



**NEW ERA UNIVERSITY COLLEGE**

**FACULTY OF COMPUTER SCIENCE AND INFORMATION COMPUTING  
TECHNOLOGY**

**DEPARTMENT OF INFORMATION COMPUTING TECHNOLOGY**

**May Semester 2024**

**TCYS133 ALGORITHMS AND DATA STRUCTURES**

**FINAL REPORT**

**PREPARED BY:**

**LIM FENG YAO 2350102-DCS**

**YAP HUI JUN 2320020-DCS**

**WONG JIE YING 2350080-DCS**

**LECTURER: Ts. LAW TENG YI**

**DATE OF SUBMISSION: 20/08/2024**

## Table of Contents

<b>1. Introduction.....</b>	<b>1</b>
<b>2. Flow of System (Flow Chart Diagram) .....</b>	<b>3</b>
<b>3. Data Structure System.....</b>	<b>4</b>
<b>3.1 Data Structure: Linked List .....</b>	<b>4</b>
<b>3.2 Database Table .....</b>	<b>4</b>
<b>3.3 Data Flow Diagram (DFD) .....</b>	<b>5</b>
<b>3.3.1 Level 0.....</b>	<b>5</b>
<b>3.3.2 Level 1.....</b>	<b>5</b>
<b>3.4 Data Table.....</b>	<b>6</b>
<b>3.4.1 User Table .....</b>	<b>6</b>
<b>3.4.2 Doctor Schedule Table .....</b>	<b>7</b>
<b>3.4.3 Type of service Table .....</b>	<b>8</b>
<b>3.4.4 Patient Appointment Table .....</b>	<b>8</b>
<b>3.4.5 Cancel Appointment Table.....</b>	<b>9</b>
<b>3.4.6 Medical Diagnosis Table .....</b>	<b>9</b>
<b>3.4.7 Patient Allergy Table .....</b>	<b>10</b>
<b>3.4.8 Vaccination Records Table.....</b>	<b>10</b>
<b>3.4.9 Patient Admission Transfer Table .....</b>	<b>11</b>
<b>3.4.10 Patient Follow Up Table .....</b>	<b>11</b>
<b>3.4.11 Feedback Table.....</b>	<b>12</b>
<b>3.4.12 Discount Table.....</b>	<b>12</b>
<b>3.4.13 Billint Table.....</b>	<b>13</b>
<b>3.4.14 Supplier Table .....</b>	<b>13</b>
<b>3.4.15 Purchase Table .....</b>	<b>13</b>
<b>3.4.16 Inventory Table .....</b>	<b>14</b>
<b>3.4.17 Medical Insurance Table .....</b>	<b>14</b>
<b>3.4.18 Notice Table.....</b>	<b>15</b>
<b>3.4.19 Emergency Tools Table .....</b>	<b>15</b>
<b>3.4.20 Project Table .....</b>	<b>16</b>
<b>3.5 Data Type.....</b>	<b>17</b>
<b>4. Hash Function.....</b>	<b>17</b>
<b>5. Pointers and References .....</b>	<b>17</b>

<b>6. Program Output.....</b>	<b>18</b>
<b>    6.4 Patient Menu .....</b>	<b>25</b>
<b>6.4.1 Profile .....</b>	<b>27</b>
<b>6.4.1.1 Update Profile .....</b>	<b>29</b>
<b>6.4.1.2 Delete Profile.....</b>	<b>31</b>
<b>6.4.1.3 Exit .....</b>	<b>32</b>
<b>6.4.2 Doctor Info.....</b>	<b>33</b>
<b>6.4.3 Make Appointment.....</b>	<b>33</b>
<b>6.4.4 Cancel Appointment .....</b>	<b>37</b>
<b>6.4.4.1 Add New Cancel Appointment.....</b>	<b>37</b>
<b>6.4.4.2 Edit Cancelled Appointment.....</b>	<b>39</b>
<b>6.4.5 Appointment History .....</b>	<b>41</b>
<b>6.4.6 Medical Diagnosis.....</b>	<b>42</b>
<b>6.4.7 Allergy .....</b>	<b>42</b>
<b>6.4.7.1 Add New Patient Allergy .....</b>	<b>42</b>
<b>6.4.7.2 Edit Patient Allergy.....</b>	<b>43</b>
<b>6.4.7.3 Delete Patient Allergy.....</b>	<b>43</b>
<b>6.4.7.4 Exit.....</b>	<b>45</b>
<b>6.4.8 Notices.....</b>	<b>46</b>
<b>6.4.9 Give Feedback .....</b>	<b>46</b>
<b>6.4.9.1 Add New Feedback .....</b>	<b>46</b>
<b>6.4.9.2 Edit Feedback .....</b>	<b>47</b>
<b>6.4.9.3 Delete Feedback .....</b>	<b>49</b>
<b>6.4.9.4 Exit.....</b>	<b>49</b>
<b>6.4.10 Vaccination History .....</b>	<b>50</b>
<b>6.4.10.1 Add New Vaccination History .....</b>	<b>50</b>
<b>6.4.10.2 Edit Vaccination History .....</b>	<b>51</b>
<b>6.4.10.3 Delete Vaccination History.....</b>	<b>51</b>
<b>6.4.10.4 Exit.....</b>	<b>53</b>
<b>6.4.11 Billing.....</b>	<b>53</b>
<b>6.4.12 Medical Insurance.....</b>	<b>54</b>
<b>6.4.12.1 Add New Medical Insurance .....</b>	<b>54</b>
<b>6.4.12.2 Edit Medical Insurance.....</b>	<b>55</b>
<b>6.4.12.3 Exit.....</b>	<b>56</b>
<b>6.4.13 Emergency Tool .....</b>	<b>56</b>

<b>6.4.14 Project.....</b>	57
<b>6.5 Doctor Menu.....</b>	58
<b>6.5.1 Profile .....</b>	58
<b>6.5.2 Patient Appointment Schedule .....</b>	61
<b>6.5.3 Patient Medical Diagnosis .....</b>	64
<b>6.5.4.1 Patient Follow-up .....</b>	69
<b>6.5.4.2 Patient Admission Transfer.....</b>	73
<b>6.5.5 Vaccination Records .....</b>	75
<b>6.5.6 Doctor Schedule.....</b>	76
<b>6.5.7 Patient Allergies History .....</b>	76
<b>6.5.8 Notices.....</b>	76
<b>6.5.8.1 Emergency Tools .....</b>	76
<b>6.5.8.2 Project .....</b>	76
<b>6.5.8.3 Back to Doctor Menu.....</b>	76
<b>6.5.9 Exit .....</b>	77
<b>6.6 Admin Menu.....</b>	78
<b>6.6.1 All User .....</b>	80
<b>6.6.1.1 Filter User .....</b>	81
<b>6.6.1.2 Add User .....</b>	82
<b>6.6.1.3 Update User .....</b>	84
<b>6.6.1.4 Delete User .....</b>	85
<b>6.6.2 Medical Insurance .....</b>	87
<b>6.6.2.1 Add Medical Insurance .....</b>	88
<b>6.6.2.2 Update Medical Insurance .....</b>	89
<b>6.6.2.3 Delete Medical Insurance .....</b>	91
<b>6.6.3 Doctor Schedule.....</b>	93
<b>6.6.3.1 Add Doctor Schedule .....</b>	94
<b>6.6.3.2 Update Doctor Schedule.....</b>	95
<b>6.6.3.3 Delete Doctor Schedule .....</b>	97
<b>6.6.4 Patient Medical Diagnosis .....</b>	99
<b>6.6.5 Type of Service .....</b>	99
<b>6.6.5.1 Add Type of Service.....</b>	100
<b>6.6.5.2 Update Type of Service .....</b>	101
<b>6.6.5.3 Delete Type of Service.....</b>	102
<b>6.6.6 Patient Appointment .....</b>	104

<b>6.6.6.1 Add Patient Appointment .....</b>	105
<b>6.6.6.2 Update Patient Appointment.....</b>	107
<b>6.6.7 Patient Cancelled Appointment .....</b>	109
<b>6.6.7.1 Add Patient Cancelled Appointment.....</b>	110
<b>6.6.7.2 Update Patient Cancelled Appointment .....</b>	112
<b>6.6.8 Inventory Management .....</b>	113
<b>6.6.8.1 Inventory .....</b>	114
<b>6.6.8.1.1 Add Inventory .....</b>	115
<b>6.6.8.1.2 Update Inventory .....</b>	116
<b>6.6.8.1.3 Delete Inventory .....</b>	118
<b>6.6.8.2 Supplier .....</b>	120
<b>6.6.8.2.1 Add Supplier.....</b>	121
<b>6.6.8.2.2 Update Supplier .....</b>	122
<b>6.6.8.2.3 Delete Supplier .....</b>	124
<b>6.6.8.3 Purchase Order .....</b>	126
<b>6.6.8.3.1 Add Purchase Order.....</b>	127
<b>6.6.8.3.2 Update Purchase Order .....</b>	128
<b>6.6.8.3.3 Delete Purchase Order .....</b>	130
<b>6.6.9 View and Publish Notices .....</b>	132
<b>6.6.9.1 Add Notices .....</b>	132
<b>6.6.9.2 Update Notices .....</b>	133
<b>6.6.9.3 Delete Notices .....</b>	135
<b>6.6.10 View and Publish Project .....</b>	137
<b>6.6.10.1 Add Project .....</b>	137
<b>6.6.10.2 Update Project.....</b>	138
<b>6.6.10.3 Delete Project .....</b>	140
<b>6.6.11 Discount .....</b>	141
<b>6.6.11.1 Add Discount.....</b>	142
<b>6.6.11.2 Update Discount .....</b>	143
<b>6.6.11.3 Delete Discount .....</b>	145
<b>6.6.12 Patient Admission Transfer .....</b>	147
<b>6.6.13 View Feedback.....</b>	148
<b>6.6.14 Patient Vaccination Records .....</b>	148
<b>6.6.15 History of Patient Allergies.....</b>	149
<b>6.6.16 Latest Patient Status .....</b>	149

<b>6.6.17 Emergency Tools.....</b>	150
<b>6.6.17.1 Add Emergency Tools .....</b>	150
<b>6.6.17.2 Update Emergency Tools.....</b>	151
<b>6.6.17.3 Delete Emergency Tools .....</b>	153
<b>6.6.18 Billing.....</b>	155
<b>6.6.18.1 Add Billing .....</b>	156
<b>6.6.18.2 Update Billing.....</b>	158
<b>6.6.18.3 Delete Billing .....</b>	160
<b>6.6.19 Exit .....</b>	162
<b>7. Example Coding.....</b>	163
<b>7.1 Patient Menu .....</b>	163
<b>7.1.1 Make appointments .....</b>	163
<b>7.1.1.1 Structure.....</b>	163
<b>7.1.1.2 Print .....</b>	164
<b>7.1.1.3 Push back.....</b>	165
<b>7.1.1.4 Create .....</b>	166
<b>7.1.1.5 Read .....</b>	169
<b>7.1.2 Cancel Appointment .....</b>	169
<b>7.1.2.1 Structure.....</b>	169
<b>7.1.2.2 Print .....</b>	170
<b>7.1.2.3 Push back.....</b>	171
<b>7.1.2.4 Create .....</b>	172
<b>7.1.2.5 Read .....</b>	174
<b>7.1.2.6 Update .....</b>	175
<b>7.2 Doctor Menu.....</b>	177
<b>7.2.1 Structure.....</b>	177
<b>7.2.2 Node.....</b>	177
<b>7.2.3 Push Back.....</b>	177
<b>7.2.4 Generates Unique ID.....</b>	178
<b>7.2.5 Create Medical Diagnosis.....</b>	178
<b>7.2.6 Read Medical Diagnosis .....</b>	181
<b>7.2.7 Update Medical Diagnosis .....</b>	182
<b>7.2.8 Delete Medical Diagnosis.....</b>	186
<b>7.3 Admin Menu.....</b>	189
<b>7.3.1 Inventory Structure .....</b>	189

<b>7.3.2 Inventory Node .....</b>	190
<b>7.3.3 Inventory Push Back.....</b>	191
<b>7.3.4 Generates Unique Inventory ID.....</b>	192
<b>7.3.5 Read Inventory .....</b>	192
<b>7.3.6 Create Inventory.....</b>	193
<b>7.3.7 Update Inventory.....</b>	195
<b>7.3.8 Delete Inventory .....</b>	198
<b>8. Database Sqlite3.....</b>	199
<b>8.1 Billing Table .....</b>	199
<b>8.2 Cancel Appointment Table.....</b>	199
<b>8.3 Discount Table .....</b>	199
<b>8.4 Doctor Schedule Table .....</b>	199
<b>8.5 Emergency Table .....</b>	200
<b>8.6 Feedback Table.....</b>	200
<b>8.7 Inventory Table .....</b>	200
<b>8.8 medical Diagnosis Table .....</b>	200
<b>8.9 Medical Insurance Table .....</b>	201
<b>8.10 Notice Table.....</b>	201
<b>8.11 Patient Admission Transfer Table.....</b>	201
<b>8.12 Patient Allergy Table.....</b>	201
<b>8.13 Patient Follow Up Table .....</b>	202
<b>8.14 Patient Appointment Table.....</b>	202
<b>8.15 Project Table .....</b>	202
<b>8.16 Purchase Table .....</b>	202
<b>8.18 Type of Service Table .....</b>	203
<b>8.19 User Table .....</b>	203
<b>8.20 Vaccination Records Table .....</b>	203
<b>9.0 Conclusion .....</b>	203
<b>10.0 Distribution .....</b>	206

## **1. Introduction**

The Best Clinic Appointment Management System is a comprehensive platform designed to streamline and enhance the efficiency of scheduling and managing clinic appointments. The system is structured to accommodate three primary user roles: patients, doctors, and administrators, each with tailored functionalities to meet their specific needs.

Patients are provided with a user-friendly menu featuring fourteen key functions that facilitate their interaction with the clinic. The primary functions include the ability to make and cancel appointments, view medical diagnoses, and manage payments. Patients can easily schedule appointments by selecting the desired service, date, time, and doctor. Additionally, they can view and settle their bills securely through the system. Other important features include the ability to update personal information, access clinic notices and doctor information, review medical reports, check allergy details, review follow-up appointments, and admission statuses. The system also offers a direct communication channel for patients to contact clinic support if needed.

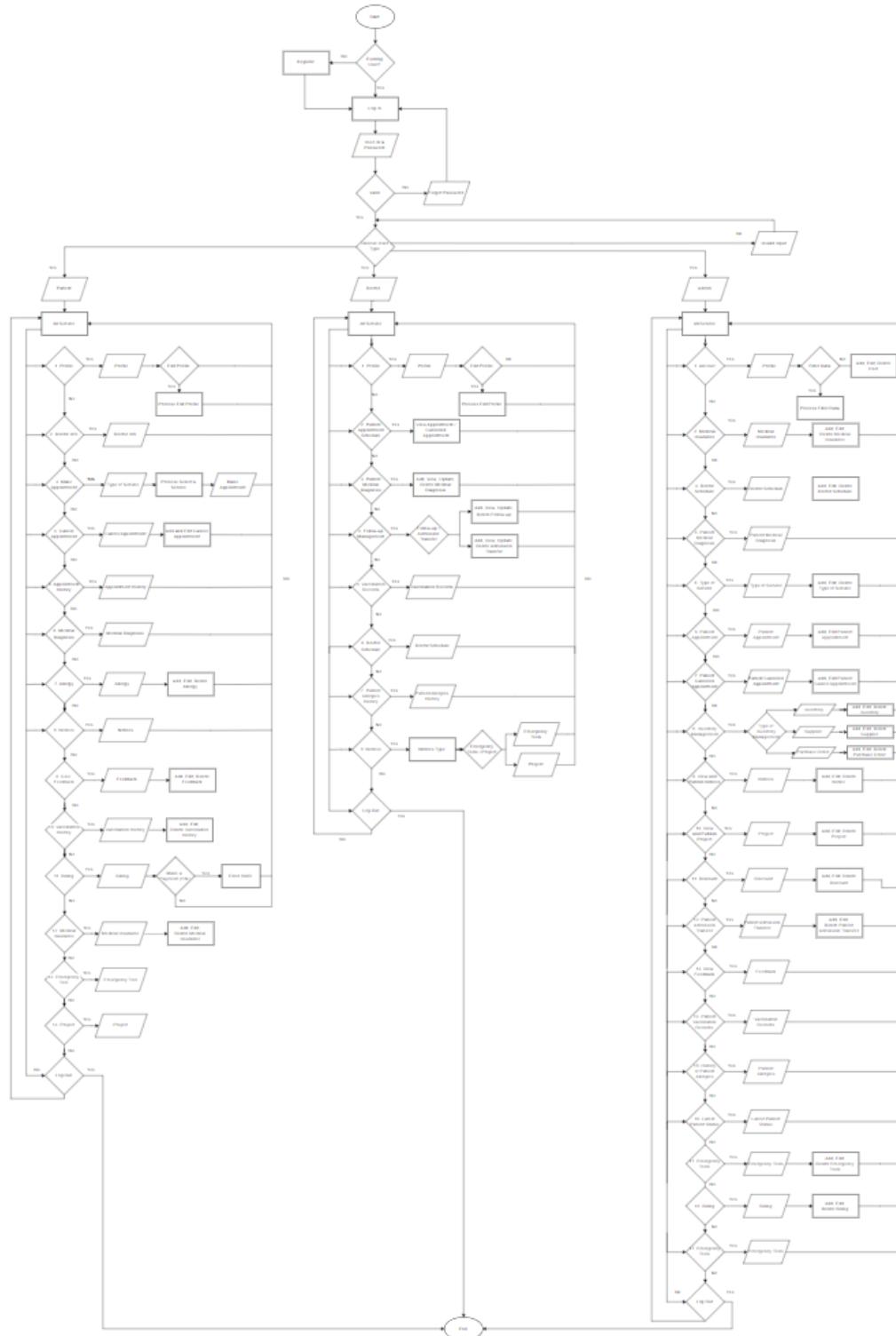
Doctors have access to an eight-function menu that supports their clinical responsibilities and patient management tasks. Their primary functions include viewing patient appointments and conducting medical diagnoses. The system allows doctors to select a patient, update medical history, record diagnoses, and document medications. It also supports the management of follow-up care, including recording dates and descriptions of follow-up actions. For patients requiring admission, doctors can record check-in and check-out times and update the patient's current status. Additionally, doctors can view clinic information, patient allergy details, and their own medical records. A support contact feature is also available for doctors to communicate with the administrative team when needed.

Administrators enjoy the highest level of access, with eighteen functions at their disposal, allowing them to manage various aspects of clinic operations. Administrators can perform CRUD operations on user accounts, ensuring that patient, doctor, and admin information is kept up to date. They are also responsible for managing clinic information, including updating services, and posting notices. The system enables admins to oversee payment records, review doctors' medical diagnoses and patient follow-ups.

Overall, the Best Clinic Appointment Management System is designed to provide a secure, user-friendly, and efficient platform for managing clinic appointments, patient care, and administrative tasks, ensuring that all users can interact with the clinic seamlessly and effectively.

## 2. Flow of System (Flow Chart Diagram)

The Flow of System (Flow Chart Diagram) shows how the Best Clinic Appointment Management System works. It maps out the steps users take from logging in to performing tasks like making appointments, updating information, and so on. This diagram helps to understand how different parts of the system interact and function together.



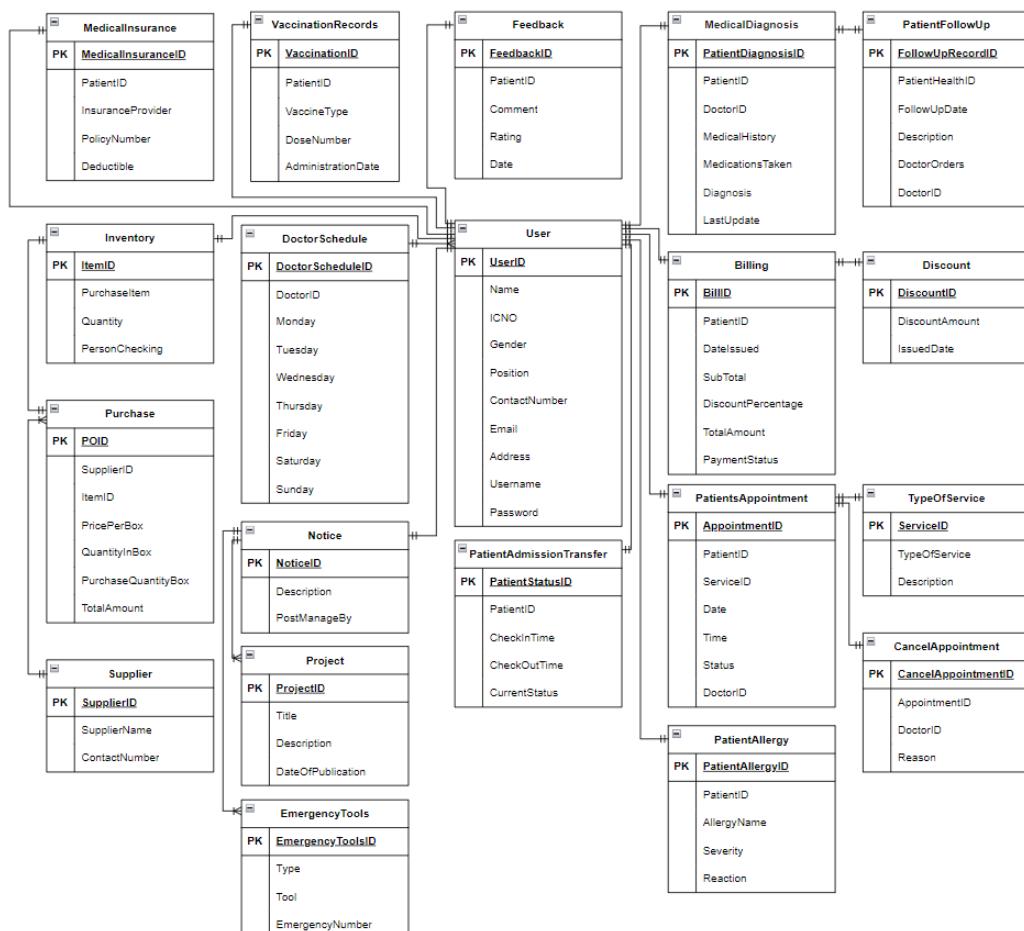
### 3. Data Structure System

#### 3.1 Data Structure: Linked List

In the Best Clinic Appointment Management System, a linked list data structure is used to efficiently manage patient records, appointments, and other clinic data. Linked lists allow for easy insertion, deletion, and updating of information, which is essential in a healthcare setting where details often change. By representing each piece of data as nodes within linked lists, the system can handle dynamic information smoothly and keep operations flexible and responsive to the clinic's needs.

#### 3.2 Database Table

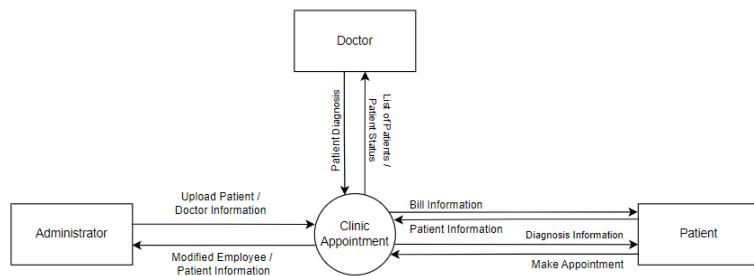
The Best Clinic Appointment Management System is supported by a database that organizes essential clinic data across various tables. These tables manage user information, appointment scheduling, medical records, billing, and inventory. This structure ensures that all clinic operations run smoothly, with data being easily accessible and well-organized.



### 3.3 Data Flow Diagram (DFD)

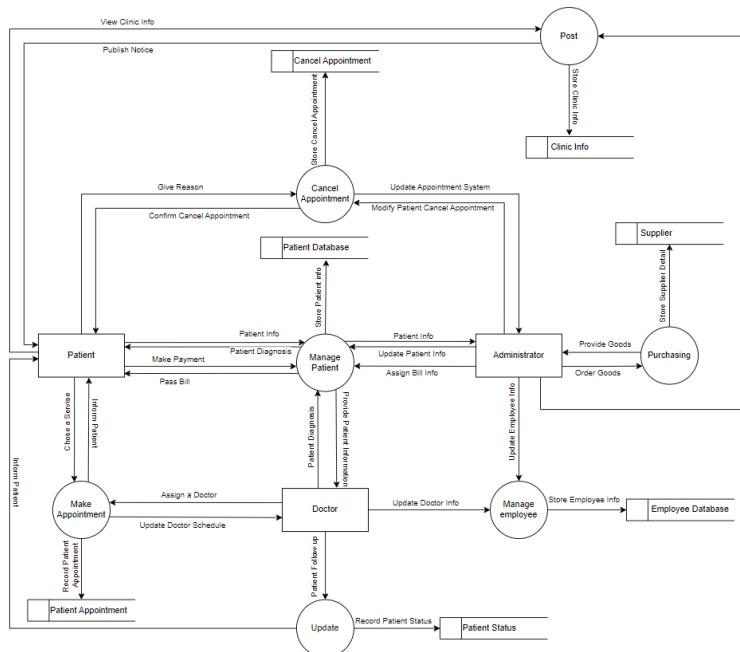
#### 3.3.1 Level 0

In our project, the Level 1 Data Flow Diagram (DFD) highlights essential functions such as managing patient diagnoses, handling patient lists and statuses, updating patient and doctor information, and processing billing details. It also covers appointment scheduling and information management, ensuring that all critical aspects of patient care and administrative tasks are efficiently handled.



#### 3.3.2 Level 1

This Data Flow Diagram (DFD) outlines how information flows within a clinic management system. It shows interactions between Patients, Administrators, Doctors, Suppliers, and various databases. Key processes include managing patient appointments and billing, scheduling and updating appointments, ordering from suppliers, updating employee records, and posting notices. The diagram illustrates how data moves between these components to ensure smooth clinic operations.



## 3.4 Data Table

### 3.4.1 User Table

User									
UserID	Name	IC NO	Gender	Position	Contact Number	Email	Address	Username	Password
P1	Johnathan Alexander Doe	800101141239	Male	Patient	60119876543	johndoe@example.com	Majestic Maxim Jalan 9/142, Taman Len Seng, Batu 9, Cheras, Kuala Lumpur	John Doe	ab7kl
P2	Jane Elizabeth Smith	850202105678	Female	Patient	60145678901	janesmith@example.com	7 Jalan Dataran Cheras 6, Balakong, Kajang, Selangor	Jane Smith	x9mn124
P3	Alice Victoria Brown	880303089102	Female	Patient	60171234567	alicebrown@example.com	Jalan Tasik Utama, Sungai Besi, Wilayah Persekutuan Kuala Lumpur	Alice Brown	1uz9x
E1	Johnathan Maxwell Smith	750123105671	Male	Doctor	60154489651	john@example.com	Jalan Setia Tropika U13/20, Setia Eco Park, Setia Eco Park, Shah	John Smith	156cd

							Alam, Selangor		
E2	Matthew Harrison Wilson	820305891723	Male	Doctor	6016654654 5	matthew@example.com	Lebuhraya Sungai Besi BESRAYA, Sungai Besi, Kuala Lumpur	Matthe w Wilson	casca 62
E3	Michael Christopher Brown	910405141122	Female	Admin	6014148655 4	michael@example.com	1A Tingkat 1 Jalan S 10, 1, Jalan Bukit Kajang, 43000 Kajang, Selangor	Michael Brown	dsCg 530

#### 3.4.2 Doctor Schedule Table

Doctor Schedule ID									
Doctor ID	Schedule ID	Doctor ID	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
D1	E1		Working	Working	Working	Off	Off	Off	Off
D2	E2		Off	Off	Off	Working	Working	Working	Off

### 3.4.3 Type of service Table

Type of service		
ServiceID	Type of Service	Description
S1	Laboratory Services	Diagnostic Services include blood and urine tests
S2	Vaccination	Immunizations cover all ages
S3	Family Medicine	Primary Care focuses on health, prevention, and chronic conditions

### 3.4.4 Patient Appointment Table

Patient's Appointment						
Appointment ID	Patient ID	ServiceID	Date	Time	Status	Doctor ID
A1	P1	S3	05/01/2024	09:00	Accessed	E1
A2	P2	S1	08/02/2024	10:00	Cancelled	E2
A3	P2	S1	10/02/2024	10:00	Accessed	E2
A4	P3	S2	15/03/2024	12:00	Accessed	E1

### 3.4.5 Cancel Appointment Table

Cancel Appointment by Patient			
Cancel Appointment ID	Appointment ID	Doctor ID	Reason
CA1	A2	E2	Scheduling conflicts

### 3.4.6 Medical Diagnosis Table

Medical Diagnosis						
Patient Diagnosis ID	Patient ID	Doctor ID	Medical History	Medications Taken	Diagnosis	Last Update
PH1	P1	E1	Hypertension	Lisinopril, Amlodipine, Losartan, Metoprolol	Dengue fever	08/01/2024
PH2	P2	E2	Diabetes	Metformin, Insulin, DPP-4 inhibitors	Iron deficiency anaemia	17/02/2024
PH3	P3	E1	Asthma	Albuterol (short-acting), Fluticasone	Get HPV vaccine	15/03/2024
PH4	P3	E1	Diabetes	Metformin, Insulin	Type 2 Diabetes	2024-03-17

### **3.4.7 Patient Allergy Table**

<b>Patient Allergy</b>				
<b>Patient Allergy ID</b>	<b>Patient ID</b>	<b>Allergy Name</b>	<b>Severity</b>	<b>Reaction</b>
AI1	P2	Peanut	Severe	Anaphylaxis (life-threatening allergic reaction)

### **3.4.8 Vaccination Records Table**

<b>Vaccination Records</b>				
<b>Vaccination ID</b>	<b>Patient ID</b>	<b>Vaccine Type</b>	<b>Dose Number</b>	<b>Administration Date</b>
V1	P1	AZ	First dose	16/09/2021
V2	P3	Pfizer	First dose	01/10/2021

### 3.4.9 Patient Admission Transfer Table

Patient Admission Transfer				
Patient Status ID	Patient ID	Check-in Time	Check-out Time	Current Status
PS1	P1	06/01/2024 10.30	08/01/2024 12.00	Discharged
PS2	P2	10/02/2024 13.00	17/02/2024 16.00	Transferred
PS3	P3	15/03/2024 13.00	15/03/2024 15.30	Discharged

### 3.4.10 Patient Follow Up Table

Patient Follow-up					
Follow-up Record ID	Patient Diagnosis ID	Follow-up Date	Description	Doctor Orders Description	Doctor ID
FU1	PH1	07/01/2024	Mild fever, joint pain; no rash	Monitor platelets, rest. Follow-up in 1 week	E1
FU2	PH2	13/02/2024	Reduced fatigue, better energy	Continue treatment, add Vitamin C. Test in 1 month	E2

#### **3.4.11 Feedback Table**

<b>Feedback</b>				
<b>Feedback ID</b>	<b>Patient ID</b>	<b>Comment</b>	<b>Rating</b>	<b>Date</b>
F1	P1	Excellent care, very satisfied with service.	5	08/01/2024
F2	P2	Good service, but waiting time was too long.	3	14/02/2024

#### **3.4.12 Discount Table**

<b>Discount</b>		
<b>Discount ID</b>	<b>Discount Amount</b>	<b>Issued Date</b>
V1	5%	05/01/2024
V2	10%	16/02/2024

### 3.4.13 Billint Table

Billing						
Bill ID	Patient ID	Date Issued	SubTotal	Discount Percentage	Total Amount	Payment Status
B1	P1	07/01/2024	RM 542.00	-	RM 542.00	Unpaid
B2	P2	16/02/2024	RM 873.33	10%	RM 786.00	Unpaid
B3	P3	18/03/2024	RM 206.00	-	RM 206.00	Unpaid

### 3.4.14 Supplier Table

Supplier		
Supplier ID	Supplier Name	Contact Number
SU1	ABC Medical	60167456215
SU2	XYZ Supplies	60145951716

### 3.4.15 Purchase Table

Purchase						
PO ID	Supplier ID	Item ID	Price/Box (RM)	Quantity in a Box	Purchase Quantity (Box)	Total Amount (RM)

PO1	SU1	I1	60	100	8	480
PO2	SU2	I1	20	50	4	80

#### 3.4.16 Inventory Table

Inventory			
Item ID	Purchase Item	Quantity	Person Checking
I1	Syringes	500	E2
I2	Bandages	100	E2

#### 3.4.17 Medical Insurance Table

Medical Insurance				
Medical Insurance ID	Patient ID	Insurance Provider	Policy Number	Deductible
INS1	P1	Allianz	A12392	400000
INS2	P2	AIA	AIA193842	600000

### **3.4.18 Notice Table**

Notice (link employee)		
Notice ID	Description	Post Manage By
N1	Update emergency tools	E3
N2	Vaccination Information	E3
N3	Know more about your health	E3

### **3.4.19 Emergency Tools Table**

Emergency Tools			
Emergency Tools ID (PK)	Type	Tool	Emergency Number
ET1	Contact Information	Emergency Number	03-86457894
ET2	Transportation	Ambulance Department Service	999

### **3.4.20 Project Table**

<b>Project</b>			
<b>Project ID (PK)</b>	<b>Title</b>	<b>Description</b>	<b>Date of Publication</b>
PRO1	New Vaccination Guidelines	Release of updated guidelines for vaccination protocols	18/01/2024
PRO3	Mental Health Awareness	Campaign to promote mental health awareness	08/03/2024

### **3.5 Data Type**

In the Best Clinic Appointment Management System, various data types are utilized to effectively manage and store information. **TEXT** is used for string data such as names, addresses, and descriptions, while **INTEGER** is employed for whole numbers like quantities and ID numbers. **DOUBLE** handles decimal values, which are essential for monetary amounts and measurements. **DATE** is used to record dates such as appointment and issuance dates, while **TIME** is used for specific times like appointment schedules. **DATETIME** combines both date and time, useful for tracking events like check-ins and check-outs. These data types ensure the system can accurately manage and process all necessary information for smooth clinic operations.

### **4. Hash Function**

In the Best Clinic Appointment Management System, hash functions are used to efficiently manage and secure various types of data. A hash function converts input data, such as user information or system records, into a fixed-size value known as a hash code. This hash code is used for quick data retrieval, ensuring that similar data is stored and accessed efficiently. For instance, when organizing and indexing records in the system, hash functions help in distributing data uniformly, which reduces the likelihood of data collisions and speeds up search operations. Additionally, hash functions support data integrity by providing a way to verify that data has not been altered or corrupted.

### **5. Pointers and References**

In our project, we use references and pointers to efficiently manage and manipulate data. **Pointers** allow us to directly access and modify memory locations, which is crucial for handling dynamic data structures and optimizing performance. **References**, on the other hand, provide a simpler way to work with variables without dealing with memory addresses explicitly, making the code cleaner and easier to understand. Together, these tools enhance the flexibility and efficiency of our project's data management and functionality.

## 6. Program Output

## 6.1 Main Menu

Show the logo and ask if the user has an account or not.

```
# # # # # # # # # # ***** * ***** * ***** * ***** * * * ***** * * * ***** * *****  
# ----- # * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  
# | | | | # * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  
# ----- # * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  
# | BEST CLINIC | # ***** * ***** * ***** * * * * * * * * * * * * * *  
# ----- # * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  
# | | | | # * * * * * * * * * * * * * * * * * * * * * * * * * *  
# ----- # * * * * * * * * * * * * * * * * * * * * * * * * * *  
# # # # # # # # # # ***** * ***** * ***** * * * ***** * * * ***** * *****  
Do you have an account? (Y/N): |
```

## 6.1 Register

If you select ‘n’ or ‘N’, it will show a Register Account to allow users to enter the username and password they want. If no error will show “Account registered successfully!”.

```
# # # # # # # # # #   *****   *****   ****   *****   *****   *   *****   *   *   *****   *****  
#   ----- #   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *  
#   |   |   |   #   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *  
#   ----- ----- #   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *  
# |  BEST CLINIC  | #   *****   *****   ****   *   *   *   *   *   *   *   *   *   *  
#   ----- ----- #   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *  
#   |   |   |   #   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *  
#   ----- ----- #   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *  
# # # # # # # # # #   *****   *****   ****   *   *****   *****   *****   *   *   *****   *****  
Do you have an account? (Y/N): n  
  
Register Account  
-----  
Enter username: John Smith  
Enter password: 156cd  
Account registered successfully!
```

Ask the user to complete their profile.

```
Complete User Profile
-----
Enter Name: Johnathan Maxwell Smith
Enter IC No (12 digits): 750123105671
Enter Gender (Male/Female): Male
Enter Position (Patient/Doctor/Admin): Doctor
Enter Contact Number (11-12 digits): 60154489651
Enter Email: john@example.com
Enter Address: Jalan Setia Tropika U13/20, Setia Eco Park, Setia Eco Park, Shah Alam, Selangor
User inserted successfully
```

Based on the user position show the following menu.

