ALBUKHARY INTERNATIONAL UNIVERSITY

CCC2133 : Database Management System
Semester 1 , Academic Session: 2023/2024

Project

Group Number: 7

Dr. Zurinahni Zainol

Hospital management system

Name	Matric No.	Email Address	Module	Role
Roua Aliman	AIU22102291	roua.aliman@student.aiu.edu.my	Doctor	Leader
Rania Kherba	AIU22102285	rania.kherba@student.aiu.edu.my	Medical record	Member
S M Asiful Islam Saky	AIU22102005	asiful.saky@student.aiu.edu.my	Patient	Member
Rukayya Sadiq	AIU21102372	rukayya.sadiq@student.aiu.edu.my	Appointment	Member

Date of submission

Table of contents

1.0 Introduction

1.1 Description of the organization

Kedah Medical Centre Sdn Bhd, situated in Pumpong, 05250 Alor Setar, Kedah Darul Aman, stands as an eminent healthcare institution in the kedah region. Kedah Medical Center deals with a considerably large number of patients from all over Kedah state everyday. The medical System is committed to deliver exceptional patient care for all of its patients. Thus, the deployment of a complete system that manages the patient's information is necessary. Kedah Hospital management system is a hospital information software that assists the hospital to manage their operations. A hospital management system will make information more accessible and easy to manage and update.

1.2 Description of the current system and its problems

To administer many parts of hospital operations, Kedah Medical Centre now uses a combination of manual record-keeping and fragmented digital systems. This strategy has resulted in a number of unique issues for Kedah Medical Centre, including:

Inefficiency: Manual processes take time and are prone to mistakes, causing delays in patient treatment and administrative activities.

Data Fragmentation: Information is distributed across several departments and systems, making real-time, aggregated data difficult to access.

Lack of Integration: Existing systems frequently function in silos, preventing smooth communication between departments and risking Kedah Medical Centre's overall coordination of healthcare services.

Security Concerns: Manual records and disparate systems may pose security risks, especially concerning patient confidentiality and data integrity, a critical consideration for Kedah Medical Centre.

1.3 Motivation

The healthcare industry works as a mainstream factor for the welfare of society. Because of the high crowd, sometimes the operational complexities within the hospitals arise at the peak and become challenging to deliver best care. The actual motivation behind developing hospital management systems is to make the process easier of all the management processes like patients registration, billing, doctors appointment, doctors prescription, etc. We see the difficulties and sufferings people face in the traditional hospital operations due to manual record-keeping, analog and disconnected systems, and unorganized information. An integrated hospital management system aims to smooth these processes by centralizing data, automating routine tasks, and optimizing resource allocation. In manual operations, we see the hospital staff, from doctors to executives, face difficulties in accessing critical patient information, diagnostic history and medical records. The hospital management system helps to access patient records, treatment history, and medication details, empowering healthcare providers to make informed decisions swiftly, thereby improving patient care and safety. The hospital management system comes with a user-friendly interface for patients with smoother interactions, simplified appointments, and greater transparency in healthcare processes. Patients can also check reports and navigate the doctors' scheduling for the consultation. It also ensures security measures in every aspect. By developing the system we can enhance the facilities of the healthcare industry and can assure the best output for the society.

1.4 How the proposed database system will benefit the organization

Effective data administration, better patient care, enhanced workflow, data security, analytics, appointment scheduling, resource optimization, telemedicine and remote access, and scalability are just a few advantages of the suggested database system for a hospital. It ensures prompt access and effective administration by centralizing administrative, medical, and patient data. Healthcare professionals can make better decisions and deliver better patient care when they have access to thorough patient records. In addition, the system complies with legal requirements, protects the privacy of user data, and offers analytics for resource allocation and trend detection. In order to adapt to changes and expansion within the hospital, it also enables scalability.

1.5 Describe each module In the hospital management system

1.5.0 Patient Module

Patients info module is the section of the system that consists of patients' personal information that will be recorded while registering for the patient. Here the required fields will be about patients' basic information and later on about patients' history. It's about the particular history ID and patients' health and clinical information. The system will also record the previous prescription that was given by the doctor. Eventually the payment details are the information that will be used for patients' release and so on whether the amount of money is already paid or due. In conclusion, the module that records and deals with patients' personal information, medical history and payment details, is the patient module.

1.5.1 Doctor Module

The module that deals with the doctors' information in the system is the doctors' info module. Here all the doctors' details get managed and stored. Information about each doctor in the hospital is stored there so reaching the doctor is more convenient. Doctors information will contain the doctor's first and last name, specialization, contact details, and address. The module contains an entity to manage and check the doctors working days and hours. Furthermore, The appointment module will save the appointment of the patients with which doctor and details about the appointment and the diagnosis.

1.5.2 Appointment Module

General information regarding the appointments will be included in the appointments module. Three tables will be included in this module: one for storing appointment information, such as the patient's ID, the appointment writer's name, and the date. The Appointment Notes table, on the other hand, will contain the notes the doctor makes regarding the patient and the appointment. Ultimately, the appointment state will be recorded in the final table, which will be the Appointment status table.

1.5.3 Medical Record Module:

The Medical Record Module stands as a cornerstone in our hospital system, with three main parts: MedicalRecords, Prescriptions, and TreatmentHistory. Its primary function ist keeping track of patient health info and making sure everything connects smoothly .
The module captures a comprehensive snapshot of a patient's medical journey. The MedicalRecords table holds key details like diagnoses and test results, while the Prescriptions and TreatmentHistory tables delve into medication specifics and treatment dynamics. This structured approach ensures that healthcare providers can access accurate and holistic patient information, leading to improved care, precise treatment planning, and streamlined hospital operations.

2.0 User Requirements and Business Rules

2.1 Patient Module

	User Requirement	Business Rule
01	Patient Registration: The system creates patient user profiles, for the new patient it registers a profile with essential information such as name, contact information, insurance, medical history and other info. It also maintains the profile and user data for the registered patients.	Patients interact with the system for various purposes like scheduling appointments, accessing medical records, updating their personal information and managing communication with the healthcare providers.
02	Appointment Scheduling: The system should be developed to allow patients to book appointments for consultation with doctors or specialists. It should be under the patients' ability to request, view, and manage appointments with healthcare providers. A clear communication among the patients, doctors and administrative individuals should be made through this system.	It is an official connection between doctors and patients about setting up the appointment schedules for the meeting. Patients seek a system to set up the upcoming possible appointment date and time. The healthcare providers can communicate with patients for consultations or follow-ups.
03	Medical Records Management: One of the most important tasks of the patient module of HMS is creating and managing the database to store patient medical histories, diagnosis, prescribed medications, lab test results, and treatment plans securely. Healthcare providers should have access for these information to view and edit for the medication purposes while patients might need the access to view these information.	All the respective healthcare providers like doctors, nurses, and medical staff access patient records, update treatment plans, store diagnosis reports and the compliments of the diagnostic lab tests which might be viewed by the patients too.
04	Billing and Payments: The system must have automatic billing functionalities that will generate the invoice once the patient pays the bill. It should also track the payments and other information such as insurance claims that is related to patient treatments. In brief, patients should receive clear and easily understandable invoices and staff should have access to manage billing, insurance claims, and payments.	Staff manage the overall system, assist patients with appointments, handle billing information, track the payments, work to generate payment invoices, and facilitate communication between patients and healthcare providers.

2.2 Doctor Module

	User Requirement	Business Rule
01	Doctor Profile Management: The system allows doctors to create and manage their profiles, allowing them to input their personal information like their name, address, contact information, and even specify their specialization. The system allows the doctors to organize their shift hours and working days.	Each doctor is associated with zero to many patients illustrating the doctor's responsibility toward their patients.
02	Appointment scheduling: The system enables doctors to schedule and manage appointments with the patients. Doctors can manage appointments and set a suitable timing to make his medical check-up on the patient.	Doctors can schedule appointments with the patient ensuring their availability. Patients on the other hand are allowed to request a specific timing with their doctor and wait for confirmation from the doctor.
03	Patient information access: The system enables the doctor to check the medical history record of the patient allowing the doctor to know about patients current medications, any previous operations and diagnosis.	Doctors should vow to keep his patients' information safe and not accessing the records is only for the purpose of diagnosing. Doctors will only be permitted to access and update their own patients.
04	Prescription Management: Through the system doctors can prescribe the treatment digitally including information about the medication dosage and additional instructions.	Doctors can prescribe medications based on their specialization field.
05	Update medical records: The doctor has the ability to update the medication notes and instructions for each patient if needed. The system should ensure the confidentiality of the medical records. Only related doctors can update the medications for their patients.	Doctors can create and update medical records during or after the check-up. Medical records are confidential and can be only accessed by a respective doctor.

2.3 Appointment Module

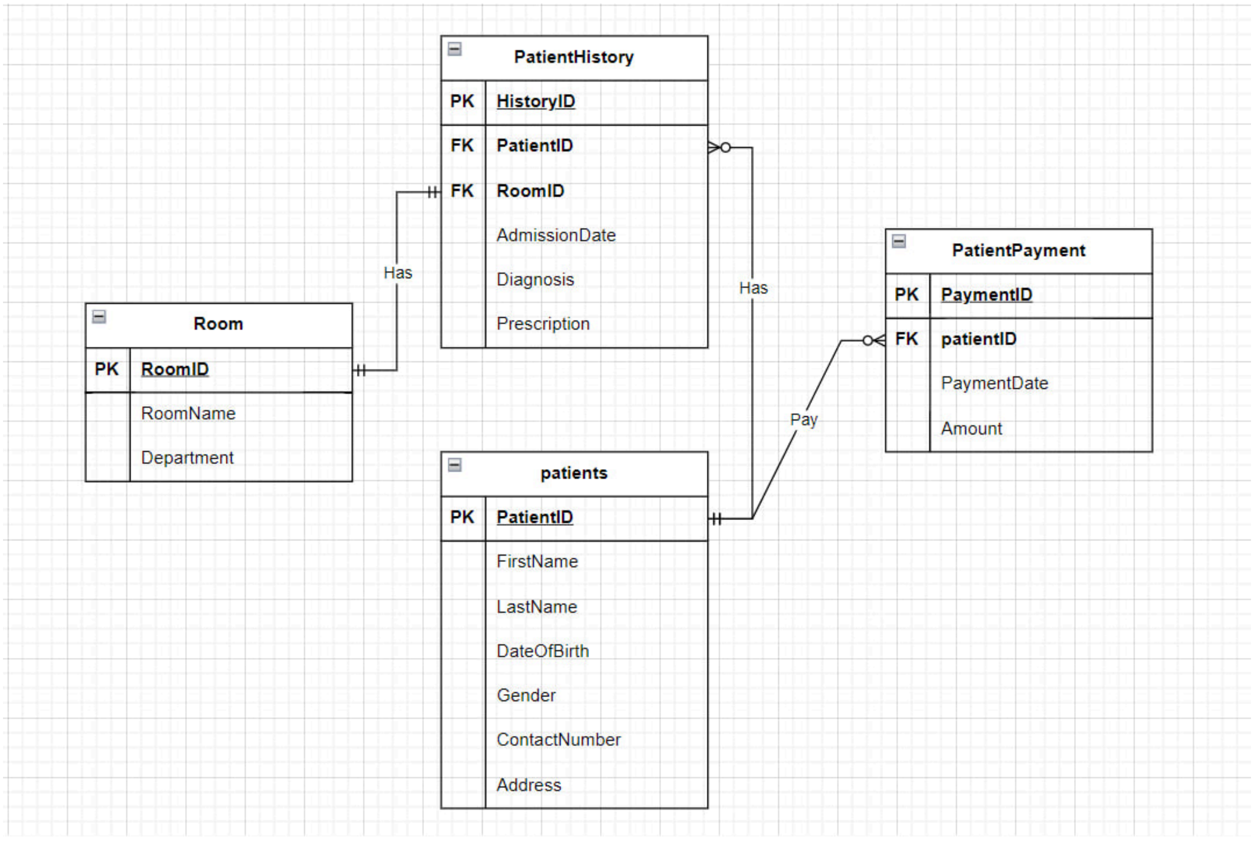
	User Requirement	Business Rule
01	**Appointment Module:** the ability to book patient visits with particular physicians. - Keep a record of every appointment's specifics, including the time, date, and medical information (status). - Assign a distinct AppointmentID to every appointment. - Using foreign keys, associate appointments with certain patients (patientID) and physicians (DoctorID).	**Primary Keys (PK):** The primary key in the appointment table is AppointmentID. FollowUpID is the principal key in the FollowUp table. The main key in the Insurance table is InsurancetID.
02	**Follow-Up Appointments:** - Keep track of any follow-up appointments related to initial appointments. - Keep track of information like the description, follow-up date, and doctor's details. - Make certain that every follow-up appointment has its own FollowUpID.	**Foreign Keys (FK):** To link appointments with patients and doctors, respectively, the appointment table employs the foreign keys PatientID and DoctorID. In order to create links between patients and appointments, the insurance table uses PatientID and AppointmentID as foreign keys.
03	**Insurance Information:** Organize appointment-related insurance data. - Keep a record of your insurance information, such as policy numbers, coverage details, and insurance companies. - Using foreign keys, associate insurance information with patients (PatientID) and appointments (AppointmentID).	**Data Type Constraints:** Establish unique data types for each entity's attributes to guarantee that only accurate data is kept.

