

ghathfan@graduate.utm.my

Home Address

Room 717 ,Block C, KLG Residence -  
UTM Johor Bahru**Medical Record****Save**

# UTM E-Clinic System

## Doctor Profile



Doctor Name

Dr. Mohd Zaki Bin Yunos

Doctor ID

D0001

Contact Number

+60 1234567890

Speciality

General Practitioners

Email Address

example@mail.com

Home Address

UTM Johor Bahru

**Save**

## Summary

This document outlines the design of the logical ERD of the system and the normalization process. The Data Dictionary is also modified to conform to the new logical ERD as well as the Relational Database schema. These changes were implemented in the SQL statements by applying both DDL and DML.