## Doctor Menu

\*\*\*\*\*

- 1. Profile**
- 2. Patient Appointment Schedule**
- 3. Patient Medical Diagnosis**
- 4. Follow-up Management**
- 5. Vaccination Records**
- 6. Doctor Schedule**
- 7. Patient Allergies History**
- 8. Notices**
- 0. Exit**

If the user puts it blank or invalid input will show a reminder and ask to enter again.

Here the IC number should be 12 digits.

```
Enter IC No (12 digits):  
IC No must be exactly 12 digits. Please try again.  
Enter IC No (12 digits): 820305891  
IC No must be exactly 12 digits. Please try again.
```

Set the gender only and can enter ‘male’, ‘female’, ‘Male’, or ‘Female’.

```
Enter Gender (Male/Female):  
Gender must be either 'male' or 'female'. Please try again.  
Enter Gender (Male/Female): malr  
Gender must be either 'male' or 'female'. Please try again.  
Enter Gender (Male/Female): male
```

Set the position only can enter ‘patient’, ‘doctor’, ‘admin’, ‘Patient’, ‘Doctor’, or ‘Admin’.

```
Enter Position (Patient/Doctor/Admin):  
Position must be either 'patient', 'doctor', or 'admin'. Please try again.  
Enter Position (Patient/Doctor/Admin): dootor  
Position must be either 'patient', 'doctor', or 'admin'. Please try again.  
Enter Position (Patient/Doctor/Admin): doctor
```

Set contact number should be entered between 11 – 12 digits.

```
Enter Contact Number (11-12 digits):  
Contact number must be between 11 to 12 digits. Please try again.  
Enter Contact Number (11-12 digits): 60166546  
Contact number must be between 11 to 12 digits. Please try again.  
Enter Contact Number (11-12 digits): 60166546545
```

Set email must include ‘@’.

```
Enter Email:  
Incorrect email format. Please try again.  
Enter Email: matthewexample.com  
Incorrect email format. Please try again.  
Enter Email: matthew@example.com
```

Address disallows put blank.

```
Enter Address:  
Address cannot be empty. Please try again.
```

If the username already exists, disallow using the username to register.

```
Register Account  
-----  
Enter username: John Smith  
Username already exists. Please try another username.  
Enter username:  
Username cannot be empty. Please enter again.  
Enter username: |
```

Using a different username and the same password is allowed to register an account.

```
Register Account
-----
Enter username: Matthew Wilson
Enter password: 156cd
Account registered successfully!
```

## 6.2 Login

Login account with the user that is already registered.

```
# # # # # # # # # # ***** ***** **** ***** ***** * * ***** * * * ***** *
# ----- # * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
# | BEST CLINIC | # ***** ***** **** * * * * * * * * * * * * * * * * * * *
# ----- # * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
# | | | # * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
# ----- # * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
# # # # # # # # # # ***** ***** **** * * ***** ***** ***** * * * ***** *
Do you have an account? (Y/N): y

Login Account
-----
Enter your Username: John Smith
Enter your password: 156cd
Login Successful
Welcome John Smith
```

The username and password are disallowed and put blank.

```
Login Account
-----
Enter your Username:
Username cannot be empty. Please enter again.
Enter your Username: John Smith
Enter your password:
Password cannot be empty. Please enter again.
Enter your password: 156cd
Login Successful
Welcome John Smith
```

Invalid three times will ask if the user wants to register an account or not.

```
Login Account
-----
Enter your Username: JOHN SMITH
Enter your password: 156cd
User not found.
Username or password incorrect. Please try again.
```

```
Login Account
-----
Enter your Username: Johnn smith
Enter your password: 145cd
User not found.
Username or password incorrect. Please try again.
```

```
Login Account
-----
Enter your Username: John smith
Enter your password: 156dc
User not found.
Username or password incorrect. Please try again.
Account not found? Register Account? (Y/N): |
```

If ‘y’ or ‘Y’ shows the register menu for the user to register a new account.

```
Account not found? Register Account? (Y/N): y

Register Account
-----
Enter username: Matthew Wilson
Enter password: casca62
Account registered successfully!
```

If ‘n’ or ‘N’ shows forgot password option for the user to select, if they forgot and would like to reset the password? If the user selects ‘y’ or ‘Y’, enter can perform a reset password action.

```
Account not found? Register Account? (Y/N): n
Forgot password? (Y/N): y
Enter username to reset password: |
```

If you select ‘n’ or ‘N’ will exit the program.

```
Account not found? Register Account? (Y/N): n
Forgot password? (Y/N): n
Exiting program...
```

### 6.3 Forgot Password

Users need to enter their username to reset the old password. They are required to enter two times to verify the new password is the same.

```
Forgot password? (Y/N): y
Enter username to reset password: John Smith
Enter new password: 159as
Confirm new password: 159as
Password changed successfully!
```

When the new password and confirm new password do not match, the user needs to enter it again.

```
Enter username to reset password: John Smith
Enter new password: 12kij
Confirm new password: 11ijk
Passwords do not match. Please try again.
Enter new password: |
```

When a user enters a username that is not found in the system, ask the user to register a new account. If the user enters ‘y’ or ‘Y’, return to the register menu.

```
-----  
Login Account  
-----  
Enter your Username: John smith  
Enter your password: 156cd  
User not found.  
Username or password incorrect. Please try again.  
Account not found? Register Account? (Y/N): n  
Forgot password? (Y/N): y  
Enter username to reset password: Wilson  
User not found. Please try again.  
Do you want to register a new account? (Y/N): y  
  
Register Account  
-----  
Enter username: |
```

For the option “Do you want to register a new account? (Y/N)”, the user enters ‘n’ or ‘N’ and returns to the login menu.

```
-----  
Login Account  
-----  
Enter your Username: John smith  
Enter your password: 156cd  
User not found.  
Username or password incorrect. Please try again.  
Account not found? Register Account? (Y/N): n  
Forgot password? (Y/N): y  
Enter username to reset password: Wilson  
User not found. Please try again.  
Do you want to register a new account? (Y/N): n  
Returning to login menu.  
Do you have an account? (Y/N): |
```

#### **6.4 Patient Menu**

The patient menu provides a user-friendly interface for patients to manage their appointments, view medical information, and access various services within the system.

- ```
Patient Menu
1. Profile
2. Doctor Info
3. Make Appointment
4. Cancel Appointment
5. Appointment History
6. Medical Diagnosis
7. Allergy
8. Notices
9. Give Feedback
10. Vaccination History
11. Billing
12. Medical Insurance
13. Emergency Tool
14. Project
0. Exit
```

```
Patient Menu
1. Profile
2. Doctor Info
3. Make Appointment
4. Cancel Appointment
5. Appointment History
6. Medical Diagnosis
7. Allergy
8. Notices
9. Give Feedback
10. Vaccination History
11. Billing
12. Medical Insurance
13. Emergency Tool
14. Project
0. Exit
```

Enter your choice: y

Invalid input. Please enter a valid integer.

```
Patient Menu
1. Profile
2. Doctor Info
3. Make Appointment
4. Cancel Appointment
5. Appointment History
6. Medical Diagnosis
7. Allergy
8. Notices
9. Give Feedback
10. Vaccination History
11. Billing
12. Medical Insurance
13. Emergency Tool
14. Project
0. Exit
```

Enter your choice:

If the user types something that isn't an integer, display 'Invalid input. Please enter a valid integer.' and then show the patient menu again.

Type '0' to exit the patient menu.

```
Patient Menu
1. Profile
2. Doctor Info
3. Make Appointment
4. Cancel Appointment
5. Appointment History
6. Medical Diagnosis
7. Allergy
8. Notices
9. Give Feedback
10. Vaccination History
11. Billing
12. Medical Insurance
13. Emergency Tool
14. Project
0. Exit
Enter your choice: 0
```

```
C:\Users\User\Documents\ClinicAppointmentSchedulingSystem>
```

#### 6.4.1 Profile

This profile belongs to Johnathan Alexander Doe. It will display the user's information.

```
Profile
(null):UserID = P1
Name = Johnathan Alexander Doe
ICNO = 800101141239
Gender = Male
Position = Patient
ContactNumber = 60119876543
Email = johndoe@example.com
Address = Majestic Maxim Jalan 9/142, Taman Len Seng, Batu 9, Cheras, Kuala Lumpur
Username = John Doe
Password = ab7kl
```

This profile belongs to Jane Elizabeth Smith. It will display the user's information.

```
Profile
(null):UserID = P2
Name = Jane Elizabeth Smith
ICNO = 850202105678
Gender = Female
Position = Patient
ContactNumber = 60145678901
Email = janesmith@example.com
Address = 7 Jalan Dataran Cheras 6, Balakong, Kajang, Selangor
Username = Jane Smith
Password = x9mn124
```

This profile belongs to Alice Victoria Brown. It will display the user's information.

```
Profile
(null):UserID = P3
Name = Alice Victoria Brown
ICNO = 880303089102
Gender = Female
Position = Patient
ContactNumber = 60171234567
Email = alicebrown@example.com
Address = Jalan Tasik Utama, Sungai Besi, Wilayah Persekutuan Kuala Lumpur
Username = Alice Brown
Password = 1uz9x
```

#### 6.4.1.1 Update Profile

This is the user SQLite.

| User ID | Name                    | IC NO        | Gender | Position | Contact Number | Email                  | Address                                 | Username    | Password |
|---------|-------------------------|--------------|--------|----------|----------------|------------------------|-----------------------------------------|-------------|----------|
| Filter  | Filter                  | Filter       | Filter | Filter   | Filter         | Filter                 | Filter                                  | Filter      | Filter   |
| 1 P1    | Johnathan Alexander Doe | 800101141239 | Male   | Patient  | 60119876543    | johndoe@example.com    | Majestic Maxim Jalan 9/142, Taman ...   | John Doe    | ab7kl    |
| 2 P2    | Jane Elizabeth Smith    | 850202105678 | Female | Patient  | 60145678901    | janesmith@example.com  | 7 Jalan Dataran Cheras 6, Balakong, ... | Jane Smith  | x9mn124  |
| 3 P3    | Alice Victoria Brown    | 880303089102 | Female | Patient  | 60171234567    | alicebrown@example.com | Jalan Tasik Utama, Sungai Besi, ...     | Alice Brown | 1uz9x    |

If the patient wants to edit the profile, prompt them with 'Do you want to take action on your Profile? (Y/N)'. If they type 'Y', proceed to the update process. If they type 'N', return to the patient menu. Then, display the available actions and prompt them to choose '1' to update the profile.

```
Profile
(null):UserID = P1
Name = Johnathan Alexander Doe
ICNO = 800101141239
Gender = Male
Position = Patient
ContactNumber = 60119876543
Email = johndoe@example.com
Address = Majestic Maxim Jalan 9/142, Taman Len Seng, Batu 9, Cheras, Kuala Lumpur
Username = John Doe
Password = ab7kl

Do you want to have action for Profile? (Y/N): Y

Select an action:
1. Edit User Profile
2. Delete User
0. Exit
Enter your choice: 1
```

Then update the contact number.

Before

```
Profile
(null):UserID = P1
Name = Johnathan Alexander Doe
ICNO = 800101141239
Gender = Male
Position = Patient
ContactNumber = 60119876543
Email = johndoe@example.com
Address = Majestic Maxim Jalan 9/142, Taman Len Seng, Batu 9, Cheras, Kuala Lumpur
Username = John Doe
Password = ab7kl
```

After

```
(null):UserID = P1
Name = Johnathan Alexander Doe
ICNO = 800101141239
Gender = Male
Position = Patient
ContactNumber = 60162323245
Email = johndoe@example.com
Address = Majestic Maxim Jalan 9/142, Taman Len Seng, Batu 9, Cheras, Kuala Lumpur
Username = John Doe
Password = ab7kl

Do you want to have action for Profile? (Y/N):
```

In SQLite, the contact number will be updated.

Before

|   | UserID | Name                    | ICNO         | Gender | Position | ContactNumber | Email                  | Address                                 | Username    | Password |
|---|--------|-------------------------|--------------|--------|----------|---------------|------------------------|-----------------------------------------|-------------|----------|
|   | Filter | Filter                  | Filter       | Filter | Filter   | Filter        | Filter                 | Filter                                  | Filter      | Filter   |
| 1 | P1     | Johnathan Alexander Doe | 800101141239 | Male   | Patient  | 60119876543   | johndoe@example.com    | Majestic Maxim Jalan 9/142, Taman ...   | John Doe    | ab7kl    |
| 2 | P2     | Jane Elizabeth Smith    | 850202105678 | Female | Patient  | 60145678901   | janesmith@example.com  | 7 Jalan Dataran Cheras 6, Balakong, ... | Jane Smith  | x9mn124  |
| 3 | P3     | Alice Victoria Brown    | 880303089102 | Female | Patient  | 60171234567   | alicebrown@example.com | Jalan Tasik Utama, Sungai Besi, ...     | Alice Brown | 1uz9x    |

After

|   | UserID | Name                    | ICNO         | Gender | Position | ContactNumber | Email                  | Address                                 | Username    | Password |
|---|--------|-------------------------|--------------|--------|----------|---------------|------------------------|-----------------------------------------|-------------|----------|
|   | Filter | Filter                  | Filter       | Filter | Filter   | Filter        | Filter                 | Filter                                  | Filter      | Filter   |
| 1 | P1     | Johnathan Alexander Doe | 800101141239 | Male   | Patient  | 60162323245   | johndoe@example.com    | Majestic Maxim Jalan 9/142, Taman ...   | John Doe    | ab7kl    |
| 2 | P2     | Jane Elizabeth Smith    | 850202105678 | Female | Patient  | 60145678901   | janesmith@example.com  | 7 Jalan Dataran Cheras 6, Balakong, ... | Jane Smith  | x9mn124  |
| 3 | P3     | Alice Victoria Brown    | 880303089102 | Female | Patient  | 60171234567   | alicebrown@example.com | Jalan Tasik Utama, Sungai Besi, ...     | Alice Brown | 1uz9x    |

If the user does not type anything, it will show 'No fields were updated.'

```
Enter new values for the following fields. Leave empty to skip updating the field.
Name:
ICNo:
Gender:
ContactNumber: 60162323245
Email:
Address:
Username:
Password:
No fields were updated.

Do you want to have action for Profile? (Y/N):
```

If the IC number does not have exactly 12 digits, it will show 'Invalid IC Number. It must be exactly 12 digits.' Please try again.

```
Name: Jane Elizabeth Smith
ICNo: 4581634478
Invalid ICNo. It must be exactly 12 digits.
ICNo:
```

If the user types a word that is not 'Male' or 'Female', it will show 'Invalid Gender. Please enter 'Male', 'Female', or any lowercase variation.'

```
Gender: lelaki
Invalid Gender. Please enter 'Male', 'Female', or any lowercase variation.
Gender:
```

If the contact number does not have 11 or 12 digits, it will show 'Invalid Contact Number. It must be 11 or 12 digits.'

```
ContactNumber: 60124578
Invalid Contact Number. It must be 11 or 12 digits.
ContactNumber: |
```

If the user types an email without the '@' symbol, it will show 'Invalid Email. It must contain an '@' symbol.'

```
Email: jane
Invalid Email. It must contain an '@' symbol.
Email: |
```

#### 6.4.1.2 Delete Profile

Select option 2. Then, you will be asked, 'Do you want to delete your account (press Enter to exit the delete process)? (Y/N):'. Type 'Y', and after that, a message will appear saying 'Data deleted successfully.'

```
Select an action:
1. Edit User Profile
2. Delete User
0. Exit
Enter your choice: 2
Do you want to delete your account (blank to exit delete process)? (Y/N): Y
Data deleted successfully!
```

In SQLite, John Doe will not be shown because the record has already been deleted.

| User ID | Name                 | IC NO        | Gender | Position | Contact Number | Email                  | Address                                 | Username    | Password |
|---------|----------------------|--------------|--------|----------|----------------|------------------------|-----------------------------------------|-------------|----------|
| Filter  | Filter               | Filter       | Filter | Filter   | Filter         | Filter                 | Filter                                  | Filter      | Filter   |
| 1 P2    | Jane Elizabeth Smith | 850202105678 | Female | Patient  | 60145678901    | janesmith@example.com  | 7 Jalan Dataran Cheras 6, Balakong, ... | Jane Smith  | x9mn124  |
| 2 P3    | Alice Victoria Brown | 880303089102 | Female | Patient  | 60171234567    | alicebrown@example.com | Jalan Tasik Utama, Sungai Besi, ...     | Alice Brown | 1uz9x    |

After deleting the profile, go to option 1. The user profile will not be shown.

```
Patient Menu
 1. Profile
 2. Doctor Info
 3. Make Appointment
 4. Cancel Appointment
 5. Appointment History
 6. Medical Diagnosis
 7. Allergy
 8. Notices
 9. Give Feedback
10. Vaccination History
11. Billing
12. Medical Insurance
13. Emergency Tool
14. Project
 0. Exit
Enter your choice: 1
```

```
Profile
```

```
Do you want to have action for Profile? (Y/N):
```

#### 6.4.1.3 Exit

When the user types '0', they will be redirected to the patient menu.

```
Profile
(null):UserID = P1
Name = Johnathan Alexander Doe
ICNO = 800101141239
Gender = Male
Position = Patient
ContactNumber = 60119876543
Email = johndoe@example.com
Address = Majestic Maxim Jalan 9/142, Taman Len Seng, Batu 9, Cheras, Kuala Lumpur
Username = John Doe
Password = ab7kl

Do you want to have action for Profile? (Y/N): y

Select an action:
1. Edit User Profile
2. Delete User
0. Exit
Enter your choice: 0

Patient Menu
 1. Profile
 2. Doctor Info
 3. Make Appointment
 4. Cancel Appointment
 5. Appointment History
 6. Medical Diagnosis
 7. Allergy
 8. Notices
 9. Give Feedback
10. Vaccination History
11. Billing
12. Medical Insurance
13. Emergency Tool
14. Project
 0. Exit
Enter your choice: |
```

#### 6.4.2 Doctor Info

Choosing option 2 will show the doctor's information.

```
Doctor Info
(null):UserID = E1
Name = Johnathan Maxwell Smith
Gender = Male
Position = Doctor
ContactNumber = 60154489651
Email = john@example.com

(null):UserID = E2
Name = Matthew Harrison Wilson
Gender = Male
Position = Doctor
ContactNumber = 60166546545
Email = matthew@example.com
```

#### 6.4.3 Make Appointment

Choose 'S2' for the Vaccination, then enter the date and time.

```
Add New PatientsAppointment
*****
(null):ServiceID = S1
TypeOfService = Laboratory Services
Description = Diagnostic Services include blood and urine tests

(null):ServiceID = S2
TypeOfService = Vaccination
Description = Immunizations cover all ages

(null):ServiceID = S3
TypeOfService = Family Medicine
Description = Primary Care focuses on health, prevention, and chronic conditions

Enter Service ID ( valid options are: S1 S2 S3 ): S2
Enter Date(YYYY-MM-DD): 2024-08-09
Enter Time(HH:MM): 11:00
```

Choose the doctor you want. The system will then confirm the action as successful. After that, you will be asked if you want to make another appointment. If you choose 'Y', you can continue making appointments. If you choose 'N', you will return to the patient menu.

```
(null):DoctorScheduleID = D1
DoctorID = E1
Monday = Working
Tuesday = Working
Wednesday = Working
Thursday = Off
Friday = Off
Saturday = Off
Sunday = Off

(null):DoctorScheduleID = D2
DoctorID = E2
Monday = Off
Tuesday = Off
Wednesday = Off
Thursday = Working
Friday = Working
Saturday = Working
Sunday = Off

Enter Doctor ID ( valid options are: E1 E2 ): E2
PatientsAppointment added successfully!
Do you want to add more appointments? (Y/N): |
```

If you choose 'Y', you can continue to make another appointment.

```
Do you want to add more appointments? (Y/N): y

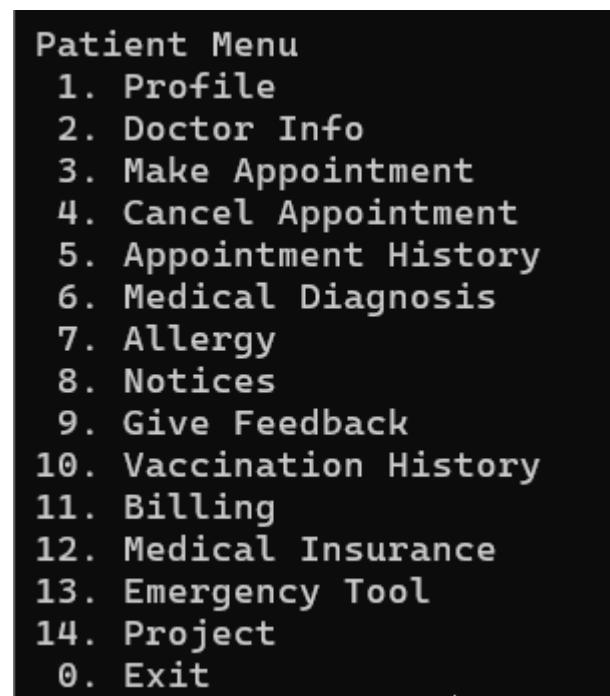
Add New PatientsAppointment
*****
(null):ServiceID = S1
TypeOfService = Laboratory Services
Description = Diagnostic Services include blood and urine tests

(null):ServiceID = S2
TypeOfService = Vaccination
Description = Immunizations cover all ages

(null):ServiceID = S3
TypeOfService = Family Medicine
Description = Primary Care focuses on health, prevention, and chronic conditions

Enter Service ID ( valid options are: S1 S2 S3 ): |
```

If you choose 'N', you will return to the patient menu.



This is the make appointment SQLite

Before

| AppointmentID | PatientID | ServiceID | Date       | Time   | Status   | DoctorID |
|---------------|-----------|-----------|------------|--------|----------|----------|
| Filter        | Filter    | Filter    | Filter     | Filter | Filter   | Filter   |
| 1 A1          | P1        | S3        | 2024-01-05 | 09:00  | Accessed | E1       |
| 2 A2          | P2        | S1        | 2024-02-08 | 10:00  | Canceled | E2       |
| 3 A3          | P2        | S1        | 2024-02-10 | 10:00  | Accessed | E2       |
| 4 A4          | P3        | S2        | 2024-03-15 | 12:00  | Accessed | E1       |

After

| AppointmentID | PatientID | ServiceID | Date       | Time   | Status   | DoctorID |
|---------------|-----------|-----------|------------|--------|----------|----------|
| Filter        | Filter    | Filter    | Filter     | Filter | Filter   | Filter   |
| 1 A1          | P1        | S3        | 2024-01-05 | 09:00  | Accessed | E1       |
| 2 A2          | P2        | S1        | 2024-02-08 | 10:00  | Canceled | E2       |
| 3 A3          | P2        | S1        | 2024-02-10 | 10:00  | Accessed | E2       |
| 4 A4          | P3        | S2        | 2024-03-15 | 12:00  | Accessed | E1       |
| 5 A5          | P2        | S2        | 2024-08-09 | 11:00  | Accessed | E2       |

If the user chooses '4', it will show 'Invalid Service ID. Please enter a valid Service ID from the list.'

```
Add New PatientsAppointment
*****
(null):ServiceID = S1
TypeOfService = Laboratory Services
Description = Diagnostic Services include blood and urine tests

(null):ServiceID = S2
TypeOfService = Vaccination
Description = Immunizations cover all ages

(null):ServiceID = S3
TypeOfService = Family Medicine
Description = Primary Care focuses on health, prevention, and chronic conditions

Enter Service ID ( valid options are: S1 S2 S3 ): 4
Invalid Service ID. Please enter a valid Service ID from the list.
Valid options are: S1 S2 S3 : Enter Service ID ( valid options are: S1 S2 S3 ): |
```

If the user does not enter the date in the correct format, it will show 'Invalid date format. Please enter a valid date (YYYY-MM-DD).

```
Enter Date(YYYY-MM-DD): 2024
Invalid date format. Please enter a valid date (YYYY-MM-DD).
Enter Date(YYYY-MM-DD): |
```

If the user does not enter the time in the correct format, it will show 'Invalid time format. Please enter a valid time (HH:MM).

```
Enter Time(HH:MM): 1212
Invalid date format. Please enter a valid date (HH:MM).
Enter Time(HH:MM): |
```

If the user types 'E4', it will show 'Invalid Doctor ID. Please enter a valid Doctor ID from the list.'

```
Enter Doctor ID ( valid options are: E1 E2 ): E4
Invalid Doctor ID. Please enter a valid Doctor ID from the list.
Valid options are: E1 E2 : Enter Doctor ID ( valid options are: E1 E2 ): |
```

## 6.4.4 Cancel Appointment

### 6.4.4.1 Add New Cancel Appointment

Choose option 1, then enter the appointment ID you want to cancel. After that, you'll be prompted to write the reason for the cancellation. Once the reason is entered, it will show 'CancelAppointment added successfully!'

```
Select an action:  
1. Add New Canceled Appoiment  
2. Edit Canceled Appoiment  
0. Exit  
Enter your choice: 1  
  
Add New Cancel Appointment  
*****  
Generated CancelAppointmentID: CA2  
Enter Appointment ID (valid options are: A2 A3 A5 ): A5  
Enter Cancel Reason: Because I booking wrong date  
CancelAppointment added successfully!
```

This is the SQLite for cancelling appointments.

Before

|   | CancelAppointmentID | AppointmentID | DoctorID | Reason               |
|---|---------------------|---------------|----------|----------------------|
|   | Filter              | Filter        | Filter   | Filter               |
| 1 | CA1                 | A2            | E2       | Scheduling conflicts |

After

|   | CancelAppointmentID | AppointmentID | DoctorID | Reason               |
|---|---------------------|---------------|----------|----------------------|
|   | Filter              | Filter        | Filter   | Filter               |
| 1 | CA1                 | A2            | E2       | Scheduling conflicts |
| 2 | CA2                 | A5            | E2       | Because I bookin...  |

After that, the system will show the cancellation appointment history. If you choose 'Y', you can continue to make a cancellation appointment. If you choose 'N', you will return to the patient menu.

```
PatientsAppointment status updated to 'Cancelled'!
Cancel Appointment ID: CA1
Appointment ID: A2
Doctor ID: E2
Reason: Scheduling conflicts
```

```
Cancel Appointment ID: CA2
Appointment ID: A5
Doctor ID: E2
Reason: Because I booking wrong date
```

```
Do you want to add another Cancel Appointment? (Y/N): |
```

If you choose 'Y', you can continue to cancel another appointment.

```
Add New Cancel Appointment
*****
Generated CancelAppointmentID: CA3
Enter Appointment ID (valid options are: A2 A3 A5 ): A3
Enter Cancel Reason:
```

If you choose 'N', you will return to the patient menu.

```
Patient Menu
1. Profile
2. Doctor Info
3. Make Appointment
4. Cancel Appointment
5. Appointment History
6. Medical Diagnosis
7. Allergy
8. Notices
9. Give Feedback
10. Vaccination History
11. Billing
12. Medical Insurance
13. Emergency Tool
14. Project
0. Exit
```

If the appointment has already been cancelled and the user tries to enter it again, it will show 'This Appointment ID has already been cancelled. Update process exited.'

```
Add New Cancel Appointment
*****
Generated CancelAppointmentID: CA3
Enter Appointment ID (valid options are: A2 A3 A5 ): A2
This Appointment ID has already been cancelled. Update process exited.
```

#### 6.4.4.2 Edit Cancelled Appointment

Choose '2' to edit an appointment. You will need to provide the appointment ID, and the edited reason for the change.

```
Select an action:
1. Add New Canceled Appoinment
2. Edit Canceled Appoiment
0. Exit
Enter your choice: 2

Enter CancelAppointment ID to update: 1
Enter new values for the following fields. Leave empty to skip updating the field.
AppointmentID: 2
Reason: closed shop
Data updated successfully!
```

This is the SQLite of cancel appointment.

Before

|   | CancelAppointmentID | AppointmentID | DoctorID | Reason               |
|---|---------------------|---------------|----------|----------------------|
|   | Filter              | Filter        | Filter   | Filter               |
| 1 | CA1                 | A2            | E2       | Scheduling conflicts |

After

|   | CancelAppointmentID | AppointmentID | DoctorID | Reason      |
|---|---------------------|---------------|----------|-------------|
|   | Filter              | Filter        | Filter   | Filter      |
| 1 | CA1                 | A2            | E2       | closed shop |

If the ID entered is not correct, it will show 'No relevant search results found.'

```
Enter CancelAppointment ID to update: 5
No relevant search results found.
```

If the user does not type anything, it will show 'No fields were updated.'

```
Enter CancelAppointment ID to update: 1
Enter new values for the following fields. Leave empty to skip updating the
field.
AppointmentID:
Reason:
No fields were updated.
```

If the user enters a non-relevant cancellation appointment ID, it will show 'No relevant search results found.'

```
Patient Cancel Appointment
Cancel Appointment ID: CA1
Appointment ID: A2
Doctor ID: E2
Reason: closed shop

Do you want to have action for Cancel Appoinment? (Y/N): Y

Select an action:
1. Add New Canceled Appoinment
2. Edit Canceled Appoinment
0. Exit
Enter your choice: 2

Enter CancelAppointment ID to update: 3
No relevant search results found.
```

If the user enters a non-relevant appointment ID, it will show 'Invalid Appointment ID. Please enter a valid Appointment ID.'

```
AppointmentID: 4
Invalid AppointmentID. Please enter a valid AppointmentID.
AppointmentID: |
```

#### 6.4.5 Appointment History

Choose '5' user can view the appointment history prior to the current appointment.

```
Patient Appointment History
(null):AppointmentID = A2
PatientID = P2
ServiceID = S1
Date = 2024-02-08
Time = 10:00
Status = Canceled
DoctorID = E2

(null):AppointmentID = A3
PatientID = P2
ServiceID = S1
Date = 2024-02-10
Time = 10:00
Status = Cancelled
DoctorID = E2

(null):AppointmentID = A5
PatientID = P2
ServiceID = S2
Date = 2024-08-09
Time = 11:00
Status = Cancelled
DoctorID = E2

(null):AppointmentID = A6
PatientID = P2
ServiceID = S2
Date = 2024-12-12
Time = 12:00
Status = Accessed
DoctorID = E2

(null):AppointmentID = A7
PatientID = P2
ServiceID = S1
Date = 2024-11-11
Time = 11:11
Status = Accessed
DoctorID = E2

(null):AppointmentID = A8
PatientID = P2
ServiceID = S2
Date = 2024-11-11
Time = 12:11
Status = Accessed
DoctorID = E1
```

#### 6.4.6 Medical Diagnosis

Choose '6' user can view the patient's medical diagnosis information.

```
Patient Medical Diagnosis Info

(null):PatientDiagnosisID = PH2
PatientID = P2
DoctorID = E2
MedicalHistory = Diabetes
MedicationsTaken = Metformin, Insulin, DPP-4 inhibitors
Diagnosis = Iron deficiency anemia
LastUpdate = 2024-02-17
```

#### 6.4.7 Allergy

##### 6.4.7.1 Add New Patient Allergy

Choose '1', then enter the allergy name, severity, and reaction. After doing so, it will show 'Patient Allergy added successfully!'

```
Select an action:
1. Add New Patient Allergy
2. Edit Patient Allergy
3. Delete Patient Allergy
0. Exit
Enter your choice: 1
Enter Allergy Name: Asthma
Enter Severity: severe
Enter Reaction: cough
Patient Allergy added successfully!
```

This is the SQLite of allergy.

Before

| PatientAllergyID | PatientID | AllergyName | Severity | Reaction                                   |
|------------------|-----------|-------------|----------|--------------------------------------------|
| Filter           | Filter    | Filter      | Filter   | Filter                                     |
| 1 AI1            | P2        | Peanut      | Severe   | Anaphylaxis (life-threatening allergic ... |

After

| PatientAllergyID | PatientID | AllergyName | Severity | Reaction                                   |
|------------------|-----------|-------------|----------|--------------------------------------------|
| Filter           | Filter    | Filter      | Filter   | Filter                                     |
| 1 AI1            | P2        | Peanut      | Severe   | Anaphylaxis (life-threatening allergic ... |
| 2 AI2            | P2        | Asthma      | severe   | cough                                      |

#### 6.4.7.2 Edit Patient Allergy

Choose '2', then enter the allergy name, severity, and update the reaction. After doing so, it will show 'Data updated successfully!'

```
Select an action:  
1. Add New Patient Allergy  
2. Edit Patient Allergy  
3. Delete Patient Allergy  
0. Exit  
Enter your choice: 2  
Enter Patient Allergy ID to update (blank to exit update process): 2  
Enter new values for the following fields. Leave empty to skip updating the field.  
AllergyName: Asthma  
Severity: severe  
Reaction: cough and difficulty getting enough air.  
Data updated successfully!
```

This is the SQLite of allergy.

| PatientAllergyID | PatientID | AllergyName | Severity | Reaction                                   |
|------------------|-----------|-------------|----------|--------------------------------------------|
| Filter           | Filter    | Filter      | Filter   | Filter                                     |
| 1 AI1            | P2        | Peanut      | Severe   | Anaphylaxis (life-threatening allergic ... |
| 2 AI2            | P2        | Asthma      | severe   | cough                                      |

After

| PatientAllergyID | PatientID | AllergyName | Severity | Reaction ▾1                                      |
|------------------|-----------|-------------|----------|--------------------------------------------------|
| Filter           | Filter    | Filter      | Filter   | Filter                                           |
| 1 AI1            | P2        | Peanut      | Severe   | Anaphylaxis (life-threatening allergic reaction) |
| 2 AI2            | P2        | Asthma      | severe   | cough and difficulty getting enough air.         |

#### 6.4.7.3 Delete Patient Allergy

Choose '2' to delete an allergy. After doing so, it will show 'Data deleted successfully!'

```
Select an action:  
1. Add New Patient Allergy  
2. Edit Patient Allergy  
3. Delete Patient Allergy  
0. Exit  
Enter your choice: 3  
Enter Patient Allergy ID to delete (blank to exit delete process): 2  
Data deleted successfully!
```

## Before

|   | PatientAllergyID | PatientID | AllergyName | Severity | Reaction ▾ <sup>1</sup>                          |
|---|------------------|-----------|-------------|----------|--------------------------------------------------|
|   | Filter           | Filter    | Filter      | Filter   | Filter                                           |
| 1 | AI1              | P2        | Peanut      | Severe   | Anaphylaxis (life-threatening allergic reaction) |
| 2 | AI2              | P2        | Asthma      | severe   | cough and difficulty getting enough air.         |

## After

|   | PatientAllergyID | PatientID | AllergyName | Severity | Reaction ▾ <sup>1</sup>                          |
|---|------------------|-----------|-------------|----------|--------------------------------------------------|
|   | Filter           | Filter    | Filter      | Filter   | Filter                                           |
| 1 | AI1              | P2        | Peanut      | Severe   | Anaphylaxis (life-threatening allergic reaction) |

#### 6.4.7.4 Exit

When the user types '0', they will be redirected to the patient menu.

```
Do you want to have action for Allergy? (Y/N): y

Select an action:
1. Add New Patient Allergy
2. Edit Patient Allergy
3. Delete Patient Allergy
0. Exit
Enter your choice: 0

Patient Menu
1. Profile
2. Doctor Info
3. Make Appointment
4. Cancel Appointment
5. Appointment History
6. Medical Diagnosis
7. Allergy
8. Notices
9. Give Feedback
10. Vaccination History
11. Billing
12. Medical Insurance
13. Emergency Tool
14. Project
0. Exit
Enter your choice: |
```

#### 6.4.8 Notices

Choose '8' to view the notice.

```
Notices
(null):NoticeID = N1
Description = Update emergency tools

(null):NoticeID = N2
Description = Vaccination Information

(null):NoticeID = N3
Description = Know more about your health
```

#### 6.4.9 Give Feedback

##### 6.4.9.1 Add New Feedback

Select '1' to add new feedback.

```
Select an action:
1. Add New Feedback
2. Edit Feedback
3. Delete Feedback
0. Exit
Enter your choice: 1

Add New Feedback
*****
Enter Comment: very good service
Enter Rating (0-5): 5
Enter Date (YYYY-MM-DD): 2024-12-12
Feedback added successfully!
```

This is SQLite of feedback.