2.4 Medical Record Module

	Requirements	Business rules
1	The system allows to create multiple medical records for different patients and it allows inputting details such as diagnosis, prescribed medications, and treatment history.	One patient can have zero or many medical records.Each medical record is linked to one patient. It allows multiple entries for prescriptions and treatment history in a single medical record. Access to medical records is restricted based on the principle of least privilege, ensuring only authorized personnel can view and modify patient data.
2	system allows adding, modifying, or removing prescriptions associated with medical records.	One medical record can have zero or many prescriptions. Each prescription is linked to only one medical record. Prescriptions must be associated with a valid medical record, and any changes made should be logged for accountability.
3	The module must integrate with the Appointments module, ensuring that medical records, prescriptions, and treatment history are linked to the corresponding appointments.	Each medical record, prescription, and treatment history entry is connected to one appointment. Integration with the Appointments module must be consistent and seamless, ensuring that all relevant data is linked accurately.
4	Users must be able to input and manage treatment records, including the date, description, prescribing doctor, and status (ongoing or completed).	Each treatment record is associated with one medical record. The status of treatments (ongoing or completed) should be updated promptly to reflect the current state of patient care.
5	The system should allow healthcare providers to easily navigate , review and input patient medical records.	All data entered into the system must adhere to accuracy standards to ensure reliable patient information.

3.0 Entity relationship modeling

3.1 Patient Module



Explanation:

The Entity-Relationship Diagram for the **Patient** module consists of the four entities that have the functionality of work for the patients to store and manage their personal details, medical history and so on. These four tables are 'Patients', 'PatientHistory', 'PatientPayment', and 'Room'.

We can see from the above diagram that the first one is the '**patients**' table where PatientID is the primary key. Every patient will be given a unique ID that will be generated by the system while the other attributes are some personal information of the patients for the proper identification. These are the patient's first and last name, date of birth, gender and contact details and address.

The second entity of this module is the '**PatientHistory**' entity that stores the patients' clinical history. This entity works as a record for storing and analyzing the diagnostic lab report of the patients. HistoryID is the primary key that is unique for every patient and PatientID and RoomID are the foreign keys of this entity. The rest of the attributes are AdmissionDate, Diagnosis and Prescription.

The third entity of this module is '**PatienPayment**' which is all about billing and payment of the patients. This entity manages and tracks the payment details of the patients where PaymentID is the primary key and PatientID is the foreign key. The other two attributes are PaymentDate and Amount. This entity also works for generating invoices for the payment done and paid.

The fourth and last entity of this module is '**Room**' that consists of data of the room details and department where the patient is admitted. Here, RoomID is the primary key that is a unique room number that is assigned differently for every room of the department. The other two attributes are RoomName and Department.

Relationships:

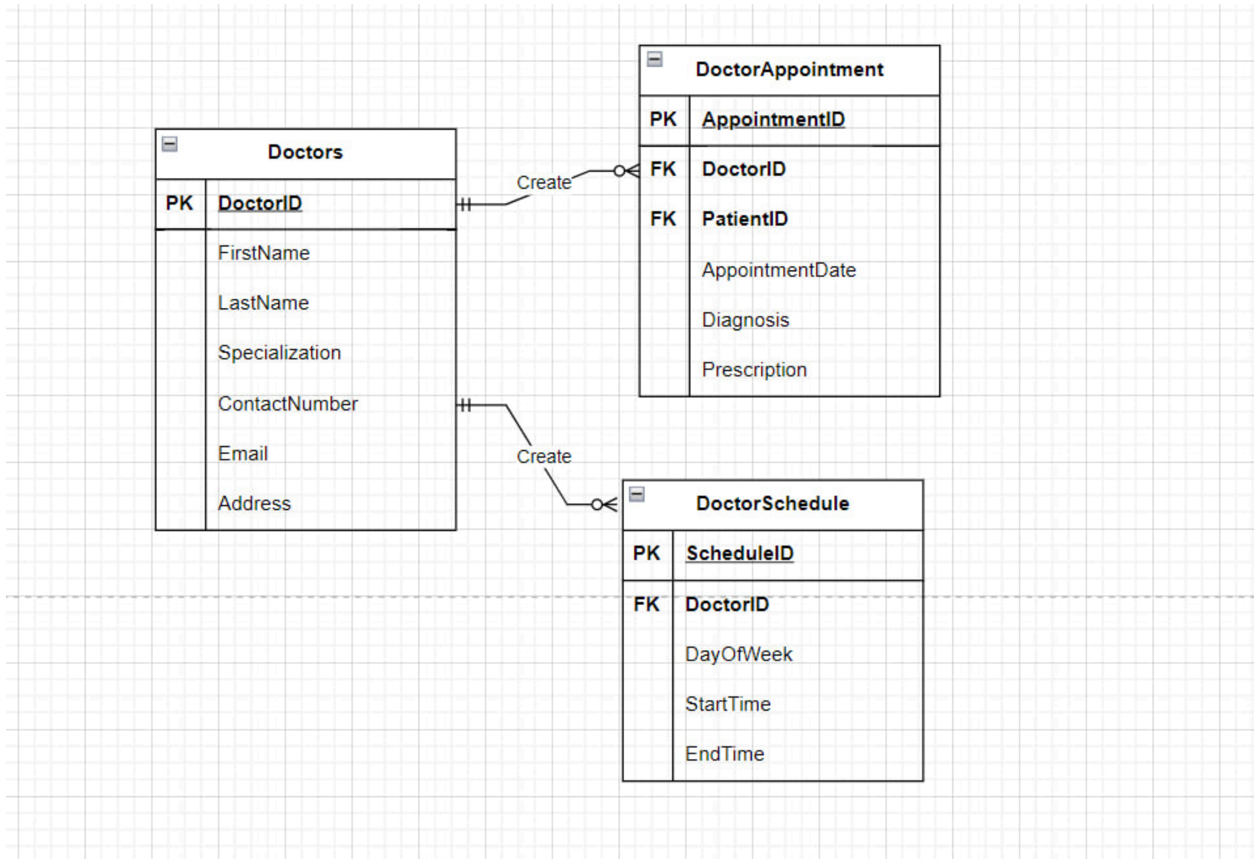
- Patients and PatientHistory holds a one to many relation where each patient can have multiple entries in the PatientHistory table, as indicated by the foreign key relationship between the \`PatientID\` in the Patients table and the \`PatientID\` in the PatientHistory table.

- The relationship between Patients and PatientPayments entities is one to many relation. Similar to PatientHistory, each patient can have multiple payment records in the PatientPayments table. The \`PatientID\` in the Patients table establishes this relationship with the \`PatientID\` in the PatientPayments table, enabling multiple payment records for a single patient.

Constraints:

- Primary Key constraint: The PatientID serves as a primary key, ensuring that every patient holds a distinct identity within the table. Similarly, the HistoryID, functioning as the primary key, secures uniqueness for each historical record. PaymentID, operating as the primary key, maintains the uniqueness of every payment record. Additionally, the RoomID, designated as a primary key, assures that each room possesses a unique identifier, distinct from any other room within the dataset.
- Foreign Key constraint: The utilization of the PatientID as a foreign key linked to the PatientID within the Patients table guarantees the correlation of every historical record and payment entry with an established patient. This practice ensures that each record in the PatientHistory table and each payment in the PatientPayments table corresponds to an existing patient entry in the Patients table, thereby maintaining the integrity of patient-related data across different records and transactions.
- Data Type constraint: All the entities of the module have its particular data types. This will help to ensure that only valid values of this data type will be stored.
- NOT NULL Constraint: NOT NULL constraints in certain fields such as \`FirstName\` and \`LastName\` in the patients table have these NOT NULL constraints that allow us to ensure that this field is compulsory and must be filled.

3.2 Doctor Module



Explanation:

The Entity-Relationship Diagram for the doctors module consists of three entities that manages the doctors activities, these tables are: \`Doctors\`, \`DoctorAppointment\`, and \`DoctorSchedule\`.

In the **\`Doctors\`** table, DoctorID is the primary key. Every doctor will be given a unique ID (number) in the database. The rest of attributes in this entity will be general information about the doctor like the Doctors first name and last name, specialization, contact number, and address.

The second entity in this module is the **\`DoctorAppointment\`** entity, which holds records of the appointments that the doctor will make. The AppointmentID is the primary key in this table, ensuring every Appointment is unique. The table has other two foreign keys that reference other tables, For example: The DoctorID is a foreign key referencing the Doctors table and PatientID is a foreign key referencing a patient table (from the patient module). The rest of attributes hold more details about the appointment like the date, the diagnosis, and prescription if there is.

The third entity in this module is the **\`DoctorSchedule\`** entity, Which stores the working hours of each doctor in the hospital. The scheduleID is the primary key in this table ensuring to have a uniquely ordered schedule for each doctor. The table has DoctorID as a foreign key that references the Doctor table. The other attributes of the tale include information about the schedule timing, like the working day, start time, and end time.

Relationships:

-The `Doctor` entity has a relationship with the `DoctorAppointment` entity. A Doctor can make many or zero appointments. Therefore, the relationship between them is one to many.

-The Doctor entity can be entitled to multiple working schedules. A Doctor can be assigned one or more schedules everyday, but a schedule only belongs to one doctor. Therefore, the relationship is one to many.

Constraints:

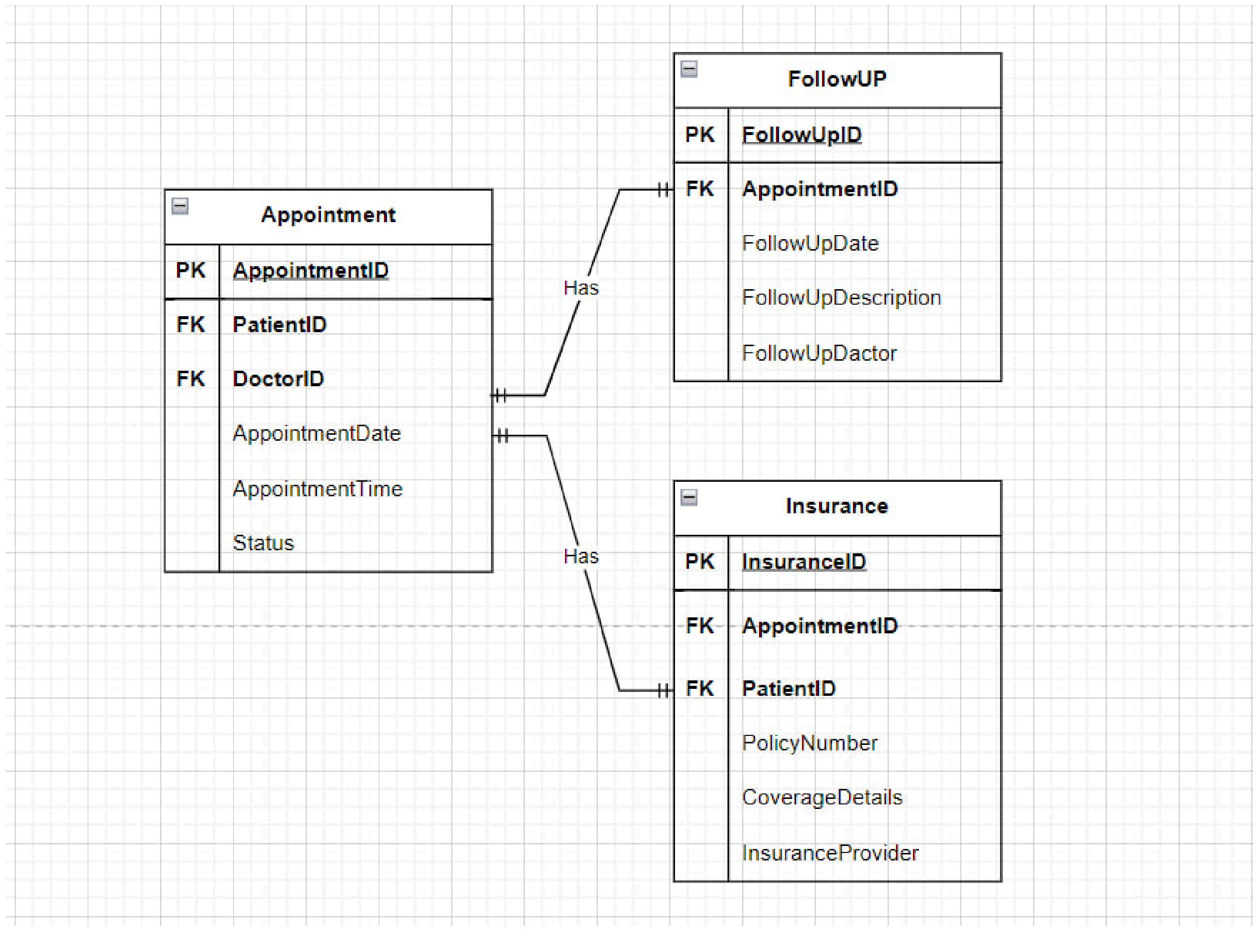
-Primary Key constraint: The DoctorID in the doctor table serves as a primary key which ensures that every doctor will have a unique ID to identify them. This constraint prevents having duplicated Doctor IDs. Furthermore, In the DoctorAppointment table The AppointmentID is the primary key ensuring the uniqueness of each appointment. The ScheduleID is also a primary key that serves as a way to ensure the uniqueness of the schedule.