Before

|   | FeedbackID | PatientID | Comment ▾1                                   | Rating | Date       |  |
|---|------------|-----------|----------------------------------------------|--------|------------|--|
|   | Filter     | Filter    | Filter                                       | Filter | Filter     |  |
| 1 | F2         | P2        | Good service, but waiting time was too long. | 3      | 2024-02-14 |  |
| 2 | F1         | P1        | Excellent care, very satisfied with service. | 5      | 2024-01-08 |  |

After

|   | FeedbackID | PatientID | Comment ▾1                                   | Rating | Date       |
|---|------------|-----------|----------------------------------------------|--------|------------|
|   | Filter     | Filter    | Filter                                       | Filter | Filter     |
| 1 | F3         | P2        | very good service                            | 5      | 2024-12-12 |
| 2 | F2         | P2        | Good service, but waiting time was too long. | 3      | 2024-02-14 |
| 3 | F1         | P1        | Excellent care, very satisfied with service. | 5      | 2024-01-08 |

#### 6.4.9.2 Edit Feedback

Select '2' to edit the feedback.

```
Select an action:  
1. Add New Feedback  
2. Edit Feedback  
3. Delete Feedback  
0. Exit  
Enter your choice: 2  
Enter Feedback ID to update (blank to exit update process): 3  
Enter new values for the following fields. Leave empty to skip updating the field.  
Comment: the doctor very detail and have the good service  
Rating: 5  
Date: 2024-12-13  
Data updated successfully!
```

Before

|   | FeedbackID | PatientID | Comment ▾1                                   | Rating | Date       |
|---|------------|-----------|----------------------------------------------|--------|------------|
|   | Filter     | Filter    | Filter                                       | Filter | Filter     |
| 1 | F3         | P2        | very good service                            | 5      | 2024-12-12 |
| 2 | F2         | P2        | Good service, but waiting time was too long. | 3      | 2024-02-14 |
| 3 | F1         | P1        | Excellent care, very satisfied with service. | 5      | 2024-01-08 |

After

|   | FeedbackID | PatientID | Comment ▾1                                       | Rating | Date       |
|---|------------|-----------|--------------------------------------------------|--------|------------|
|   | Filter     | Filter    | Filter                                           | Filter | Filter     |
| 1 | F1         | P1        | Excellent care, very satisfied with service.     | 5      | 2024-01-08 |
| 2 | F2         | P2        | Good service, but waiting time was too long.     | 5      | 2024-12-12 |
| 3 | F3         | P2        | the doctor very detail and have the good service | 5      | 2024-12-13 |

If the feedback ID is not relevant, it will show 'No relevant search results found.'

```
Select an action:  
1. Add New Feedback  
2. Edit Feedback  
3. Delete Feedback  
0. Exit  
Enter your choice: 2  
Enter Feedback ID to update (blank to exit update process): 6  
No relevant search results found.
```

Rating wrong

If the user enters a rating that is not between 0 and 5, it will show 'Invalid rating value. Please enter a value between 0 and 5.'

```
Do you want to have action for Feedback? (Y/N): y  
  
Select an action:  
1. Add New Feedback  
2. Edit Feedback  
3. Delete Feedback  
0. Exit  
Enter your choice: 2  
Enter Feedback ID to update (blank to exit update process): 1  
Enter new values for the following fields. Leave empty to skip updating the field.  
Comment: nice  
Rating: 15  
Invalid Rating value. Please enter a value between 0 and 5.  
Rating: 4  
Date: 2024-02-02  
Data updated successfully!
```

If the date is incorrect due to an invalid month, it will show 'Invalid Date. Please enter a valid date in the format YYYY-MM-DD.'

```
Date: 204  
Invalid Date. Please enter a valid date in the format YYYY-MM-DD.  
Date: 2024-20-20  
Invalid Date. Please enter a valid date in the format YYYY-MM-DD.  
Date: 2024-03-03
```

#### **6.4.9.3 Delete Feedback**

Select '3' to delete the feedback.

```
Select an action:  
1. Add New Feedback  
2. Edit Feedback  
3. Delete Feedback  
0. Exit  
Enter your choice: 3  
Enter Feedback ID to delete (blank to exit delete process): 3  
Data deleted successfully!
```

#### **6.4.9.4 Exit**

When the user types '0', they will be redirected to the patient menu.

```
Do you want to have action for Feedback? (Y/N): y  
  
Select an action:  
1. Add New Feedback  
2. Edit Feedback  
3. Delete Feedback  
0. Exit  
Enter your choice: 0  
  
Patient Menu  
1. Profile  
2. Doctor Info  
3. Make Appointment  
4. Cancel Appointment  
5. Appointment History  
6. Medical Diagnosis  
7. Allergy  
8. Notices  
9. Give Feedback  
10. Vaccination History  
11. Billing  
12. Medical Insurance  
13. Emergency Tool  
14. Project  
0. Exit  
Enter your choice: |
```

## 6.4.10 Vaccination History

### 6.4.10.1 Add New Vaccination History

Select '1' to add new vaccination records.

```
Select an action:  
1. Add New Vaccination History  
2. Edit Vaccination History  
3. Delete Vaccination History  
0. Exit  
Enter your choice: 1  
Enter Vaccine Type: Sinovac  
Enter Dose Number: 3  
Enter Administration Date (YYYY-MM-DD): 2024-11-11  
Patient Vaccination Records added successfully!
```

This is the SQLite of vaccination records.

Before

|   | VaccinationID | PatientID | VaccineType | DoseNumber | AdministrationDate |
|---|---------------|-----------|-------------|------------|--------------------|
|   | Filter        | Filter    | Filter      | Filter     | Filter             |
| 1 | V1            | P1        | AZ          | First dose | 2021-09-16         |
| 2 | V2            | P3        | Pfizer      | First dose | 2021-10-01         |

After

|   | VaccinationID | PatientID | VaccineType | DoseNumber | AdministrationDate |
|---|---------------|-----------|-------------|------------|--------------------|
|   | Filter        | Filter    | Filter      | Filter     | Filter             |
| 1 | V1            | P1        | AZ          | First dose | 2021-09-16         |
| 2 | V2            | P3        | Pfizer      | First dose | 2021-10-01         |
| 3 | V3            | P2        | Sinovac     | 3          | 2024-11-11         |

#### 6.4.10.2 Edit Vaccination History

Select '2' to update the vaccination details.

```
Select an action:  
1. Add New Vaccination History  
2. Edit Vaccination History  
3. Delete Vaccination History  
0. Exit  
Enter your choice: 2  
Enter Vaccination ID to update (blank to exit update process): 3  
Enter new values for the following fields. Leave empty to skip updating the field.  
VaccineType: Sinovac China  
DoseNumber: 3  
AdministrationDate: 2024-11-11  
Data updated successfully!
```

Before

|   | VaccinationID | PatientID | VaccineType | DoseNumber | AdministrationDate |
|---|---------------|-----------|-------------|------------|--------------------|
|   | Filter        | Filter    | Filter      | Filter     | Filter             |
| 1 | V1            | P1        | AZ          | First dose | 2021-09-16         |
| 2 | V2            | P3        | Pfizer      | First dose | 2021-10-01         |
| 3 | V3            | P2        | Sinovac     | 3          | 2024-11-11         |

After

|   | VaccinationID | PatientID | VaccineType   | DoseNumber | AdministrationDate |
|---|---------------|-----------|---------------|------------|--------------------|
|   | Filter        | Filter    | Filter        | Filter     | Filter             |
| 1 | V1            | P1        | AZ            | First dose | 2021-09-16         |
| 2 | V2            | P3        | Pfizer        | First dose | 2021-10-01         |
| 3 | V3            | P2        | Sinovac China | 3          | 2024-11-11         |

#### 6.4.10.3 Delete Vaccination History

Select '3' to delete the vaccination history.

```
Select an action:  
1. Add New Vaccination History  
2. Edit Vaccination History  
3. Delete Vaccination History  
0. Exit  
Enter your choice: 3  
Enter Patient Allergy ID to delete (blank to exit delete process): 3  
Data deleted successfully!
```

Before

|   | VaccinationID | PatientID | VaccineType   | DoseNumber | AdministrationDate |
|---|---------------|-----------|---------------|------------|--------------------|
|   | Filter        | Filter    | Filter        | Filter     | Filter             |
| 1 | V1            | P1        | AZ            | First dose | 2021-09-16         |
| 2 | V2            | P3        | Pfizer        | First dose | 2021-10-01         |
| 3 | V3            | P2        | Sinovac China | 3          | 2024-11-11         |

After

|   | VaccinationID | PatientID | VaccineType | DoseNumber | AdministrationDate |
|---|---------------|-----------|-------------|------------|--------------------|
|   | Filter        | Filter    | Filter      | Filter     | Filter             |
| 1 | V1            | P1        | AZ          | First dose | 2021-09-16         |
| 2 | V2            | P3        | Pfizer      | First dose | 2021-10-01         |

#### **6.4.10.4 Exit**

When the user types '0', they will be redirected to the patient menu.

```
Do you want to have action for Vaccination History? (Y/N): y

Select an action:
1. Add New Vaccination History
2. Edit Vaccination History
3. Delete Vaccination History
0. Exit
Enter your choice: 0

Patient Menu
1. Profile
2. Doctor Info
3. Make Appointment
4. Cancel Appointment
5. Appointment History
6. Medical Diagnosis
7. Allergy
8. Notices
9. Give Feedback
10. Vaccination History
11. Billing
12. Medical Insurance
13. Emergency Tool
14. Project
0. Exit
Enter your choice: |
```

#### **6.4.11 Billing**

Choose '11' to view the billing details. To confirm payment, type the bill ID. After doing so, it will show 'Payment successful.'

```
Billing
(null):BillID = B2
PatientID = P2
DateIssued = 2024-02-16
SubTotal = RM 873.33
DiscountPercentage = 10%
TotalAmount = RM 786.00
PaymentStatus = Unpaid

Do you want to make a payment? (Y/N): y
Enter the BillID to make payment: 2
Payment successful.
```

Before

|   | BillID | PatientID | DateIssued | SubTotal  | DiscountPercentage | TotalAmount | PaymentStatus |
|---|--------|-----------|------------|-----------|--------------------|-------------|---------------|
|   | Filter | Filter    | Filter     | Filter    | Filter             | Filter      | Filter        |
| 1 | B1     | P1        | 2024-07-01 | RM 542.00 | NULL               | RM 542.00   | Unpaid        |
| 2 | B2     | P2        | 2024-02-16 | RM 873.33 | 10%                | RM 786.00   | Unpaid        |
| 3 | B3     | P3        | 2024-03-18 | RM 206.00 | NULL               | RM 206.00   | Unpaid        |

After

|   | BillID | PatientID | DateIssued | SubTotal  | DiscountPercentage | TotalAmount | PaymentStatus |
|---|--------|-----------|------------|-----------|--------------------|-------------|---------------|
|   | Filter | Filter    | Filter     | Filter    | Filter             | Filter      | Filter        |
| 1 | B1     | P1        | 2024-07-01 | RM 542.00 | NULL               | RM 542.00   | Unpaid        |
| 2 | B2     | P2        | 2024-02-16 | RM 873.33 | 10%                | RM 786.00   | Paid          |
| 3 | B3     | P3        | 2024-03-18 | RM 206.00 | NULL               | RM 206.00   | Unpaid        |

If the billing ID has already been paid and is typed again, it will show 'Payment successful'.

```
Enter the BillID to make payment: 2
Payment successful.
```

If the billing ID is not relevant, it will show 'Bill ID not found'.

```
Enter the BillID to make payment: 4
BillID not found.
```

#### 6.4.12 Medical Insurance

##### 6.4.12.1 Add New Medical Insurance

Choose '1' to add new medical insurance.

```
Select an action:
1. Add New Medical Insurance
2. Edit Medical Insurance
3. Delete Medical Insurance
0. Exit
Enter your choice: 1
Enter Insurance Provider: prudential
Enter Policy Number: 123456789
Enter Deductible: 9000000
Patient Medical Insurance added successfully!
```

This is SQLite of medical insurance

Before

|   | MedicalInsuranceID | PatientID | InsuranceProvider | PolicyNumber | Deductible |
|---|--------------------|-----------|-------------------|--------------|------------|
|   | Filter             | Filter    | Filter            | Filter       | Filter     |
| 1 | INS1               | P1        | Allianz           | A12392       | 400000     |
| 2 | INS2               | P2        | AIA               | AIA193842    | 600000     |

After

|   | MedicalInsuranceID | PatientID | InsuranceProvider | PolicyNumber | Deductible |
|---|--------------------|-----------|-------------------|--------------|------------|
|   | Filter             | Filter    | Filter            | Filter       | Filter     |
| 1 | INS1               | P1        | Allianz           | A12392       | 400000     |
| 2 | INS2               | P2        | AIA               | AIA193842    | 600000     |
| 3 | INS3               | P2        | prudential        | 123456789    | 9000000    |

#### 6.4.12.2 Edit Medical Insurance

Choose '2' to edit the medical insurance.

```
Select an action:  
1. Add New Medical Insurance  
2. Edit Medical Insurance  
3. Delete Medical Insurance  
0. Exit  
Enter your choice: 2  
Enter MedicalInsurance ID to update (blank to exit update process): 3  
Enter new values for the following fields. Leave empty to skip updating the field.  
InsuranceProvider: prudential  
PolicyNumber: 123456789  
Deductible: 900000000  
Data updated successfully!
```

Before

|   | MedicalInsuranceID | PatientID | InsuranceProvider | PolicyNumber | Deductible |
|---|--------------------|-----------|-------------------|--------------|------------|
|   | Filter             | Filter    | Filter            | Filter       | Filter     |
| 1 | INS1               | P1        | Allianz           | A12392       | 400000     |
| 2 | INS2               | P2        | AIA               | AIA193842    | 600000     |
| 3 | INS3               | P2        | prudential        | 123456789    | 9000000    |

After

|   | MedicalInsuranceID | PatientID | InsuranceProvider | PolicyNumber | Deductible |
|---|--------------------|-----------|-------------------|--------------|------------|
|   | Filter             | Filter    | Filter            | Filter       | Filter     |
| 1 | INS1               | P1        | Allianz           | A12392       | 400000     |
| 2 | INS2               | P2        | AIA               | AIA193842    | 600000     |
| 3 | INS3               | P2        | prudential        | 123456789    | 9000000000 |

If the ID typed is not relevant, it will show 'No relevant search results found.'

```
Enter MedicalInsurance ID to update (blank to exit update process): 6
No relevant search results found.
```

If nothing is typed, it will show 'No fields were updated.'

```
InsuranceProvider:
PolicyNumber:
Deductible:
No fields were updated.
```

#### 6.4.12.3 Exit

#### 6.4.13 Emergency Tool

Choose '13' to view the emergency tool information.

```
Emergency Tool
(null):EmergencyToolsID = ET1
Type = Contact Information
Tool = Emergency Number
EmergencyNumber = 03-86457894

(null):EmergencyToolsID = ET2
Type = Transportation
Tool = Ambulance Department Service
EmergencyNumber = 999
```

#### **6.4.14 Project**

Choose '14' to view the project.

```
Project
(null):ProjectID = PRO1
Title = New Vaccination Guidelines
Description = Release of updated guidelines for vaccination protocols
DateOfPublication = 2024-01-18

(null):ProjectID = PRO3
Title = Mental Health Awareness
Description = Campaign to promote mental health awareness
DateOfPublication = 2024-03-08
```

## 6.5 Doctor Menu

### 6.5.1 Profile

Select 1 for profile

```
Doctor Menu
*****
1. Profile
2. Patient Appointment Schedule
3. Patient Medical Diagnosis
4. Follow-up Management
5. Vaccination Records
6. Doctor Schedule
7. Patient Allergies History
8. Notices
0. Exit
Enter your choice: 1

Profile
*****
1. View Profile
2. Edit Profile
3. Delete Profile
0. Exit
Enter your choice:
```

Select 1 to view the doctor profile. The profile includes the user ID, name, IC number, gender, position, contact number, email, address, username, and password.

```
Profile
*****
1. View Profile
2. Edit Profile
3. Delete Profile
0. Exit
Enter your choice: 1
-----
User ID      : E1
Name         : Johnathan Maxwell Smith
IC No        : 750123105671
Gender       : Male
Position     : Doctor
Contact Number : 60154489651
Email        : john@example.com
Address      : Jalan Setia Tropika U13/20, Setia Eco Park, Setia Eco Park, Shah Alam, Selangor
Username     : John Smith
Password     : 156cd
-----
```

Select 2 to edit the doctor profile. All the information of the doctor can be edited except the position where the user doesn't have the privileges to modify. Here I edit the contract number from 60154489651 to 60101574826.

```
1. View Profile
2. Edit Profile
3. Delete Profile
0. Exit
Enter your choice: 2
Enter new/current Name: Johnathan Maxwell Smith
Enter new/current ICNO (12 digits): 750123105671
Enter new Gender (valid options are: Male, Female): Male
Enter new/current Contact Number (11-12 digits): 60101574826
Enter new/current Email (must include '@'): john@example.com
Enter new/current Address: Jalan Setia Tropika U13/20, Setia Eco Park, Setia Eco Park, Shah Alam, Selangor
Enter new/current Username: John Smith
Enter new/current Password: 156cd
Updated user with ID 'E1' successfully.
```

## Result

|                |   |                                                                                 |
|----------------|---|---------------------------------------------------------------------------------|
| User ID        | : | E1                                                                              |
| Name           | : | Johnathan Maxwell Smith                                                         |
| IC No          | : | 750123105671                                                                    |
| Gender         | : | Male                                                                            |
| Position       | : | Doctor                                                                          |
| Contact Number | : | 60101574826                                                                     |
| Email          | : | john@example.com                                                                |
| Address        | : | Jalan Setia Tropika U13/20, Setia Eco Park, Setia Eco Park, Shah Alam, Selangor |
| Username       | : | John Smith                                                                      |
| Password       | : | 156cd                                                                           |

The data stored on database Sqlite3 also will update when user edit their profile.

| User ID | Name                    | IC NO        | Gender | Position | Contact Number | Email            | Address                                  | Username   | Password |
|---------|-------------------------|--------------|--------|----------|----------------|------------------|------------------------------------------|------------|----------|
| Filter  | Filter                  | Filter       | Filter | Filter   | Filter         | Filter           | Filter                                   | Filter     | Filter   |
| E1      | Johnathan Maxwell Smith | 750123105671 | Male   | Doctor   | 60101574826    | john@example.com | Jalan Setia Tropika U13/20, Setia Eco... | John Smith | 156cd    |

## Invalid Input

```
Enter new/current Name:
Name cannot be empty. Please enter a valid name.
```

The Name can't be blank.

```
Invalid ICNO. It must be exactly 12 digits.
Enter new/current ICNO (12 digits): 7501231056
```

The IC number needs to enter 12 digits.

```
Enter new Gender (valid options are: Male, Female): maal
Invalid gender. Please enter 'Male' or 'Female'.
```

The gender only can enter ‘male’, ‘female’, ‘Male’ , or ‘Female’.

```
Enter new/current Contact Number (11-12 digits): 0101574826
Invalid contact number. It must be between 11 and 12 digits.
```

The contact number should be entered 11-12 digits.

```
Enter new/current Email (must include '@'): johnexample.com
Invalid email. It must include '@'.
```

The email must include ‘@’.

```
Enter new/current Address:
Address cannot be empty. Please enter a valid address.
```

The Address can't be blank.

```
Enter new/current Username:
Username cannot be empty. Please enter a valid username.
```

The Username can't be blank.

```
Enter new/current Username:
Username cannot be empty. Please enter a valid username.
```

The Password can't be blank.

```
Enter new/current Password:
Password cannot be empty. Please enter a valid password.
```

Select 3 for delete profile. Delete profile means delete the account. After deletion, the user can't login with the current username, and password.

- 1. View Profile
- 2. Edit Profile
- 3. Delete Profile
- 0. Exit

```
Enter your choice: 3
```

```
Deleted user with ID 'E1' successfully.
```

Select 0 for exit.

## Profile

\*\*\*\*\*

1. View Profile
2. Edit Profile
3. Delete Profile
0. Exit

Enter your choice: 0

## Doctor Menu

\*\*\*\*\*

1. Profile
2. Patient Appointment Schedule
3. Patient Medical Diagnosis
4. Follow-up Management
5. Vaccination Records
6. Doctor Schedule
7. Patient Allergies History
8. Notices
0. Exit

### 6.5.2 Patient Appointment Schedule

Select 1 for viewing patient appointments. Here the doctor only can view own patient appointments. Now I login to the doctor account and the doctor ID is “E1”. So, here only will show the patient appointment that selected this doctor.

#### Patient Appointment Schedule

\*\*\*\*\*

1. View Appointment
2. View Cancelled Appointment
0. Exit

Enter your choice: 1

| Appointment ID | Patient ID | Service ID | Date       | Time  | Status   | Doctor ID |
|----------------|------------|------------|------------|-------|----------|-----------|
| A1             | P1         | S3         | 2024-01-05 | 09:00 | Accessed | E1        |
| A4             | P3         | S2         | 2024-03-15 | 12:00 | Accessed | E1        |

All the patient appointment records on database Sqlite3.

| AppointmentID | PatientID | ServiceID | Date       | Time   | Status   | DoctorID |
|---------------|-----------|-----------|------------|--------|----------|----------|
| Filter        | Filter    | Filter    | Filter     | Filter | Filter   | Filter   |
| A1            | P1        | S3        | 2024-01-05 | 09:00  | Accessed | E1       |
| A2            | P2        | S1        | 2024-02-08 | 10:00  | Canceled | E2       |
| A3            | P2        | S1        | 2024-02-10 | 10:00  | Accessed | E2       |
| A4            | P3        | S2        | 2024-03-15 | 12:00  | Accessed | E1       |

Select 2 for view cancelled appointment. Due to the patient which selected this doctor didn't have to cancel any appointment, so here will not show anything.

|                                                           |
|-----------------------------------------------------------|
| Patient Appointment Schedule                              |
| *****                                                     |
| 1. View Appointment                                       |
| 2. View Cancelled Appointment                             |
| 0. Exit                                                   |
| Enter your choice: 2                                      |
| -----                                                     |
| Cancel Appointment ID   Appointment ID  Doctor ID  Reason |
| -----                                                     |

The other doctor with doctor ID “E2”, which can view the cancelled appointment by his patient.

|                                                           |
|-----------------------------------------------------------|
| Patient Appointment Schedule                              |
| *****                                                     |
| 1. View Appointment                                       |
| 2. View Cancelled Appointment                             |
| 0. Exit                                                   |
| Enter your choice: 2                                      |
| -----                                                     |
| Cancel Appointment ID   Appointment ID  Doctor ID  Reason |
| -----                                                     |
| CA1   A2   E2   Scheduling conflicts                      |

All the patients cancelled appointment records on database Sqlite3.

| CancelAppointmentID | AppointmentID | DoctorID | Reason               |
|---------------------|---------------|----------|----------------------|
| Filter              | Filter        | Filter   | Filter               |
| CA1                 | A2            | E2       | Scheduling conflicts |

Select 0 for exit.

## Patient Appointment Schedule

---

- 1. View Appointment
- 2. View Cancelled Appointment
- 0. Exit

Enter your choice: 0

## Doctor Menu

---

- 1. Profile
- 2. Patient Appointment Schedule
- 3. Patient Medical Diagnosis
- 4. Follow-up Management
- 5. Vaccination Records
- 6. Doctor Schedule
- 7. Patient Allergies History
- 8. Notices
- 0. Exit

Enter your choice: |

### 6.5.3 Patient Medical Diagnosis

Select 1 to add a patient medical diagnosis. For the patient ID doctor only needs to enter the integer, it will auto generate the ‘P’ at front.

```
Patient Medical Diagnosis
*****
1. Add Medical Diagnosis
2. View Medical Diagnosis
3. Update Medical Diagnosis
4. Delete Medical Diagnosis
0. Exit
Enter your choice: 1

Add Patient Medical Diagnosis
*****
Enter new Patient ID ( valid options are: P1 P3 ): 1
Enter Medical History: Hypertension
Enter Medications Taken: Lisinopril, Amlodipine, Losartan, Metoprolol
Enter Diagnosis: Dengue fever
Enter Last Update (YYYY-MM-DD): 2024-01-08
Medical diagnosis 'PH1' added successfully.
Do you want to add another medical diagnosis? (Y/N): y
```

For the asking “Do you want to add another medical diagnosis? (Y/N)”, if a user enters ‘y’ or ‘Y’, the user can add another more patient medical diagnosis.

```
Do you want to add another medical diagnosis? (Y/N): y

Add Patient Medical Diagnosis
*****
Enter new Patient ID ( valid options are: P1 P3 ): 3
Enter Medical History: Diabetes
Enter Medications Taken: Metformin, Insulin
Enter Diagnosis: Type 2 Diabetes
Enter Last Update (YYYY-MM-DD): 2024-08-15
Medical diagnosis 'PH4' added successfully.
Do you want to add another medical diagnosis? (Y/N): |
```

Instead, the doctor selects ‘n’ or ‘N’, and returns the patient to the medical diagnosis menu.

```
Do you want to add another medical diagnosis? (Y/N): n

Patient Medical Diagnosis
*****
1. Add Medical Diagnosis
2. View Medical Diagnosis
3. Update Medical Diagnosis
4. Delete Medical Diagnosis
0. Exit
Enter your choice: |
```

Invalid Input

Disallow to enter other patient IDs and only can enter integers.

```
Enter new Patient ID ( valid options are: P1 P3 ): 2
Invalid patient ID. Please enter a valid patient ID from the list.
Valid options are: P1 P3
Enter new Patient ID ( valid options are: P1 P3 ): q
Invalid patient ID. Please enter a valid patient ID from the list.
Valid options are: P1 P3
```

Date must enter in (YYYY-MM-DD) format.

```
Enter Last Update (YYYY-MM-DD): 2024-13-29
Invalid date format. Please enter a valid date (YYYY-MM-DD).
Enter Last Update (YYYY-MM-DD): 20240815
Invalid date format. Please enter a valid date (YYYY-MM-DD).
```

Select 2 to view the patient's medical diagnosis. Doctors are only allowed to view their own patient medical diagnosis.

```
Patient Medical Diagnosis
*****
1. Add Medical Diagnosis
2. View Medical Diagnosis
3. Update Medical Diagnosis
4. Delete Medical Diagnosis
0. Exit
Enter your choice: 2

Patient Diagnosis ID | Patient ID | Doctor ID | Medical History | Medications Taken | Diagnosis | Last Update
-----|-----|-----|-----|-----|-----|-----|-----|
PH1 | P1 | E1 | Hypertension | Lisinopril, Amlodipine, Losartan, Metoprolol | Dengue fever | 2024-01-08
PH3 | P3 | E1 | Asthma | Albuterol (short-acting), Fluticasone | Get HPV vaccine | 2024-03-15
PH4 | P3 | E1 | Diabetes | Metformin, Insulin | Type 2 Diabetes | 2024-03-17
```

Put blank can exit the delete process.

| Patient Diagnosis ID | Patient ID | Doctor ID | Medical History | Medications Taken                            | Diagnosis       | Last Update |
|----------------------|------------|-----------|-----------------|----------------------------------------------|-----------------|-------------|
| PH1                  | P1         | E1        | Hypertension    | Lisinopril, Amlodipine, Losartan, Metoprolol | Dengue fever    | 2024-01-08  |
| PH3                  | P3         | E1        | Asthma          | Albuterol (short-acting), Fluticasone        | Get HPV vaccine | 2024-03-15  |

Enter Patient Diagnosis ID to update (Blank for exit):

Patient Medical Diagnosis  
\*\*\*\*\*  
1. Add Medical Diagnosis  
2. View Medical Diagnosis  
3. Update Medical Diagnosis  
4. Delete Medical Diagnosis  
0. Exit  
Enter your choice: |

Select 3 for updating patient medical diagnosis. Here has updated the PH4's date from 2024-03-17 to 2024-03-28.

| Patient Diagnosis ID | Patient ID | Doctor ID | Medical History | Medications Taken                            | Diagnosis       | Last Update |
|----------------------|------------|-----------|-----------------|----------------------------------------------|-----------------|-------------|
| PH1                  | P1         | E1        | Hypertension    | Lisinopril, Amlodipine, Losartan, Metoprolol | Dengue fever    | 2024-01-08  |
| PH3                  | P3         | E1        | Asthma          | Albuterol (short-acting), Fluticasone        | Get HPV vaccine | 2024-03-15  |
| PH4                  | P3         | E1        | Diabetes        | Metformin, Insulin                           | Type 2 Diabetes | 2024-03-17  |

Enter Patient Diagnosis ID to update (Blank for exit): 4

Enter new Patient ID ( valid options are: P1 P3 ): 3

Enter new Doctor ID ( valid options are: E1 E2 ): 1

Enter new Medical History: Diabetes

Enter new Medications Taken: Metformin, Insulin

Enter new Disease: Type 2 Diabetes

Enter Last Update (YYYY-MM-DD): 2024-03-28

Updated diagnosis with ID PH4 successfully.

Result on database Sqlite3.

| PatientDiagnosisID | PatientID | DoctorID | MedicalHistory | MedicationsTaken                      | Diagnosis       | LastUpdate |
|--------------------|-----------|----------|----------------|---------------------------------------|-----------------|------------|
| Filter             | Filter    | Filter   | Filter         | Filter                                | Filter          | Filter     |
| 1 PH1              | P1        | E1       | Hypertension   | Lisinopril, Amlodipine, Losartan, ... | Dengue fever    | 2024-01-08 |
| 2 PH2              | P2        | E2       | Diabetes       | Metformin, Insulin, DPP-4 inhibitors  | Iron deficiency | 2024-02-17 |
| 3 PH3              | P3        | E1       | Asthma         | Albuterol (short-acting), Fluticasone | Get HPV vaccine | 2024-03-15 |
| 4 PH4              | P3        | E1       | Diabetes       | Metformin, Insulin                    | Type 2 Diabetes | 2024-03-28 |

If a doctor tries to enter the patient diagnosis ID not in the list will show he is not authorized to perform an update.

| Patient Diagnosis ID | Patient ID | Doctor ID | Medical History | Medications Taken                            | Diagnosis       | Last Update |
|----------------------|------------|-----------|-----------------|----------------------------------------------|-----------------|-------------|
| PH1                  | P1         | E1        | Hypertension    | Lisinopril, Amlodipine, Losartan, Metoprolol | Dengue fever    | 2024-01-08  |
| PH3                  | P3         | E1        | Asthma          | Albuterol (short-acting), Fluticasone        | Get HPV vaccine | 2024-03-15  |

Enter Patient Diagnosis ID to update (Blank for exit): 2

You are not authorized to update this diagnosis.

When the patient diagnosis ID is updated as “E2” which is another doctor ID, the current doctor can’t view the patient diagnosis information any more.

| Patient Medical Diagnosis                                |            |           |                 |                                              |                 |             |
|----------------------------------------------------------|------------|-----------|-----------------|----------------------------------------------|-----------------|-------------|
| *****                                                    |            |           |                 |                                              |                 |             |
| 1. Add Medical Diagnosis                                 |            |           |                 |                                              |                 |             |
| 2. View Medical Diagnosis                                |            |           |                 |                                              |                 |             |
| 3. Update Medical Diagnosis                              |            |           |                 |                                              |                 |             |
| 4. Delete Medical Diagnosis                              |            |           |                 |                                              |                 |             |
| 0. Exit                                                  |            |           |                 |                                              |                 |             |
| Enter your choice: 3                                     |            |           |                 |                                              |                 |             |
| Patient Diagnosis ID                                     | Patient ID | Doctor ID | Medical History | Medications Taken                            | Diagnosis       | Last Update |
| PH1                                                      | P1         | E1        | Hypertension    | Lisinopril, Amlodipine, Losartan, Metoprolol | Dengue fever    | 2024-01-08  |
| PH3                                                      | P3         | E1        | Asthma          | Albuterol (short-acting), Fluticasone        | Get HPV vaccine | 2024-03-15  |
| PH4                                                      | P3         | E1        | Diabetes        | Metformin, Insulin                           | Type 2 Diabetes | 2024-03-28  |
| Enter Patient Diagnosis ID to update (Blank for exit): 4 |            |           |                 |                                              |                 |             |
| Enter new Patient ID ( valid options are: P1 P3 ): 3     |            |           |                 |                                              |                 |             |
| Enter new Doctor ID ( valid options are: E1 E2 ): 2      |            |           |                 |                                              |                 |             |
| Enter new Medical History: Diabetes                      |            |           |                 |                                              |                 |             |
| Enter new Medications Taken: Metformin, Insulin          |            |           |                 |                                              |                 |             |
| Enter new Disease: Type 2 Diabetes                       |            |           |                 |                                              |                 |             |
| Enter Last Update (YYYY-MM-DD): 2024-03-28               |            |           |                 |                                              |                 |             |
| Updated diagnosis with ID 'PH4' successfully.            |            |           |                 |                                              |                 |             |

Here are the results.

| Patient Diagnosis ID | Patient ID | Doctor ID | Medical History | Medications Taken                            | Diagnosis       | Last Update |
|----------------------|------------|-----------|-----------------|----------------------------------------------|-----------------|-------------|
| PH1                  | P1         | E1        | Hypertension    | Lisinopril, Amlodipine, Losartan, Metoprolol | Dengue fever    | 2024-01-08  |
| PH3                  | P3         | E1        | Asthma          | Albuterol (short-acting), Fluticasone        | Get HPV vaccine | 2024-03-15  |

Put blank can exit the delete process.

| Patient Diagnosis ID                                                                       | Patient ID | Doctor ID | Medical History | Medications Taken                            | Diagnosis       | Last Update |
|--------------------------------------------------------------------------------------------|------------|-----------|-----------------|----------------------------------------------|-----------------|-------------|
| PH1                                                                                        | P1         | E1        | Hypertension    | Lisinopril, Amlodipine, Losartan, Metoprolol | Dengue fever    | 2024-01-08  |
| PH3                                                                                        | P3         | E1        | Asthma          | Albuterol (short-acting), Fluticasone        | Get HPV vaccine | 2024-03-15  |
| Enter Patient Diagnosis ID to delete (number only, will auto-prefix PH, or blank to exit): |            |           |                 |                                              |                 |             |
| Patient Medical Diagnosis                                                                  |            |           |                 |                                              |                 |             |
| *****                                                                                      |            |           |                 |                                              |                 |             |

Invalid Input

Doctor ID only allows entering valid ID which is “E1” and “E2”.

```
Enter new Doctor ID ( valid options are: E1 E2 ): 3
Invalid doctor ID. Please enter a valid patient ID from the list.
Valid options are: E1 E2
```

Select 0 for exit.

```
Patient Medical Diagnosis
*****
1. Add Medical Diagnosis
2. View Medical Diagnosis
3. Update Medical Diagnosis
4. Delete Medical Diagnosis
0. Exit
Enter your choice: 0
```

```
Doctor Menu
*****
1. Profile
2. Patient Appointment Schedule
3. Patient Medical Diagnosis
4. Follow-up Management
5. Vaccination Records
6. Doctor Schedule
7. Patient Allergies History
8. Notices
0. Exit
Enter your choice: |
```

#### **6.5.4.1 Patient Follow-up**

Select 4 for follow-up management. After this will show the follow-up management menu for doctor to select patient follow-up.

```
Doctor Menu
*****
1. Profile
2. Patient Appointment Schedule
3. Patient Medical Diagnosis
4. Follow-up Management
5. Vaccination Records
6. Doctor Schedule
7. Patient Allergies History
8. Notices
0. Exit
Enter your choice: 4
```

```
Follow-up Management
*****
1. Patient Follow-up
2. Patient Admission Transfer
0. Back to Doctor Menu
Enter your choice: 1
```

Select 1 for add patient follow-up. Doctors only allow to add own patient's diagnosis ID.

```
Patient Follow-up
*****
1. Add Patient Follow-up
2. View Patient Follow-up
3. Update Patient Follow-up
4. Delete Patient Follow-up
0. Back to Follow-up Management
Enter your choice: 1

Add Patient Follow Up Record
*****
Enter Patient DiagnosisID (valid options are: PH1 PH3 ): 1
Enter Last Update (YYYY-MM-DD): 2024-01-07
Enter Description: Mild fever, joint pain; no rash
Enter Doctor Orders: Monitor platelets, rest. Follow-up in 1 week
Patient follow up 'FU1' added successfully.
Do you want to add another follow-up record? (Y/N): n
```

Invalid input

Patient diagnosis ID only can enter valid ID which are “PH1” and “PH3”.

```
Enter Patient DiagnosisID (valid options are: PH1 PH3 ): 2
Invalid Patient DiagnosisID. Please enter a valid ID from the list.
Valid options are: PH1 PH3
```

The date must enter in (YYYY-MM-DD) format.

```
Enter Last Update (YYYY-MM-DD): 20240321
Invalid date format. Please enter a valid date (YYYY-MM-DD).
```

Select 2 for view patient follow-up. Doctors only will vie own patient follow-up information.

```
Patient Follow-up
*****
1. Add Patient Follow-up
2. View Patient Follow-up
3. Update Patient Follow-up
4. Delete Patient Follow-up
0. Back to Follow-up Management
Enter your choice: 2

Follow Up Record ID	Patient Diagnosis ID	Follow-up Date	Description	Doctor Orders	Doctor ID
FU1 | PH1 | 2024-01-07 | Mild fever, joint pain; no rash | Monitor platelets, rest. Follow-up in 1 week | E1
```

Select 3 for update patient follow-up. Before performing an update, they will show the doctor the following record he added before. Update last update from “2024-01-15” to “2024-01-20” and “Respiratory pain; ongoing meds” to “Respiratory pain”.

```

Patient Follow-up
*****
1. Add Patient Follow-up
2. View Patient Follow-up
3. Update Patient Follow-up
4. Delete Patient Follow-up
0. Back to Follow-up Management
Enter your choice: 3
Follow Up Record ID |Patient Diagnosis ID |Follow-up Date| Description | Doctor Orders |Doctor ID
-----|-----|-----|-----|-----|-----|
FU1 | PH1 | 2024-01-07 | Mild fever, joint pain | Monitor platelets, rest. Follow-up in 1 week | E1
FU2 | PH3 | 2024-01-15 | Respiratory pain; ongoing meds | Continue meds; give HPV vaccine | E1
Available follow-ups for your patients:
Enter Follow Up Record ID to update (Blank for exit): 2
Enter new Patient Diagnosis ID (valid options are: PH1 PH3 ): 3
Enter new Last Update (YYYY-MM-DD): 2024-01-20
Enter new Description: Respiratory pain
Enter new Doctor Orders: Continue meds; give HPV vaccine
Enter new Doctor ID ( valid options are: E1 E2 ): 1
Updated patient follow up with ID 'FU2' successfully.

```

Result after updating successfully.

| Follow Up Record ID | Patient Diagnosis ID | Follow-up Date | Description            | Doctor Orders                                | Doctor ID |
|---------------------|----------------------|----------------|------------------------|----------------------------------------------|-----------|
| FU1                 | PH1                  | 2024-01-07     | Mild fever, joint pain | Monitor platelets, rest. Follow-up in 1 week | E1        |
| FU2                 | PH3                  | 2024-01-20     | Respiratory pain       | Continue meds; give HPV vaccine              | E1        |

data on database Sqlite3

| FollowUpRecordID | PatientDiagnosisID | FollowUpDate | Description            | DoctorOrders                                 | DoctorID |
|------------------|--------------------|--------------|------------------------|----------------------------------------------|----------|
| Filter           | Filter             | Filter       | Filter                 | Filter                                       | Filter   |
| FU1              | PH1                | 2024-01-07   | Mild fever, joint pain | Monitor platelets, rest. Follow-up in 1 week | E1       |
| FU2              | PH3                | 2024-01-20   | Respiratory pain       | Continue meds; give HPV vaccine              | E1       |

Invalid input

Enter the follow up record ID didn't in the list will return not found.

```

Follow Up Record ID	Patient Diagnosis ID	Follow-up Date	Description	Doctor Orders	Doctor ID
FU1 | PH1 | 2024-01-07 | Mild fever, joint pain; no rash | Monitor platelets, rest. Follow-up in 1 week | E1
Available follow-ups for your patients:
Enter Follow Up Record ID to update (Blank for exit): 2
Follow-Up Record ID 'FU2' not found. No update performed.

```

Date needs to be entered in (YYYY-MM-DD) format.

```

Enter new Last Update (YYYY-MM-DD): 2024-13-20
Invalid date format. Please enter a valid date (YYYY-MM-DD)

```

Can't enter the invalid doctor ID.

```

Enter new Doctor ID ( valid options are: E1 E2 ): 3
Invalid doctor ID. Please enter a valid patient ID from the list.
Valid options are: E1 E2

```

Put blank can exit

```

Follow Up Record ID	Patient Diagnosis ID	Follow-up Date	Description	Doctor Orders	Doctor ID
FU1 | PH1 | 2024-01-07 | Mild fever, joint pain; no rash | Monitor platelets, rest. Follow-up in 1 week | E1
Available follow-ups for your patients:
Enter Follow Up Record ID to update (Blank for exit):

```

Select 4 for delete patient follow-up.

| Patient Follow-up                                                                           |                        |                                              |    |               |           |
|---------------------------------------------------------------------------------------------|------------------------|----------------------------------------------|----|---------------|-----------|
| *****                                                                                       |                        |                                              |    |               |           |
| 1. Add Patient Follow-up                                                                    |                        |                                              |    |               |           |
| 2. View Patient Follow-up                                                                   |                        |                                              |    |               |           |
| 3. Update Patient Follow-up                                                                 |                        |                                              |    |               |           |
| 4. Delete Patient Follow-up                                                                 |                        |                                              |    |               |           |
| 0. Back to Follow-up Management                                                             |                        |                                              |    |               |           |
| Enter your choice: 4                                                                        |                        |                                              |    |               |           |
| Follow Up Record ID   Patient Diagnosis ID   Follow-up Date                                 |                        | Description                                  |    | Doctor Orders | Doctor ID |
| FU1   PH1   2024-01-07                                                                      | Mild fever, joint pain | Monitor platelets, rest. Follow-up in 1 week | E1 | a   E1        |           |
| FU2   PH3   2024-01-20                                                                      | a                      |                                              |    |               |           |
| Enter Follow-Up Record ID to delete (number only, will auto-prefix FU, or blank to exit): 2 |                        |                                              |    |               |           |
| Deleted patient follow up with ID 'FU2' successfully.                                       |                        |                                              |    |               |           |

Enter the follow up record ID didn't in the list will return not found.

| Patient Follow-up                                                                           |                        |                                              |    |               |           |
|---------------------------------------------------------------------------------------------|------------------------|----------------------------------------------|----|---------------|-----------|
| *****                                                                                       |                        |                                              |    |               |           |
| 1. Add Patient Follow-up                                                                    |                        |                                              |    |               |           |
| 2. View Patient Follow-up                                                                   |                        |                                              |    |               |           |
| 3. Update Patient Follow-up                                                                 |                        |                                              |    |               |           |
| 4. Delete Patient Follow-up                                                                 |                        |                                              |    |               |           |
| 0. Back to Follow-up Management                                                             |                        |                                              |    |               |           |
| Enter your choice: 4                                                                        |                        |                                              |    |               |           |
| Follow Up Record ID   Patient Diagnosis ID   Follow-up Date                                 |                        | Description                                  |    | Doctor Orders | Doctor ID |
| FU1   PH1   2024-01-07                                                                      | Mild fever, joint pain | Monitor platelets, rest. Follow-up in 1 week | E1 | a   E1        |           |
| FU2   PH3   2024-01-20                                                                      | a                      |                                              |    |               |           |
| Enter Follow-Up Record ID to delete (number only, will auto-prefix FU, or blank to exit): 3 |                        |                                              |    |               |           |
| Patient follow up with ID 'FU3' not found. No deletion performed.                           |                        |                                              |    |               |           |

Put blank can exit

| Patient Follow-up                                                                         |                        |                                              |    |               |           |
|-------------------------------------------------------------------------------------------|------------------------|----------------------------------------------|----|---------------|-----------|
| *****                                                                                     |                        |                                              |    |               |           |
| 1. Add Patient Follow-up                                                                  |                        |                                              |    |               |           |
| 2. View Patient Follow-up                                                                 |                        |                                              |    |               |           |
| 3. Update Patient Follow-up                                                               |                        |                                              |    |               |           |
| 4. Delete Patient Follow-up                                                               |                        |                                              |    |               |           |
| 0. Back to Follow-up Management                                                           |                        |                                              |    |               |           |
| Enter your choice:                                                                        |                        |                                              |    |               |           |
| Follow Up Record ID   Patient Diagnosis ID   Follow-up Date                               |                        | Description                                  |    | Doctor Orders | Doctor ID |
| FU1   PH1   2024-01-07                                                                    | Mild fever, joint pain | Monitor platelets, rest. Follow-up in 1 week | E1 | a   E1        |           |
| FU2   PH3   2024-01-20                                                                    | a                      |                                              |    |               |           |
| Enter Follow-Up Record ID to delete (number only, will auto-prefix FU, or blank to exit): |                        |                                              |    |               |           |
| Patient follow up with ID '' not found. No deletion performed.                            |                        |                                              |    |               |           |

Select 0 for exit

```
Patient Follow-up
*****
1. Add Patient Follow-up
2. View Patient Follow-up
3. Update Patient Follow-up
4. Delete Patient Follow-up
0. Back to Follow-up Management
Enter your choice: 0

Follow-up Management
*****
1. Patient Follow-up
2. Patient Admission Transfer
0. Back to Doctor Menu
Enter your choice: |
```

#### 6.5.4.2 Patient Admission Transfer

Select 1 for add patient admission transfer.

```
Patient Admission Transfer
*****
1. Add Patient Admission Transfer
2. View Patient Admission Transfer
3. Update Patient Admission Transfer
4. Delete Patient Admission Transfer
0. Back to Follow-up Management
Enter your choice: 1

Add Patient Admission Transfer
*****
Enter new Patient ID ( valid options are: P1 P3 ): 3
Enter new Check-in Date and Time (YYYY-MM-DD HH:MM): 2024-03-15 13:00
Enter new Check-out Date and Time (YYYY-MM-DD HH:MM): 224-03-15 15:30
Invalid date and time format. Please enter in the format YYYY-MM-DD HH:MM
Enter new Check-out Date and Time (YYYY-MM-DD HH:MM): 2024-03-15 15:30
Enter Current Status (Discharged/Transferred/Other): Discharged
Patient Status 'PS3' added successfully.
Do you want to add another admission transfer record? (Y/N): n
```

Select 2 for view patient admission transfer.

| Patient Status ID | Patient ID | Check-in Time    | Check-out Time   | Current Status |
|-------------------|------------|------------------|------------------|----------------|
| PS1               | P1         | 2024-01-06 10:30 | 2024-01-08 12:00 | Discharged     |

Select 3 for update patient admission transfer. Doctors can select the patient status ID they want to perform an update. Update check-in time from 2024-03-15 13:00 to 2024-03-15 14:00.

```
Enter your choice: 3
-----
Patient Status ID | Patient ID | Check-in Time | Check-out Time | Current Status
-----
PS1 | P1 | 2024-01-06 10:30 | 2024-01-08 12:00 | Discharged
PS3 | P3 | 2024-03-15 13:00 | 2024-03-15 15:30 | Discharged
Enter Patient Status ID to update (Blank for exit): 3
Enter new Patient ID (valid options are: P1 P2 P3 ): 3
Enter new Check-in Date and Time (YYYY-MM-DD HH:MM): 2024-03-15 14 : 00
Enter new Check-out Date and Time (YYYY-MM-DD HH:MM): 2024-03-15 15 : 30
Enter new Current Status (Discharged/Transferred/Other): Discharged
Updated patient status with ID 'PS3' successfully.
```

Invalid input

The check-in date and time must enter in (YYYY-MM-DD HH:MM) format.

```
Enter new Check-in Date and Time (YYYY-MM-DD HH:MM): 20240106 10 : 30
Invalid date and time format. Please enter in the format YYYY-MM-DD HH:MM.
```

The check-out date and time must enter in (YYYY-MM-DD HH:MM) format.

```
Enter new Check-out Date and Time (YYYY-MM-DD HH:MM): 2024/01/08 12 : 00
Invalid date and time format. Please enter in the format YYYY-MM-DD HH:MM.
```

If the doctor entered the patient status ID didn't in the list return not found.

```
Enter your choice: 3
-----
Patient Status ID | Patient ID | Check-in Time | Check-out Time | Current Status
-----
PS1      | P1      | 2024-01-06 10:30 | 2024-01-08 12:00 | Discharged
PS3      | P3      | 2024-03-15 13:00 | 2024-03-15 15:30 | Discharged
Enter Patient Status ID to update (Blank for exit): 4
Patient Status with ID 'PS4' not found. No update performed.
```

Put blank for exit

```
-----
Patient Status ID | Patient ID | Check-in Time | Check-out Time | Current Status
-----
PS1      | P1      | 2024-01-06 10:30 | 2024-01-08 12:00 | Discharged
PS3      | P3      | 2024-03-15 13:00 | 2024-03-15 15:30 | Discharged
Enter Patient Status ID to update (Blank for exit):
Patient Admission Transfer
*****
1. Add Patient Admission Transfer
2. View Patient Admission Transfer
3. Update Patient Admission Transfer
4. Delete Patient Admission Transfer
0. Back to Follow-up Management
```

Select 4 for delete patient admission transfer. Enter patient status ID to perform deletion.

```
Enter your choice: 4
-----
Patient Status ID | Patient ID | Check-in Time | Check-out Time | Current Status
-----
PS1      | P1      | 2024-01-06 10:30 | 2024-01-08 12:00 | Discharged
PS3      | P3      | 2024-03-15 14:00 | 2024-03-15 15:30 | Discharged
Enter Admission Transfer Record ID to delete (number only, will auto-prefix PH, or blank to exit): 3
Deleted patient status with ID 'PS3' successfully.
```

If a doctor entered the patient status ID didn't in the list return not found.

```
-----
Patient Status ID | Patient ID | Check-in Time | Check-out Time | Current Status
-----
PS1      | P1      | 2024-01-06 10:30 | 2024-01-08 12:00 | Discharged
PS3      | P3      | 2024-03-15 14:00 | 2024-03-15 15:30 | Discharged
Enter Admission Transfer Record ID to delete (number only, will auto-prefix PH, or blank to exit): 5
Patient status with ID 'PS5' not found. No deletion performed.
```

Put blank for exit

| Patient Status ID | Patient ID | Check-in Time    | Check-out Time   | Current Status |
|-------------------|------------|------------------|------------------|----------------|
| PS1               | P1         | 2024-01-06 10:30 | 2024-01-08 12:00 | Discharged     |
| PS3               | P3         | 2024-03-15 14:00 | 2024-03-15 15:30 | Discharged     |

Enter Admission Transfer Record ID to delete (number only, will auto-prefix PH, or blank to exit):

Patient Admission Transfer  
\*\*\*\*\*  
1. Add Patient Admission Transfer  
2. View Patient Admission Transfer  
3. Update Patient Admission Transfer  
4. Delete Patient Admission Transfer  
0. Back to Follow-up Management

Select 0 for exit

| Patient Admission Transfer |                                   |
|----------------------------|-----------------------------------|
| *****                      |                                   |
| 1.                         | Add Patient Admission Transfer    |
| 2.                         | View Patient Admission Transfer   |
| 3.                         | Update Patient Admission Transfer |
| 4.                         | Delete Patient Admission Transfer |
| 0.                         | Back to Follow-up Management      |

Enter your choice: 0

| Follow-up Management |                            |
|----------------------|----------------------------|
| *****                |                            |
| 1.                   | Patient Follow-up          |
| 2.                   | Patient Admission Transfer |
| 0.                   | Back to Doctor Menu        |

Enter your choice: |

### 6.5.5 Vaccination Records

Choose '5' to view the vaccination records.

| Vaccination ID | Patient ID | Vaccine Type | Dose Number | Administration Date |
|----------------|------------|--------------|-------------|---------------------|
| V1             | P1         | AZ           | First dose  | 2021-09-16          |
| V2             | P3         | Pfizer       | First dose  | 2021-10-01          |

## 6.5.6 Doctor Schedule

Choose '6' to view the doctor schedule.

| Doctor Schedule ID | Doctor ID | Monday  | Tuesday | Wednesday | Thursday | Friday  | Saturday | Sunday |
|--------------------|-----------|---------|---------|-----------|----------|---------|----------|--------|
| D1                 | E1        | Working | Working | Working   | Off      | Off     | Off      | Off    |
| D2                 | E2        | Off     | Off     | Off       | Working  | Working | Working  | Off    |

## 6.5.7 Patient Allergies History

Choose '7' to view the patient's allergies history.

| Patient Allergy ID | Patient ID | Allergy Name | Severity | Reaction                                         |
|--------------------|------------|--------------|----------|--------------------------------------------------|
| AII                | P2         | Peanut       | Severe   | Anaphylaxis (life-threatening allergic reaction) |

## 6.5.8 Notices

### 6.5.8.1 Emergency Tools

Choose '1' to view the emergency tools.

| Emergency Tools ID | Type                | Tool                         | Emergency Number |
|--------------------|---------------------|------------------------------|------------------|
| ET1                | Contact Information | Emergency Number             | 03-86457894      |
| ET2                | Transportation      | Ambulance Department Service | 999              |

### 6.5.8.2 Project

Choose '2' to view the project.

| Project ID | Title                      | DateOfPublication                                       | Description |
|------------|----------------------------|---------------------------------------------------------|-------------|
| PRO1       | New Vaccination Guidelines | Release of updated guidelines for vaccination protocols | 2024-01-18  |
| PRO3       | Mental Health Awareness    | Campaign to promote mental health awareness             | 2024-03-08  |

### 6.5.8.3 Back to Doctor Menu

Choose '0' to return to the doctor menu.

| Doctor Menu                     |
|---------------------------------|
| *****                           |
| 1. Profile                      |
| 2. Patient Appointment Schedule |
| 3. Patient Medical Diagnosis    |
| 4. Follow-up Management         |
| 5. Vaccination Records          |
| 6. Doctor Schedule              |
| 7. Patient Allergies History    |
| 8. Notices                      |
| 0. Exit                         |

If the ID chosen is not the relevant id, it will show 'Invalid choice. Please try again.'

```
Notices
*****
1. Emergency Tools
2. Project
0. Back to Doctor Menu
Enter your choice for more action of the notice: 3
Invalid choice. Please try again.
```

If the ID chosen is not relevant, it will show 'Invalid choice. Please try again.

```
Notices
*****
1. Emergency Tools
2. Project
0. Back to Doctor Menu
Enter your choice for more action of the notice: ,
Invalid input. Please enter a valid integer.
```

### 6.5.9 Exit

Choose '0' to exit the doctor menu.

```
Doctor Menu
*****
1. Profile
2. Patient Appointment Schedule
3. Patient Medical Diagnosis
4. Follow-up Management
5. Vaccination Records
6. Doctor Schedule
7. Patient Allergies History
8. Notices
0. Exit
Enter your choice: 0
Exiting Doctor Menu
```

## **6.6 Admin Menu**

The admin menu provides a simple interface for admin managing different functions within the system.

```
Admin Menu
1. All User
2. Medical Insurance
3. Doctor Schedule
4. Patient Medical Diagnosis
5. Type of Service
6. Patient Appointment
7. Patient Canceled Appointment
8. Inventory Management
9. View and Publish Notices
10. View and Publish Project
11. Discount
12. Patient Admission Transfer
13. View Feedback
14. Patient Vaccination Records
15. History of Patient Allergies
16. Latest Patient Status
17. Emergency Tools
18. Billing
0. Exit
Enter your choice: |
```

If an invalid input is entered, the system prompts the user with an "Invalid input" message and asks them to enter a valid integer. The menu then reappears for the user to select a valid option.

**Admin Menu**

- 1. All User
- 2. Medical Insurance
- 3. Doctor Schedule
- 4. Patient Medical Diagnosis
- 5. Type of Service
- 6. Patient Appointment
- 7. Patient Canceled Appointment
- 8. Inventory Management
- 9. View and Publish Notices
- 10. View and Publish Project
- 11. Discount
- 12. Patient Admission Transfer
- 13. View Feedback
- 14. Patient Vaccination Records
- 15. History of Patient Allergies
- 16. Latest Patient Status
- 17. Emergency Tools
- 18. Billing
- 0. Exit

Enter your choice: j

Invalid input. Please enter a valid integer.

**Admin Menu**

- 1. All User
- 2. Medical Insurance
- 3. Doctor Schedule
- 4. Patient Medical Diagnosis
- 5. Type of Service
- 6. Patient Appointment
- 7. Patient Canceled Appointment
- 8. Inventory Management
- 9. View and Publish Notices
- 10. View and Publish Project
- 11. Discount
- 12. Patient Admission Transfer
- 13. View Feedback
- 14. Patient Vaccination Records
- 15. History of Patient Allergies
- 16. Latest Patient Status
- 17. Emergency Tools
- 18. Billing
- 0. Exit

Enter your choice: |

### 6.6.1 All User

All User is a list of all users within the hospital management system displayed, including their details such as UserID, Name, ICNO, Gender, Position, Contact Number, Email, and Address.

```
All User
(null):UserID = E1
Name = Johnathan Maxwell Smith
ICNO = 750123105671
Gender = Male
Position = Doctor
ContactNumber = 60154489651
Email = john@example.com
Address = Jalan Setia Tropika U13/20, Setia Eco Park, Setia Eco Park, Shah Alam, Selangor

(null):UserID = E2
Name = Matthew Harrison Wilson
ICNO = 820305891723
Gender = Male
Position = Doctor
ContactNumber = 60166546545
Email = matthew@example.com
Address = Lebuhraya Sungai Besi BESRAYA, Sungai Besi, Kuala Lumpur

(null):UserID = E3
Name = Michael Christopher Brown
ICNO = 910405141122
Gender = Female
Position = Admin
ContactNumber = 60141486554
Email = michael@example.com
Address = 1A Tingkat 1 Jalan S 10, 1, Jalan Bukit Kajang, 43000 Kajang, Selangor

(null):UserID = P1
Name = Johnathan Alexander Doe
ICNO = 800101141239
Gender = Male
Position = Patient
ContactNumber = 60119876543
Email = johndoe@example.com
Address = Majestic Maxim Jalan 9/142, Taman Len Seng, Batu 9, Cheras, Kuala Lumpur

(null):UserID = P2
Name = Jane Elizabeth Smith
ICNO = 850202105678
Gender = Female
Position = Patient
ContactNumber = 60145678901
Email = janessmith@example.com
Address = 7 Jalan Dataran Cheras 6, Balakong, Kajang, Selangor
```

### 6.6.1.1 Filter User

Admin can retrieve specific user data based on chosen criteria. The system prompts the admin to filter data by various attributes.

```
Do you want to filter the data? (Y/N): y
Filter by (Name, ICNo, Gender, Position, ContactNumber, Email, Address, Username): Position
Enter value for Position: Admin
UserID: E3
Name: Michael Christopher Brown
ICNO: 910405141122
Gender: Female
Position: Admin
ContactNumber: 60141486554
Email: michael@example.com
Address: 1A Tingkat 1 Jalan S 10, 1, Jalan Bukit Kajang, 43000 Kajang, Selangor
Username: Michael Brown
Password: dsCg530

Do you want to filter the data? (Y/N): y
Filter by (Name, ICNo, Gender, Position, ContactNumber, Email, Address, Username): Gender
Enter value for Gender: Female
UserID: P2
Name: Jane Elizabeth Smith
ICNO: 850202105678
Gender: Female
Position: Patient
ContactNumber: 60145678901
Email: janeshsmith@example.com
Address: 7 Jalan Dataran Cheras 6, Balakong, Kajang, Selangor
Username: Jane Smith
Password: x9mn124

UserID: E3
Name: Michael Christopher Brown
ICNO: 910405141122
Gender: Female
Position: Admin
ContactNumber: 60141486554
Email: michael@example.com
Address: 1A Tingkat 1 Jalan S 10, 1, Jalan Bukit Kajang, 43000 Kajang, Selangor
Username: Michael Brown
Password: dsCg530
```

If an invalid field is entered, the system notifies the admin and requests a valid field.

```
Do you want to filter the data? (Y/N): y
Filter by (Name, ICNo, Gender, Position, ContactNumber, Email, Address, Username): k
Invalid field. Please try again.
Filter by (Name, ICNo, Gender, Position, ContactNumber, Email, Address, Username): |
```

If it finds no matching results, it will show **No relevant search results found.**

```
Do you want to filter the data? (Y/N): y
Filter by (Name, ICNo, Gender, Position, ContactNumber, Email, Address, Username): Email
Enter value for Email: Jerry
No relevant search results found.
```

### 6.6.1.2 Add User

The input is validated at each step, ensuring correct formats (e.g., IC number must be 12 digits, contact number at least 10 digits, etc.). After successful validation and data entry, the user is created with a unique UserID based on their position (e.g., 'P' for patients, 'E' for doctors/admins). The system displays all existing users after creation and asks if another user should be added.

```
Do you want to have action for User Info? (Y/N): y

Select an action:
1. Add New User
2. Edit User
3. Delete User
0. Exit
Enter your choice: 1

Add New User
*****
Enter User Name: Alice Victoria Brown
Enter IC Number (12 digits): 88030308910
Input is not valid. Please try again.
Enter IC Number (12 digits): 880303089102
Enter Gender (Female/Male): emale
Input is not valid. Please try again.
Enter Gender (Female/Male): Female
Enter Position (Patient/Doctor/Admin): tient
Input is not valid. Please try again.
Enter Position (Patient/Doctor/Admin): Patient
Enter User Contact Number (at least 10 digits): 60171234
Input is not valid. Please try again.
Enter User Contact Number (at least 10 digits): 60171234567
Enter Email: alicebrown@example.com
Input is not valid. Please try again.
Enter Email: alicebrown@example.com
Enter Address: Jalan Tasik Utama, Sungai Besi, Wilayah Persekutuan Kuala Lumpur
Enter Username: Alice Brown
Enter Password: 1uz9x
```

```

(null):UserID = P2
Name = Jane Elizabeth Smith
ICNO = 850202105678
Gender = Female
Position = Patient
ContactNumber = 60145678901
Email = janesmith@example.com
Address = 7 Jalan Dataran Cheras 6, Balakong, Kajang, Selangor

(null):UserID = P3
Name = Alice Victoria Brown
ICNO = 880303089102
Gender = Female
Position = Patient
ContactNumber = 60171234567
Email = alicebrown@example.com
Address = Jalan Tasik Utama, Sungai Besi, Wilayah Persekutuan Kuala Lumpur

Do you want to add another user? (Y/N): |

```

SQLite shows add the new user data successfully.

|   | UserID | Name                      | ICNO         | Gender | Position | ContactNumber | Email                   | Address                            | Username       | Password |
|---|--------|---------------------------|--------------|--------|----------|---------------|-------------------------|------------------------------------|----------------|----------|
|   | Filter | Filter                    | Filter       | Filter | Filter   | Filter        | Filter                  | Filter                             | Filter         | Filter   |
| 1 | P1     | Johnathan Alexander Doe   | 800101141239 | Male   | Patient  | 60119876543   | johndoe@example.com     | Majestic Maxim Jalan 9/142, ...    | John Doe       | ab7kl    |
| 2 | P2     | Jane Elizabeth Smith      | 850202105678 | Female | Patient  | 60145678901   | janesmith@example.com   | 7 Jalan Dataran Cheras 6, ...      | Jane Smith     | x9mn124  |
| 3 | E1     | Johnathan Maxwell Smith   | 750123105671 | Male   | Doctor   | 60154489651   | john@example.com        | Jalan Setia Tropika U13/20, ...    | John Smith     | 156cd    |
| 4 | E2     | Matthew Harrison Wilson   | 820305891723 | Male   | Doctor   | 60166546545   | matthew@example.com     | Lebuhraya Sungai Besi ...          | Matthew Wilson | casca62  |
| 5 | E3     | Michael Christopher Brown | 910405141122 | Female | Admin    | 60141486554   | michael@example.com     | 1A Tingkat 1 Jalan S 10, 1, Jal... | Michael Brown  | dsCg530  |
| 6 | P3     | Alice Victoria Brown      | 880303089102 | Female | Patient  | 60171234567   | alicebrown@example.c... | Jalan Tasik Utama, Sungai Bes...   | Alice Brown    | 1uz9x    |

### 6.6.1.3 Update User

The system prompts for the **User ID**. If blank, the update process exits. If it doesn't match any records, a **no relevant search results found** message is shown, and the update process exits.

```
Do you want to have action for User Info? (Y/N): y

Select an action:
1. Add New User
2. Edit User
3. Delete User
0. Exit
Enter your choice: 2
Enter Entire User ID to update (blank to exit update process):
Deletion process exited.
```

```
Do you want to have action for User Info? (Y/N): y

Select an action:
1. Add New User
2. Edit User
3. Delete User
0. Exit
Enter your choice: 2
Enter Entire User ID to update (blank to exit update process): j
No relevant search results found.
```

After entering a valid **User ID**, the system lets you update fields, leaving blanks unchanged. It confirms successful updates and shows the updated information.

```
Do you want to have action for User Info? (Y/N): y

Select an action:
1. Add New User
2. Edit User
3. Delete User
0. Exit
Enter your choice: 2
Enter Entire User ID to update (blank to exit update process): P3
Enter new values for the following fields. Leave empty to skip updating the field.
Name: Lilith Victoria Brown
ICNo:
Gender:
Position:
ContactNumber:
Email: lilithbrown@patient.com
Address:
Username:
Password:
Data updated successfully!
```

SQLite is showing the updated user information.

| UserID | Name                      | ICNO         | Gender | Position | ContactNumber | Email                   | Address                            | Username       | Password |
|--------|---------------------------|--------------|--------|----------|---------------|-------------------------|------------------------------------|----------------|----------|
| Filter | Filter                    | Filter       | Filter | Filter   | Filter        | Filter                  | Filter                             | Filter         | Filter   |
| P1     | Johnathan Alexander Doe   | 800101141239 | Male   | Patient  | 60119876543   | johndoe@example.com     | Majestic Maxim Jalan 9/142, ...    | John Doe       | ab7kl    |
| P2     | Jane Elizabeth Smith      | 850202105678 | Female | Patient  | 60145678901   | janesmith@example.com   | 7 Jalan Dataran Cheras 6, ...      | Jane Smith     | x9mn124  |
| E1     | Johnathan Maxwell Smith   | 750123105671 | Male   | Doctor   | 60154489651   | john@example.com        | Jalan Setia Tropika U13/20, ...    | John Smith     | 156cd    |
| E2     | Matthew Harrison Wilson   | 820305891723 | Male   | Doctor   | 60166546545   | matthew@example.com     | Lebuhraya Sungai Besi ...          | Matthew Wilson | casca62  |
| E3     | Michael Christopher Brown | 910405141122 | Female | Admin    | 60141486554   | michael@example.com     | 1A Tingkat 1 Jalan S 10, 1, Jal... | Michael Brown  | dsCg530  |
| P3     | Lilith Victoria Brown     | 880303089102 | Female | Patient  | 60171234567   | lilithbrown@patient.com | Jalan Tasik Utama, Sungai Bes...   | Alice Brown    | 1uz9x    |

#### 6.6.1.4 Delete User

The system prompts for the **User ID**. If blank, the deletion process exits. If it doesn't match any records, a **no relevant search results found** message is shown, and the deletion process exits.

```
Do you want to have action for User Info? (Y/N): y

Select an action:
1. Add New User
2. Edit User
3. Delete User
0. Exit
Enter your choice: 3
Enter Entire User ID to delete (blank to exit delete process):
Deletion process exited.
```

```
Do you want to have action for User Info? (Y/N): y

Select an action:
1. Add New User
2. Edit User
3. Delete User
0. Exit
Enter your choice: 3
Enter Entire User ID to delete (blank to exit delete process): l
No relevant search results found.
```

After entering a valid **User ID**, the system successfully deletes the user and confirms the deletion. Then it displays the remaining users and asks if you want to perform another action.

```
Do you want to have action for User Info? (Y/N): y

Select an action:
1. Add New User
2. Edit User
3. Delete User
0. Exit
Enter your choice: 3
Enter Entire User ID to delete (blank to exit delete process): P3
Data delete successfully!
```

```
(null):UserID = E1
Name = Johnathan Maxwell Smith
ICNO = 750123105671
Gender = Male
Position = Doctor
ContactNumber = 60154489651
Email = john@example.com
Address = Jalan Setia Tropika U13/20, Setia Eco Park, Setia Eco Park, Shah Alam, Selangor

(null):UserID = E2
Name = Matthew Harrison Wilson
ICNO = 820305891723
Gender = Male
Position = Doctor
ContactNumber = 60166546545
Email = matthew@example.com
Address = Lebuhraya Sungai Besi BESRAYA, Sungai Besi, Kuala Lumpur

(null):UserID = E3
Name = Michael Christopher Brown
ICNO = 910405141122
Gender = Female
Position = Admin
ContactNumber = 60141486554
Email = michael@example.com
Address = 1A Tingkat 1 Jalan S 10, 1, Jalan Bukit Kajang, 43000 Kajang, Selangor

(null):UserID = P1
Name = Johnathan Alexander Doe
ICNO = 800101141239
Gender = Male
Position = Patient
ContactNumber = 60119876543
Email = johndoe@example.com
Address = Majestic Maxim Jalan 9/142, Taman Len Seng, Batu 9, Cheras, Kuala Lumpur

(null):UserID = P2
Name = Jane Elizabeth Smith
ICNO = 850202105678
Gender = Female
Position = Patient
ContactNumber = 60145678901
Email = janeshsmith@example.com
Address = 7 Jalan Dataran Cheras 6, Balakong, Kajang, Selangor
```

```
Do you want to have action for User Info? (Y/N): |
```

SQLite shows the user data is deleted.

| User ID | Name                      | IC NO        | Gender | Position | Contact Number | Email                 | Address                            | Username       | Password |
|---------|---------------------------|--------------|--------|----------|----------------|-----------------------|------------------------------------|----------------|----------|
| 1 P1    | Johnathan Alexander Doe   | 800101141239 | Male   | Patient  | 60119876543    | johndoe@example.com   | Majestic Maxim Jalan 9/142, ...    | John Doe       | ab7kl    |
| 2 P2    | Jane Elizabeth Smith      | 850202105678 | Female | Patient  | 60145678901    | janesmith@example.com | 7 Jalan Dataran Cheras 6, ...      | Jane Smith     | x9mn124  |
| 3 E1    | Johnathan Maxwell Smith   | 750123105671 | Male   | Doctor   | 60154489651    | john@example.com      | Jalan Setia Tropika U13/20, ...    | John Smith     | 156cd    |
| 4 E2    | Matthew Harrison Wilson   | 820305891723 | Male   | Doctor   | 60166546545    | matthew@example.com   | Lebuhraya Sungai Besi ...          | Matthew Wilson | casca62  |
| 5 E3    | Michael Christopher Brown | 910405141122 | Female | Admin    | 60141486554    | michael@example.com   | 1A Tingkat 1 Jalan S 10, 1, Jal... | Michael Brown  | dsCg530  |

### 6.6.2 Medical Insurance

The system displays details of the medical insurance record. After showing this information, it prompts whether you want to perform another action related to medical insurance.

```
Medical Insurance
(null):MedicalInsuranceID = INS1
PatientID = P1
InsuranceProvider = Allianz
PolicyNumber = A12392
Deductible = 400000
```

Do you want to have action for Medical Insurance? (Y/N):

#### 6.6.2.1 Add Medical Insurance

If an invalid **Patient ID** or non-integer value for **Deductible** is entered, the system requests correct input. Upon successful addition, the system displays the updated list of medical insurance records and asks if you want to add another.

```
Do you want to have action for Medical Insurance? (Y/N): y

Select an action:
1. Add New Medical Insurance
2. Edit Medical Insurance
3. Delete Medical Insurance
0. Exit
Enter your choice: 1

Add New Medical Insurance
*****
Enter Patient ID (valid options are: P1 P2 ): P3
Invalid Patient ID. Please enter a valid Patient ID from the list.
Valid options are: P1 P2
Enter Patient ID (valid options are: P1 P2 ): P2
Enter Insurance Provider: AIA
Enter Policy Number: AIA193842
Enter Deductible: L
Invalid input. Please enter a valid integer.
600000
Medical Insurance added successfully!

(null):MedicalInsuranceID = INS1
PatientID = P1
InsuranceProvider = Allianz
PolicyNumber = A12392
Deductible = 400000

(null):MedicalInsuranceID = INS2
PatientID = P2
InsuranceProvider = AIA
PolicyNumber = AIA193842
Deductible = 600000

Do you want to add another Medical Insurance? (Y/N): |
```