-Foreign key constraint: Foreign keys are used in the three tables in this module for the purpose of referencing. Referencing ensures that there's a link or a connection between the three tables. For example, The doctorID attribute is a foreign key in the DoctorAppointment table referencing a Doctor. This constraint establishes a link between appointments and the corresponding doctor.

-Data type Constraint: Each entity has its specific data types. This will help to ensure that only valid values of this data type will be stored.

-NOT NULL Constraint: NOT NULL constraints in certain fields such as `FirstName` and `LastName` in the Doctor's table have these NOT NULL constraints that allow us to ensure that this field is compulsory and must be filled.

3.3 Appointment Module



Explanation:

The Entity-Relationship Diagram for theAppointment module consists of three entities that manages the Appointment activities, these tables are: `Appointment `, `FollowUp`, and `Insurance`.

In the Appointment table, AppointmentID is the primary key. Every appointment will be given a unique ID (AppointmentID) in the database. patientID and DoctorID serve as foreign keys. The patientID foreign key connects each patient to a specific appointment . The DoctorID foreign key connects each appointment to a specific doctor, ensuring data consistency. Attributes like appointmentDate, appointmentTime and status capture relevant medical information.

The second entity in this module is the `**Insurance**` entity, which holds records of the Insurance. The InsurancetID is the primary key in this table, ensuring every data redundancy . The table has other two foreign keys that reference other tables, For example: The PatientID is a foreign key referencing the patient table and AppointmentID is a foreign key referencing a Appointment table (from the Appointment module). The rest of attributes hold more details about the Insurance like Policy number, coverage details and insurance provider .

The third entity in this module is the `**FollowUP**` entity, Which stores the followup appointment details. The followUpID is the primary key in this table ensuring to have a uniqueness of data. The table has AppointmentID as a foreign key that references the Appointment table. The other attributes of the table include information about the FollowUp date , description and doctor .

Relationships:

-Appointment to FollowUp (AppointmentID): One-to-one relationship, as one Appointment can have only one FollowUp .

-Appointment to Insurance (AppointmentID):one-to-one, indicating that one Appointment can be associated with one insurance source .

Constraints:

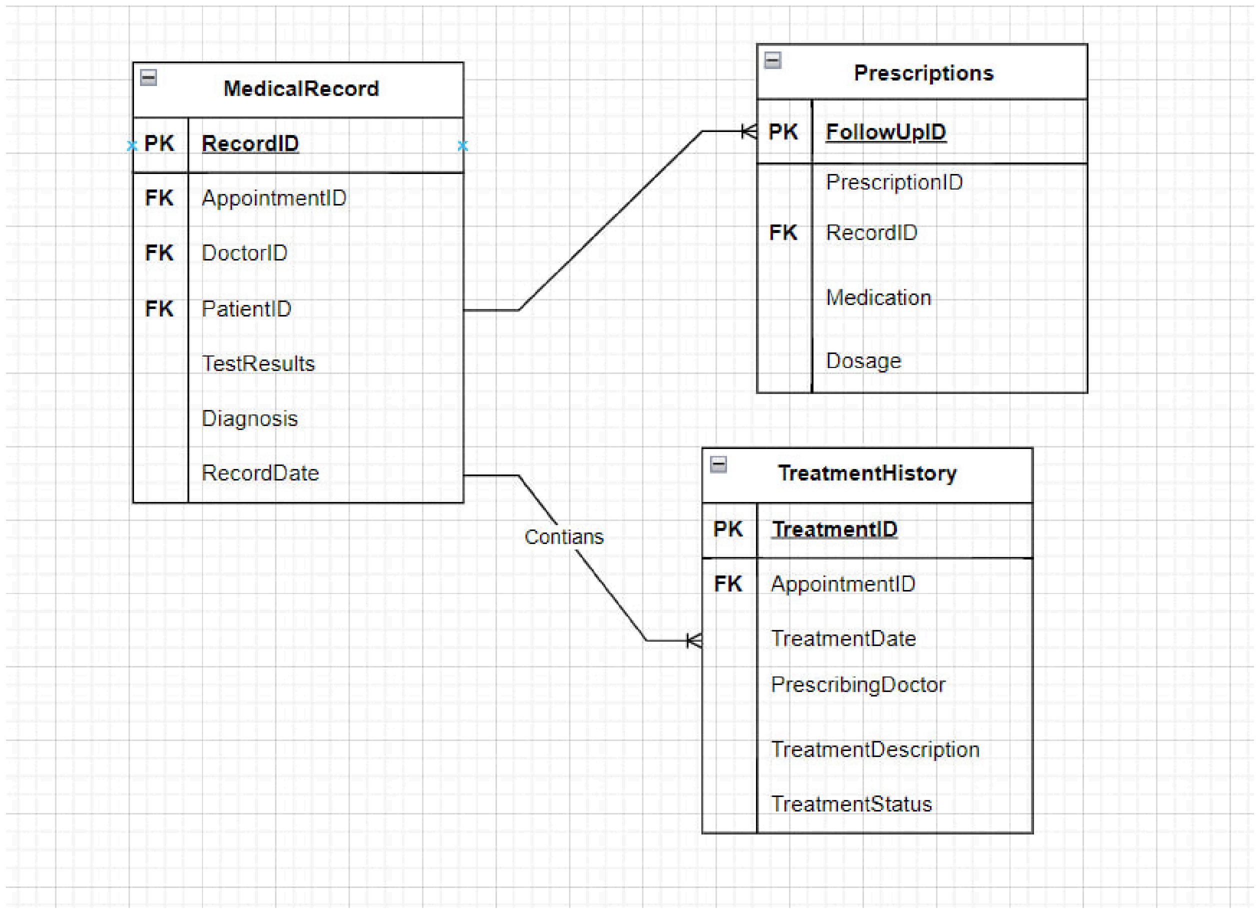
-Primary Keys constraint (PK): primary key uniquely identifies a record within a table. Each entity (table) has a primary key, such as AppointmentID in the Appointment entity, followUpID in the followUp entity, and InsurancetID in the Insurance entity. This constraint prevents data redundancy .

-Foreign Keys (FK): A foreign key establishes a link between tables by referencing the primary key of another table. In the Appointment entity, foreign keys include PatientID and DoctorID. In the Insurance entity, the foreign keys are PatientID and AppointmentID .

-Data type Constraint: Each entity has its specific data types. This will help to ensure that only valid values of this data type will be stored.

-NOT NULL Constraint: NOT NULL constraints in certain fields such as `FirstName` and `LastName` in the Doctor's table have these NOT NULL constraints that allow us to ensure that this field is compulsory and must be filled.

3.4 Medical Record Module



Explanation:

The Entity-Relationship Diagram (ERD) for the Medical Record Module follows the Crow's Foot notation and consists of three interconnected entities (tables) : **MedicalRecords** , **Prescriptions**, and **TreatmentHistory**.

In the MedicalRecords entity (table) : RecordID is the primary key, uniquely identifying each medical record. TreatementID , PatientID and DoctorID serve as foreign keys. The TreatmentID foreign key connects each medical record to a specific treatment history, ensuring data consistency. Attributes like RecordDate, Diagnosis, Prescription, and TestResults capture relevant medical information.

The Prescriptions entity (table) features PrescriptionID as the primary key, and a RecordID foreign key establishes a one-to-many relationship with the MedicalRecords table, indicating that a medical record may have multiple prescriptions. Attributes such as FollowUpID , RecordID , Medication and Dosage store details about prescribed medications.

The TreatmentHistory entity (table) uses TreatmentID as the primary key, and an AppointmentID foreign key links to the Appointments table. Attributes like TreatmentDate, TreatmentDescription, PrescribingDoctor, and TreatmentStatus capture information about treatment records.

Relationships:

MedicalRecords to Prescriptions (RecordID): One-to-many relationship, as one medical record can have multiple prescriptions . But each prescription can be related to one medical record

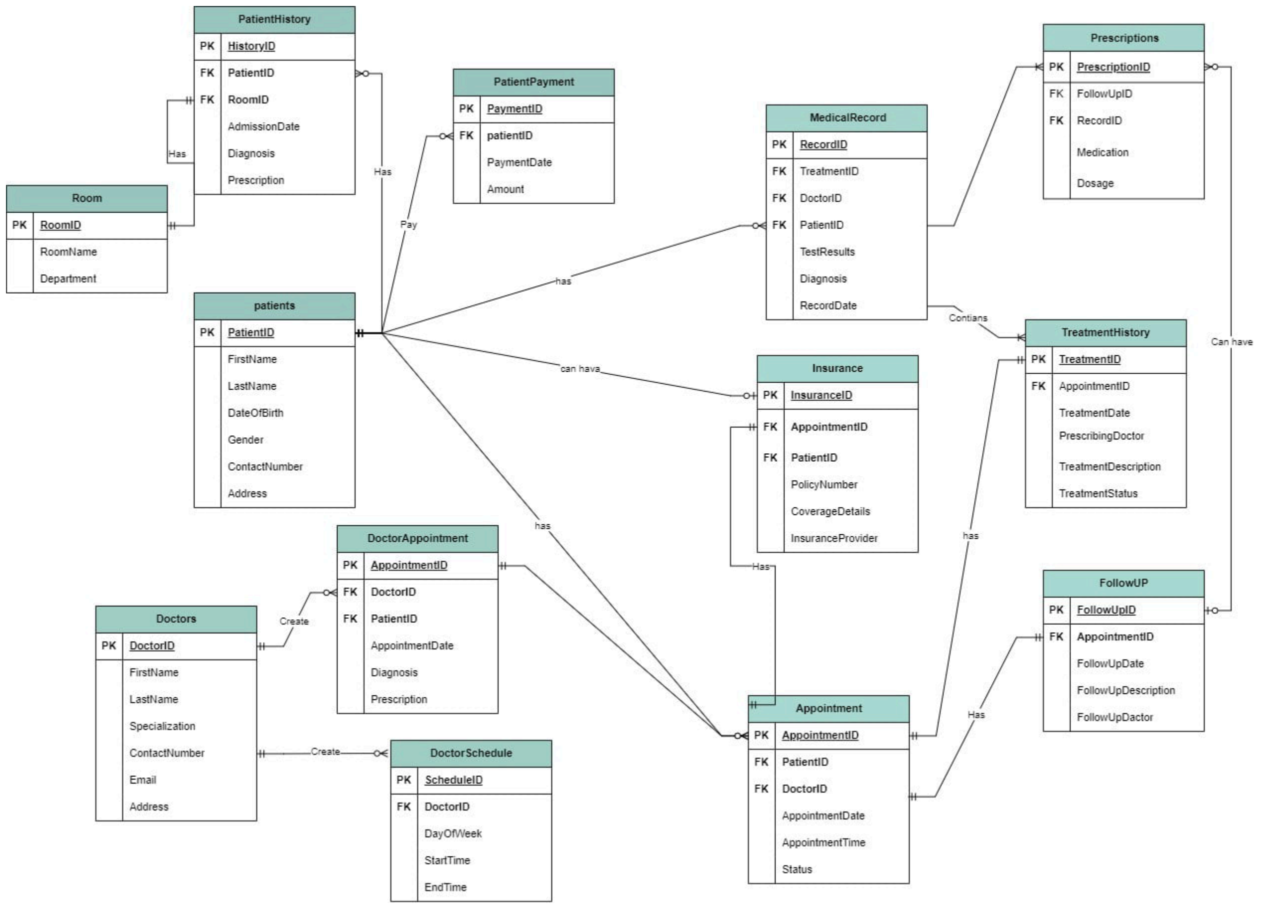
MedicalRecords to TreatmentHistory(TreatmentID):one-to-many, indicating that one medical record can be associated with multiple treatment history entries.

Keys constraints :

-Primary Keys (PK):A primary key uniquely identifies a record within a table. Each entity (table) has a primary key, such as RecordID in the MedicalRecords entity, PrescriptionID in the Prescriptions entity, and TreatmentID in the TreatmentHistory entity.

-Foreign Keys (FK): A foreign key establishes a link between tables by referencing the primary key of another table. In the MedicalRecords entity, foreign keys include TreatmentID, PatientID, and DoctorID. In the Prescriptions entity, the foreign key is RecordID, linking to the MedicalRecords table.

3.5 Hospital management database whole modules



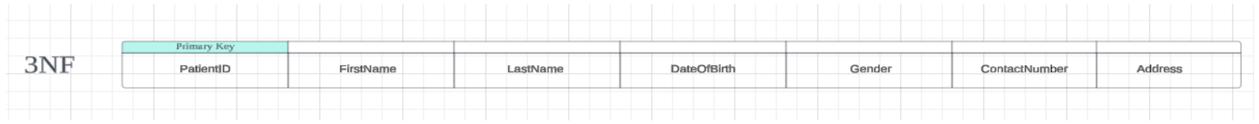
Explanation:
The whole hospital management database is a combination of the 4 modules containing 13 entities and 15 relationships.