SQLite shows the new medical insurance data successfully.

|   | MedicalInsuranceID | PatientID | InsuranceProvider | PolicyNumber | Deductible |
|---|--------------------|-----------|-------------------|--------------|------------|
|   | Filter             | Filter    | Filter            | Filter       | Filter     |
| 1 | INS1               | P1        | Allianz           | A12392       | 400000     |
| 2 | INS2               | P2        | AIA               | AIA193842    | 600000     |

### 6.6.2.2 Update Medical Insurance

The system prompts for the **Medical Insurance ID**. If blank, the update process exits. If it doesn't match any records, a **no relevant search results found** message is shown, and the update process exits.

```
Do you want to have action for Medical Insurance? (Y/N): y
Select an action:
1. Add New Medical Insurance
2. Edit Medical Insurance
3. Delete Medical Insurance
0. Exit
Enter your choice: 2
Enter Medical Insurance ID to update (blank to exit update process):
Update process exited.

Do you want to have action for Medical Insurance? (Y/N): y
Select an action:
1. Add New Medical Insurance
2. Edit Medical Insurance
3. Delete Medical Insurance
0. Exit
Enter your choice: 2
Enter Medical Insurance ID to update (blank to exit update process): INS3
No relevant search results found.
```

After entering a valid **Medical Insurance ID**, the system lets you update fields, leaving blanks unchanged. It confirms successful updates and shows the updated information.

```
Do you want to have action for Medical Insurance? (Y/N): y
Select an action:
1. Add New Medical Insurance
2. Edit Medical Insurance
3. Delete Medical Insurance
0. Exit
Enter your choice: 2
Enter Medical Insurance ID to update (blank to exit update process): INS2
Enter new values for the following fields. Leave empty to skip updating the field.
PatientID:
InsuranceProvider: Great Eastern
PolicyNumber:
Deductible:
Data updated successfully!
(null):MedicalInsuranceID = INS1
PatientID = P1
InsuranceProvider = Allianz
PolicyNumber = A12392
Deductible = 400000

(null):MedicalInsuranceID = INS2
PatientID = P2
InsuranceProvider = Great Eastern
PolicyNumber = AIA193842
Deductible = 600000
```

SQLite is showing the updated medical insurance information.

|   | MedicalInsuranceID | PatientID | InsuranceProvider | PolicyNumber | Deductible |
|---|--------------------|-----------|-------------------|--------------|------------|
|   | Filter             | Filter    | Filter            | Filter       | Filter     |
| 1 | INS1               | P1        | Allianz           | A12392       | 400000     |
| 2 | INS2               | P2        | Great Eastern     | AIA193842    | 600000     |

### 6.6.2.3 Delete Medical Insurance

The system prompts for the **Medical Insurance ID**. If blank, the deletion process exits. If it doesn't match any records, a **no relevant search results found** message is shown, and the deletion process exits.

```
Do you want to have action for Medical Insurance? (Y/N): y

Select an action:
1. Add New Medical Insurance
2. Edit Medical Insurance
3. Delete Medical Insurance
0. Exit
Enter your choice: 3
Enter MedicalInsurance ID to delete (blank to exit delete process):
Deletion process exited.

Do you want to have action for Medical Insurance? (Y/N): y

Select an action:
1. Add New Medical Insurance
2. Edit Medical Insurance
3. Delete Medical Insurance
0. Exit
Enter your choice: 3
Enter MedicalInsurance ID to delete (blank to exit delete process): m
No relevant search results found.
```

After entering a valid **Medical Insurance ID**, the system successfully deletes the user and confirms the deletion. Then it displays the remaining users and asks if you want to perform another action.

```
Do you want to have action for Medical Insurance? (Y/N): y

Select an action:
1. Add New Medical Insurance
2. Edit Medical Insurance
3. Delete Medical Insurance
0. Exit
Enter your choice: 3
Enter MedicalInsurance ID to delete (blank to exit delete process): INS2
Data deleted successfully!

(null):MedicalInsuranceID = INS1
PatientID = P1
InsuranceProvider = Allianz
PolicyNumber = A12392
Deductible = 400000

Do you want to have action for Medical Insurance? (Y/N): |
```

SQLite shows the medical insurance data is deleted.

|   | MedicalInsuranceID | PatientID | InsuranceProvider | PolicyNumber | Deductible |
|---|--------------------|-----------|-------------------|--------------|------------|
|   | Filter             | Filter    | Filter            | Filter       | Filter     |
| 1 | INS1               | P1        | Allianz           | A12392       | 400000     |

### **6.6.3 Doctor Schedule**

The system displays details of the doctor's schedule. After showing this information, it prompts whether you want to perform another action related to the doctor schedule.

```
Doctor Schedule
(null):DoctorScheduleID = D1
DoctorID = E1
Monday = Working
Tuesday = Working
Wednesday = Working
Thursday = Off
Friday = Off
Saturday = Off
Sunday = Off
```

```
Do you want to have action for Doctor Schedule? (Y/N): |
```

### 6.6.3.1 Add Doctor Schedule

The system prompts for a valid **Doctor ID**. If the **Doctor ID** is invalid, the system asks for a valid input. After entering valid details, the system confirms successful addition and displays the updated schedule. Then it asks if you want to add another schedule.

```
Add New Doctor Schedule
*****
Enter Doctor ID ( valid options are: E1 E2 ): E3
Invalid Doctor ID. Please enter a valid Doctor ID from the list.
Valid options are: E1 E2 : Enter Doctor ID ( valid options are: E1 E2 ): E2
Enter Monday Schedule: Off
Enter Tuesday Schedule: Off
Enter Wednesday Schedule: Off
Enter Thursday Schedule: Working
Enter Friday Schedule: Working
Enter Saturday Schedule: Working
Enter Sunday Schedule: Off
DoctorSchedule added successfully!
(null):DoctorScheduleID = D1
DoctorID = E1
Monday = Working
Tuesday = Working
Wednesday = Working
Thursday = Off
Friday = Off
Saturday = Off
Sunday = Off

(null):DoctorScheduleID = D2
DoctorID = E2
Monday = Off
Tuesday = Off
Wednesday = Off
Thursday = Working
Friday = Working
Saturday = Working
Sunday = Off

Do you want to add another DoctorSchedule? (Y/N): |
```

SQLite shows add the new doctor schedule data successfully.

|   | DoctorScheduleID | DoctorID | Monday  | Tuesday | Wednesday | Thursday | Friday   | Saturday | Sunday |
|---|------------------|----------|---------|---------|-----------|----------|----------|----------|--------|
|   | Filter           | Filter   | Filter  | Filter  | Filter    | Filter   | Filter   | Filter   | Filter |
| 1 | D1               | E1       | Working | Working | Working   | Off      | Off      | Off      | Off    |
| 2 | D2               | E2       | Off     | Off     | Off       | Working  | Worki... | Working  | Off    |

### 6.6.3.2 Update Doctor Schedule

The system prompts for the **Doctor Schedule ID**. If blank, the update process exits. If it doesn't match any records, a **no relevant search results found** message is shown, and the update process exits.

```
Do you want to have action for Doctor Schedule? (Y/N): y
Select an action:
1. Add New Doctor Scheduler
2. Edit Doctor Scheduler
3. Delete Doctor Scheduler
0. Exit
Enter your choice: 2
Enter Doctor Schedule ID to update (blank to exit update process):
Update process exited.

Do you want to have action for Doctor Schedule? (Y/N): y
Select an action:
1. Add New Doctor Scheduler
2. Edit Doctor Scheduler
3. Delete Doctor Scheduler
0. Exit
Enter your choice: 2
Enter Doctor Schedule ID to update (blank to exit update process): 3
No relevant search results found.
```

After entering a valid **Doctor Schedule ID**, the system lets you update fields, leaving blanks unchanged. It confirms successful updates and shows the updated information.

```
Do you want to have action for Doctor Schedule? (Y/N): y

Select an action:
1. Add New Doctor Scheduler
2. Edit Doctor Scheduler
3. Delete Doctor Scheduler
0. Exit
Enter your choice: 2
Enter Doctor Schedule ID to update (blank to exit update process): D2
Enter new values for the following fields. Leave empty to skip updating the field.
DoctorID:
Monday:
Tuesday:
Wednesday:
Thursday:
Friday:
Saturday:
Sunday: Working
Data updated successfully!

(null):DoctorScheduleID = D1
DoctorID = E1
Monday = Working
Tuesday = Working
Wednesday = Working
Thursday = Off
Friday = Off
Saturday = Off
Sunday = Off

(null):DoctorScheduleID = D2
DoctorID = E2
Monday = Off
Tuesday = Off
Wednesday = Off
Thursday = Working
Friday = Working
Saturday = Working
```

SQLite is showing the updated doctor schedule information.

|   | DoctorScheduleID | DoctorID | Monday  | Tuesday | Wednesday | Thursday | Friday   | Saturday | Sunday  |
|---|------------------|----------|---------|---------|-----------|----------|----------|----------|---------|
|   | Filter           | Filter   | Filter  | Filter  | Filter    | Filter   | Filter   | Filter   | Filter  |
| 1 | D1               | E1       | Working | Working | Working   | Off      | Off      | Off      | Off     |
| 2 | D2               | E2       | Off     | Off     | Off       | Working  | Worki... | Working  | Working |

### 6.6.3.3 Delete Doctor Schedule

The system prompts for the **Doctor Schedule ID**. If blank, the deletion process exits. If it doesn't match any records, a **no relevant search results found** message is shown, and the deletion process exits.

```
Do you want to have action for Doctor Schedule? (Y/N): y
Select an action:
1. Add New Doctor Scheduler
2. Edit Doctor Scheduler
3. Delete Doctor Scheduler
0. Exit
Enter your choice: 3
Enter Doctor Schedule ID to delete (blank to exit delete process):
Deletion process exited.

Do you want to have action for Doctor Schedule? (Y/N): y
Select an action:
1. Add New Doctor Scheduler
2. Edit Doctor Scheduler
3. Delete Doctor Scheduler
0. Exit
Enter your choice: 3
Enter Doctor Schedule ID to delete (blank to exit delete process): D3
No relevant search results found.
```

After entering a valid **Doctor Schedule ID**, the system successfully deletes the user and confirms the deletion. Then it displays the remaining users and asks if you want to perform another action.

```
Do you want to have action for Doctor Schedule? (Y/N): y

Select an action:
1. Add New Doctor Scheduler
2. Edit Doctor Scheduler
3. Delete Doctor Scheduler
0. Exit
Enter your choice: 3
Enter Doctor Schedule ID to delete (blank to exit delete process): D2
Data deleted successfully!

(null):DoctorScheduleID = D1
DoctorID = E1
Monday = Working
Tuesday = Working
Wednesday = Working
Thursday = Off
Friday = Off
Saturday = Off
Sunday = Off

Do you want to have action for Doctor Schedule? (Y/N): |
```

SQLite shows the user doctor schedule data is deleted.

|   | DoctorScheduleID | DoctorID | Monday  | Tuesday | Wednesday | Thursday | Friday | Saturday | Sunday |
|---|------------------|----------|---------|---------|-----------|----------|--------|----------|--------|
|   | Filter           | Filter   | Filter  | Filter  | Filter    | Filter   | Filter | Filter   | Filter |
| 1 | D1               | E1       | Working | Working | Working   | Off      | Off    | Off      | Off    |

#### 6.6.4 Patient Medical Diagnosis

The system displays details of the patient's medical diagnosis. Admin only has read rights.

```
Patient Medical Diagnosis Info

(null):PatientDiagnosisID = PH1
PatientID = P1
DoctorID = E1
MedicalHistory = Hypertension
MedicationsTaken = Lisinopril, Amlodipine, Losartan, Metoprolol
Diagnosis = Dengue fever
LastUpdate = 08/01/2024

(null):PatientDiagnosisID = PH2
PatientID = P2
DoctorID = E2
MedicalHistory = Diabetes
MedicationsTaken = Metformin, Insulin, DPP-4 inhibitors
Diagnosis = Iron deficiency anemia
LastUpdate = 17/02/2024

(null):PatientDiagnosisID = PH3
PatientID = P3
DoctorID = E1
MedicalHistory = Asthma
MedicationsTaken = Albuterol (short-acting), Fluticasone
Diagnosis = Get HPV vaccine
LastUpdate = 15/03/2024
```

#### 6.6.5 Type of Service

The system displays details of the type of service for the clinic. After showing this information, it prompts whether you want to perform another action related to the type of service.

```
Type Of Service
(null):ServiceID = S1
TypeOfService = Laboratory Services
Description = Diagnostic Services include blood and urine tests

(null):ServiceID = S2
TypeOfService = Vaccination
Description = Immunizations cover all ages
```

Do you want to have action for Type of Service? (Y/N): |

### 6.6.5.1 Add Type of Service

The system prompts for the service type and description. Once entered, it confirms the addition and displays the updated list of services. Then it asks if you want to add another service.

```
Add New Service
*****
Enter Type of Service: Family Medicine
Enter Description: Primary Care focuses on health, prevention, and chronic conditions
Service added successfully!

(null):ServiceID = S1
TypeOfService = Laboratory Services
Description = Diagnostic Services include blood and urine tests

(null):ServiceID = S2
TypeOfService = Vaccination
Description = Immunizations cover all ages

(null):ServiceID = S3
TypeOfService = Family Medicine
Description = Primary Care focuses on health, prevention, and chronic conditions

Do you want to add another Service? (Y/N): |
```

SQLite shows the new type of service data successfully.

|   | ServiceID | TypeOfService       | Description                      |
|---|-----------|---------------------|----------------------------------|
|   | Filter    | Filter              | Filter                           |
| 1 | S1        | Laboratory Services | Diagnostic Services include ...  |
| 2 | S2        | Vaccination         | Immunizations cover all ages     |
| 3 | S3        | Family Medicine     | Primary Care focuses on healt... |

### 6.6.5.2 Update Type of Service

The system prompts for the **Service ID**. If blank, the update process exits. If it doesn't match any records, a **no relevant search results found** message is shown, and the update process exits.

```
Do you want to have action for Type of Service? (Y/N): y

Select an action:
1. Add New Type of Service
2. Edit Type of Service
3. Delete Type of Service
0. Exit
Enter your choice: 2
Enter Service ID to update (blank to exit update process):
Update process exited.

Do you want to have action for Type of Service? (Y/N): y

Select an action:
1. Add New Type of Service
2. Edit Type of Service
3. Delete Type of Service
0. Exit
Enter your choice: 2
Enter Service ID to update (blank to exit update process): 5
No relevant search results found.
```

After entering a valid **Service ID**, the system lets you update fields, leaving blanks unchanged. It confirms successful updates and shows the updated information.

```
Do you want to have action for Type of Service? (Y/N): y

Select an action:
1. Add New Type of Service
2. Edit Type of Service
3. Delete Type of Service
0. Exit
Enter your choice: 2
Enter Service ID to update (blank to exit update process): S2
Enter new values for the following fields. Leave empty to skip updating the field.
TypeofService: Blood Test
Description: A blood test is a diagnostic procedure in which a sample of blood is analyzed to assess health, detect diseases, or monitor conditions.
Data updated successfully!

(null):ServiceID = S1
TypeofService = Laboratory Services
Description = Diagnostic Services include blood and urine tests

(null):ServiceID = S2
TypeofService = Blood Test
Description = A blood test is a diagnostic procedure in which a sample of blood is analyzed to assess health, detect diseases, or monitor conditions.

(null):ServiceID = S3
TypeofService = Family Medicine
Description = Primary Care focuses on health, prevention, and chronic conditions
```

SQLite is showing the updated service information.

| ServiceID | TypeOfService       | Description                                                        |
|-----------|---------------------|--------------------------------------------------------------------|
| Filter    | Filter              | Filter                                                             |
| 1 S1      | Laboratory Services | Diagnostic Services include blood and urine tests                  |
| 2 S2      | Blood Test          | A blood test is a diagnostic procedure in which a sample of blo... |
| 3 S3      | Family Medicine     | Primary Care focuses on health, prevention, and chronic ...        |

#### 6.6.5.3 Delete Type of Service

The system prompts for the **Service ID**. If blank, the deletion process exits. If it doesn't match any records, a **no relevant search results found** message is shown, and the deletion process exits.

```
Do you want to have action for Type of Service? (Y/N): y
Select an action:
1. Add New Type of Service
2. Edit Type of Service
3. Delete Type of Service
0. Exit
Enter your choice: 3
Enter Service ID to delete (blank to exit delete process):
Deletion process exited.
```

```
Do you want to have action for Type of Service? (Y/N): y
Select an action:
1. Add New Type of Service
2. Edit Type of Service
3. Delete Type of Service
0. Exit
Enter your choice: 3
Enter Service ID to delete (blank to exit delete process): 5
No relevant search results found.
```

After entering a valid **Service ID**, the system successfully deletes the user and confirms the deletion. Then it displays the remaining users and asks if you want to perform another action.

```
Do you want to have action for Type of Service? (Y/N): y
Select an action:
1. Add New Type of Service
2. Edit Type of Service
3. Delete Type of Service
0. Exit
Enter your choice: 3
Enter Service ID to delete (blank to exit delete process): S3
Data deleted successfully!

(null):ServiceID = S1
TypeOfService = Laboratory Services
Description = Diagnostic Services include blood and urine tests

(null):ServiceID = S2
TypeOfService = Blood Test
Description = A blood test is a diagnostic procedure in which a sample of blood is analyzed to assess health, detect diseases, or monitor conditions.

Do you want to have action for Type of Service? (Y/N): |
```

SQLite shows the service data is deleted.

| ServiceID | TypeOfService       | Description                                                         |
|-----------|---------------------|---------------------------------------------------------------------|
| Filter    | Filter              | Filter                                                              |
| 1 S1      | Laboratory Services | Diagnostic Services include blood and urine tests                   |
| 2 S2      | Blood Test          | A blood test is a diagnostic procedure in which a sample of bloo... |

### **6.6.6 Patient Appointment**

The system displays details of the patient's appointment, also history. After showing this information, it prompts whether you want to perform another action related to patient appointment.

```
Patient Appointment
(null):AppointmentID = A1
PatientID = P1
ServiceID = S3
Date = 05/01/2024
Time = 09:00
Status = Accessed
DoctorID = E1

(null):AppointmentID = A2
PatientID = P2
ServiceID = S1
Date = 08/02/2024
Time = 10:00
Status = Accessed
DoctorID = E2
```

```
Do you want to have action for Patient Appointment? (Y/N): |
```

### 6.6.6.1 Add Patient Appointment

The system allows the admin to help patients to make appointments for them. When adding a new patient appointment, the system prompts for the Patient ID, Service ID, date, time, and Doctor ID. If invalid inputs are provided (e.g., non-existent ID or incorrect date formats), the system requests correct values. Once valid inputs are given, the system confirms the appointment and shows the updated list. It then asks if you want to add more appointments.

```
Do you want to have action for Patient Appointment? (Y/N): y
Select an action:
1. Add New Patients Appointment
2. Edit Patients Appointment
0. Exit
Enter your choice: 1

Add New PatientsAppointment
*****
Enter Patient ID (valid options are: P1 P2 ): p3
Invalid patient ID. Please enter a valid patient ID from the list.
Valid options are: P1 P2
Enter Patient ID (valid options are: P1 P2 ): P1

(null):ServiceID = S1
TypeOfService = Laboratory Services
Description = Diagnostic Services include blood and urine tests

(null):ServiceID = S2
TypeOfService = Blood Test
Description = A blood test is a diagnostic procedure in which a sample of blood is analyzed to assess health, detect diseases, or monitor conditions.

Enter Service ID ( valid options are: S1 S2 ): S3
Invalid Service ID. Please enter a valid Service ID from the list.
Valid options are: S1 S2 :
Enter Service ID ( valid options are: S1 S2 ): S2
Enter Last Update(YYYY-MM-DD): 026
Invalid date format. Please enter a valid date (YYYY-MM-DD).
Enter Last Update(YYYY-MM-DD): 2024-08-19
Enter Last Update(HH:MM): 30
Invalid date format. Please enter a valid date (HH:MM).
Enter Last Update(HH:MM): 10:30

(null):DoctorScheduleID = D1
DoctorID = E1
Monday = Working
Tuesday = Working
Wednesday = Working
Thursday = Off
Friday = Off
Saturday = Off
Sunday = Off

Enter Doctor ID ( valid options are: E1 E2 ): E3
Invalid Doctor ID. Please enter a valid Doctor ID from the list.
Valid options are: E1 E2 : Enter Doctor ID ( valid options are: E1 E2 ): E1
```

```
(null):AppointmentID = A1
PatientID = P1
ServiceID = S3
Date = 05/01/2024
Time = 09:00
Status = Accessed
DoctorID = E1
```

```
(null):AppointmentID = A2
PatientID = P2
ServiceID = S1
Date = 08/02/2024
Time = 10:00
Status = Accessed
DoctorID = E2
```

```
(null):AppointmentID = A3
PatientID = P1
ServiceID = S2
Date = 2024-08-19
Time = 10:30
Status = Accessed
DoctorID = E1
```

```
Do you want to add more appointments? (Y/N): |
```

SQLite shows the new patient's appointment data successfully.

|   | AppointmentID | PatientID | ServiceID | Date       | Time   | Status   | DoctorID |  |
|---|---------------|-----------|-----------|------------|--------|----------|----------|--|
|   | Filter        | Filter    | Filter    | Filter     | Filter | Filter   | Filter   |  |
| 1 | A1            | P1        | S3        | 05/01/2024 | 09:00  | Accessed | E1       |  |
| 2 | A2            | P2        | S1        | 08/02/2024 | 10:00  | Accessed | E2       |  |
| 3 | A3            | P1        | S2        | 2024-08-19 | 10:30  | Accessed | E1       |  |

### 6.6.6.2 Update Patient Appointment

The system prompts for the **Patient's Appointment ID**. If blank, the update process exits. If it doesn't match any records, a **no relevant search results found** message is shown, and the update process exits.

```
Do you want to have action for Patient Appointment? (Y/N): y

Select an action:
1. Add New Patients Appointment
2. Edit Patients Appointment
0. Exit
Enter your choice: 2
Enter PatientsAppointment ID to update (blank to exit update process):
Update process exited.

Do you want to have action for Patient Appointment? (Y/N): y

Select an action:
1. Add New Patients Appointment
2. Edit Patients Appointment
0. Exit
Enter your choice: 2
Enter PatientsAppointment ID to update (blank to exit update process): A6
No relevant search results found.
```

After entering a valid **Patient's Appointment ID**, the system lets you update fields, leaving blanks unchanged. It confirms successful updates and shows the updated information.

```
Do you want to have action for Patient Appointment? (Y/N): y

Select an action:
1. Add New Patients Appointment
2. Edit Patients Appointment
0. Exit
Enter your choice: 2
Enter PatientsAppointment ID to update (blank to exit update process): A3
Enter new values for the following fields. Leave empty to skip updating the field.
PatientID:
ServiceID: S1
Date: 2024-08-20
Time:
DoctorID:
Data updated successfully!

(null):AppointmentID = A1
PatientID = P1
ServiceID = S3
Date = 05/01/2024
Time = 09:00
Status = Accessed
DoctorID = E1

(null):AppointmentID = A2
PatientID = P2
ServiceID = S1
Date = 08/02/2024
Time = 10:00
Status = Accessed
DoctorID = E2

(null):AppointmentID = A3
PatientID = P1
ServiceID = S1
Date = 2024-08-20
Time = 10:30
Status = Accessed
```

SQLite is showing the updated patient's appointment information.

|   | AppointmentID | PatientID | ServiceID | Date       | Time   | Status   | DoctorID |
|---|---------------|-----------|-----------|------------|--------|----------|----------|
|   | Filter        | Filter    | Filter    | Filter     | Filter | Filter   | Filter   |
| 1 | A1            | P1        | S3        | 05/01/2024 | 09:00  | Accessed | E1       |
| 2 | A2            | P2        | S1        | 08/02/2024 | 10:00  | Accessed | E2       |
| 3 | A3            | P1        | S1        | 2024-08-20 | 10:30  | Accessed | E1       |

### **6.6.7 Patient Cancelled Appointment**

The system will display details of the patient's cancelled appointment. Now it shows nothing because the record is empty. After showing this information, it prompts whether you want to perform another action related to patient's cancelled appointment.

```
Admin Menu
1. All User
2. Medical Insurance
3. Doctor Schedule
4. Patient Medical Diagnosis
5. Type of Service
6. Patient Appointment
7. Patient Canceled Appointment
8. Inventory Management
9. View and Publish Notices
10. View and Publish Project
11. Discount
12. Patient Admission Transfer
13. View Feedback
14. Patient Vaccination Records
15. History of Patient Allergies
16. Latest Patient Status
17. Emergency Tools
18. Billing
0. Exit
Enter your choice: 7
```

**Patient Canceled Appoinment**

Do you want to have action for Canceled Appoinment? (Y/N): y|

### 6.6.7.1 Add Patient Cancelled Appointment

The system allows the admin to help patients to cancel appointments. It first verifies the validity of the Appointment ID and Doctor ID. If an invalid IDs is entered, it asks again. After providing valid inputs, the system updates the appointment status to 'Cancelled' and confirms that the cancellation was successful.

```
Do you want to have action for Canceled Appoinment? (Y/N): y

Select an action:
1. Add New Canceled Appoinment
2. Edit Canceled Appoinment
0. Exit
Enter your choice: 1

Add New Cancel Appointment
*****
Enter Appointment ID (valid options are: A1 A2 A3 ): A5
Invalid Appointment ID. Please enter a valid Appointment ID from the list.
Valid options are: A1 A2 A3 : Enter Appointment ID (valid options are: A1 A2 A3 ): A3
Enter Doctor ID ( valid options are: E1 E2 ): E3
Invalid Doctor ID. Please enter a valid Doctor ID from the list.
Valid options are: E1 E2 : Enter Doctor ID ( valid options are: E1 E2 ): E1
Enter Cancel Reason: Personal Reason
CancelAppointment added successfully!

PatientsAppointment status updated to 'Cancelled'!

(null):CancelAppointmentID = CA1
AppointmentID = A3
DoctorID = E1
Reason = Personal Reason

Do you want to add another Cancel Appointment? (Y/N): |
```

SQLite shows add the new cancel appointment data successfully.

|   | CancelAppointmentID | AppointmentID | DoctorID | Reason          |
|---|---------------------|---------------|----------|-----------------|
|   | Filter              | Filter        | Filter   | Filter          |
| 1 | CA1                 | A3            | E1       | Personal Reason |

Patient's Appointment shows the A3's status is Cancelled.

```
Patient Appointment
(null):AppointmentID = A1
PatientID = P1
ServiceID = S3
Date = 05/01/2024
Time = 09:00
Status = Accessed
DoctorID = E1

(null):AppointmentID = A2
PatientID = P2
ServiceID = S1
Date = 08/02/2024
Time = 10:00
Status = Accessed
DoctorID = E2

(null):AppointmentID = A3
PatientID = P1
ServiceID = S1
Date = 2024-08-20
Time = 10:30
Status = Cancelled
DoctorID = E1
```

SQLite Patient's Appointment also shows the A3's status is Cancelled.

|   | AppointmentID | PatientID | ServiceID | Date       | Time   | Status    | DoctorID |
|---|---------------|-----------|-----------|------------|--------|-----------|----------|
|   | Filter        | Filter    | Filter    | Filter     | Filter | Filter    | Filter   |
| 1 | A1            | P1        | S3        | 05/01/2024 | 09:00  | Accessed  | E1       |
| 2 | A2            | P2        | S1        | 08/02/2024 | 10:00  | Accessed  | E2       |
| 3 | A3            | P1        | S1        | 2024-08-20 | 10:30  | Cancelled | E1       |

### 6.6.7.2 Update Patient Cancelled Appointment

The system prompts for the **Cancel Appointment ID**. If blank, the update process exits. If it doesn't match any records, a **no relevant search results found** message is shown, and the update process exits.

```
Do you want to have action for Canceled Appoinment? (Y/N): y

Select an action:
1. Add New Canceled Appoinment
2. Edit Canceled Appoinment
0. Exit
Enter your choice: 2
Enter CancelAppointment ID to update (blank to exit update process):
Update process exited.

Do you want to have action for Canceled Appoinment? (Y/N): y

Select an action:
1. Add New Canceled Appoinment
2. Edit Canceled Appoinment
0. Exit
Enter your choice: 2
Enter CancelAppointment ID to update (blank to exit update process): CA2
No relevant search results found.
```

After entering a valid **Cancel Appointment ID**, the system lets you update fields, leaving blanks unchanged. It confirms successful updates and shows the updated information.

```
Do you want to have action for Canceled Appoinment? (Y/N): y

Select an action:
1. Add New Canceled Appoinment
2. Edit Canceled Appoinment
0. Exit
Enter your choice: 2
Enter CancelAppointment ID to update (blank to exit update process): CA1
Enter new values for the following fields. Leave empty to skip updating the field.
AppointmentID:
DoctorID:
Reason: out of trip
Data updated successfully!
(null):CancelAppointmentID = CA1
AppointmentID = A3
DoctorID = E1
Reason = out of trip
```

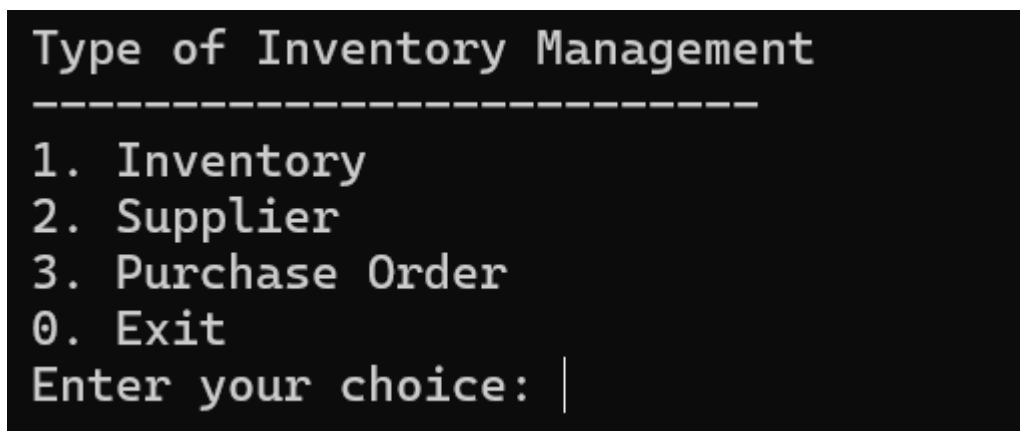
SQLite is showing the updated cancel appointment information.

| CancelAppointmentID | AppointmentID | DoctorID | Reason      |
|---------------------|---------------|----------|-------------|
| Filter              | Filter        | Filter   | Filter      |
| 1 CA1               | A3            | E1       | out of trip |

**Note:** The Patient Appointment and Patient Cancelled Appointment tables do not allow data deletion. When a patient cancels an appointment, it is recorded in the Cancelled Appointment table, and the corresponding appointment in the Patient Appointment table is updated with a status of "Cancelled." This approach maintains the integrity of appointment records and preserves a complete history of all appointments.

#### **6.6.8 Inventory Management**

The Inventory Management presents choices for managing inventory, suppliers, and purchase orders. Select Exit will go back to the admin menu.



### **6.6.8.1 Inventory**

The system displays details of the status of inventory. After showing this information, it prompts whether you want to perform another action related to inventory.

#### **Type of Inventory Management**

---

- 1. Inventory
- 2. Supplier
- 3. Purchase Order
- 0. Exit

Enter your choice: 1

#### **Inventory**

```
(null):ItemID = I1
PurchaseItem = Syringes
Quantity = 500
PersonChecking = E3
```

Do you want to have action for Inventory? (Y/N): |

#### **6.6.8.1.1 Add Inventory**

The system requests details such as the purchase item, quantity, and user ID. After successfully adding the new inventory, the updated list is displayed. The user is then asked if they want to add more inventory.

```
Do you want to have action for Inventory? (Y/N): y
```

```
Select an action:
```

- 1. Add New Inventory
- 2. Edit Inventory
- 3. Delete Inventory
- 0. Exit

```
Enter your choice: 1
```

```
Add New Inventory
```

```
*****
```

```
Enter Purchase Item: Bandages
```

```
Enter Quantity: 100
```

```
Enter User ID ( valid options are: E1 E2 E3 ): E3
```

```
Inventory added successfully!
```

```
(null):ItemID = I1
```

```
PurchaseItem = Syringes
```

```
Quantity = 500
```

```
PersonChecking = E3
```

```
(null):ItemID = I2
```

```
PurchaseItem = Bandages
```

```
Quantity = 100
```

```
PersonChecking = E3
```

```
Do you want to add another Inventory? (Y/N): |
```

SQLite shows the new inventory data successfully.

|   | ItemID | PurchaseItem | Quantity | PersonChecking |
|---|--------|--------------|----------|----------------|
|   | Filter | Filter       | Filter   | Filter         |
| 1 | I1     | Syringes     | 500      | E3             |
| 2 | I2     | Bandages     | 100      | E3             |

#### 6.6.8.1.2 Update Inventory

The system prompts for the **Inventory ID**. If blank, the update process exits. If it doesn't match any records, a **no relevant search results found** message is shown, and the update process exits.

```
Do you want to have action for Inventory? (Y/N): y
Select an action:
1. Add New Inventory
2. Edit Inventory
3. Delete Inventory
0. Exit
Enter your choice: 2
Enter Inventory ID to update (blank to exit update process):
Update process exited.

Do you want to have action for Inventory? (Y/N): y
Select an action:
1. Add New Inventory
2. Edit Inventory
3. Delete Inventory
0. Exit
Enter your choice: 2
Enter Inventory ID to update (blank to exit update process): I3
No relevant search results found.
```

After entering a valid **Inventory ID**, the system lets you update fields, leaving blanks unchanged. It confirms successful updates and shows the updated information.

```
Do you want to have action for Inventory? (Y/N): y
Select an action:
1. Add New Inventory
2. Edit Inventory
3. Delete Inventory
0. Exit
Enter your choice: 2
Enter Inventory ID to update (blank to exit update process): I2
Enter new values for the following fields. Leave empty to skip updating the field.
PurchaseItem: Face mask
Quantity: 200
PersonChecking:
Data updated successfully!

(null):ItemID = I1
PurchaseItem = Syringes
Quantity = 500
PersonChecking = E3

(null):ItemID = I2
PurchaseItem = Face mask
Quantity = 200
PersonChecking = E3
```

SQLite is showing the updated inventory information.

|   | ItemID | PurchaseItem | Quantity | PersonChecking |
|---|--------|--------------|----------|----------------|
|   | Filter | Filter       | Filter   | Filter         |
| 1 | I1     | Syringes     | 500      | E3             |
| 2 | I2     | Face mask    | 200      | E3             |

#### 6.6.8.1.3 Delete Inventory

The system prompts for the **Inventory ID**. If blank, the deletion process exits. If it doesn't match any records, a **no relevant search results found** message is shown, and the deletion process exits.

```
Do you want to have action for Inventory? (Y/N): y

Select an action:
1. Add New Inventory
2. Edit Inventory
3. Delete Inventory
0. Exit
Enter your choice: 3
Enter Inventory ID to delete (blank to exit delete process):
Deletion process exited.

Do you want to have action for Inventory? (Y/N): y

Select an action:
1. Add New Inventory
2. Edit Inventory
3. Delete Inventory
0. Exit
Enter your choice: 3
Enter Inventory ID to delete (blank to exit delete process): I3
No relevant search results found.
```

After entering a valid **Inventory ID**, the system successfully deletes the user and confirms the deletion. Then it displays the remaining users and asks if you want to perform another action.

```
Do you want to have action for Inventory? (Y/N): y

Select an action:
1. Add New Inventory
2. Edit Inventory
3. Delete Inventory
0. Exit
Enter your choice: 3
Enter Inventory ID to delete (blank to exit delete process): I2
Data deleted successfully!

(null):ItemID = I1
PurchaseItem = Syringes
Quantity = 500
PersonChecking = E3

Do you want to have action for Inventory? (Y/N): |
```

SQLite shows the inventory data is deleted.

| ItemID | PurchaseItem | Quantity | PersonChecking |
|--------|--------------|----------|----------------|
| Filter | Filter       | Filter   | Filter         |
| 1 I1   | Syringes     | 500      | E3             |

### **6.6.8.2 Supplier**

The system displays details of the suppliers. After showing this information, it prompts whether you want to perform another action related to supplier.

#### **Type of Inventory Management**

---

- 1. Inventory
- 2. Supplier
- 3. Purchase Order
- 0. Exit

Enter your choice: 2

Supplier  
(null):SupplierID = SU1  
SupplierName = ABC Medical  
ContactNumber = 60167456215

Do you want to have action for Supplier? (Y/N): |

#### 6.6.8.2.1 Add Supplier

When adding a new supplier, the user is asked to enter the supplier name and contact number. If an invalid contact number (less than 10 digits) is entered, an error message is shown. Once valid information is provided, the supplier is added successfully, and the updated list is displayed. The user is then asked if they wish to add another supplier.

```
Do you want to have action for Supplier? (Y/N): y

Select an action:
1. Add New Supplier
2. Edit Supplier
3. Delete Supplier
0. Exit
Enter your choice: 1

Add New Supplier
*****
Enter Supplier Name: XYZ Supplies
Enter Contact Number: 2
Invalid Contact Number. Please enter a number with at least 10 digits.
Enter Contact Number: 60145951716
Supplier added successfully!

(null):SupplierID = SU1
SupplierName = ABC Medical
ContactNumber = 60167456215

(null):SupplierID = SU2
SupplierName = XYZ Supplies
ContactNumber = 60145951716

Do you want to add another Supplier? (Y/N): |
```

SQLite shows add the new supplier data successfully.

|   | SupplierID | SupplierName | ContactNumber |
|---|------------|--------------|---------------|
|   | Filter     | Filter       | Filter        |
| 1 | SU1        | ABC Medical  | 60167456215   |
| 2 | SU2        | XYZ Supplies | 60145951716   |

### 6.6.8.2.2 Update Supplier

The system prompts for the **Supplier ID**. If blank, the update process exits. If it doesn't match any records, a **no relevant search results found** message is shown, and the update process exits.

```
Do you want to have action for Supplier? (Y/N): y

Select an action:
1. Add New Supplier
2. Edit Supplier
3. Delete Supplier
0. Exit
Enter your choice: 2
Enter Supplier ID to update (blank to exit update process):
Update process exited.

Do you want to have action for Supplier? (Y/N): y

Select an action:
1. Add New Supplier
2. Edit Supplier
3. Delete Supplier
0. Exit
Enter your choice: 2
Enter Supplier ID to update (blank to exit update process): 3
No relevant search results found.
```

After entering a valid **Supplier ID**, the system lets you update fields, leaving blanks unchanged. It confirms successful updates and shows the updated information.

```
Do you want to have action for Supplier? (Y/N): y
Select an action:
1. Add New Supplier
2. Edit Supplier
3. Delete Supplier
0. Exit
Enter your choice: 2
Enter Supplier ID to update (blank to exit update process): SU2
Enter new values for the following fields. Leave empty to skip updating the field.
SupplierName: YZ Supplier
ContactNumber:
Data updated successfully!

(null):SupplierID = SU1
SupplierName = ABC Medical
ContactNumber = 60167456215

(null):SupplierID = SU2
SupplierName = YZ Supplier
ContactNumber = 60145951716
```

SQLite is showing the updated supplier information.

|   | SupplierID | SupplierName | ContactNumber |
|---|------------|--------------|---------------|
|   | Filter     | Filter       | Filter        |
| 1 | SU1        | ABC Medical  | 60167456215   |
| 2 | SU2        | YZ Supplier  | 60145951716   |

#### 6.6.8.2.3 Delete Supplier

The system prompts for the **Supplier ID**. If blank, the deletion process exits. If it doesn't match any records, a **no relevant search results found** message is shown, and the deletion process exits.

```
Do you want to have action for Supplier? (Y/N): y
Select an action:
1. Add New Supplier
2. Edit Supplier
3. Delete Supplier
0. Exit
Enter your choice: 3
Enter Supplier ID to delete (blank to exit delete process):
Deletion process exited.

Do you want to have action for Supplier? (Y/N): y
Select an action:
1. Add New Supplier
2. Edit Supplier
3. Delete Supplier
0. Exit
Enter your choice: 3
Enter Supplier ID to delete (blank to exit delete process): 6
No relevant search results found.
```

After entering a valid **Supplier ID**, the system successfully deletes the user and confirms the deletion. Then it displays the remaining users and asks if you want to perform another action.

```
Do you want to have action for Supplier? (Y/N): y
Select an action:
1. Add New Supplier
2. Edit Supplier
3. Delete Supplier
0. Exit
Enter your choice: 3
Enter Supplier ID to delete (blank to exit delete process): SU2
Data deleted successfully!

(null):SupplierID = SU1
SupplierName = ABC Medical
ContactNumber = 60167456215

Do you want to have action for Supplier? (Y/N): |
```

SQLite shows the supplier data is deleted.

|   | SupplierID | SupplierName | ContactNumber |
|---|------------|--------------|---------------|
|   | Filter     | Filter       | Filter        |
| 1 | SU1        | ABC Medical  | 60167456215   |
|   |            |              |               |

### **6.6.8.3 Purchase Order**

The system displays details of purchase order records. After showing this information, it prompts whether you want to perform another action related to purchase order.

```
Type of Inventory Management
```

- 
- 1. Inventory
  - 2. Supplier
  - 3. Purchase Order
  - 0. Exit

```
Enter your choice: 3
```

```
Purchase Order
```

```
(null):POID = P01
```

```
SupplierID = SU1
```

```
ItemID = I1
```

```
PricePerBox = 60.0
```

```
QuantityInBox = 100
```

```
PurchaseQuantityBox = 8
```

```
TotalAmount = 480.0
```

```
Do you want to have action for Purchase Order? (Y/N): |
```

#### 6.6.8.3.1 Add Purchase Order

If invalid `Supplier ID` `Supplier ID` are entered, error messages prompt the user to select valid options. Once the correct IDs are entered, the user provides the price per box, quantity in each box, and the number of boxes to be purchased. The system calculates the total amount and confirms the successful addition of the purchase order. Finally, it asks if the user wants to add another purchase order.

```
Do you want to have action for Purchase Order? (Y/N): y

Select an action:
1. Add New Purchase Order
2. Edit Purchase Order
3. Delete Purchase Order
0. Exit
Enter your choice: 1
Enter Supplier ID (valid options are: SU1 ): SU2
Invalid Supplier ID. Please enter a valid Supplier ID from the list.
Valid options are: SU1
Enter Supplier ID (valid options are: SU1 ): SU1
Enter Item ID (valid options are: I1 ): I2
Invalid Item ID. Please enter a valid Item ID from the list.
Valid options are: I1
Enter Item ID (valid options are: I1 ): I1
Enter Price per Box: 60
Enter Quantity in Box: 100
Enter Purchase Quantity in Box: 3
Purchase added successfully!

(null):POID = P01
SupplierID = SU1
ItemID = I1
PricePerBox = 60.0
QuantityInBox = 100
PurchaseQuantityBox = 3
TotalAmount = 480.0

(null):POID = P02
SupplierID = SU1
ItemID = I1
PricePerBox = 60.0
QuantityInBox = 100
PurchaseQuantityBox = 3
TotalAmount = 180.0

Do you want to add another Purchase? (Y/N): |
```

SQLite shows the new purchase order data successfully.

|   | POID   | SupplierID | ItemID | PricePerBox | QuantityInBox | PurchaseQuantityBox | TotalAmount |
|---|--------|------------|--------|-------------|---------------|---------------------|-------------|
|   | Filter | Filter     | Filter | Filter      | Filter        | Filter              | Filter      |
| 1 | PO1    | SU1        | I1     | 60.0        | 100           | 8                   | 480.0       |
| 2 | PO2    | SU1        | I1     | 60          | 100           | 3                   | 180         |

#### 6.6.8.3.2 Update Purchase Order

The system prompts for the **Purchase ID**. If blank, the update process exits. If it doesn't match any records, a **no relevant search results found** message is shown, and the update process exits.

```
Do you want to have action for Purchase Order? (Y/N): y
Select an action:
1. Add New Purchase Order
2. Edit Purchase Order
3. Delete Purchase Order
0. Exit
Enter your choice: 2
Enter Purchase ID to update (blank to exit update process):
Update process exited.

Do you want to have action for Purchase Order? (Y/N): y
Select an action:
1. Add New Purchase Order
2. Edit Purchase Order
3. Delete Purchase Order
0. Exit
Enter your choice: 2
Enter Purchase ID to update (blank to exit update process): 3
No relevant search results found.
```

After entering a valid **Purchase ID**, the system lets you update fields, leaving blanks unchanged. It confirms successful updates and shows the updated information.

```
Do you want to have action for Purchase Order? (Y/N): y
Select an action:
1. Add New Purchase Order
2. Edit Purchase Order
3. Delete Purchase Order
0. Exit
Enter your choice: 2
Enter Purchase ID to update (blank to exit update process): P02
Enter new values for the following fields. Leave empty to skip updating the field.
SupplierID:
ItemID:
PricePerBox: 50
QuantityInBox:
PurchaseQuantityBox: 5
Calculated TotalAmount: 250
Data updated successfully!

(null):POID = P01
SupplierID = SU1
ItemID = I1
PricePerBox = 60.0
QuantityInBox = 100
PurchaseQuantityBox = 8
TotalAmount = 480.0

(null):POID = P02
SupplierID = SU1
ItemID = I1
PricePerBox = 50.0
QuantityInBox = 100
PurchaseQuantityBox = 5
TotalAmount = 250.0
```

SQLite is showing the updated purchase order information.

|   | POID   | SupplierID | ItemID | PricePerBox | QuantityInBox | PurchaseQuantityBox | TotalAmount |
|---|--------|------------|--------|-------------|---------------|---------------------|-------------|
|   | Filter | Filter     | Filter | Filter      | Filter        | Filter              | Filter      |
| 1 | PO1    | SU1        | I1     | 60.0        | 100           | 8                   | 480.0       |
| 2 | PO2    | SU1        | I1     | 50.0        | 100           | 5                   | 250.0       |

#### 6.6.8.3.3 Delete Purchase Order

The system prompts for the **Purchase ID**. If blank, the deletion process exits. If it doesn't match any records, a **no relevant search results found** message is shown, and the deletion process exits.

```
Do you want to have action for Purchase Order? (Y/N): y
Select an action:
1. Add New Purchase Order
2. Edit Purchase Order
3. Delete Purchase Order
0. Exit
Enter your choice: 3
Enter Purchase ID to delete (blank to exit delete process):
Deletion process exited.

Do you want to have action for Purchase Order? (Y/N): y3
Select an action:
1. Add New Purchase Order
2. Edit Purchase Order
3. Delete Purchase Order
0. Exit
Enter your choice: Enter Purchase ID to delete (blank to exit delete process): 6
No relevant search results found.
```

After entering a valid **Purchase ID**, the system successfully deletes the user and confirms the deletion. Then it displays the remaining users and asks if you want to perform another action.

```
Do you want to have action for Purchase Order? (Y/N): y

Select an action:
1. Add New Purchase Order
2. Edit Purchase Order
3. Delete Purchase Order
0. Exit
Enter your choice: 3
Enter Purchase ID to delete (blank to exit delete process): P02
Data deleted successfully!
```

```
(null):POID = P01
SupplierID = SU1
ItemID = I1
PricePerBox = 60.0
QuantityInBox = 100
PurchaseQuantityBox = 8
TotalAmount = 480.0
```

```
Do you want to have action for Purchase Order? (Y/N): |
```

SQLite shows the purchase order data is deleted.

|   | POID   | SupplierID | ItemID | PricePerBox | QuantityInBox | PurchaseQuantityBox | TotalAmount |
|---|--------|------------|--------|-------------|---------------|---------------------|-------------|
|   | Filter | Filter     | Filter | Filter      | Filter        | Filter              | Filter      |
| 1 | PO1    | SU1        | I1     | 60.0        | 100           | 8                   | 480.0       |

## 6.6.9 View and Publish Notices

The system displays a list of notices with their IDs, descriptions, and the person who posted them. It then prompts the user to choose whether to take further action regarding notices.

```
View and Publish Notices
(null):NoticeID = N1
Description = Update emergency tools
PostManageBy = E3

(null):NoticeID = N2
Description = Vaccination Information
PostManageBy = E3
```

```
Do you want to have action for Notice? (Y/N): |
```

### 6.6.9.1 Add Notices

Ensuring the description and the User ID of the person managing the post are provided. If the User ID is invalid, the system prompts for a valid ID. After successfully adding the notice, it displays the updated list of notices and asks if the user wants to add another.

```
Do you want to have action for Notice? (Y/N): y
Select an action:
1. Add New Notice
2. Edit Notice
3. Delete Notice
0. Exit
Enter your choice: 1

Add New Notice
*****
Enter Description: Know more about your health
Enter User ID of the person who manages the post ( valid options are: E1 E2 E3 ): E6
Invalid User ID. Please enter a valid User ID from the list.
Valid options are: E1 E2 E3 : Enter User ID of the person who manages the post ( valid options are: E1 E2 E3 ): E3
Notice added successfully!

(null):NoticeID = N1
Description = Update emergency tools
PostManageBy = E3

(null):NoticeID = N2
Description = Vaccination Information
PostManageBy = E3

(null):NoticeID = N3
Description = Know more about your health
PostManageBy = E3

Do you want to add another notice? (Y/N): |
```

SQLite shows add the new notice data successfully.

|   | NoticeID | Description             | PostManageBy |
|---|----------|-------------------------|--------------|
|   | Filter   | Filter                  | Filter       |
| 1 | N1       | Update emergency tools  | E3           |
| 2 | N2       | Vaccination Information | E3           |
| 3 | N3       | Know more about you...  | E3           |

#### 6.6.9.2 Update Notices

The system prompts for the **Notice ID**. If blank, the update process exits. If it doesn't match any records, a **no relevant search results found** message is shown, and the update process exits.

```
Do you want to have action for Notice? (Y/N): y

Select an action:
1. Add New Notice
2. Edit Notice
3. Delete Notice
0. Exit
Enter your choice: 2
Enter Notice ID to update (blank to exit update process):
Update process exited.

Do you want to have action for Notice? (Y/N): y

Select an action:
1. Add New Notice
2. Edit Notice
3. Delete Notice
0. Exit
Enter your choice: 2
Enter Notice ID to update (blank to exit update process): 7
No relevant search results found.
```

After entering a valid **Notice ID**, the system lets you update fields, leaving blanks unchanged. It confirms successful updates and shows the updated information.

```
Do you want to have action for Notice? (Y/N): y
Select an action:
1. Add New Notice
2. Edit Notice
3. Delete Notice
0. Exit
Enter your choice: 2
Enter Notice ID to update (blank to exit update process): N3
Enter new values for the following fields. Leave empty to skip updating the field.
Description: Know more about body health
PostManageBy:
Data updated successfully!

(null):NoticeID = N1
Description = Update emergency tools
PostManageBy = E3

(null):NoticeID = N2
Description = Vaccination Information
PostManageBy = E3

(null):NoticeID = N3
Description = Know more about body health
PostManageBy = E3
```

SQLite is showing the updated notice information.

|   | NoticeID | Description                 | PostManageBy |
|---|----------|-----------------------------|--------------|
|   | Filter   | Filter                      | Filter       |
| 1 | N1       | Update emergency tools      | E3           |
| 2 | N2       | Vaccination Information     | E3           |
| 3 | N3       | Know more about body health | E3           |

### 6.6.9.3 Delete Notices

The system prompts for the **Notice ID**. If blank, the deletion process exits. If it doesn't match any records, a **no relevant search results found** message is shown, and the deletion process exits.

```
Do you want to have action for Notice? (Y/N): y

Select an action:
1. Add New Notice
2. Edit Notice
3. Delete Notice
0. Exit
Enter your choice: 3
Enter Notice ID to delete (blank to exit delete process):
Deletion process exited.

Do you want to have action for Notice? (Y/N): y3

Select an action:
1. Add New Notice
2. Edit Notice
3. Delete Notice
0. Exit
Enter your choice: Enter Notice ID to delete (blank to exit delete process): 9
No relevant search results found.
```

After entering a valid **Notice ID**, the system successfully deletes the user and confirms the deletion. Then it displays the remaining users and asks if you want to perform another action.

```
Do you want to have action for Notice? (Y/N): y
Select an action:
1. Add New Notice
2. Edit Notice
3. Delete Notice
0. Exit
Enter your choice: 3
Enter Notice ID to delete (blank to exit delete process): N3
Data deleted successfully!

(null):NoticeID = N1
Description = Update emergency tools
PostManageBy = E3

(null):NoticeID = N2
Description = Vaccination Information
PostManageBy = E3

Do you want to have action for Notice? (Y/N): |
```

SQLite shows the notice data is deleted.

|   | NoticeID | Description             | PostManageBy |
|---|----------|-------------------------|--------------|
|   | Filter   | Filter                  | Filter       |
| 1 | N1       | Update emergency tools  | E3           |
| 2 | N2       | Vaccination Information | E3           |

### **6.6.10 View and Publish Project**

The system displays the current project details and prompts whether the user wants to take further action on projects.

```
View and Publish Project
(null):ProjectID = PR01
Title = New Vaccination Guidelines
Description = Release of updated guidelines for vaccination protocols
DateOfPublication = 18/01/2024
```

```
Do you want to have action for Project? (Y/N): |
```

#### **6.6.10.1 Add Project**

The system allows adding a new project by entering the title, description, and publication date. It validates the date format and confirms successful addition of the project.

```
Do you want to have action for Project? (Y/N): y

Select an action:
1. Add New Project
2. Edit Project
3. Delete Project
0. Exit
Enter your choice: 1

Add New Project
*****
Enter Title: Mental Health Awareness
Enter Description: Campaign to promote mental health awareness
Enter Date Of Publication (YYYY-MM-DD): 60
Invalid date format. Please enter a valid date (YYYY-MM-DD).
Enter Date Of Publication (YYYY-MM-DD): 2024/08/24
Invalid date format. Please enter a valid date (YYYY-MM-DD).
Enter Date Of Publication (YYYY-MM-DD): 2024-08-24
Project added successfully!

(null):ProjectID = PR01
Title = New Vaccination Guidelines
Description = Release of updated guidelines for vaccination protocols
DateOfPublication = 18/01/2024

(null):ProjectID = PR02
Title = Mental Health Awareness
Description = Campaign to promote mental health awareness
DateOfPublication = 2024-08-24

Do you want to add another Project? (Y/N): |
```

SQLite shows the new project data successfully.

|   | ProjectID | Title                      | Description                       | DateOfPublication |
|---|-----------|----------------------------|-----------------------------------|-------------------|
|   | Filter    | Filter                     | Filter                            | Filter            |
| 1 | PRO1      | New Vaccination Guidelines | Release of updated guidelines ... | 18/01/2024        |
| 2 | PRO2      | Mental Health Awareness    | Campaign to promote mental ...    | 2024-08-24        |

#### 6.6.10.2 Update Project

The system prompts for the **Project ID**. If blank, the update process exits. If it doesn't match any records, a **no relevant search results found** message is shown, and the update process exits.

```
Do you want to have action for Project? (Y/N): y

Select an action:
1. Add New Project
2. Edit Project
3. Delete Project
0. Exit
Enter your choice: 2
Enter Project ID to update (blank to exit update process):
Update process exited.

Do you want to have action for Project? (Y/N): y

Select an action:
1. Add New Project
2. Edit Project
3. Delete Project
0. Exit
Enter your choice: 2
Enter Project ID to update (blank to exit update process): 6
No relevant search results found.
```

After entering a valid **Project ID**, the system lets you update fields, leaving blanks unchanged. It confirms successful updates and shows the updated information.

```
Do you want to have action for Project? (Y/N): y

Select an action:
1. Add New Project
2. Edit Project
3. Delete Project
0. Exit
Enter your choice: 2
Enter Project ID to update (blank to exit update process): PRO2
Enter new values for the following fields. Leave empty to skip updating the field.
Title:
Description:
DateOfPublication: 2024-10-20
Data updated successfully!
(null):ProjectID = PRO1
Title = New Vaccination Guidelines
Description = Release of updated guidelines for vaccination protocols
DateOfPublication = 18/01/2024

(null):ProjectID = PRO2
Title = Mental Health Awareness
Description = Campaign to promote mental health awareness
DateOfPublication = 2024-10-20
```

SQLite is showing the updated project information.

|   | ProjectID | Title                      | Description                       | DateOfPublication |
|---|-----------|----------------------------|-----------------------------------|-------------------|
|   | Filter    | Filter                     | Filter                            | Filter            |
| 1 | PRO1      | New Vaccination Guidelines | Release of updated guidelines ... | 18/01/2024        |
| 2 | PRO2      | Mental Health Awareness    | Campaign to promote mental ...    | 2024-10-20        |

### 6.6.10.3 Delete Project

The system prompts for the **Project ID**. If blank, the deletion process exits. If it doesn't match any records, a **no relevant search results found** message is shown, and the deletion process exits.

```
Do you want to have action for Project? (Y/N): y

Select an action:
1. Add New Project
2. Edit Project
3. Delete Project
0. Exit
Enter your choice: 3
Enter Project ID to delete (blank to exit delete process):
Deletion process exited.
```

```
Do you want to have action for Project? (Y/N): y

Select an action:
1. Add New Project
2. Edit Project
3. Delete Project
0. Exit
Enter your choice: 3
Enter Project ID to delete (blank to exit delete process): 9
No relevant search results found.
```

After entering a valid **Project ID**, the system successfully deletes the user and confirms the deletion. Then it displays the remaining users and asks if you want to perform another action.

```
Do you want to have action for Project? (Y/N): y

Select an action:
1. Add New Project
2. Edit Project
3. Delete Project
0. Exit
Enter your choice: 3
Enter Project ID to delete (blank to exit delete process): PRO2
Data deleted successfully!
(null):ProjectID = PRO1
Title = New Vaccination Guidelines
Description = Release of updated guidelines for vaccination protocols
DateOfPublication = 18/01/2024

Do you want to have action for Project? (Y/N): |
```

SQLite shows the project data is deleted.

| ProjectID | Title                      | Description                       | DateOfPublication |
|-----------|----------------------------|-----------------------------------|-------------------|
| Filter    | Filter                     | Filter                            | Filter            |
| 1 PRO1    | New Vaccination Guidelines | Release of updated guidelines ... | 18/01/2024        |

### 6.6.11 Discount

The system displays existing discount information and prompts whether to proceed with actions related to discounts.

```
Discount
(null):DiscountID = V1
DiscountAmount = 5%
IssuedDate = 05/01/2024

Do you want to have action for Discount? (Y/N): |
```

#### 6.6.11.1 Add Discount

The system prompts for a discount amount and issued date. If invalid input is given, it requests valid information. After successful addition, it confirms the new discount and displays existing discounts.

```
Add New Discount
*****
Enter Discount Amount (1-100 %): 0
Invalid input. Please enter a valid integer.
Enter Discount Amount (1-100 %): 20
Enter Issued Date (YYYY-MM-DD): 23
Invalid date format. Please enter a valid date (YYYY-MM-DD).
Enter Issued Date (YYYY-MM-DD): 2024-08-24
Discount added successfully!

(null):DiscountID = V1
DiscountAmount = 5%
IssuedDate = 05/01/2024

(null):DiscountID = V2
DiscountAmount = 20.000000%
IssuedDate = 2024-08-24

Do you want to add another Discount? (Y/N): |
```

SQLite shows adding the new discount data successfully.

|   | DiscountID | DiscountAmount | IssuedDate |
|---|------------|----------------|------------|
|   | Filter     | Filter         | Filter     |
| 1 | V1         | 5%             | 05/01/2024 |
| 2 | V2         | 20.000000%     | 2024-08-24 |

### 6.6.11.2 Update Discount

The system prompts for the **Discount ID**. If blank, the update process exits. If it doesn't match any records, a **no relevant search results found** message is shown, and the update process exits.

```
Do you want to have action for Discount? (Y/N): y

Select an action:
1. Add New Discount
2. Edit Discount
3. Delete Discount
0. Exit
Enter your choice: 2
Enter Discount ID to update (blank to exit update process):
Update process exited.

Do you want to have action for Discount? (Y/N): y

Select an action:
1. Add New Discount
2. Edit Discount
3. Delete Discount
0. Exit
Enter your choice: 2
Enter Discount ID to update (blank to exit update process): 3
No relevant search results found.
```

After entering a valid **Discount ID**, the system lets you update fields, leaving blanks unchanged. It confirms successful updates and shows the updated information.

```
Do you want to have action for Discount? (Y/N): y
Select an action:
1. Add New Discount
2. Edit Discount
3. Delete Discount
0. Exit
Enter your choice: 2
Enter Discount ID to update (blank to exit update process): V2
Enter new values for the following fields. Leave empty to skip updating the field.

DiscountAmount: 10
IssuedDate:
Data updated successfully!

(null):DiscountID = V1
DiscountAmount = 5%
IssuedDate = 05/01/2024

(null):DiscountID = V2
DiscountAmount = 10%
IssuedDate = 2024-08-24
```

SQLite is showing the updated discount information.

|   | DiscountID | DiscountAmount | IssuedDate |
|---|------------|----------------|------------|
|   | Filter     | Filter         | Filter     |
| 1 | V1         | 5%             | 05/01/2024 |
| 2 | V2         | 10%            | 2024-08-24 |

### 6.6.11.3 Delete Discount

The system prompts for the **Discount ID**. If blank, the deletion process exits. If it doesn't match any records, a **no relevant search results found** message is shown, and the deletion process exits.

```
Do you want to have action for Discount? (Y/N): y

Select an action:
1. Add New Discount
2. Edit Discount
3. Delete Discount
0. Exit
Enter your choice: 3
Enter Discount ID to delete (blank to exit delete process):
Deletion process exited.
```

```
Do you want to have action for Discount? (Y/N): y

Select an action:
1. Add New Discount
2. Edit Discount
3. Delete Discount
0. Exit
Enter your choice: 3
Enter Discount ID to delete (blank to exit delete process): 9
No relevant search results found.
```

After entering a valid **Discount ID**, the system successfully deletes the user and confirms the deletion. Then it displays the remaining users and asks if you want to perform another action.

```
Do you want to have action for Discount? (Y/N): y

Select an action:
1. Add New Discount
2. Edit Discount
3. Delete Discount
0. Exit
Enter your choice: 3
Enter Discount ID to delete (blank to exit delete process): V2
Data deleted successfully!

(null):DiscountID = V1
DiscountAmount = 5%
IssuedDate = 05/01/2024
```

```
Do you want to have action for Discount? (Y/N): |
```

SQLite shows the discount data is deleted.

|   | DiscountID | DiscountAmount | IssuedDate |
|---|------------|----------------|------------|
|   | Filter     | Filter         | Filter     |
| 1 | V1         | 5%             | 05/01/2024 |
|   |            |                |            |

### **6.6.12 Patient Admission Transfer**

The system displays details of the patient's admission transfer. Admin only has read rights.

```
Patient Admission Transfer
(null):PatientStatusID = PS1
PatientID = P1
CheckInTime = 06/01/2024 10.30
CheckOutTime = 08/01/2024 12.00
CurrentStatus = Discharged

(null):PatientStatusID = PS2
PatientID = P2
CheckInTime = 10/02/2024 13.00
CheckOutTime = 17/02/2024 16.00
CurrentStatus = Transferred

(null):PatientStatusID = PS3
PatientID = P3
CheckInTime = 15/03/2024 13.00
CheckOutTime = 15/03/2024 15.30
CurrentStatus = Discharged
```

### **6.6.13 View Feedback**

The system displays details of the patient's feedback for the clinic. Admin only has read rights.

```
View Feedback
(null):FeedbackID = F1
PatientID = P1
Comment = Excellent care, very satisfied with service.
Rating = 5
Date = 08/01/2024
```

### **6.6.14 Patient Vaccination Records**

The system displays details of the patient's vaccination records. Admin only has read rights.

```
Patient Vaccination Records
(null):VaccinationID = V1
PatientID = P1
VaccineType = AZ
DoseNumber = First dose
AdministrationDate = 16/09/2021

(null):VaccinationID = V2
PatientID = P3
VaccineType = Pfizer
DoseNumber = First dose
AdministrationDate = 01/10/2021
```

### **6.6.15 History of Patient Allergies**

The system displays details of the history of patient allergies. Admin only has read rights.

```
History of Patient Allergies
(null):PatientAllergyID = AI1
PatientID = P2
AllergyName = Peanut
Severity = Severe
Reaction = Anaphylaxis (life-threatening allergic reaction)
```

### **6.6.16 Latest Patient Status**

The system displays details of the patient's admission transfer. Admin only has read rights.

```
Latest Patient Status
(null):FollowUpRecordID = FU1
PatientHealthID = PH1
FollowUpDate = 07/01/2024
Description = Mild fever, joint pain; no rash
DoctorOrders = Monitor platelets, rest. Follow-up in 1 week
DoctorID = E1

(null):FollowUpRecordID = FU2
PatientHealthID = PH2
FollowUpDate = 13/02/2024
Description = Reduced fatigue, better energy
DoctorOrders = Continue treatment, add Vitamin C. Test in 1 month
DoctorID = E2
```

### 6.6.17 Emergency Tools

The system displays existing emergency tools and asks if you want to perform actions related to them.

```
Emergency Tools
(null):EmergencyToolsID = ET1
Type = Contact Information
Tool = Emergency Number
EmergencyNumber = 03-86457894
```

```
Do you want to have action for Emergency Tools? (Y/N): |
```

#### 6.6.17.1 Add Emergency Tools

The system will verify the Contact Number, it needs at least 3 digits. After adding data successfully, it displays the updated list of emergency tools and asks if you want to add more.

```
Add New Emergency Tools
*****
Enter Type: Transportation
Enter Tool: Ambulance Department Service
Enter Emergency Number: 0
Invalid Contact Number. Please enter a number with at least 3 digits.
Enter Emergency Number: 999
EmergencyTools added successfully!
(null):EmergencyToolsID = ET1
Type = Contact Information
Tool = Emergency Number
EmergencyNumber = 03-86457894

(null):EmergencyToolsID = ET2
Type = Transportation
Tool = Ambulance Department Service
EmergencyNumber = 999

Do you want to add another Emergency Tools? (Y/N): |
```

SQLite shows the new emergency tool data successfully.

|   | EmergencyToolsID | Type                | Tool                         | EmergencyNumber |
|---|------------------|---------------------|------------------------------|-----------------|
|   | Filter           | Filter              | Filter                       | Filter          |
| 1 | ET1              | Contact Information | Emergency Number             | 03-86457894     |
| 2 | ET2              | Transportation      | Ambulance Department Service | 999             |

### 6.6.17.2 Update Emergency Tools

The system prompts for the **Emergency Tools ID**. If blank, the update process exits. If it doesn't match any records, a **no relevant search results found** message is shown, and the update process exits.

```
Do you want to have action for Emergency Tools? (Y/N): y
Select an action:
1. Add New Emergency Tools
2. Edit Emergency Tools
3. Delete Emergency Tools
0. Exit
Enter your choice: 2
Enter Emergency Tools ID to update (blank to exit update process):
Update process exited.

Do you want to have action for Emergency Tools? (Y/N): y
Select an action:
1. Add New Emergency Tools
2. Edit Emergency Tools
3. Delete Emergency Tools
0. Exit
Enter your choice: 2
Enter Emergency Tools ID to update (blank to exit update process): 6
No relevant search results found.
```

After entering a valid **Emergency Tools ID**, the system lets you update fields, leaving blanks unchanged. It confirms successful updates and shows the updated information.

```
Do you want to have action for Emergency Tools? (Y/N): y

Select an action:
1. Add New Emergency Tools
2. Edit Emergency Tools
3. Delete Emergency Tools
0. Exit
Enter your choice: 2
Enter Emergency Tools ID to update (blank to exit update process): ET2
Enter new values for the following fields. Leave empty to skip updating the field.
Type:
Tool:
EmergencyNumber: 110
Data updated successfully!

(null):EmergencyToolsID = ET1
Type = Contact Information
Tool = Emergency Number
EmergencyNumber = 03-86457894

(null):EmergencyToolsID = ET2
Type = Transportation
Tool = Ambulance Department Service
EmergencyNumber = 110
```

SQLite is showing the updated emergency tools information.

|   | EmergencyToolsID | Type                | Tool                         | EmergencyNumber |
|---|------------------|---------------------|------------------------------|-----------------|
|   | Filter           | Filter              | Filter                       | Filter          |
| 1 | ET1              | Contact Information | Emergency Number             | 03-86457894     |
| 2 | ET2              | Transportation      | Ambulance Department Service | 110             |

### 6.6.17.3 Delete Emergency Tools

The system prompts for the **Emergency Tools ID**. If blank, the deletion process exits. If it doesn't match any records, a **no relevant search results found** message is shown, and the deletion process exits.

```
Do you want to have action for Emergency Tools? (Y/N): y
Select an action:
1. Add New Emergency Tools
2. Edit Emergency Tools
3. Delete Emergency Tools
0. Exit
Enter your choice: 3
Enter Emergency Tools ID to delete (blank to exit delete process):
Deletion process exited.

Do you want to have action for Emergency Tools? (Y/N): y
Select an action:
1. Add New Emergency Tools
2. Edit Emergency Tools
3. Delete Emergency Tools
0. Exit
Enter your choice: 3
Enter Emergency Tools ID to delete (blank to exit delete process): 9
No relevant search results found.
```

After entering a valid **Emergency Tools ID**, the system successfully deletes the user and confirms the deletion. Then it displays the remaining users and asks if you want to perform another action.

```
Do you want to have action for Emergency Tools? (Y/N): y

Select an action:
1. Add New Emergency Tools
2. Edit Emergency Tools
3. Delete Emergency Tools
0. Exit
Enter your choice: 3
Enter Emergency Tools ID to delete (blank to exit delete process): ET2
Data deleted successfully!

(null):EmergencyToolsID = ET1
Type = Contact Information
Tool = Emergency Number
EmergencyNumber = 03-86457894

Do you want to have action for Emergency Tools? (Y/N): |
```

SQLite shows the emergency tools data is deleted.

|   | EmergencyToolsID | Type                | Tool             | EmergencyNumber |
|---|------------------|---------------------|------------------|-----------------|
|   | Filter           | Filter              | Filter           | Filter          |
| 1 | ET1              | Contact Information | Emergency Number | 03-86457894     |

### **6.6.18 Billing**

The system displays details of the billing, also the invoices for patients. After showing this information, it prompts whether you want to perform another action related to billing.

```
Billing
(null):BillID = B1
PatientID = P1
DateIssued = 07/01/2024
SubTotal = RM 542.00
DiscountPercentage = -
TotalAmount = RM 542.00
PaymentStatus = Unpaid

(null):BillID = B2
PatientID = P2
DateIssued = 16/02/2024
SubTotal = RM 873.33
DiscountPercentage = 10%
TotalAmount = RM 786.00
PaymentStatus = Unpaid
```

Do you want to have action for Billing? (Y/N): |

### 6.6.18.1 Add Billing

The system verifies the patient ID and date format, calculates the total amount considering any discount percentage, and records the payment status. Invalid inputs, such as incorrect patient IDs or date formats, are flagged for correction. For discounts, you can use '-' if there's no discount or specify a percentage (e.g., 10%). The system calculates the total amount by applying the discount to the subtotal.

```
Add New Billing
*****
Enter Patient ID (valid options are: P1 P2 ): P3
Invalid Patient ID. Please enter a valid Patient ID from the list.
Valid options are: P1 P2
Enter Patient ID (valid options are: P1 P2 ): P1
Enter Date Issued (YYYY-MM-DD): 30
Invalid date format. Please enter a valid date (YYYY-MM-DD).
Enter Date Issued (YYYY-MM-DD): 2024-08-24
Enter Sub Total: RM 60
Enter Discount Percentage (use '-' if no discount, or percentage e.g., 10%): -
Enter Payment Status (Unpaid/Paid): Paid
Billing added successfully!

(null):BillID = B1
PatientID = P1
DateIssued = 07/01/2024
SubTotal = RM 542.00
DiscountPercentage = -
TotalAmount = RM 542.00
PaymentStatus = Unpaid

(null):BillID = B2
PatientID = P2
DateIssued = 16/02/2024
SubTotal = RM 873.33
DiscountPercentage = 10%
TotalAmount = RM 786.00
PaymentStatus = Unpaid

(null):BillID = B3
PatientID = P1
DateIssued = 2024-08-24
SubTotal = RM60.00
DiscountPercentage = 0.00%
TotalAmount = RM60.00
PaymentStatus = Paid

Do you want to add another Billing? (Y/N): y
```

```
Add New Billing
*****
Enter Patient ID (valid options are: P1 P2 ): P2
Enter Date Issued (YYYY-MM-DD): 2024-08-24
Enter Sub Total: RM 80
Enter Discount Percentage (use '-' if no discount, or percentage e.g., 10%): 20
Enter Payment Status (Unpaid/Paid): Unpaid
Billing added successfully!

(null):BillID = B1
PatientID = P1
DateIssued = 07/01/2024
SubTotal = RM 542.00
DiscountPercentage = -
TotalAmount = RM 542.00
PaymentStatus = Unpaid

(null):BillID = B2
PatientID = P2
DateIssued = 16/02/2024
SubTotal = RM 873.33
DiscountPercentage = 10%
TotalAmount = RM 786.00
PaymentStatus = Unpaid

(null):BillID = B3
PatientID = P1
DateIssued = 2024-08-24
SubTotal = RM60.00
DiscountPercentage = 0.00%
TotalAmount = RM60.00
PaymentStatus = Paid

(null):BillID = B4
PatientID = P2
DateIssued = 2024-08-24
SubTotal = RM80.00
DiscountPercentage = 20.00%
TotalAmount = RM64.00
PaymentStatus = Unpaid
```

SQLite shows add the new cancel billing data successfully.