Relationships between modules:
- **Patient- Insurance** (one to one relationship): Every patient has optionaly one insurance.
- Patient- Appointment (one to many): One patient can have one or many appointments.
- Patient- MedicalReport (one to many): Every patient has one or many medical records.
- DoctorAppointment- Appointment (one to many): Doctor can record one or many appointments.
- Appointment- TreatmentHistory (one to one): Every appointment will be associated with a treatment history.
- **FollowUp - Prescriptions** (One to many): Every follow up can have or many prescriptions.

4.0 Normalization

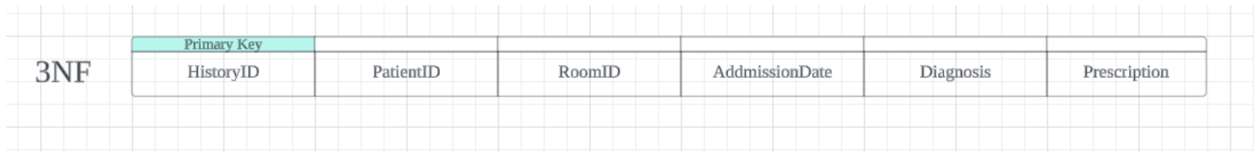
4.1 Normalization of Patient Module

4.1.1 Patients Table



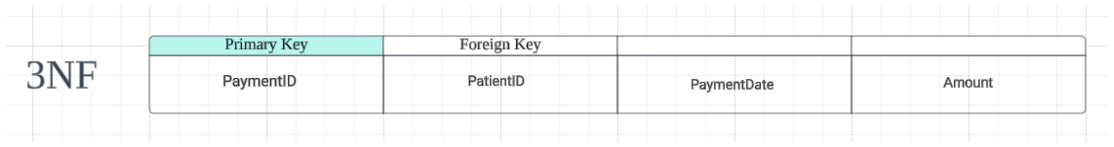
Primary Key						
PatientID	FirstName	LastName	DateOfBirth	Gender	ContactNumber	Address

4.1.2 PatientsHistory Table



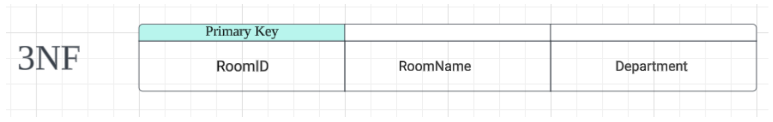
Primary Key					
HistoryID	PatientID	RoomID	AddmissionDate	Diagnosis	Prescription

4.1.3 PatientsPayment Table



Primary Key	Foreign Key		
PaymentID	PatientID	PaymentDate	Amount

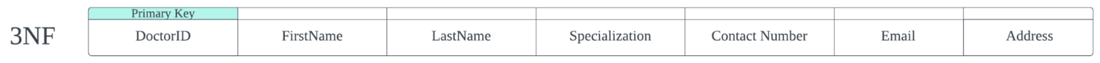
4.1.4 Room Table



Primary Key		
RoomID	RoomName	Department

4.2 Normalization of Doctor Module

4.2.1 Doctors Table

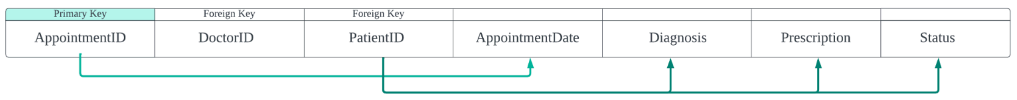


Primary Key						
DoctorID	FirstName	LastName	Specialization	Contact Number	Email	Address

4.2.2 DoctorAppointments table

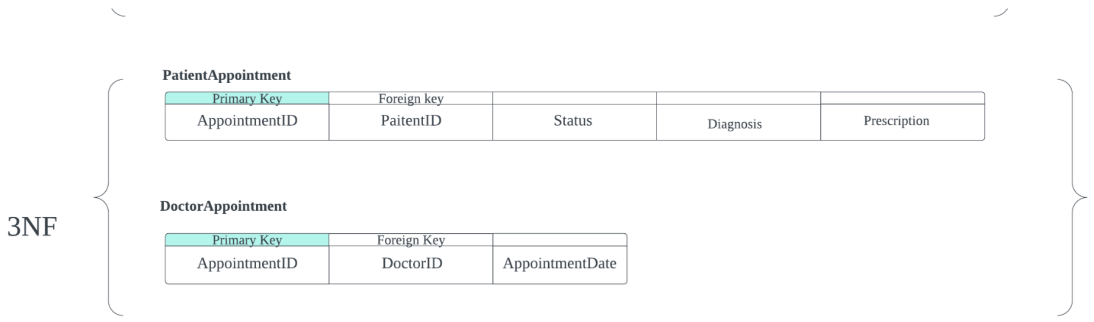
2NF:

The tables have a primary key and all the values are atomic.



Primary Key	Foreign Key	Foreign Key				
AppointmentID	DoctorID	PatientID	AppointmentDate	Diagnosis	Prescription	Status

3NF:



3NF

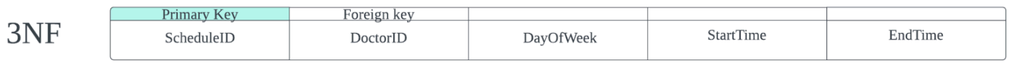
PatientAppointment

Primary Key	Foreign key			
AppointmentID	PaitentID	Status	Diagnosis	Prescription

DoctorAppointment

Primary Key	Foreign Key	
AppointmentID	DoctorID	AppointmentDate

4.2.3 DoctorSchedule table

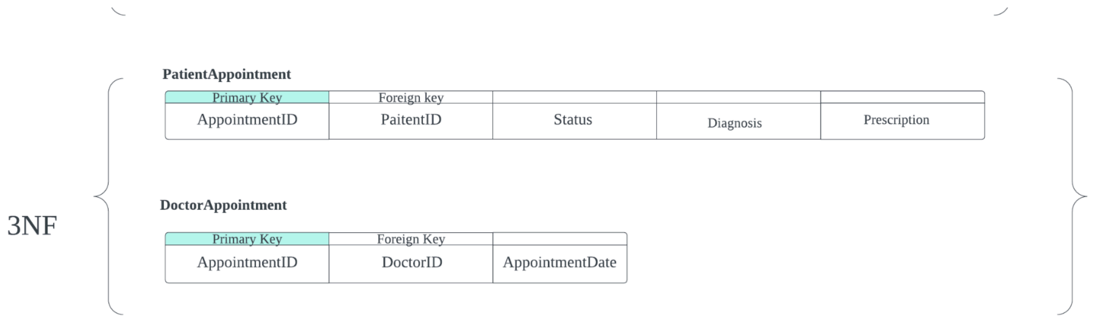


3NF

Primary Key	Foreign key			
ScheduleID	DoctorID	DayOfWeek	StartTime	EndTime

4.3 Normalization of Appointment Module

4.3.1 Patient Appointment table



3NF

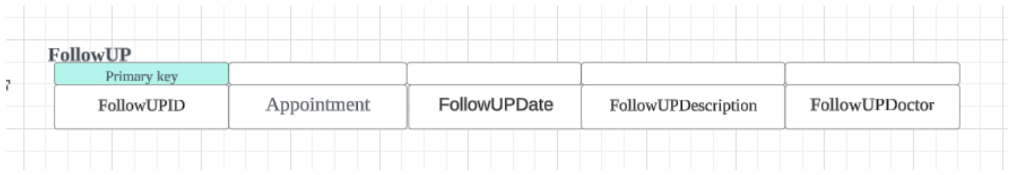
PatientAppointment

Primary Key	Foreign key			
AppointmentID	PaitentID	Status	Diagnosis	Prescription

DoctorAppointment

Primary Key	Foreign Key	
AppointmentID	DoctorID	AppointmentDate

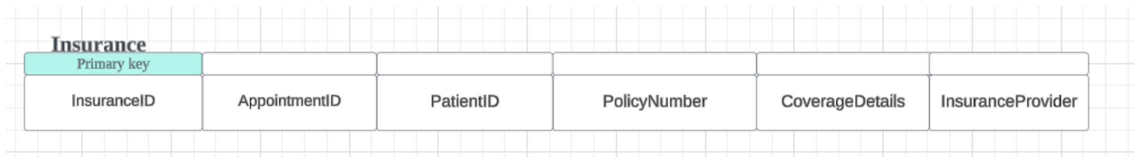
4.3.2 Follow UP table



FollowUP

Primary key				
FollowUPID	Appointment	FollowUPDate	FollowUPDescription	FollowUPDoctor

4.3.3 Insurance Table

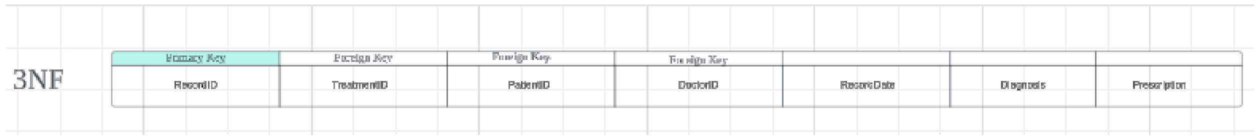


Insurance

Primary key					
InsuranceID	AppointmentID	PatientID	PolicyNumber	CoverageDetails	InsuranceProvider

4.4 Normalization of Medical Record Module

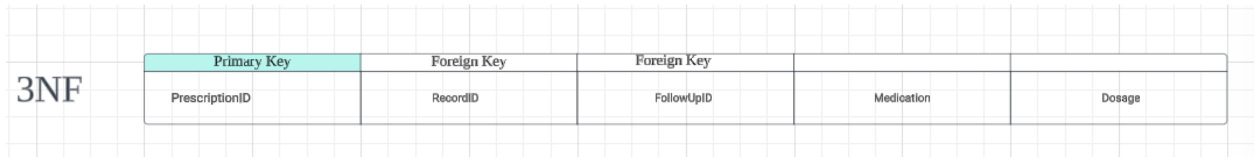
4.4.1 Medical records Table



3NF

Primary Key	Foreign Key	Foreign Key	Foreign Key			
RecordID	TreatmentID	PatientID	DoctorID	RecordDate	Diagnosis	Prescription

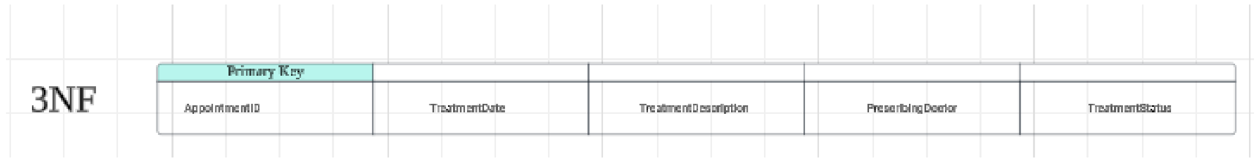
4.4.2 Prescription Table



3NF

Primary Key	Foreign Key	Foreign Key		
PrescriptionID	RecordID	FollowUpID	Medication	Dosage

4.4.1 Treatment History Table



3NF

Primary Key				
AppointmentID	TreatmentDate	TreatmentDescription	PrescribingDoctor	TreatmentStatus

5.0 Data dictionary

5.1 Patient Module

Table Name	Attributes	Datatype	Format	Null able	PK or FK	Reference Table
Patients	PatientID	INT	1	No	PK	

	FirstName	VARCHAR(50)	Rahim	No		
	LastName	VARCHAR(50)	Mia	No		
	DateOfBirth	DATE	15-09-2002	No		
	Gender	VARCHAR(15)	Male	Yes		
	ContactNumber	INT(50)	+603857289	No		
	Address	VARCHAR(100)	Kedah, jalan tun razak, alor setar 05200.	yes		
PatientHistory	HIstoryID	INT	50	No	PK	
	PatientID	INT	1	No	FK	Patients
	RoomID	INT	203	No	FK	
	AdmissionDate	DATE	01/12/2014	No		
	Diagnosis	VARCHAR(50)	Ultrasonogram	No		
	Prescription	VARCHAR(50)	Bed Rest	No		
PatientPayment	PaymentID	INT	45	No	PK	
	PatientID	INT	1	No	FK	Patients

	PaymentDate	DATE	01/12/2014	No		
	Amount	INT	4000	No		
Room	RoomID	INT	120	No	PK	PatientsHistory
	RoomName	VARCHAR(50)	Male Ward	No		
	Department	VARCHAR(50)	ENT	No		

5.2 Doctor Module

Table Name	Attributes	Datatype	Format	Null able	PK or FK	Reference Table
Doctors	DoctorID	INT	1	No	PK	
	FirstName	VARCHAR(50)	Ahmad	No		
	LastName	VARCHAR(50)	Ali	No		
	Specialization	VARCHAR(50)	dermatologist	No		
	ContactNumber	VARCHAR(15)	+603857294	Yes		
	Email	VARCHAR(50)	ahmad.ali@doctor.com	No		
	Address	VARCHAR(100)	Kedah, jalan tun razak, alor setar 05200.	yes		

DoctorAppointment	AppointmentID	INT	50	No	PK	
	DoctorID	INT	1	No	FK	Doctors
	AppointmentDate	DATETIME	01/12/2014	No		
DoctorSchedule	ScheduleID	INT	45	No		
	DoctorID	INT	1	No	FK	Doctors
	DayOfWeek	VARCHAR(15)	Sunday	No		
	StartTime	VARCHAR(6)	9:00am	No		
	EndTime	VARCHAR(6)	5:00pm	No		

5.3 Appointment Module

Table Name	Attributes	Datatype	Format	Null able	PK or FK	Reference Table
PatientAppointment	AppointmentID	INT	1029	No	PK	
	PatientID	INT	1	No	FK	Patient
	Status	Varchar(20)	Pending	Yes		
	Diagnosis	VARCHAR(20)	Cavity	Yes		

Table Name	Attributes	Datatype	Format	Null able	PK or FK	Reference Table
	Prescription	VARCHAR(100)	Mouthwash	Yes		
Insurance	InsuranceID	INT	10	NOT NULL	PK	
	AppointmentID	INT	10		FK	AppointmentI D
	PatientID	INT	10	NOT NULL	FK	Patients
	PolicyNumber	VARCHAR(30)	01/12/2014	No		
	CoverageDetails	VARCHAR(6)	100	No		
	InsuranceProvid er	VARCHAR(50)	50	No		

Table Name	Attributes	Datatype	Format	Null able	PK or FK	Reference Table
FollowUP	FollowUPID	INT	10	NOT NULL	PK	
	AppointmentI D	INT	10		FK	Appointmen tID
	FollowUPDat e	INT	02/07/2023	NOT NULL	FK	Date
	FollowUPDes cribetion	DATETI ME	100	No		
	FollowUPDoc tor	VARCHA R(20)	9:00am	No		

5.4 Medical Record Module

Table Name	Attributes	Datatype	Format	Null able	PK or FK	Reference Table
MedicalRecords	RecordID	INT		NOT NUll	PK	
	TreatmentID	INT			FK	Treatment History
	PatientID	INT		NOT NUL L	FK	Patients
	DoctorID	INT		NOT NUL L	FK	Doctors
	RecordDate	DATE	YYYY-MM-DD			
	Diagnosis	VARCHA R(255)				
	Prescription	VARCHA R(255)				
	TestResults	VARCHA R(255)				
Prescriptions	PrescriptionID	INT		NOT NUL L	PK	
	RecordID	INT		NOT NUL L	FK	MedicalRe cords
	FollowUpID	INT		NOT NUL L		

Table Name	Attributes	Datatype	Format	Nullable	PK or FK	Reference Table
	Medication	VARCHAR(100)				
	Dosage	VARCHAR(50)				
TreatmentHistory	TreatmentID	INT		NOT NULL	PK	
	AppointmentID	INT		NOT NULL	FK	Appointments
	TreatmentDate	DATE	YYYY-MM-DD			
	TreatmentDescription	VARCHAR(250)				
	PrescribingDoctor	VARCHAR(100)				
	TreatmentStatus	VARCHAR(50)				

6.0 Data Implementation

6.1 Patient Module

```
CREATE TABLE Patients (
   PatientID INT PRIMARY KEY,
   FirstName VARCHAR2(50),
   LastName VARCHAR2(50),
   DateOfBirth DATE,
   Gender VARCHAR2(15),
   ContactNumber VARCHAR2(15),
   Address VARCHAR2(100)
);

CREATE TABLE PatientHistory (
   HistoryID INT PRIMARY KEY,
   PatientID INT,
   RoomID INT,
   AdmissionDate DATE,
   Diagnosis VARCHAR2(50),
   Prescription VARCHAR2(50),
   FOREIGN KEY (PatientID) REFERENCES Patients(PatientID),
   FOREIGN KEY (RoomID) REFERENCES Rooms(RoomID) -- 'Room' should be 'Rooms' if that is the
name of the table.
);

CREATE TABLE PatientPayments (
   PaymentID INT PRIMARY KEY,
   PatientID INT,
   PaymentDate DATE,
   Amount NUMBER(10,2), -- Changed to NUMBER for currency representation.
   FOREIGN KEY (PatientID) REFERENCES Patients(PatientID)
);

CREATE TABLE Rooms (
   RoomID INT PRIMARY KEY,
   RoomName VARCHAR2(50),
   Department VARCHAR2(50)
);
```

6.2 Doctors Module

```
CREATE TABLE Doctors (
    DoctorID INT PRIMARY KEY,
    FirstName VARCHAR2(50),
    LastName VARCHAR2(50),
    Specialization VARCHAR2(50),
    ContactNumber VARCHAR2(15),
    Email VARCHAR2(50),
    Address VARCHAR2(100)
);

CREATE TABLE DoctorSchedules (
    ScheduleID INT PRIMARY KEY,
    DoctorID INT NOT NULL,
    DayOfWeek VARCHAR2(15) NOT NULL CHECK (DayOfWeek IN ('Sunday', 'Monday', 'Tuesday',
'Wednesday', 'Thursday', 'Friday', 'Saturday')),
    StartTime VARCHAR2(6), -- Use 'HH24:MI' format for time.
    EndTime VARCHAR2(6), -- Use 'HH24:MI' format for time.
    FOREIGN KEY (DoctorID) REFERENCES Doctors(DoctorID)
);

CREATE TABLE DoctorAppointments (
    AppointmentID INT PRIMARY KEY,
    DoctorID INT NOT NULL,
    AppointmentDate DATE NOT NULL,
    FOREIGN KEY (DoctorID) REFERENCES Doctors(DoctorID)
    -- Removed the reference to PatientAppointment, as it creates a circular reference.
);
```

6.3 Appointment Module

```
CREATE TABLE PatientAppointment (

AppointmentID INT PRIMARY KEY,

PatientID INT,

DoctorID INT,

AppointmentDate DATE,

AppointmentTime TIME,

Status VARCHAR(50),

);


CREATE TABLE FollowUp(
   FollowUpID INT PRIMARY KEY,
   AppointmentID INT,
   FollowUpDate DATE,
   FollowUpDescription VARCHAR(50),
   FollowUpDoctor VARCHAR(20),
 );
```

```
CREATE TABLE Insurances (
   InsuranceID INT PRIMARY KEY,
   AppointmentID INT,
   PatientID INT,
   PolicyNumber VARCHAR(50),
   CoverageDetails VARCHAR(255),
   InsuranceProvider VARCHAR(100),
  );
```

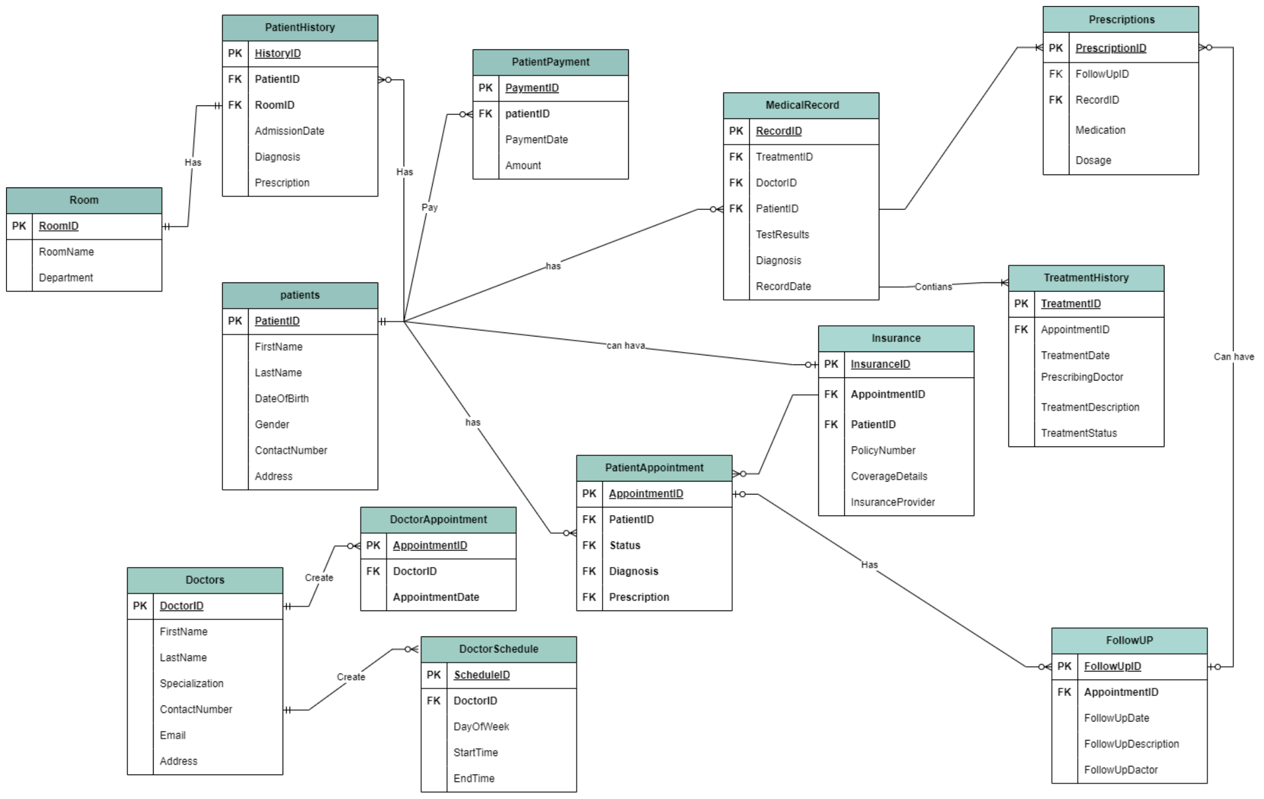
6.4 Medical Record Module

```
CREATE TABLE MedicalRecords (
   RecordID INT PRIMARY KEY,
   TreatmentID INT,
   PatientID INT,
   DoctorID INT,
   RecordDate DATE,
   Diagnosis VARCHAR2(255),
   Prescription VARCHAR2(255),
   TestResults VARCHAR2(255),
   FOREIGN KEY (TreatmentID) REFERENCES TreatmentHistory(TreatmentID),
   FOREIGN KEY (PatientID) REFERENCES Patients(PatientID),
   FOREIGN KEY (DoctorID) REFERENCES Doctors(DoctorID)
);

CREATE TABLE Prescriptions (
   PrescriptionID INT PRIMARY KEY,
   RecordID INT,
   FollowUpID INT,
   Medication VARCHAR2(100),
   Dosage VARCHAR2(50),
   FOREIGN KEY (RecordID) REFERENCES MedicalRecords(RecordID)
);

CREATE TABLE TreatmentHistory (
   TreatmentID INT PRIMARY KEY,
   AppointmentID INT,
   TreatmentDate DATE,
   TreatmentDescription VARCHAR2(255),
   PrescribingDoctor VARCHAR2(100),
   TreatmentStatus VARCHAR2(50),
   FOREIGN KEY (AppointmentID) REFERENCES PatientAppointments(AppointmentID)
);
```

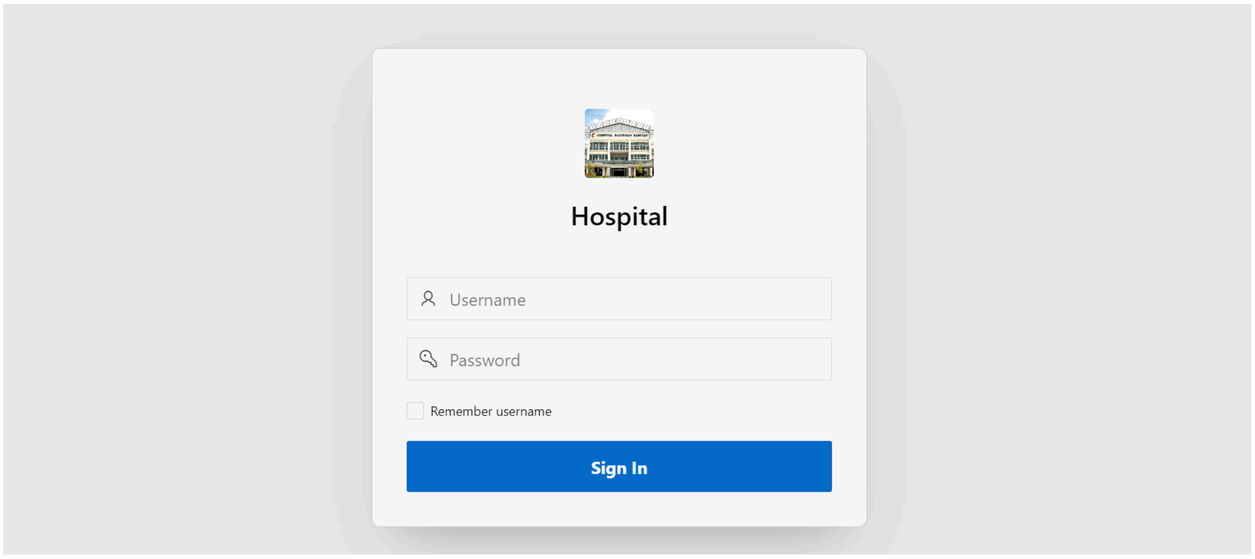
7.0 The Updated ERD



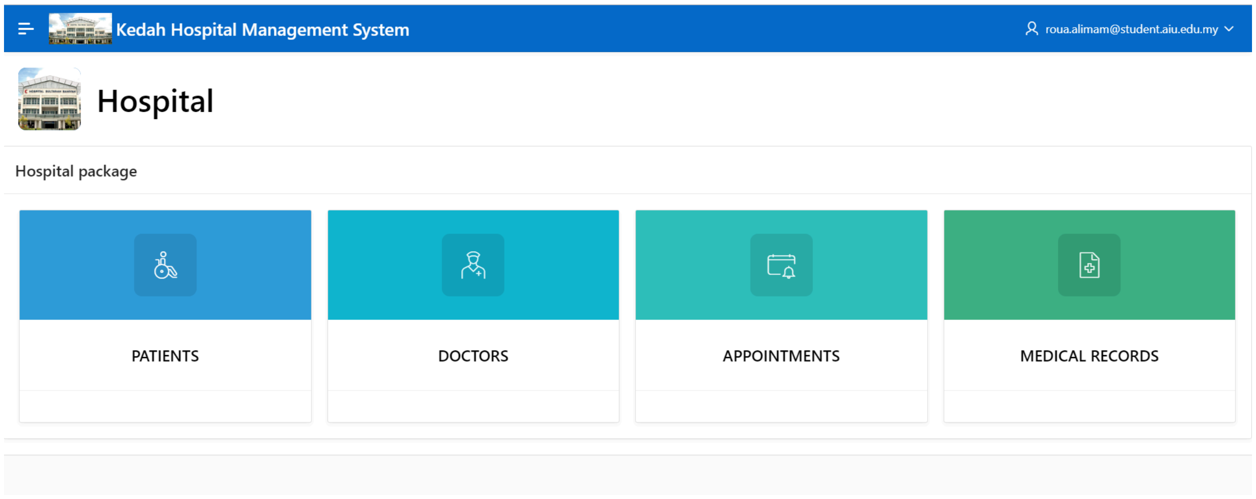
8.0 Front-End and system implementation

8.1 Generic Pages

Log_in page:

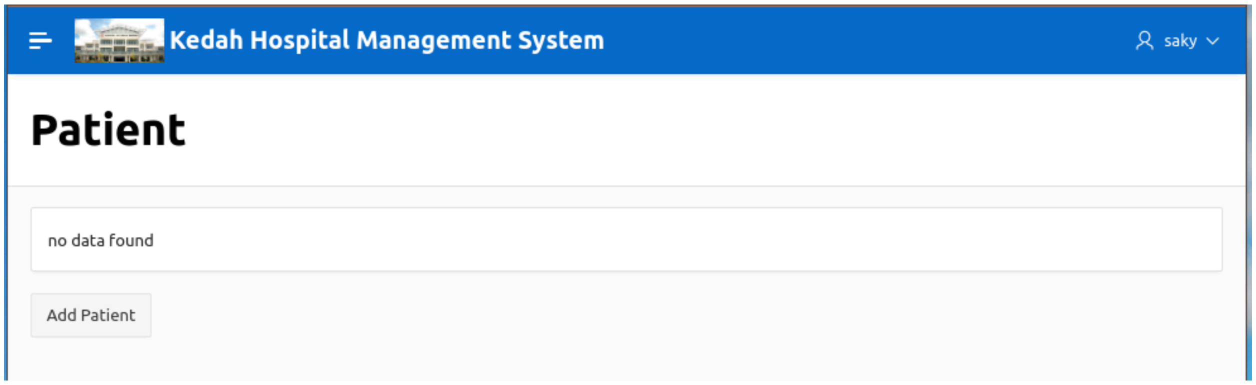


The home page:



8.2 Patient Module

The main Patient page where we can add patients, delete patients, and edit the patient information. That's in addition to viewing the insurance package for each patient.

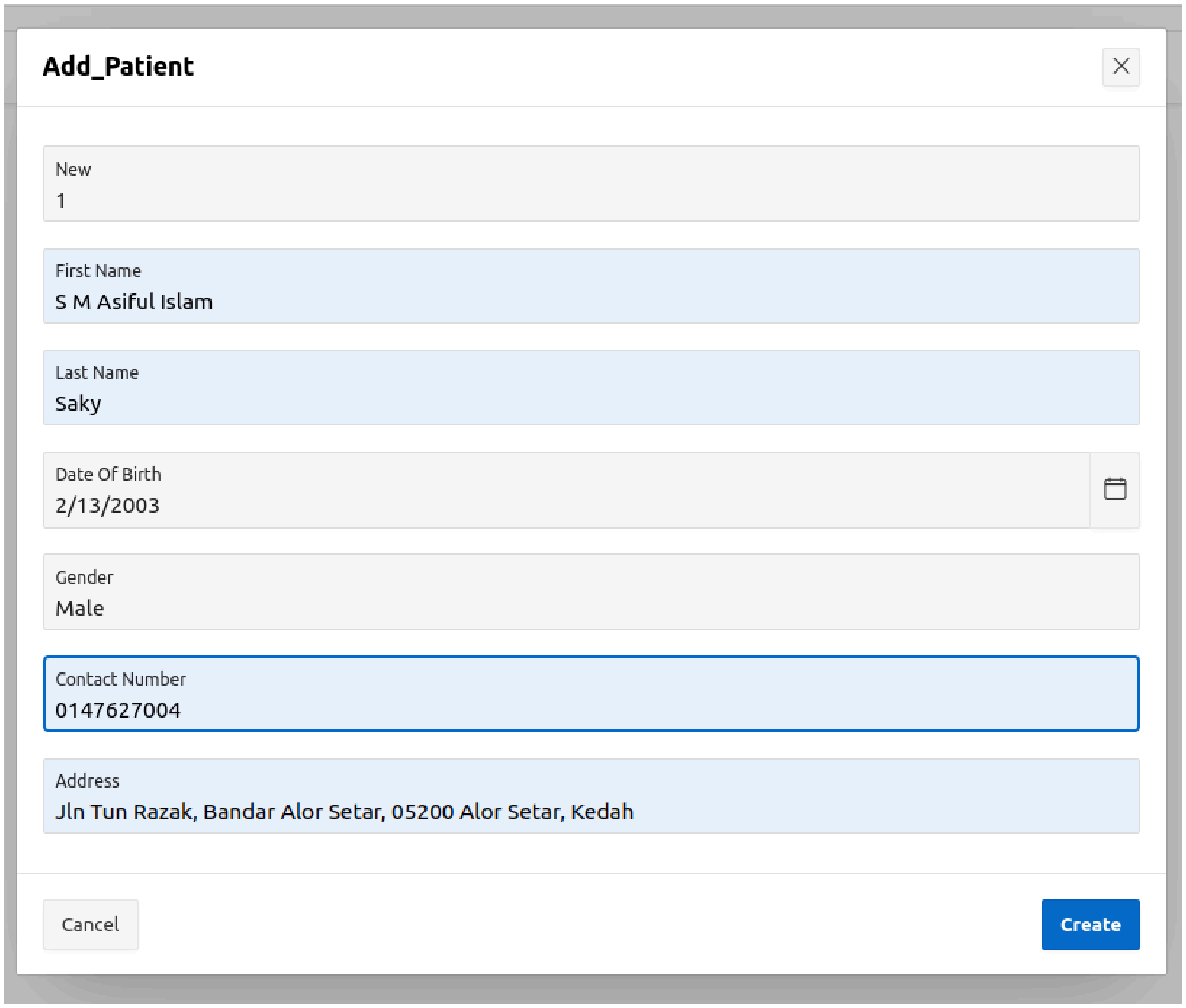

Kedah Hospital Management System

saky

Patient

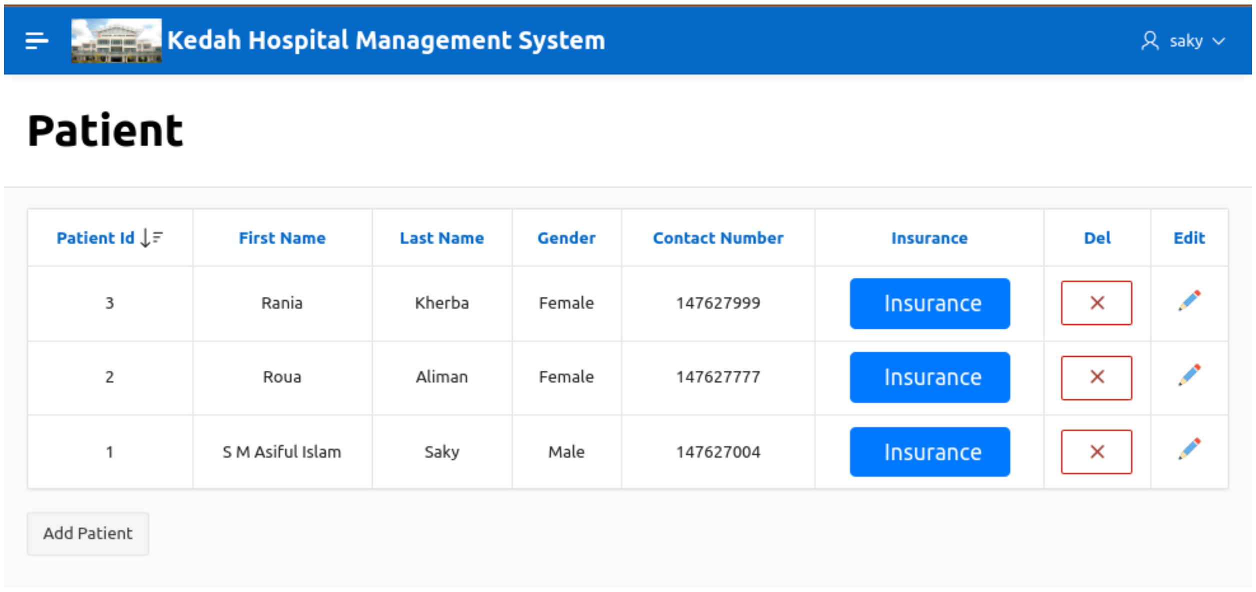
no data found

Figure 8.2.1 : The Patient Module with Zero Patient (initial state)


Add_Patient

New	1
First Name	S M Asiful Islam
Last Name	Saky
Date Of Birth	2/13/2003
Gender	Male
Contact Number	0147627004
Address	Jln Tun Razak, Bandar Alor Setar, 05200 Alor Setar, Kedah

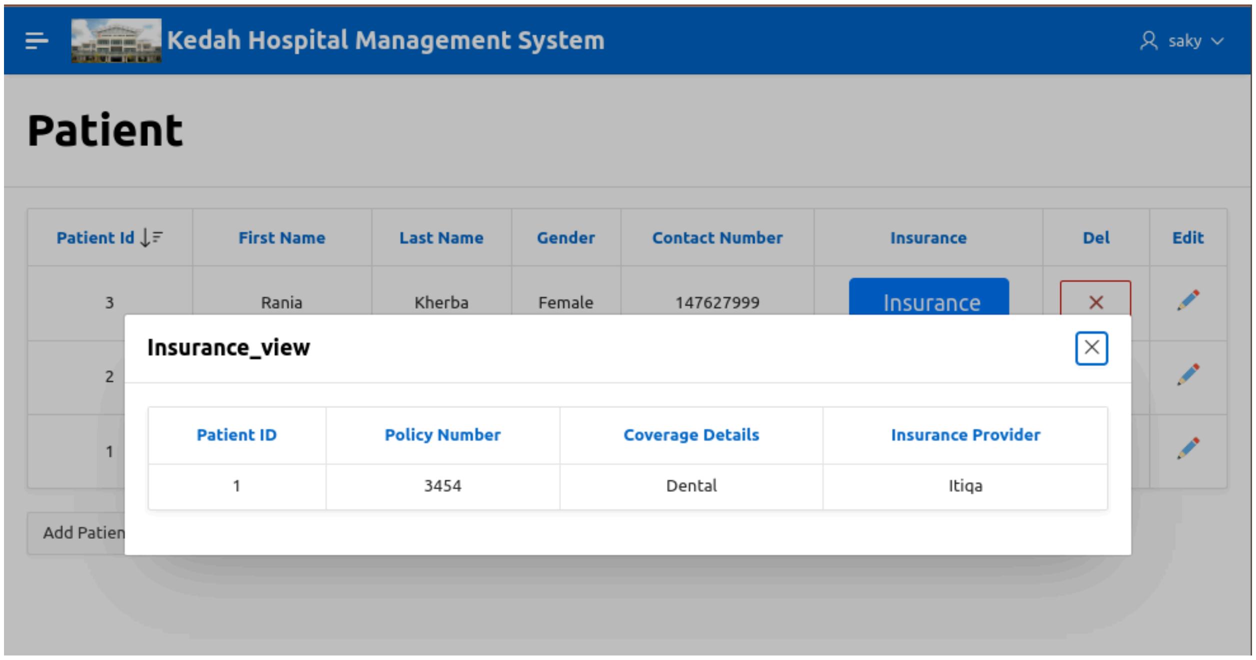
Figure 8.2.2 : Adding or registering a new patient.