### 6.6.18.2 Update Billing

The system prompts for the **Bill ID**. If blank, the update process exits. If it doesn't match any records, a **no relevant search results found** message is shown, and the update process exits.

```
Do you want to have action for Billing? (Y/N): y

Select an action:
1. Add New Billing
2. Edit Billing
3. Delete Billing
0. Exit
Enter your choice: 2
Enter Billing ID to update (blank to exit update process):
Update process exited.

Do you want to have action for Billing? (Y/N): y

Select an action:
1. Add New Billing
2. Edit Billing
3. Delete Billing
0. Exit
Enter your choice: 2
Enter Billing ID to update (blank to exit update process): 9
No relevant search results found.
```

After entering a valid **Bill ID**, the system lets you update fields, leaving blanks unchanged. It confirms successful updates and shows the updated information.

```
Do you want to have action for Billing? (Y/N): y

Select an action:
1. Add New Billing
2. Edit Billing
3. Delete Billing
0. Exit
Enter your choice: 2
Enter Billing ID to update (blank to exit update process): B4
Enter new values for the following fields. Leave empty to skip updating the field.
PatientID:
DateIssued (YYYY/MM//DD):
Enter Sub Total: 100
Enter Discount Percentage (use '-' if no discount, or percentage e.g., 10%): 10
Payment Status (Unpaid/Paid):
Data updated successfully!
(null):BillID = B1
PatientID = P1
DateIssued = 07/01/2024
SubTotal = RM542.00
DiscountPercentage = -
TotalAmount = RM542.00
PaymentStatus = Unpaid

(null):BillID = B2
PatientID = P2
DateIssued = 16/02/2024
SubTotal = RM873.33
DiscountPercentage = 10%
TotalAmount = RM786.00
PaymentStatus = Unpaid

(null):BillID = B3
PatientID = P1
DateIssued = 2024-08-24
SubTotal = RM60.00
DiscountPercentage = 0.00%
TotalAmount = RM60.00
PaymentStatus = Paid

(null):BillID = B4
PatientID = P2
DateIssued = 2024-08-24
SubTotal = RM100.00
DiscountPercentage = 10.00%
TotalAmount = RM90.00
PaymentStatus = Unpaid
```

SQLite is showing the updated user information.

### 6.6.18.3 Delete Billing

The system prompts for the **Bill ID**. If blank, the deletion process exits. If it doesn't match any records, a **no relevant search results found** message is shown, and the deletion process exits.

```
Do you want to have action for Billing? (Y/N): y
Select an action:
1. Add New Billing
2. Edit Billing
3. Delete Billing
0. Exit
Enter your choice: 3
Enter Billing ID to delete (blank to exit delete process):
Deletion process exited.

Do you want to have action for Billing? (Y/N): y
Select an action:
1. Add New Billing
2. Edit Billing
3. Delete Billing
0. Exit
Enter your choice: 3
Enter Billing ID to delete (blank to exit delete process): 9
No relevant search results found.
```

After entering a valid **Bill ID**, the system successfully deletes the user and confirms the deletion. Then it displays the remaining users and asks if you want to perform another action.

```
Do you want to have action for Billing? (Y/N): y

Select an action:
1. Add New Billing
2. Edit Billing
3. Delete Billing
0. Exit
Enter your choice: 3
Enter Billing ID to delete (blank to exit delete process): B4
Data deleted successfully!

(null):BillID = B1
PatientID = P1
DateIssued = 07/01/2024
SubTotal = RM542.00
DiscountPercentage = -
TotalAmount = RM542.00
PaymentStatus = Unpaid

(null):BillID = B2
PatientID = P2
DateIssued = 16/02/2024
SubTotal = RM873.33
DiscountPercentage = 10%
TotalAmount = RM786.00
PaymentStatus = Unpaid

(null):BillID = B3
PatientID = P1
DateIssued = 2024-08-24
SubTotal = RM60.00
DiscountPercentage = 0.00%
TotalAmount = RM60.00
PaymentStatus = Paid

Do you want to have action for Billing? (Y/N): |
```

SQLite shows the billing data is deleted.

|   | BillID | PatientID | DateIssued | SubTotal | DiscountPercentage | TotalAmount | PaymentStatus |
|---|--------|-----------|------------|----------|--------------------|-------------|---------------|
|   | Filter | Filter    | Filter     | Filter   | Filter             | Filter      | Filter        |
| 1 | B1     | P1        | 07/01/2024 | RM542.00 | -                  | RM542.00    | Unpaid        |
| 2 | B2     | P2        | 16/02/2024 | RM873.33 | 10%                | RM786.00    | Unpaid        |
| 3 | B3     | P1        | 2024-08-24 | RM60.00  | 0.00%              | RM60.00     | Paid          |

### 6.6.19 Exit

Select Exit to exit the program.

## 7. Example Coding

### 7.1 Patient Menu

#### 7.1.1 Make appointments

##### 7.1.1.1 Structure

This structure defines a PatientAppointment node, containing details such as appointment ID, patient ID, service ID, date, time, status, and doctor ID. It also includes a pointer (next) that links to the next appointment in the list, facilitating the management of a series of patient appointments in a linked list format.

```
// structure
struct PatientAppointment {
    string AppointmentID;
    string PatientID;
    string ServiceID;
    string Date;
    string Time;
    string Status;
    string DoctorID;
    PatientAppointment* next;
};
```

### 7.1.1.2 Print

This function printPatientAppointment prints a table of patient appointments. It first outputs a header with columns for Appointment ID, Patient ID, Service ID, Date, Time, Status, and Doctor ID. Then, it iterates through the linked list, printing each appointment's details in the corresponding columns.

```
PatientAppointment* PatientAppointmenthead = NULL;

//print table
void printPatientAppointment(PatientAppointment* n) {
    // Print table header
    cout << setw(5) << left << "Appointment ID" << " | "
        << setw(5) << left << "Patient ID" << " | "
        << setw(5) << left << "Service ID" << " | "
        << setw(5) << left << "Date" << " | "
        << setw(5) << left << "Time" << " | "
        << setw(5) << left << "Status" << " | "
        << setw(5) << left << "DoctorID" << endl;
    //call data
    while (n != NULL) {
        cout << setw(5) << left << n->AppointmentID << " | "
            << setw(5) << left << n->PatientID << " | "
            << setw(5) << left << n->ServiceID << " | "
            << setw(5) << left << n->Date << " | "
            << setw(5) << left << n->Time << " | "
            << setw(5) << left << n->Status << " | "
            << setw(5) << left << n->DoctorID << endl;
        n = n->next;
    }
    cout << endl;
}
```

### 7.1.1.3 Push back

This function PushBackPatientAppointment adds a new PatientAppointment node to the end of a linked list. It first creates and initializes a new PatientAppointment node with the provided details and sets its next pointer to NULL. If the list (PatientAppointmenthead) is empty, it sets the head of the list to the new node. Otherwise, it traverses the list to find the last node and appends the new node to it.

```
//push back function
void PushBackPatientAppointment(
    string appointmentID,
    string patientID,
    string serviceID,
    string date,
    string time,
    string status,
    string doctorID) {
    PatientAppointment* newPatientAppointment = new PatientAppointment();
    newPatientAppointment->AppointmentID = appointmentID;
    newPatientAppointment->PatientID = patientID;
    newPatientAppointment->ServiceID = serviceID;
    newPatientAppointment->Date = date;
    newPatientAppointment->Time = time;
    newPatientAppointment->Status = status;
    newPatientAppointment->DoctorID = doctorID;
    newPatientAppointment->next = NULL;

    if (PatientAppointmenthead == NULL) {
        PatientAppointmenthead = newPatientAppointment;
    } else {
        PatientAppointment* temp = PatientAppointmenthead;
        while (temp->next != NULL)
            temp = temp->next;

        temp->next = newPatientAppointment;
    }
}
```

#### 7.1.1.4 Create

The `patientCreatePatientsAppointment` function allows users to schedule a new appointment. It starts by generating a unique `AppointmentID` and then prompts the user to enter required details including Service ID, Date, Time, and Doctor ID. It verifies the validity of these inputs against predefined sets of valid IDs. Once all information is validated, it inserts the appointment into the database with the status set as "Accessed". The function continues to prompt the user if they wish to add more appointments until they choose to stop.

```
// Function to add new data
void patientCreatePatientsAppointment(const string& UserID, const set<string>& validServiceIDs, const set<string>& validDoctorIDs) {
    string addMore;
    do {
        string PatientID, ServiceID, Date, Time, DoctorID, Status;

        cout << "\nAdd New PatientsAppointment" << endl;
        cout << "*****" << endl;

        // Automatically generate PatientsAppointment ID
        string AppointmentID = getNextAppointmentID();

        //cout << "Enter Patient ID: ";
        //cin >> ws, PatientID;
        //adminReadService();
        adminReadService();
        bool validServiceID = false;
        do {
            cout << "Enter Service ID ( valid options are: ";
            for (const auto &id : validServiceIDs) {
                cout << id << " ";
            }
            cout << "): ";
            getline(cin >> ws, ServiceID);

            // Check if the input is numeric and prefix with 'E' if true
            if (isNumeric(ServiceID) && !ServiceID.empty()) {
                ServiceID = "S" + ServiceID;
            }

            ServiceID = toUpperCase(ServiceID);

            // Direct validation check
            validServiceID = validServiceIDs.find(ServiceID) != validServiceIDs.end();

            if (!validServiceID) {
                cout << "Invalid Service ID. Please enter a valid Service ID from the list." << endl;
                cout << "Valid options are: ";
                for (const auto &id : validServiceIDs) {
                    cout << id << " ";
                }
                cout << ":" ;
            }
        } while (!validServiceID);

        if (ServiceID.find('S') != 0) {
            ServiceID = 'S' + ServiceID;
        }
    } while (addMore == "Y");
}
```

```
//cin.ignore();
do {
    cout << "Enter Date(YYYY-MM-DD): ";
    getline(cin, Date);
    if (!isValidDatePA(Date)) {
        cout << "Invalid date format. Please enter a valid date (YYYY-MM-DD)." << endl;
    }
} while (!isValidDatePA(Date));

do {
    cout << "Enter Time(HH:MM): ";
    getline(cin, Time);
    if (!isValidTime(Time)) {
        cout << "Invalid date format. Please enter a valid date (HH:MM)." << endl;
    }
} while (!isValidTime(Time));
cout << endl;

adminReadDoctorSchedule();

bool validDoctorID = false;
do {
    cout << "Enter Doctor ID ( valid options are: ";
    for (const auto &id : validDoctorIDs) {
        cout << id << " ";
    }
    cout << "): ";

    getline(cin >> ws, DoctorID);

    // Check if the input is numeric and prefix with 'E' if true
    if (isNumeric(DoctorID) && !DoctorID.empty()) {
        DoctorID = "E" + DoctorID;
    }

    DoctorID = toUpperCase(DoctorID);

    // Direct validation check
    validDoctorID = validDoctorIDs.find(DoctorID) != validDoctorIDs.end();

    if (!validDoctorID) {
        cout << "Invalid Doctor ID. Please enter a valid Doctor ID from the list." << endl;
        cout << "Valid options are: ";
        for (const auto &id : validDoctorIDs) {
            cout << id << " ";
        }
        cout << ":";
```

```
        }

    } while (!validDoctorID);

    // Set the Status as "Accessed" for new appointments
    Status = "Accessed";

    // Insert the new PatientsAppointment into the database
    string sql = "INSERT INTO PatientsAppointment (AppointmentID, PatientID, "
                 "ServiceID, Date, Time, Status, DoctorID) VALUES ('" +
                 AppointmentID + "'", "'"+ UserID + "'", "'"+ ServiceID + "'", "'"+
                 Date + "'", "'"+ Time + "'", "'"+ Status + "'", "'"+ DoctorID + "'");

    char *messageError;
    int exit = sqlite3_exec(DB, sql.c_str(), NULL, 0, &messageError);
    if (exit != SQLITE_OK) {
        cerr << "Error inserting data: " << messageError << endl;
        sqlite3_free(messageError);
    } else {
        cout << "PatientsAppointment added successfully!" << endl;
    }

    cout << "Do you want to add more appointments? (Y/N): ";
    getline(cin >> ws, addMore);

} while (addMore == "Y" || addMore == "y");
}
```

### 7.1.1.5 Read

The `patientReadAppointment` function retrieves and displays all appointments for a specified patient. It executes a SQL query to select all records from the `PatientsAppointment` table where the `PatientID` matches the provided parameter. If the query execution fails, it prints an error message. The `callback` function, defined elsewhere, processes and formats the results.

```
void patientReadAppointment(const string& PatientID) {
    string query = "SELECT * FROM PatientsAppointment WHERE PatientID = '" + PatientID + "' ";
    char* errMsg;
    if (sqlite3_exec(DB, query.c_str(), callback, nullptr, &errMsg) != SQLITE_OK) {
        cerr << "Failed to read PatientsAppointment: " << errMsg << endl;
        sqlite3_free(errMsg);
    }
}
```

## 7.1.2 Cancel Appointment

### 7.1.2.1 Structure

The `CancelAppointment` structure tracks cancelled appointments, storing details like `CancelAppointmentID`, `AppointmentID`, `DoctorID`, and `Reason`. It uses a `next` pointer to link each record in a dynamic list, enabling efficient management and retrieval of cancellation information.

```
// structure
struct CancelAppointment {
    string CancelAppointmentID;
    string AppointmentID;
    string DoctorID;
    string Reason;
    CancelAppointment *next;
};
```

### 7.1.2.2 Print

The `printCancelAppointment` function displays the list of canceled appointments in a formatted table. It begins by printing the column headers, then iterates through the linked list of `CancelAppointment` records, outputting each record's details. The `setw` function ensures that the columns are properly aligned for readability.

```
CancelAppointment *CancelAppointmenthead = NULL;

// print table
void printCancelAppointment(CancelAppointment *n) {
    // Print table header
    cout << setw(5) << left << "Cancel Appointment ID"
        << " | " << setw(5) << left << "Appointment ID"
        << " | " << setw(5) << left << "Doctor ID"
        << " | " << setw(5) << left << "Reason" << endl;
    // call data
    while (n != NULL) {
        cout << setw(5) << left << n->CancelAppointmentID << " | "
            << setw(5) << left << n->AppointmentID << " | "
            << setw(5) << left << n->DoctorID << " | "
            << setw(5) << left << n->Reason << endl;
        n = n->next;
    }
    cout << endl;
}
```

### 7.1.2.3 Push back

The `PushBackCancelAppointment` function adds a new `CancelAppointment` record to the end of the linked list. It creates a new `CancelAppointment` instance, initializes its fields with the provided values, and appends it to the list. If the list is empty, the new record becomes the head of the list; otherwise, it is added after the last existing record.

```
// push back function
void PushBackCancelAppointment(
    string cancelAppointmentID,
    string appointmentID,
    string doctorID,
    string reason) {
    CancelAppointment *newCancelAppointment = new CancelAppointment();
    newCancelAppointment->CancelAppointmentID = cancelAppointmentID;
    newCancelAppointment->AppointmentID = appointmentID;
    newCancelAppointment->DoctorID = doctorID;
    newCancelAppointment->Reason = reason;
    newCancelAppointment->next = NULL;

    if (CancelAppointmenthead == NULL) {
        CancelAppointmenthead = newCancelAppointment;
    } else {
        CancelAppointment *temp = CancelAppointmenthead;
        while (temp->next != NULL)
            temp = temp->next;

        temp->next = newCancelAppointment;
    }
}
```

#### 7.1.2.4 Create

The `patientCreateCancelAppointment` function allows adding a new cancellation appointment. It generates a unique `CancelAppointmentID` and validates the `AppointmentID` against a set of valid IDs. It ensures that the `AppointmentID` hasn't been previously canceled. The function retrieves the corresponding `DoctorID`, prompts for a cancellation reason, and inserts the new record into the `CancelAppointment` table. Additionally, it updates the status of the related appointment to "Cancelled" in the `PatientsAppointment` table, handling any errors encountered during database operations.

```
void patientCreateCancelAppointment(const set<string>& validAppointmentIDs, const string& PatientID)
{
    string addMore;
    do {
        string AppointmentID, DoctorID, Reason;

        cout << "\nAdd New Cancel Appointment" << endl;
        cout << "*****" << endl;

        string CancelAppointmentID = getNextCancelAppointmentID();

        // DEBUG: Print CancelAppointmentID to confirm it's generated
        cout << "Generated CancelAppointmentID: " << CancelAppointmentID << endl;

        if (CancelAppointmentID.empty()) {
            cerr << "Failed to generate CancelAppointmentID. Please try again." << endl;
            return;
        }

        bool validAppointmentID = false; // Fixed variable name spelling
        do {
            cout << "Enter Appointment ID (valid options are: ";
            for (const auto& id : validAppointmentIDs) {
                cout << id << " ";
            }
            cout << "): ";

            getline(cin >> ws, AppointmentID); // Read input with leading whitespace stripped

            // Check if the input is numeric and prefix with 'A' if true
            if (isNumeric(AppointmentID) && !AppointmentID.empty()) {
                AppointmentID = "A" + AppointmentID;
            }

            AppointmentID = toUpperCase(AppointmentID);

            // Direct validation check
            validAppointmentID = validAppointmentIDs.find(AppointmentID) != validAppointmentIDs.end();

            if (!validAppointmentID) {
                cout << "Invalid Appointment ID. Please enter a valid Appointment ID from the list." << endl;
                cout << "Valid options are: ";
                for (const auto& id : validAppointmentIDs) {
                    cout << id << " ";
                }
                cout << ":" ;
            }
        } while (!validAppointmentID);

        // Check if the AppointmentID already exists in the CancelAppointment table
        string sqlCheck =
            "SELECT COUNT(*) FROM CancelAppointment WHERE AppointmentID = '" +
            AppointmentID + "'";
        sqlite3_stmt *stmt;
        int result = sqlite3_prepare_v2(DB, sqlCheck.c_str(), -1, &stmt, NULL);

        if (result != SQLITE_OK) {
            cerr << "Failed to prepare statement: " << sqlite3_errmsg(DB) << endl;
            return;
        }
    }
```

```
result = sqlite3_step(stmt);

if (result == SQLITE_ROW && sqlite3_column_int(stmt, 0) > 0) {
    cout << "This Appointment ID has already been cancelled. Update process "
        "exited."
        << endl;
    sqlite3_finalize(stmt);
    return;
}

sqlite3_finalize(stmt);

DoctorID = get("Enter Doctor ID: ");
inline bool std::string::empty() const noexcept
if (DoctorID.empty()) {
    cerr << "No matching DoctorID found for AppointmentID: " << AppointmentID << endl;
    continue; // Skip to next iteration if DoctorID is not found
}

cout << "Enter Cancel Reason: ";
getline(cin >> ws, Reason);

// Insert the new CancelAppointment into the database
string sqlInsert = "INSERT INTO CancelAppointment (CancelAppointmentID, AppointmentID, DoctorID, Reason) VALUES ('" +
    CancelAppointmentID + "', '" + AppointmentID + "', '" + DoctorID + "', '" + Reason + "')";

char *messageError;
int exit = sqlite3_exec(DB, sqlInsert.c_str(), NULL, 0, &messageError);
if (exit != SQLITE_OK) {
    cerr << "Error inserting data: " << messageError << endl;
    sqlite3_free(messageError);
} else {
    cout << "CancelAppointment added successfully!" << endl;

    // Update Status in PatientsAppointment table
    string sqlUpdate = "UPDATE PatientsAppointment SET Status = 'Cancelled' WHERE AppointmentID = '" +
        AppointmentID + "'";
    exit = sqlite3_exec(DB, sqlUpdate.c_str(), NULL, 0, &messageError);
    if (exit != SQLITE_OK) {
        cerr << "Error updating PatientsAppointment status: " << messageError
            << endl;
        sqlite3_free(messageError);
    } else {
        cout << "PatientsAppointment status updated to 'Cancelled'!" << endl;
    }
}
```

```
// Optionally read and print all CancelAppointments to confirm the addition
patientReadCancelAppointment(PatientID);

cout << "Do you want to add another Cancel Appointment? (Y/N): ";
getline(cin >> ws, addMore);

} while (addMore == "Y" || addMore == "y");
}
```

### 7.1.2.5 Read

The `patientReadCancelAppointment` function fetches and displays cancel appointment details for a given patient. It uses a parameterized SQL query to join `CancelAppointment` and `PatientsAppointment` tables, retrieves relevant records based on `PatientID`, and prints the cancel appointment ID, appointment ID, doctor ID, and reason. It handles errors and ensures resources are properly released.

```
void patientReadCancelAppointment(const string& PatientID) {
    // Define the SQL query with a placeholder for PatientID
    string query = R"(

        SELECT CA.CancelAppointmentID, CA.AppointmentID, CA.DoctorID, CA.Reason
        FROM CancelAppointment CA
        JOIN PatientsAppointment PA ON CA.AppointmentID = PA.AppointmentID
        WHERE PA.PatientID = ?

    )";

    // Prepare the SQL statement
    sqlite3_stmt* stmt;
    int result = sqlite3_prepare_v2(DB, query.c_str(), -1, &stmt, NULL);

    if (result != SQLITE_OK) {
        cerr << "Failed to prepare statement: " << sqlite3_errmsg(DB) << endl;
        return;
    }

    // Bind the PatientID parameter to the SQL query
    result = sqlite3_bind_text(stmt, 1, PatientID.c_str(), -1, SQLITE_STATIC);

    if (result != SQLITE_OK) {
        cerr << "Failed to bind parameter: " << sqlite3_errmsg(DB) << endl;
        sqlite3_finalize(stmt);
        return;
    }

    // Execute the query and handle results
    while ((result = sqlite3_step(stmt)) == SQLITE_ROW) {
        // Extract data from the current row
        string cancelAppointmentID = reinterpret_cast<const char*>(sqlite3_column_text(stmt, 0));
        string appointmentID = reinterpret_cast<const char*>(sqlite3_column_text(stmt, 1));
        string doctorID = reinterpret_cast<const char*>(sqlite3_column_text(stmt, 2));
        string reason = reinterpret_cast<const char*>(sqlite3_column_text(stmt, 3));

        // Display or process the data
        cout << "Cancel Appointment ID: " << cancelAppointmentID << endl;
        cout << "Appointment ID: " << appointmentID << endl;
        cout << "Doctor ID: " << doctorID << endl;
        cout << "Reason: " << reason << endl << endl;
    }

    if (result != SQLITE_DONE) {
        cerr << "Failed to fetch results: " << sqlite3_errmsg(DB) << endl;
    }

    // Finalize the statement to release resources
    sqlite3_finalize(stmt);
}
```

### 7.1.2.6 Update

The `patientUpdateCancelAppointment` function updates existing cancellation records by allowing the user to modify the `AppointmentID` and `Reason` fields. It first checks if the provided `CancelAppointmentID` exists in the database. If valid, it prompts the user to enter new values for the specified fields, leaving fields empty if they don't need updating. The function then constructs an SQL `UPDATE` statement based on the user's input and executes it to update the database. Finally, it displays the updated records to confirm the changes.

```
void patientUpdateCancelAppointment(const set<string>& validAppointmentIDs, const set<string>& validDoctorIDs) {
    string CancelAppointmentID, PatientID, AppointmentID, DoctorID;

    cout << "Enter CancelAppointment ID to update: ";
    getline(cin, CancelAppointmentID);
    //cin.ignore();

    if (CancelAppointmentID.empty()) {
        cout << "Update process exited.\n";
        return;
    }

    if (CancelAppointmentID.find("CA") != 0) {
        CancelAppointmentID = "CA" + CancelAppointmentID;
    }

    // Check if CancelAppointmentID exists
    string checkSql = "SELECT COUNT(*) FROM CancelAppointment WHERE CancelAppointmentID = '" + CancelAppointmentID + "'";
    sqlite3_stmt *stmt;
    int result = sqlite3_prepare_v2(DB, checkSql.c_str(), -1, &stmt, NULL);

    if (result != SQLITE_OK) {
        cerr << "Error checking CancelAppointmentID: " << sqlite3_errmsg(DB)
            << endl;
        return;
    }

    sqlite3_bind_text(stmt, 1, CancelAppointmentID.c_str(), -1, SQLITE_TRANSIENT);
    result = sqlite3_step(stmt);
    if (result == SQLITE_ROW) {
        int count = sqlite3_column_int(stmt, 0);
        if (count == 0) {
            cout << "No relevant search results found." << endl;
            sqlite3_finalize(stmt);
            return;
        }
    }
    sqlite3_finalize(stmt);

    string fields[] = {"AppointmentID", "Reason"};
    string updates = "";

    cout << "Enter new values for the following fields. Leave empty to skip updating the field."
        << endl;

    for (const string &field : fields) {
        string newValue;
        bool validInput = false;

        do {
            cout << field << ": ";
            getline(cin, newValue);

            if (field == "AppointmentID" && !newValue.empty()) {
                if (!isDigit(newValue[0])) {
                    newValue = "A" + newValue;
                }
                newValue = toUpperCase(newValue);
            }

            if (validAppointmentIDs.find(newValue) != validAppointmentIDs.end()) {
                AppointmentID = newValue;
                validInput = true;
            }
        } while (!validInput);
    }
}
```

```
    }
    cout << "Invalid AppointmentID. Please enter a valid AppointmentID." << endl;
}
} else if (field == "Reason" && !newValue.empty()) {
    validInput = true;
} else {
    validInput = true; // Skip empty fields without validation
}
} while (!validInput);

if (!newValue.empty()) {
    if (!updates.empty()) {
        updates += ", ";
    }
    updates += field + " = '" + newValue + "'";
}
}

if (updates.empty()) {
    cout << "No fields were updated." << endl;
    return;
}

string sql = "UPDATE CancelAppointment SET " + updates + " WHERE CancelAppointmentID = '" + CancelAppointmentID + "'";
char *messageError;
int exit = sqlite3_exec(DB, sql.c_str(), NULL, 0, &messageError);

if (exit != SQLITE_OK) {
    cerr << "Error Update: " << messageError << endl;
    sqlite3_free(messageError);
} else {
    cout << "Data updated successfully!\n" << endl;
}

patientReadCancelAppointment(PatientID);
}

#endif // ends this file conditional inclusion (must in the last line)
```

## 7.2 Doctor Menu

### 7.2.1 Structure

This structure defines a medicalDiagnosis node, containing details like patient ID, doctor ID, medical history, medications, diagnosis, last update, and a pointer (next) that links to the next diagnosis in the list.

```
▽ struct medicalDiagnosis {  
    string patientDiagnosisID;  
    string patientID;  
    string doctorID;  
    string medicalHistory;  
    string medicationsTaken;  
    string diagnosis;  
    string lastUpdate;  
    medicalDiagnosis *next;  
};
```

### 7.2.2 Node

This function creates a new medicalDiagnosis node by allocating memory for it and initializing it with the provided diagnosis details.

```
medicalDiagnosis *createMedicalDiagnosisNode(string patientDiagnosisID, string patientID, string doctorID, string medicalHistory, string medicationsTaken, string diagnosis, string lastUpdate) {  
    medicalDiagnosis *newNode = new medicalDiagnosis;  
    newNode->patientDiagnosisID = patientDiagnosisID;  
    newNode->patientID = patientID;  
    newNode->doctorID = doctorID;  
    newNode->medicalHistory = medicalHistory;  
    newNode->medicationsTaken = medicationsTaken;  
    newNode->diagnosis = diagnosis;  
    newNode->lastUpdate = lastUpdate;  
  
    return newNode;  
}
```

### 7.2.3 Push Back

This function appends a newly created medicalDiagnosis node to the end of the linked list. If the list is empty, the new node becomes the head; otherwise, it's added as the last node (tail).

```
void push_back(medicalDiagnosis *&head, medicalDiagnosis *&tail, const string &patientDiagnosisID, const string &patientID, const string &doctorID, const string &medicalHistory,  
               const string &medicationsTaken, const string &diagnosisText, const string &lastUpdate) {  
    medicalDiagnosis *medicalDiagnosisNode = new medicalDiagnosis();  
    medicalDiagnosisNode->patientDiagnosisID = patientDiagnosisID;  
    medicalDiagnosisNode->patientID = patientID;  
    medicalDiagnosisNode->doctorID = doctorID;  
    medicalDiagnosisNode->medicalHistory = medicalHistory;  
    medicalDiagnosisNode->medicationsTaken = medicationsTaken;  
    medicalDiagnosisNode->diagnosis = diagnosisText;  
    medicalDiagnosisNode->lastUpdate = lastUpdate;  
    medicalDiagnosisNode->next = nullptr;  
  
    if (tail == nullptr) {  
        head = tail = medicalDiagnosisNode;  
    } else {  
        tail->next = medicalDiagnosisNode;  
        tail = medicalDiagnosisNode;  
    }  
}
```

## 7.2.4 Generates Unique ID

This function generates a unique PatientDiagnosisID by querying the database for the current highest ID and incrementing it. It ensures that each diagnosis has a distinct identifier.

```
string generateDiagnosisID(sqlite3 *DB) {
    sqlite3_stmt* stmt;
    string sql = "SELECT MAX(CAST(SUBSTR(PatientDiagnosisID, 3) AS INTEGER)) FROM MedicalDiagnosis";
    int exit = sqlite3_prepare_v2(DB, sql.c_str(), -1, &stmt, NULL);

    if (exit != SQLITE_OK) {
        cerr << "Failed to prepare statement for ID generation: " << sqlite3_errmsg(DB) << endl;
        return "";
    }

    int maxID = 0;
    exit = sqlite3_step(stmt);
    if (exit == SQLITE_ROW) {
        maxID = sqlite3_column_int(stmt, 0);
    }

    sqlite3_finalize(stmt);

    maxID++;

    return "PH" + to_string(maxID);
}
```

## 7.2.5 Create Medical Diagnosis

This function inserts a new medical diagnosis into the database if the patient ID and doctor ID match an existing appointment in the PatientsAppointment table. It binds the diagnosis details to the SQL query parameters and executes the insertion.

```
void createMedicalDiagnosis(const string &patientDiagnosisID, const string &patientID, const string &doctorID, const string &medicalHistory, const string &medicationsTaken,
                           const string &diagnosis, const string &lastUpdate) {
    sqlite3_stmt* stmt;

    // SQL query to insert a new diagnosis, but only if the doctorID matches in PatientsAppointment
    string sql = "INSERT INTO MedicalDiagnosis (PatientDiagnosisID, PatientID, DoctorID, MedicalHistory, MedicationsTaken, Diagnosis, LastUpdate) "
                "SELECT ?, ?, ?, ?, ?, ?"
                "WHERE EXISTS (SELECT 1 FROM PatientsAppointment WHERE PatientID = ? AND DoctorID = ?);";

    DBExit = sqlite3_prepare_v2(DB, sql.c_str(), -1, &stmt, NULL);

    if (DBExit != SQLITE_OK) {
        cerr << "Failed to prepare statement: " << sqlite3_errmsg(DB) << endl;
        return;
    }

    // Bind parameters
    sqlite3_bind_text(stmt, 1, patientDiagnosisID.c_str(), -1, SQLITE_TRANSIENT);
    sqlite3_bind_text(stmt, 2, patientID.c_str(), -1, SQLITE_TRANSIENT);
    sqlite3_bind_text(stmt, 3, doctorID.c_str(), -1, SQLITE_TRANSIENT);
    sqlite3_bind_text(stmt, 4, medicalHistory.c_str(), -1, SQLITE_TRANSIENT);
    sqlite3_bind_text(stmt, 5, medicationsTaken.c_str(), -1, SQLITE_TRANSIENT);
    sqlite3_bind_text(stmt, 6, diagnosis.c_str(), -1, SQLITE_TRANSIENT);
    sqlite3_bind_text(stmt, 7, lastUpdate.c_str(), -1, SQLITE_TRANSIENT);
    sqlite3_bind_text(stmt, 8, patientID.c_str(), -1, SQLITE_TRANSIENT);
    sqlite3_bind_text(stmt, 9, doctorID.c_str(), -1, SQLITE_TRANSIENT);

    // Execute the query
    DBExit = sqlite3_step(stmt);
    if (DBExit != SQLITE_DONE) {
        cerr << "Failed to execute statement: " << sqlite3_errmsg(DB) << endl;
    } else {
        cout << "Medical diagnosis '" << patientDiagnosisID << "' added successfully." << endl;
    }

    // Finalize the statement
    sqlite3_finalize(stmt);
}
```

This function interacts with the user (doctor) to add a new diagnosis. It validates that the patient ID belongs to a patient assigned to the doctor, collects the diagnosis details, and then uses the push\_back function to add the diagnosis to the linked list and the createMedicalDiagnosis function to store it in the database. The process is repeated if the doctor wants to add more diagnoses.

```
void addDiagnosis(medicalDiagnosis *&head, medicalDiagnosis *&tail, user *currentUser, sqlite3* DB) {
    if (currentUser == nullptr) {
        cerr << "Error: currentUser is null" << endl;
        return;
    }

    // Set to store valid patient IDs for the current doctor
    set<string> validPatientIDs;

    // SQL query to get the Patient IDs for the current doctor
    string sql = "SELECT PatientID FROM PatientsAppointment WHERE DoctorID = ?;";
    sqlite3_stmt* stmt;

    int DBExit = sqlite3_prepare_v2(DB, sql.c_str(), -1, &stmt, NULL);
    if (DBExit != SQLITE_OK) {
        cerr << "Failed to prepare statement: " << sqlite3_errmsg(DB) << endl;
        return;
    }

    // Bind the current doctor's ID
    sqlite3_bind_text(stmt, 1, currentUser->userID.c_str(), -1, SQLITE_TRANSIENT);

    // Execute the query and populate the set with patient IDs
    while ((DBExit = sqlite3_step(stmt)) == SQLITE_ROW) {
        validPatientIDs.insert(reinterpret_cast<const char*>(sqlite3_column_text(stmt, 0)));
    }

    sqlite3_finalize(stmt);

    // Ensure that the doctor has valid patients
    if (validPatientIDs.empty()) {
        cout << "No patients assigned to this doctor." << endl;
        return;
    }
}
```

```
string addMore;
do {
    string patientID, medicalHistory, medicationsTaken, diagnosisText, lastUpdate;

    cout << "\nAdd Patient Medical Diagnosis" << endl;
    cout << "*****" << endl;

    // Automatically generate Patient Diagnosis ID
    string patientDiagnosisID = generateDiagnosisID(DB);

    bool validPatientID = false;
    do {
        cout << "Enter new Patient ID ( valid options are: ";
        for (const auto &id : validPatientIDs) {
            cout << id << " ";
        }
        cout << "): ";

        getline(cin >> ws, patientID);

        // Check if the input is numeric and prefix with 'E' if true
        if (isNumeric(patientID)) {
            patientID = "P" + patientID;
        }

        patientID = toUpperCase(patientID);

        // Direct validation check
        validPatientID = validPatientIDs.find(patientID) != validPatientIDs.end();

        if (!validPatientID) {
            cout << "Invalid patient ID. Please enter a valid patient ID from the list." << endl;
            cout << "Valid options are: ";
            for (const auto &id : validPatientIDs) {
                cout << id << " ";
            }
            cout << endl;
        }
    } while (!validPatientID);

    string doctorID = currentUser->userID; // Set doctorID from currentUser

    cout << "Enter Medical History: ";
    getline(cin >> ws, medicalHistory);

    cout << "Enter Medications Taken: ";
    getline(cin >> ws, medicationsTaken);
```

```

cout << "Enter Diagnosis: ";
getline(cin >> ws, diagnosisText);

do {
    cout << "Enter Last Update (YYYY-MM-DD): ";
    getline(cin >> ws, lastUpdate);
    if (!isValidDate(lastUpdate)) {
        cout << "Invalid date format. Please enter a valid date (YYYY-MM-DD)." << endl;
    }
} while (!isValidDate(lastUpdate));

push_back(head, tail, patientDiagnosisID, patientID, doctorID, medicalHistory, medicationsTaken, diagnosisText, lastUpdate);

createMedicalDiagnosis(patientDiagnosisID, patientID, doctorID, medicalHistory, medicationsTaken, diagnosisText, lastUpdate);

cout << "Do you want to add another medical diagnosis? (Y/N): ";
getline(cin >> ws, addMore);

} while (addMore == "Y" || addMore == "y");