Patient

Patient Id	First Name	Last Name	Gender	Contact Number	Insurance	Del	Edit
3	Rania	Kherba	Female	147627999	Insurance	✕	✎
2	Roua	Aliman	Female	147627777	Insurance	✕	✎
1	S M Asiful Islam	Saky	Male	147627004	Insurance	✕	✎

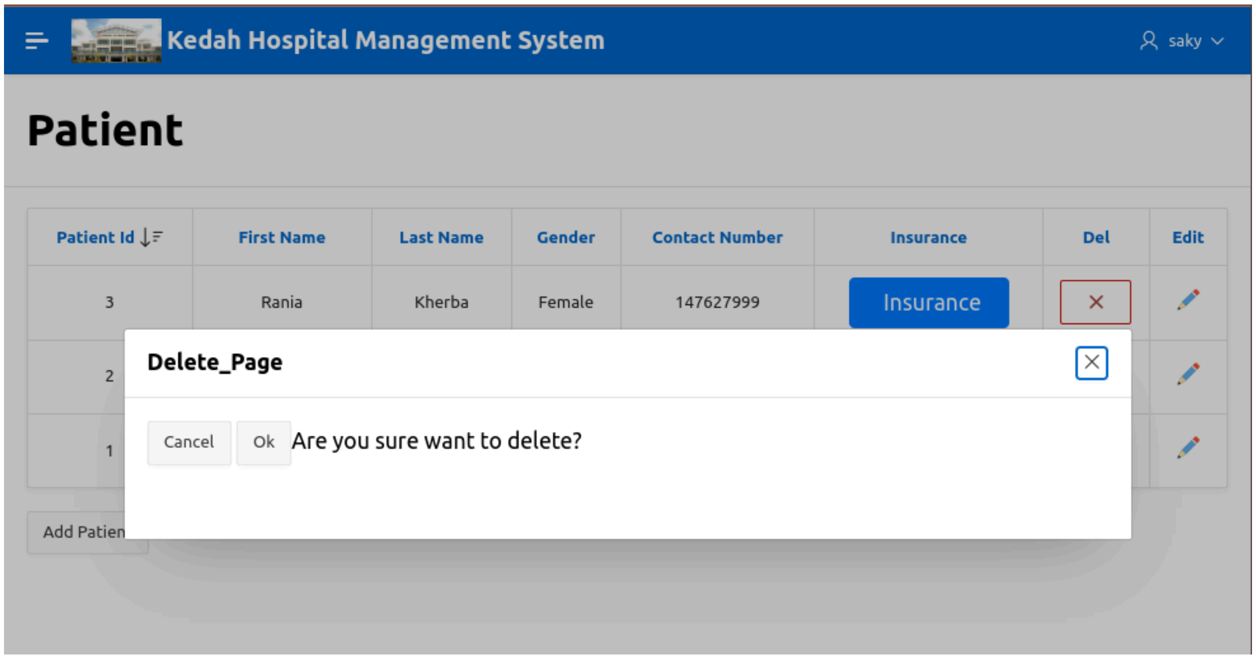
Figure 8.2.3 : After adding three patients.



Insurance_view

Patient ID	Policy Number	Coverage Details	Insurance Provider
1	3454	Dental	Itiqa

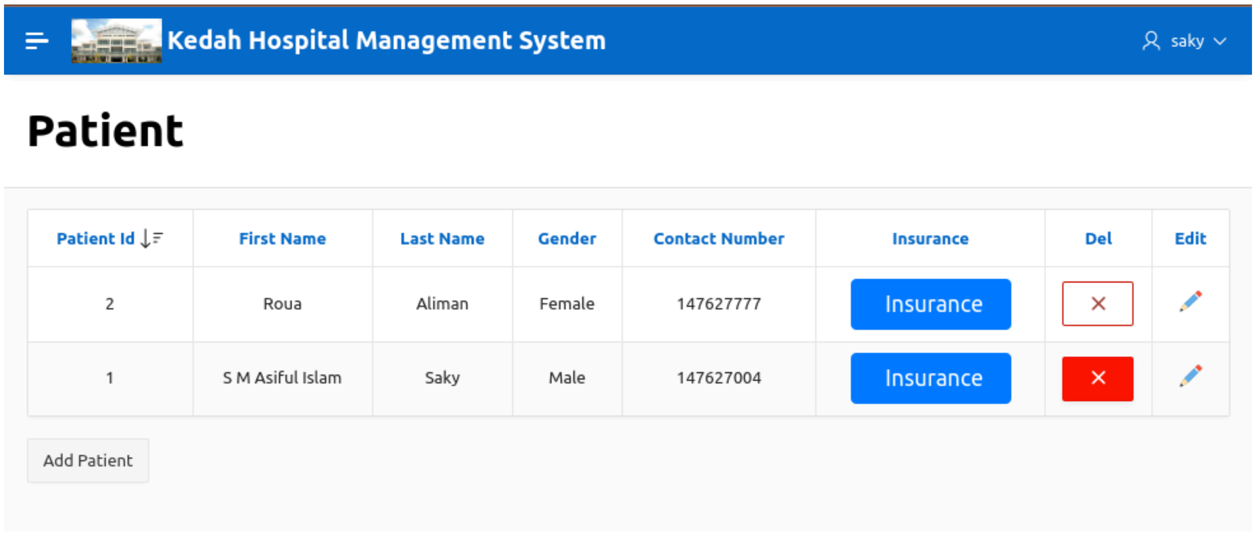
Figure 8.2.4 : Patients' insurance details



Patient Id	First Name	Last Name	Gender	Contact Number	Insurance	Del	Edit
3	Rania	Kherba	Female	147627999	Insurance	✕	✎
2							✎
1							✎

Are you sure want to delete?

Figure 8.2.5 : Deleting or removing a patient

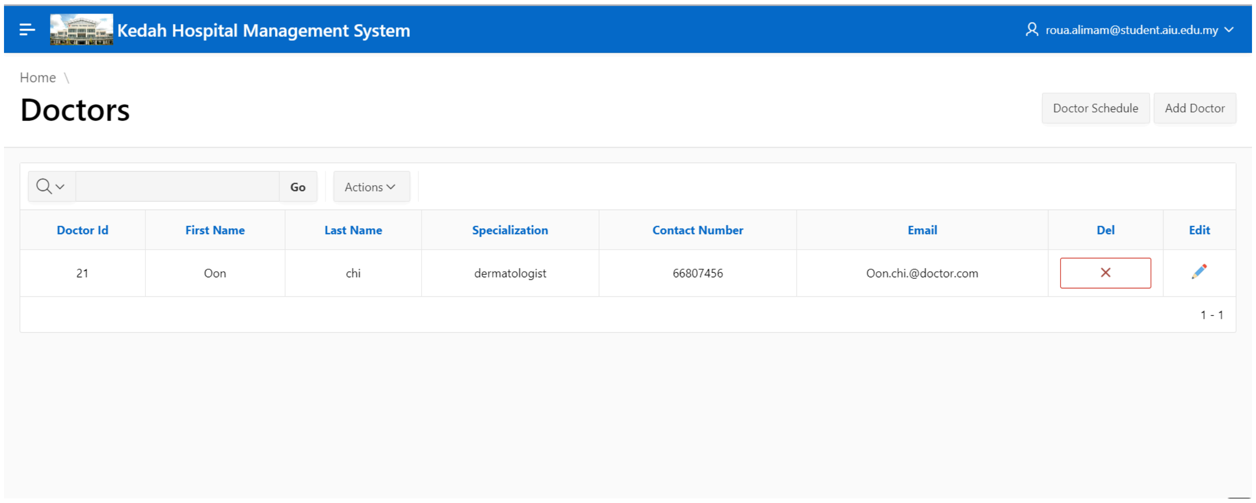


Patient Id	First Name	Last Name	Gender	Contact Number	Insurance	Del	Edit
2	Roua	Aliman	Female	147627777	Insurance	✕	✎
1	S M Asiful Islam	Saky	Male	147627004	Insurance	✕	✎

Figure 8.2.6 : After removing a patient

8.2 Doctors Module

The main Doctor list page. It includes Crud operations and functionalities. Here, We can add a new doctor, Delete a doctor record, edit Doctor's information, and navigate to Doctor schedule.

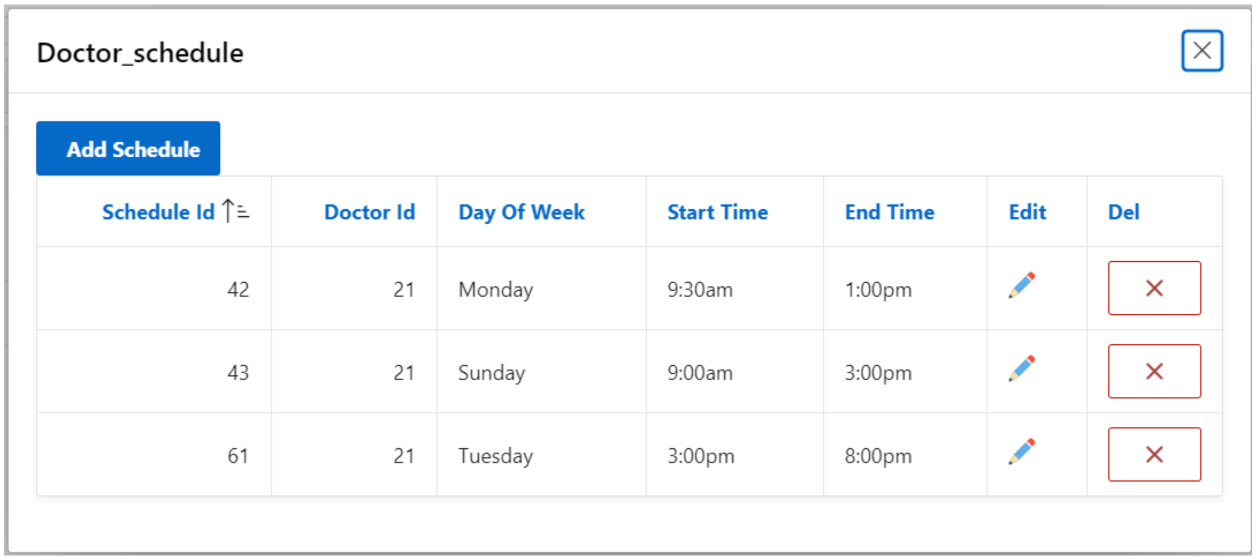

Kedah Hospital Management System
roua.alimam@student.aiu.edu.my
Home \
Doctor Schedule
Add Doctor
Go
Actions

Doctors

Doctor Id	First Name	Last Name	Specialization	Contact Number	Email	Del	Edit
21	Oon	chi	dermatologist	66807456	Oon.chi.@doctor.com	✕	✎

1 - 1

Here We can view, edit, and delete the doctor schedule.

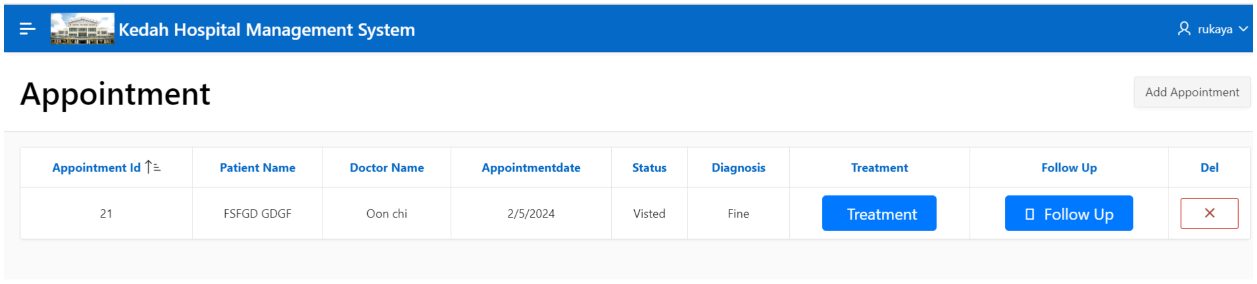


Doctor_schedule

Add Schedule

Schedule Id	Doctor Id	Day Of Week	Start Time	End Time	Edit	Del
42	21	Monday	9:30am	1:00pm	✎	✕
43	21	Sunday	9:00am	3:00pm	✎	✕
61	21	Tuesday	3:00pm	8:00pm	✎	✕

8.3 Appointment Module



Kedah Hospital Management System

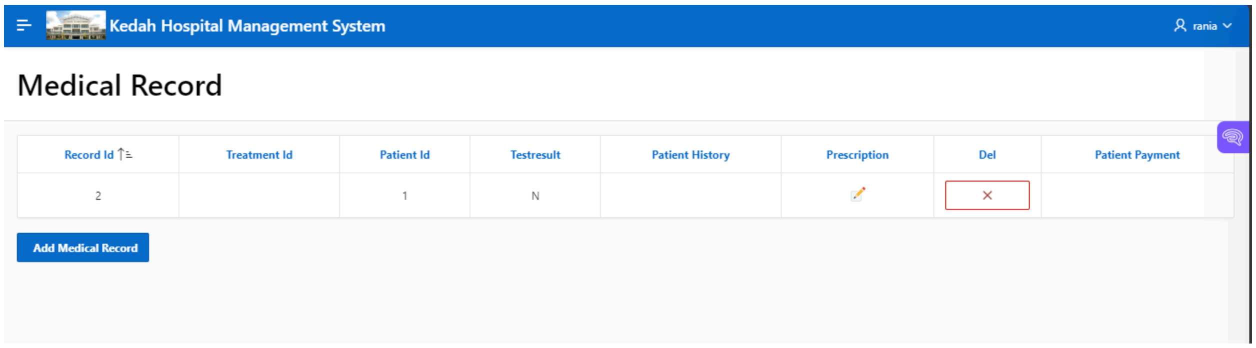
rukaya

Appointment

Add Appointment

Appointment Id	Patient Name	Doctor Name	Appointmentdate	Status	Diagnosis	Treatment	Follow Up	Del
21	FSFGD GDGF	Oon chi	2/5/2024	Visted	Fine	Treatment	Follow Up	✕

8.4 Medical Records Module



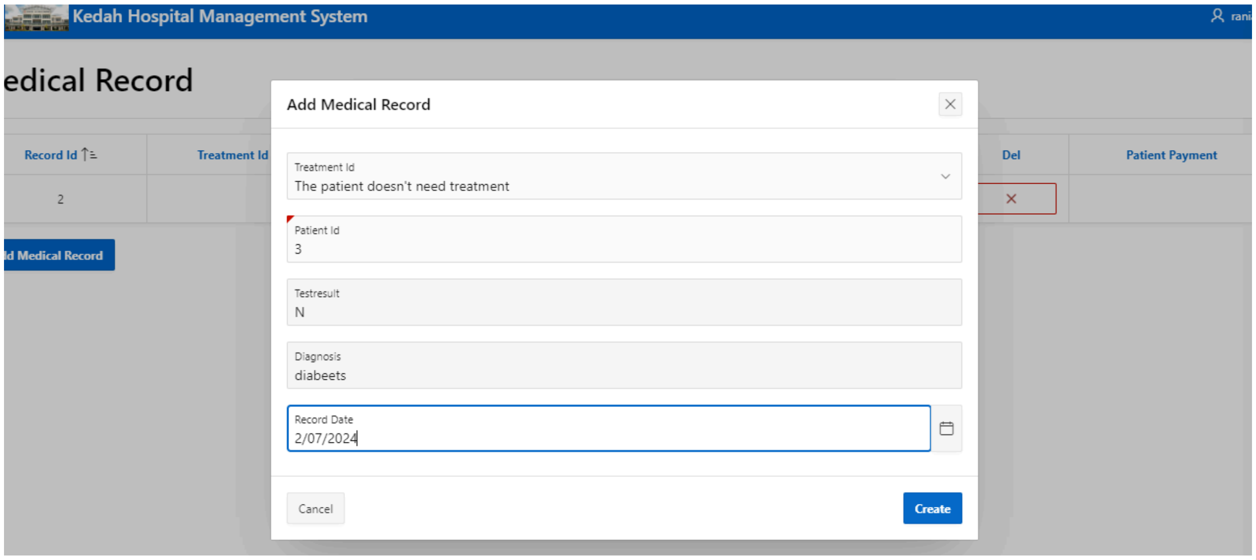
Kedah Hospital Management System

rania

Medical Record

Record Id	Treatment Id	Patient Id	Testresult	Patient History	Prescription	Del	Patient Payment
2		1	N		✎	✕	

Add Medical Record

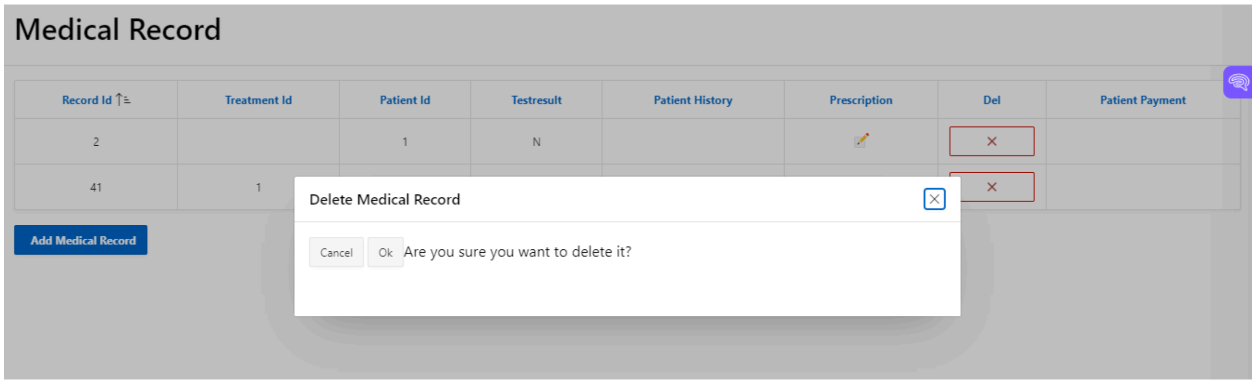


Kedah Hospital Management System

edical Record

Add Medical Record

Treatment Id	The patient doesn't need treatment
Patient Id	3
Testresult	N
Diagnosis	diabeets
Record Date	2/07/2024

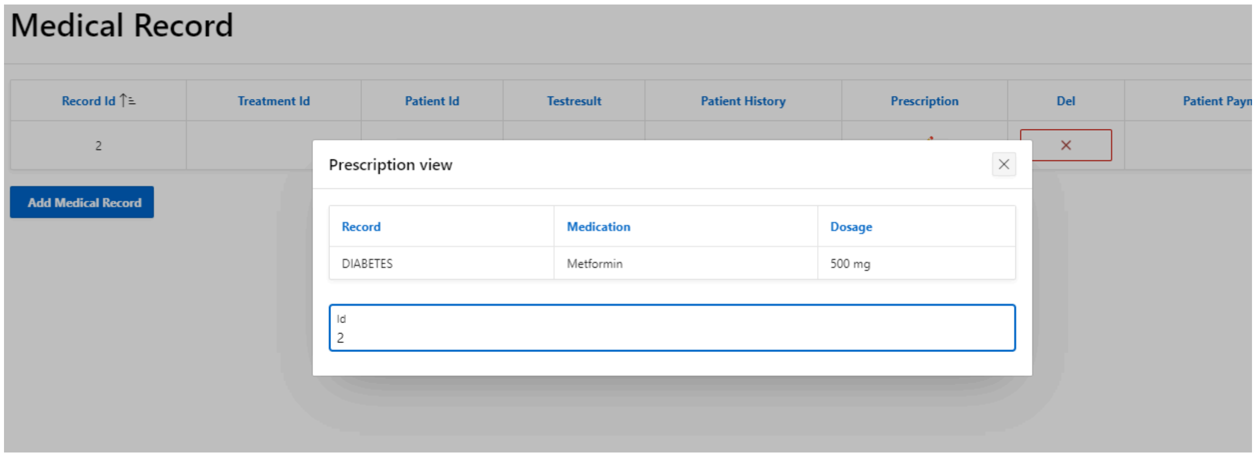


Medical Record

Record Id	Treatment Id	Patient Id	Testresult	Patient History	Prescription	Del	Patient Payment
2		1	N			×	
41	1					×	

Delete Medical Record

Are you sure you want to delete it?



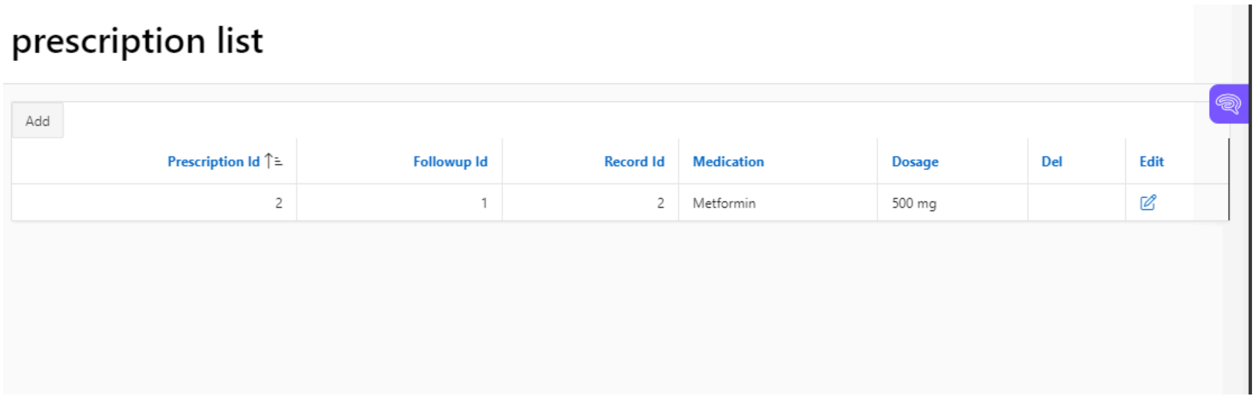
Medical Record

Record Id	Treatment Id	Patient Id	Testresult	Patient History	Prescription	Del	Patient Payn
2						×	

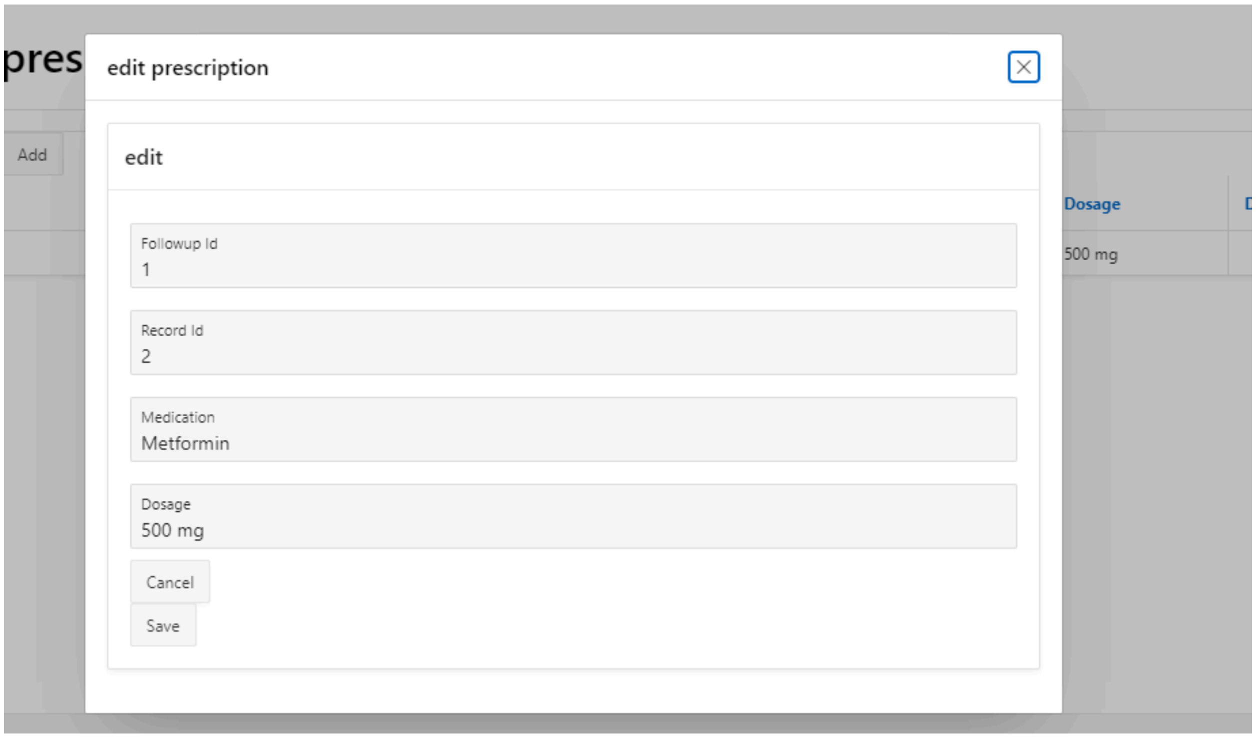
Prescription view

Record	Medication	Dosage
DIABETES	Metformin	500 mg

Id
2


prescription list

Prescription Id	Followup Id	Record Id	Medication	Dosage	Del	Edit
2	1	2	Metformin	500 mg		


edit prescription

edit

Followup Id	1
Record Id	2
Medication	Metformin
Dosage	500 mg

9.0 Pitfalls and obstacles

Building a web-based hospital management system was the most challenging task for us, as we lacked web development skills. However, during our research, we discovered a software solution for creating database web applications, and we decided to take the risk and learn how to build our system using Oracle APEX. Even though Oracle APEX is designed to be user-friendly, there is a significant learning curve. This is compounded by the inherent complexity of hospital management systems. Despite these challenges, we succeeded in designing a basic hospital management system based on our database, which includes numerous functionalities. Obtaining data was another hurdle because hospital-related data is sensitive and should not be publicly accessible. Overall, it was a rewarding experience where we learned how to use a powerful tool like Oracle APEX, as well as how to design a robust database and handle its operations.

10.0 Conclusion

In conclusion, the implementation of a comprehensive hospital management system is crucial for addressing the existing challenges faced by Kedah Medical Centre Sdn Bhd. The current manual and fragmented digital systems have led to inefficiencies, data fragmentation, lack of integration, and security concerns within the organization.

The proposed database system offers a solution by centralizing patient information, automating routine tasks, and optimizing resource allocation. It aims to improve operational efficiency, enhance patient care, ensure data security, and enable scalability.

By integrating modules such as patient management, doctor scheduling, appointment management, and medical records, the system will streamline processes, improve communication between departments, and provide healthcare providers with access to comprehensive patient information.

Overall, the adoption of this hospital management system will not only address the current challenges but also pave the way for a more efficient, secure, and patient-centered healthcare delivery system at Kedah Medical Centre Sdn Bhd.