```

## 7.2.6 Read Medical Diagnosis

The `readDiagnosis` function fetches and displays medical diagnosis records from the database for a specific patient. It executes a query to retrieve and format details such as diagnosis ID, patient ID, doctor ID, and medical history.

```

void readDiagnosis(const string &patientDiagnosisID){
    sqlite3_stmt* stmt;

    string sql = "SELECT * FROM medicalDiagnosis WHERE DoctorID = ?";
    DBExit = sqlite3_prepare_v2(DB, sql.c_str(), -1, &stmt, NULL);

    if (DBExit != SQLITE_OK) {
        cerr << "Failed to prepare statement: " << sqlite3_errmsg(DB) << endl;
        return;
    }

    sqlite3_bind_text(stmt, 1, patientDiagnosisID.c_str(), -1, SQLITE_TRANSIENT);

    if (DBExit != SQLITE_OK) {
        cerr << "Failed to bind parameter: " << sqlite3_errmsg(DB) << endl;
        sqlite3_finalize(stmt);
        return;
    }

    cout << "-----" << endl;
    cout << setw(18) << "Patient Diagnosis ID" << " | " << setw(10) << "Patient ID" << " | " << setw(10) << "Doctor ID" << " | " << setw(15) << "Medical History"
    | << " | " << setw(32) << "Medications Taken" << setw(16) << " | " << setw(16) << "Diagnosis" << setw(10) << " | " << setw(10) << "Last Update" << endl;
    cout << "-----" << endl;

    while ((DBExit = sqlite3_step(stmt)) == SQLITE_ROW) {
        cout << setw(11) << (const char*)sqlite3_column_text(stmt, 0)
            << setw(12) << " | " << setw(6) << (const char*)sqlite3_column_text(stmt, 1)
            << setw(7) << " | " << setw(6) << (const char*)sqlite3_column_text(stmt, 2)
            << setw(7) << " | " << setw(15) << (const char*)sqlite3_column_text(stmt, 3)
            << " | " << setw(45) << (const char*)sqlite3_column_text(stmt, 4)
            << " | " << setw(23) << (const char*)sqlite3_column_text(stmt, 5)
            << " | " << setw(10) << (const char*)sqlite3_column_text(stmt, 6) << endl;
    }

    if (DBExit != SQLITE_DONE) {
        cerr << "Execution failed: " << sqlite3_errmsg(DB) << endl;
    }

    sqlite3_finalize(stmt);
}

```

The `printMedicalDiagnoses` function calls `readDiagnosis` to display medical diagnoses based on the current user's ID. It passes the user's ID to `readDiagnosis` to fetch and print the relevant records.

```

void printMedicalDiagnoses(medicalDiagnosis *medicalDiagnosisHead, user *currentUser) {
    readDiagnosis(currentUser->userID);
}

```

## 7.2.7 Update Medical Diagnosis

The `updateDiagnosis` function updates a medical diagnosis record in the SQLite database by preparing and executing an `UPDATE` SQL statement with new values for various fields. It binds the new values to the statement, executes it, and handles any errors. If successful, it confirms the update.

```
void updateDiagnosis(const string& patientDiagnosisID, const string& patientID, const string& doctorID, const string& medicalHistory, const string& medicationsTaken,
                     const string& diagnosis, const string& lastUpdate, sqlite3* DB){
    sqlite3_stmt* stmt;

    string sql = "UPDATE medicalDiagnosis SET PatientID = ?, DoctorID = ?, MedicalHistory = ?, MedicationsTaken = ?, Diagnosis = ?, LastUpdate = ? WHERE PatientDiagnosisID = ?";

    DBExit = sqlite3_prepare_v2(DB, sql.c_str(), -1, &stmt, NULL);
    if (DBExit != SQLITE_OK) {
        cerr << "Failed to prepare statement: " << sqlite3_errmsg(DB) << endl;
        return;
    }

    //medicalDiagnosis *current = NULL;
    sqlite3_bind_text(stmt, 1, patientID.c_str(), -1, SQLITE_TRANSIENT);
    sqlite3_bind_text(stmt, 2, doctorID.c_str(), -1, SQLITE_TRANSIENT);
    sqlite3_bind_text(stmt, 3, medicalHistory.c_str(), -1, SQLITE_TRANSIENT);
    sqlite3_bind_text(stmt, 4, medicationsTaken.c_str(), -1, SQLITE_TRANSIENT);
    sqlite3_bind_text(stmt, 5, diagnosis.c_str(), -1, SQLITE_TRANSIENT);
    sqlite3_bind_text(stmt, 6, lastUpdate.c_str(), -1, SQLITE_TRANSIENT);
    sqlite3_bind_text(stmt, 7, patientDiagnosisID.c_str(), -1, SQLITE_TRANSIENT);

    DBExit = sqlite3_step(stmt);
    if (DBExit != SQLITE_DONE) {
        cerr << "Execution failed: " << sqlite3_errmsg(DB) << endl;
    } else {
        cout << "Updated diagnosis with ID '" << patientDiagnosisID << "' successfully." << endl;
    }
    sqlite3_finalize(stmt);
}
```

The `updateMedicalDiagnosis` function updates a diagnosis record for a logged-in doctor. It retrieves the list of patients assigned to the doctor, checks if the given diagnosis ID exists and belongs to those patients, and then prompts for new details. It validates these details before updating the record in the database. If the diagnosis ID is not found, it informs the user.

```
void updateMedicalDiagnosis(const string &loggedInDoctorID, const set<string> &validDoctorIDs, sqlite3 *DB) {
    // Set to store valid patient IDs for the current doctor
    set<string> validPatientIDs;

    // SQL query to get the Patient IDs for the current doctor
    string sql = "SELECT PatientID FROM PatientsAppointment WHERE DoctorID = ?;";
    sqlite3_stmt* stmt;

    int DBExit = sqlite3_prepare_v2(DB, sql.c_str(), -1, &stmt, NULL);
    if (DBExit != SQLITE_OK) {
        cerr << "Failed to prepare statement: " << sqlite3_errmsg(DB) << endl;
        return;
    }

    // Bind the current doctor's ID
    sqlite3_bind_text(stmt, 1, loggedInDoctorID.c_str(), -1, SQLITE_TRANSIENT);

    // Execute the query and populate the set with patient IDs
    while ((DBExit = sqlite3_step(stmt)) == SQLITE_ROW) {
        validPatientIDs.insert(reinterpret_cast<const char*>(sqlite3_column_text(stmt, 0)));
    }

    sqlite3_finalize(stmt);

    // Ensure that the doctor has valid patients
    if (validPatientIDs.empty()) {
        cout << "No patients assigned to this doctor." << endl;
        return;
    }

    string patientDiagnosisID;
    bool found = true;

    do {
        cout << "Enter Patient Diagnosis ID to update (Blank for exit): ";
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        getline(cin, patientDiagnosisID);

        if (patientDiagnosisID.empty()) {
            return; // Exit the function to return to the menu
        }
    }
```

```

patientDiagnosisID = formatDiagnosisID(patientDiagnosisID);

// Prepare the SQL query to check if the ID exists
string selectSQL = "SELECT PatientID FROM medicalDiagnosis WHERE PatientDiagnosisID = ?";
sqlite3_stmt *stmt;
int exit = sqlite3_prepare_v2(DB, selectSQL.c_str(), -1, &stmt, NULL);

if (exit != SQLITE_OK) {
    cerr << "Failed to prepare statement: " << sqlite3_errmsg(DB) << endl;
    return;
}

// Bind the diagnosis ID
sqlite3_bind_text(stmt, 1, patientDiagnosisID.c_str(), -1, SQLITE_TRANSIENT);

// Execute the query
exit = sqlite3_step(stmt);
if (exit == SQLITE_ROW) {
    string patientID = reinterpret_cast<const char*>(sqlite3_column_text(stmt, 0));

    // Check if the patient belongs to the logged-in doctor
    if (validPatientIDs.find(patientID) != validPatientIDs.end()) {
        found = true;
    }
}

string newPatientID, newDoctorID, newMedicalHistory, newMedicationsTaken, newDiagnosis, newLastUpdate;

bool validPatientID = false;
do {
    cout << "Enter new Patient ID ( valid options are: ";
    for (const auto &id : validPatientIDs) {
        cout << id << " ";
    }
    cout << "): ";

    getline(cin >> ws, newPatientID);

    // Check if the input is numeric and prefix with 'E' if true
    if (isNumeric(newPatientID) && !newPatientID.empty()) {
        newPatientID = "P" + newPatientID;
    }

    newPatientID = toUpperCase(newPatientID);

    // Direct validation check
    validPatientID = validPatientIDs.find(newPatientID) != validPatientIDs.end();
}

```

```

    if (!validPatientID) {
        cout << "Invalid patient ID. Please enter a valid patient ID from the list." << endl;
        cout << "Valid options are: ";
        for (const auto &id : validPatientIDs) {
            cout << id << " ";
        }
        cout << endl;
    }
} while (!validPatientID);

bool validDoctorID = false;
do {
    cout << "Enter new Doctor ID ( valid options are: ";
    for (const auto &id : validDoctorIDs) {
        cout << id << " ";
    }
    cout << "): ";
    getline(cin >> ws, newDoctorID);

    // Check if the input is numeric and prefix with 'E' if true
    if (isNumeric(newDoctorID) && !newDoctorID.empty()) {
        newDoctorID = "E" + newDoctorID;
    }

    newDoctorID = toUpperCase(newDoctorID);

    // Direct validation check
    validDoctorID = validDoctorIDs.find(newDoctorID) != validDoctorIDs.end();

    if (!validDoctorID) {
        cout << "Invalid doctor ID. Please enter a valid patient ID from the list." << endl;
        cout << "Valid options are: ";
        for (const auto &id : validDoctorIDs) {
            cout << id << " ";
        }
        cout << endl;
    }
} while (!validDoctorID);

cout << "Enter new Medical History: ";
getline(cin, newMedicalHistory);

cout << "Enter new Medications Taken: ";
getline(cin, newMedicationsTaken);

```

```

cout << "Enter new Disease: ";
getline(cin, newDiagnosis);

do {
    cout << "Enter Last Update (YYYY-MM-DD): ";
    getline(cin >> ws, newLastUpdate);
    if (!isValidDate(newLastUpdate)) {
        cout << "Invalid date format. Please enter a valid date (YYYY-MM-DD)." << endl;
    }
} while (!isValidDate(newLastUpdate));

updateDiagnosis(patientDiagnosisID, newPatientID, newDoctorID, newMedicalHistory, newMedicationsTaken, newDiagnosis, newLastUpdate, DB);

} else {
    cout << "Diagnosis with ID " << patientDiagnosisID << " not found. No update performed." << endl;
}
} else {
    cout << "Diagnosis with ID " << patientDiagnosisID << " not found. No update performed." << endl;
}

sqlite3_finalize(stmt);
} while (!found);
}

```

## 7.2.8 Delete Medical Diagnosis

The `deleteDiagnosis` function removes a diagnosis record from the database if the logged-in doctor is authorized. It first checks if the diagnosis ID exists and retrieves the associated patient ID. It then verifies if the logged-in doctor is authorized to manage the patient's diagnosis. If authorized, it deletes the diagnosis record. If the diagnosis is not found or the doctor is not authorized, it informs the user.

```
void deleteDiagnosis(const string& loggedInDoctorID, const string& patientDiagnosisID, sqlite3* DB) {
    sqlite3_stmt* stmt;
    bool authorized = false;

    string selectSQL = "SELECT PatientID FROM medicalDiagnosis WHERE PatientDiagnosisID = ?";
    int DBExit = sqlite3_prepare_v2(DB, selectSQL.c_str(), -1, &stmt, NULL);

    if (DBExit != SQLITE_OK) {
        cerr << "Failed to prepare statement: " << sqlite3_errmsg(DB) << endl;
        return;
    }

    sqlite3_bind_text(stmt, 1, patientDiagnosisID.c_str(), -1, SQLITE_TRANSIENT);

    // Execute the query
    DBExit = sqlite3_step(stmt);
    if (DBExit == SQLITE_ROW) {
        string patientID = reinterpret_cast<const char*>(sqlite3_column_text(stmt, 0));
        sqlite3_finalize(stmt); // Finalize the statement before preparing a new one

        // Check if the patient belongs to the logged-in doctor
        string patientCheckSQL = "SELECT 1 FROM PatientsAppointment WHERE PatientID = ? AND DoctorID = ?";
        DBExit = sqlite3_prepare_v2(DB, patientCheckSQL.c_str(), -1, &stmt, NULL);

        if (DBExit != SQLITE_OK) {
            cerr << "Failed to prepare patient authorization statement: " << sqlite3_errmsg(DB) << endl;
            return;
        }

        // Bind patient ID and doctor ID
        sqlite3_bind_text(stmt, 1, patientID.c_str(), -1, SQLITE_TRANSIENT);
        sqlite3_bind_text(stmt, 2, loggedInDoctorID.c_str(), -1, SQLITE_TRANSIENT);

        // Execute the query
        DBExit = sqlite3_step(stmt);
        if (DBExit == SQLITE_ROW) {
            authorized = true; // Authorization successful
        } else {
            cout << "You are not authorized to delete this diagnosis." << endl;
        }
        sqlite3_finalize(stmt); // Finalize the statement before preparing a new one

        if (authorized) {
            // Proceed to delete the diagnosis
            string deleteSQL = "DELETE FROM medicalDiagnosis WHERE PatientDiagnosisID = ?";
            DBExit = sqlite3_prepare_v2(DB, deleteSQL.c_str(), -1, &stmt, NULL);
        }
    }
}
```

```
    if (DBExit != SQLITE_OK) {
        cerr << "Failed to prepare DELETE statement: " << sqlite3_errmsg(DB) << endl;
        return;
    }

    // Bind the diagnosis ID
    sqlite3_bind_text(stmt, 1, patientDiagnosisID.c_str(), -1, SQLITE_TRANSIENT);

    // Execute the delete query
    DBExit = sqlite3_step(stmt);
    if (DBExit != SQLITE_DONE) {
        cerr << "Failed to delete diagnosis: " << sqlite3_errmsg(DB) << endl;
    } else {
        cout << "Deleted diagnosis with ID '" << patientDiagnosisID << "' successfully." << endl;
    }
} else {
    cout << "Diagnosis with ID '" << patientDiagnosisID << "' not found. No deletion performed." << endl;
}

sqlite3_finalize(stmt); // Ensure statement is finalized at the end
}
```

This function allows a doctor to delete a diagnosis if they are authorized. It checks if the diagnosis exists, retrieves the associated patient ID, verifies if the doctor is authorized for that patient, and then performs the deletion. If any conditions are not met, it provides appropriate feedback.

```
void deleteMedicalDiagnosis(const string &loggedInDoctorID, const set<string> &validDoctorID) {
    string patientDiagnosisID;
    bool found = false;

    cout << "Enter Patient Diagnosis ID to delete (number only, will auto-prefix PH, or blank to exit): ";
    cin.ignore();
    getline(cin, patientDiagnosisID);

    if (patientDiagnosisID.empty()) {
        return; // Exit if the user input is blank
    }

    patientDiagnosisID = formatDiagnosisID(patientDiagnosisID);

    // Prepare SQL query to check if the ID exists and get the corresponding PatientID
    string selectSQL = "SELECT PatientID FROM medicalDiagnosis WHERE PatientDiagnosisID = ?";
    sqlite3_stmt *stmt;
    int exit = sqlite3_prepare_v2(DB, selectSQL.c_str(), -1, &stmt, NULL);

    if (exit != SQLITE_OK) {
        cerr << "Failed to prepare statement: " << sqlite3_errmsg(DB) << endl;
        return;
    }

    // Bind the diagnosis ID
    sqlite3_bind_text(stmt, 1, patientDiagnosisID.c_str(), -1, SQLITE_TRANSIENT);

    // Execute the query
    exit = sqlite3_step(stmt);
    if (exit == SQLITE_ROW) {
        string patientID = reinterpret_cast<const char*>(sqlite3_column_text(stmt, 0));
        bool authorized = false;

        // Check if the patient belongs to the logged-in doctor
        // SQL query to check if the patient ID is valid for the doctor
        string patientCheckSQL = "SELECT 1 FROM PatientsAppointment WHERE PatientID = ? AND DoctorID = ?";
        sqlite3_stmt *patientStmt;
        exit = sqlite3_prepare_v2(DB, patientCheckSQL.c_str(), -1, &patientStmt, NULL);

        if (exit != SQLITE_OK) {
            cerr << "Failed to prepare statement: " << sqlite3_errmsg(DB) << endl;
            sqlite3_finalize(stmt);
            return;
        }

        // Bind the patient ID and doctor ID
        sqlite3_bind_text(patientStmt, 1, patientID.c_str(), -1, SQLITE_TRANSIENT);
        sqlite3_bind_text(patientStmt, 2, loggedInDoctorID.c_str(), -1, SQLITE_TRANSIENT);
```

```

    // Execute the query
    exit = sqlite3_step(patientStmt);
    if (exit == SQLITE_ROW) {
        authorized = true; // Authorization successful
    } else {
        cout << "Diagnosis with ID '" << patientDiagnosisID << "' not found. No deletion performed." << endl;
    }

    sqlite3_finalize(patientStmt);

    if (authorized) {
        deleteDiagnosis(loggedInDoctorID, patientDiagnosisID, DB);
    }
} else {
    cout << "Diagnosis with ID '" << patientDiagnosisID << "' not found. No deletion performed." << endl;
}

sqlite3_finalize(stmt);
}

```

## 7.3 Admin Menu

### 7.3.1 Inventory Structure

This structure defines an inventory node, containing details like item ID, purchase item, quantity, person checking, and a pointer (next) that links to the next inventory item in the list.

```

// structure
struct Inventory {
    int ItemID;
    string PurchaseItem;
    int Quantity;
    string PersonChecking;
    Inventory *next;
};

Inventory *Inventoryhead = NULL;

```

### 7.3.2 Inventory Node

This function prints the inventory table by first displaying the table headers and then iterating through the linked list to print the details of each inventory item.

```
// print table
void printInventory(Inventory *n) {
    // Print table header
    cout << setw(5) << left << "Item ID"
    << " | " << setw(5) << left << "PurchaseItem"
    << " | " << setw(5) << left << "Quantity"
    << " | " << setw(5) << left << "Person Checking" << endl;
    // call data
    while (n != NULL) {
        cout << setw(5) << left << n->ItemID << " | "
        << setw(5) << left << n->PurchaseItem << " | "
        << setw(5) << left << n->Quantity << " | "
        << setw(5) << left << n->PersonChecking << endl;
        n = n->next;
    }
    cout << endl;
}
```

### 7.3.3 Inventory Push Back

This function adds a new inventory item to the end of the inventory list by creating a new node with the provided details and linking it to the last node in the list.

```
// push back funtion
void PushBackInventory(
    int itemID,
    string purchaseItem,
    int quantity,
    string personChecking) {
    Inventory *newInventory = new Inventory();
    newInventory->ItemID = itemID;
    newInventory->PurchaseItem = purchaseItem;
    newInventory->Quantity = quantity;
    newInventory->PersonChecking = personChecking;
    newInventory->next = NULL;

    if (Inventoryhead == NULL) {
        Inventoryhead = newInventory;
    } else {
        Inventory *temp = Inventoryhead;
        while (temp->next != NULL)
            temp = temp->next;

        temp->next = newInventory;
    }
}
```

### 7.3.4 Generates Unique Inventory ID

This function generates the next inventory ID by querying the maximum numeric part of the current IDs from the database, incrementing it, and returning the new ID with a prefix "I".

```
string getNextInventoryID() {
    sqlite3_stmt *stmt;
    string sql =
        "SELECT MAX(CAST(SUBSTR(ItemID, 2) AS INTEGER)) FROM Inventory"; // Update table name to 'Inventory'
    int exit = sqlite3_prepare_v2(DB, sql.c_str(), -1, &stmt, NULL);

    if (exit != SQLITE_OK) {
        cerr << "Failed to prepare statement for ID generation: "
        << sqlite3_errmsg(DB) << endl;
        return "";
    }

    int maxID = 0;
    exit = sqlite3_step(stmt);
    if (exit == SQLITE_ROW) {
        maxID = sqlite3_column_int(stmt, 0); // Get the max ID (numeric part) from the table
    }

    sqlite3_finalize(stmt);

    maxID++; // Increment the ID for the new record

    return "I" + to_string(maxID);
}
```

### 7.3.5 Read Inventory

This function retrieves and displays all records from the `Inventory` table by executing a SQL `SELECT` query and using a callback function to process the results.

```
void adminReadInventory() {
    string query = "SELECT * FROM Inventory;";
    sqlite3_exec(DB, query.c_str(), callback, NULL, NULL);
}
```

### 7.3.6 Create Inventory

This function adds new inventory data to the database by prompting the user for details such as `PurchaseItem`, `Quantity`, and `PersonChecking`, validating the user ID, and then inserting the record into the `Inventory` table. It repeats the process if the user chooses to add more entries and displays the updated inventory list after each addition.

```
// Function to add new data
void adminCreateInventory(const set<string>& validUserIDs) {
    string addMore;
    do {
        string PurchaseItem, PersonChecking;
        int Quantity;

        cout << "\nAdd New Inventory" << endl;
        cout << "*****" << endl;

        string ItemID = getNextInventoryID();

        cout << "Enter Purchase Item: ";
        getline(cin >> ws, PurchaseItem);

        cout << "Enter Quantity: ";
        cin >> Quantity;

        while (cin.fail()) {
            handleInvalidInput();
            cin >> Quantity;
        }

        bool validUserID = false;
        do {
            cout << "Enter User ID ( valid options are: ";
            for (const auto &id : validUserIDs) {
                cout << id << " ";
            }
            cout << "): ";

            getline(cin >> ws, PersonChecking);

            // Check if the input is numeric and prefix with 'E' if true
            if (isNumeric(PersonChecking) && !PersonChecking.empty()) {
                PersonChecking = "E" + PersonChecking;
            }

            PersonChecking = toUpperCase(PersonChecking);
```

```
// Direct validation check
validUserID = validUserIDs.find(PersonChecking) != validUserIDs.end();

if (!validUserID) {
    cout << "Invalid User ID. Please enter a valid User ID from the list." << endl;
    cout << "Valid options are: ";
    for (const auto &id : validUserIDs) {
        cout << id << " ";
    }
    cout << ":" ;
}
} while (!validUserID);

// Insert the new Inventory into the database
string sql = "INSERT INTO Inventory (ItemID, PurchaseItem, Quantity, PersonChecking) VALUES ('" +
    ItemID + "', '" + PurchaseItem + "', '" + to_string(Quantity) + "', '" +
    PersonChecking + "')";

char *messageError;
int exit = sqlite3_exec(DB, sql.c_str(), NULL, 0, &messageError);
if (exit != SQLITE_OK) {
    cerr << "Error inserting data: " << messageError << endl;
    sqlite3_free(messageError);
} else {
    cout << "Inventory added successfully!\n" << endl;
}

// Optionally read and print all Inventories to confirm the addition
adminReadInventory();

cout << "Do you want to add another Inventory? (Y/N): ";
getline(cin >> ws, addMore);

} while (addMore == "Y" || addMore == "y");
}
```

### 7.3.7 Update Inventory

This function updates an inventory record in the database by prompting the user for an `ItemID`, verifying its existence, and allowing updates to fields such as `PurchaseItem`, `Quantity`, and `PersonChecking`. The function constructs the SQL update query based on user input, executes it, and displays the updated inventory list.

```
// Function to update Inventory in the database
void adminUpdateInventory(const set<string>& validUserIDs) {
    string ItemID;

    cout << "Enter Inventory ID to update (blank to exit update process): ";
    getline(cin, ItemID);

    if (ItemID.empty()) {
        cout << "Update process exited.\n";
        return;
    }

    if (ItemID.find("I") != 0) {
        ItemID = "I" + ItemID;
    }

    // Check if ItemID exists
    string checkSql =
        "SELECT COUNT(*) FROM Inventory WHERE ItemID = '" + ItemID + "' ";
    sqlite3_stmt *stmt;
    int result = sqlite3_prepare_v2(DB, checkSql.c_str(), -1, &stmt, NULL);

    if (result != SQLITE_OK) {
        cerr << "Error checking ItemID: " << sqlite3_errmsg(DB) << endl;
        return;
    }

    result = sqlite3_step(stmt);
    if (result == SQLITE_ROW) {
        int count = sqlite3_column_int(stmt, 0);
        if (count == 0) {
            cout << "No relevant search results found." << endl;
            sqlite3_finalize(stmt);
            return;
        }
    }
    sqlite3_finalize(stmt);
```

```
string fields[] = {"PurchaseItem", "Quantity", "PersonChecking"};
string updates = "";

cout << "Enter new values for the following fields. Leave empty to skip updating the field."
     << endl;

for (const string &field : fields) {
    string newValue;
    bool validInput = false;

    do {
        cout << field << ": ";
        getline(cin, newValue);

        if (field == "Quantity" && !newValue.empty()) {
            try {
                stof(newValue);
                validInput = true;
            } catch (exception& e) {
                cout << "Invalid Quantity value. Please enter a valid integer." << endl;
            }
        } else if (field == "UserID" && !newValue.empty()) {
            if (isdigit(newValue[0])) {
                newValue = "E" + newValue;
            }
            newValue = toUpperCase(newValue);

            if (validUserIDs.find(newValue) != validUserIDs.end()) {
                validInput = true;
            } else {
                cout << "Invalid UserID. Please enter a valid UserID." << endl;
            }
        } else if (!newValue.empty()) {
            validInput = true;
        } else {
            validInput = true; // Skip empty fields without validation
        }
    } while (!validInput);
```

```
        if (!newValue.empty()) {
            if (!updates.empty()) {
                |   |   updates += ", ";
            }
            updates += field + " = '" + newValue + "'";
        }
    }

    if (updates.empty()) {
        cout << "No fields were updated." << endl;
        return;
    }

    string sql = "UPDATE Inventory SET " + updates + " WHERE ItemID = '" + ItemID + "'";

    char *messageError;
    int exit = sqlite3_exec(DB, sql.c_str(), NULL, 0, &messageError);
    if (exit != SQLITE_OK) {
        cerr << "Error Update: " << messageError << endl;
        sqlite3_free(messageError);
    } else {
        cout << "Data updated successfully!\n" << endl;
    }

    adminReadInventory();
}
```

### 7.3.8 Delete Inventory

This function deletes an inventory record from the database by prompting the user for an `ItemID`, checking if the ID exists, and executing a SQL `DELETE` statement. It then displays the updated inventory list.

```
// Function to delete Inventory from the database
void adminDeleteInventory() {
    string ItemID;

    cout << "Enter Inventory ID to delete (blank to exit delete process): ";
    getline(cin, ItemID);

    if (ItemID.empty()) {
        cout << "Deletion process exited." << endl;
        return;
    }

    // Prepend "I" if it's not present
    if (ItemID.find("I") != 0) {
        ItemID = "I" + ItemID;
    }

    // Check if ItemID exists
    string checkSql = "SELECT COUNT(*) FROM Inventory WHERE ItemID = '" + ItemID + "';";
    sqlite3_stmt *stmt;
    int result = sqlite3_prepare_v2(DB, checkSql.c_str(), -1, &stmt, NULL);

    if (result != SQLITE_OK) {
        cerr << "Error checking ItemID: " << sqlite3_errmsg(DB) << endl;
        return;
    }

    result = sqlite3_step(stmt);
    if (result == SQLITE_ROW) {
        int count = sqlite3_column_int(stmt, 0);
        if (count == 0) {
            cout << "No relevant search results found." << endl;
            sqlite3_finalize(stmt);
            return;
        }
    }
    sqlite3_finalize(stmt);

    string sql = "DELETE FROM Inventory WHERE ItemID = '" + ItemID + "'";

    int exit = sqlite3_exec(DB, sql.c_str(), NULL, 0, &messageError);
    if (exit != SQLITE_OK) {
        cerr << "Error Delete: " << messageError << endl;
        sqlite3_free(messageError);
    } else {
        cout << "Data deleted successfully!\n" << endl;
    }
    adminReadInventory();
}
```

## 8. Database Sqlite3

### 8.1 Billing Table

|   | BillID | PatientID | DateIssued | SubTotal | DiscountPercentage | TotalAmount | PaymentStatus |
|---|--------|-----------|------------|----------|--------------------|-------------|---------------|
|   | Filter | Filter    | Filter     | Filter   | Filter             | Filter      | Filter        |
| 1 | B1     | P1        | 2024-01-07 | 542.0    | NULL               | 542.0       | Unpaid        |
| 2 | B2     | P2        | 2024-02-16 | 873.33   | 10%                | 786.0       | Paid          |
| 3 | B3     | P3        | 2024-03-18 | 206.0    | NULL               | 206.0       | Unpaid        |

### 8.2 Cancel Appointment Table

|   | CancelAppointmentID | AppointmentID | DoctorID | Reason               |
|---|---------------------|---------------|----------|----------------------|
|   | Filter              | Filter        | Filter   | Filter               |
| 1 | CA1                 | A2            | E2       | Scheduling conflicts |
| 2 | CA2                 | A5            | E2       | Because I bookin...  |

### 8.3 Discount Table

|   | DiscountID | DiscountAmount | IssuedDate |
|---|------------|----------------|------------|
|   | Filter     | Filter         | Filter     |
| 1 | V1         | 5%             | 2024-01-05 |
| 2 | V2         | 10%            | 2024-02-16 |
| 3 | D1         | 5%             | 2024-01-05 |
| 4 | D2         | 10%            | 2024-02-16 |

### 8.4 Doctor Schedule Table

|   | DoctorScheduleID | DoctorID | Monday  | Tuesday | Wednesday | Thursday | Friday  | Saturday | Sunday  |
|---|------------------|----------|---------|---------|-----------|----------|---------|----------|---------|
|   | Filter           | Filter   | Filter  | Filter  | Filter    | Filter   | Filter  | Filter   | Filter  |
| 1 | D1               | E1       | Working | Working | Working   | Off      | Off     | Off      | Off     |
| 2 | D2               | E2       | Off     | Off     | Off       | Working  | Working | Working  | Working |

## 8.5 Emergency Table

Table: **EmergencyTools**

| EmergencyToolsID |        | Type                | Tool                         | EmergencyNumber |
|------------------|--------|---------------------|------------------------------|-----------------|
|                  | Filter | Filter              | Filter                       | Filter          |
| 1                | ET1    | Contact Information | Emergency Number             | 03-86457894     |
| 2                | ET2    | Transportation      | Ambulance Department Service | 999             |

## 8.6 Feedback Table

Table: **Feedback**

|   | FeedbackID | PatientID | Comment                                 | Rating | Date       |
|---|------------|-----------|-----------------------------------------|--------|------------|
|   | Filter     | Filter    | Filter                                  | Filter | Filter     |
| 1 | F1         | P1        | Excellent care, very satisfied with ... | 5      | 2024-01-08 |
| 2 | F2         | P2        | Good service, but waiting time was t... | 3      | 2024-02-14 |
| 3 | F3         | P1        | Excellent care, very satisfied with ... | 5      | 2024-01-08 |

## 8.7 Inventory Table

Table: **Inventory**

|   | ItemID | PurchaseItem | Quantity | PersonChecking |
|---|--------|--------------|----------|----------------|
|   | Filter | Filter       | Filter   | Filter         |
| 1 | I1     | Syringes     | 500      | E2             |
| 2 | I2     | Face mask    | 200      | E2             |

## 8.8 medical Diagnosis Table

Table: **MedicalDiagnosis**

|   | PatientDiagnosisID | PatientID | DoctorID | MedicalHistory | MedicationsTaken                      | Diagnosis              | LastUpdate |
|---|--------------------|-----------|----------|----------------|---------------------------------------|------------------------|------------|
|   | Filter             | Filter    | Filter   | Filter         | Filter                                | Filter                 | Filter     |
| 1 | PH1                | P1        | E1       | Hypertension   | Lisinopril, Amlodipine, Losartan, ... | Dengue fever           | 2024-01-08 |
| 2 | PH2                | P2        | E2       | Diabetes       | Metformin, Insulin, DPP-4 inhibitors  | Iron deficiency anemia | 2024-02-17 |
| 3 | PH3                | P3        | E1       | Asthma         | Albuterol (short-acting), Fluticasone | Get HPV vaccine        | 2024-03-15 |
| 4 | PH4                | P3        | E1       | Diabetes       | Metformin, Insulin                    | Type 2 Diabetes        | 2024-03-28 |

## 8.9 Medical Insurance Table

Table: **MedicalInsurance** Filter in any column

| MedicalInsuranceID | PatientID | InsuranceProvider | PolicyNumber | Deductible |
|--------------------|-----------|-------------------|--------------|------------|
| 1 INS1             | P1        | Allianz           | A12392       | 400000     |
| 2 INS2             | P2        | AIA               | AIA193842    | 600000     |
| 3 INS3             | P2        | prudential        | 123456789    | 9000000    |

## 8.10 Notice Table

Table: **Notice** Filter in any column

| NoticeID | Description                 | PostManageBy |
|----------|-----------------------------|--------------|
| 1 N1     | Update emergency tools      | E3           |
| 2 N2     | Vaccination Information     | E3           |
| 3 N3     | Know more about your health | E3           |

## 8.11 Patient Admission Transfer Table

Table: **PatientAdmissionTransfer** Filter in any column

| PatientStatusID | PatientID | CheckInTime      | CheckOutTime     | CurrentStatus |
|-----------------|-----------|------------------|------------------|---------------|
| 1 PS1           | P1        | 2024-01-06 10:30 | 2024-01-08 12:00 | Discharged    |
| 2 PS2           | P2        | 2024-02-10 13:00 | 2024-02-17 16:00 | Transferred   |
| 3 PS3           | P3        | 2024-03-15 14:00 | 2024-03-15 15:30 | Discharged    |

## 8.12 Patient Allergy Table

Table: **PatientAllergy** Filter in any column

| PatientAllergyID | PatientID | AllergyName | Severity | Reaction                                    |
|------------------|-----------|-------------|----------|---------------------------------------------|
| 1 AI1            | P2        | Peanut      | Severe   | Anaphylaxis (life-threatening allergic ...) |
| 2 AI2            | P2        | Asthma      | severe   | cough                                       |

## 8.13 Patient Follow Up Table

| FollowUpRecordID | PatientDiagnosisID | FollowUpDate | Description |                                 | DoctorOrders                               | DoctorID |
|------------------|--------------------|--------------|-------------|---------------------------------|--------------------------------------------|----------|
| Filter           | Filter             | Filter       | Filter      | Filter                          | Filter                                     | Filter   |
| 1                | FU1                | PH1          | 2024-01-07  | Mild fever, joint pain; no rash | Monitor platelets, rest. Follow-up in 1... | E1       |
| 2                | FU2                | PH2          | 2024-01-20  | Respiratory pain                | Continue treatment, add Vitamin C. ...     | E2       |

## 8.14 Patient Appointment Table

| AppointmentID | PatientID | ServiceID | Date   | Time       | Status | DoctorID |
|---------------|-----------|-----------|--------|------------|--------|----------|
| Filter        | Filter    | Filter    | Filter | Filter     | Filter | Filter   |
| 1             | A1        | P1        | S3     | 2024-01-05 | 09:00  | Accessed |
| 2             | A2        | P2        | S1     | 2024-02-08 | 10:00  | Canceled |
| 3             | A3        | P2        | S1     | 2024-02-10 | 10:00  | Accessed |
| 4             | A4        | P3        | S2     | 2024-03-15 | 12:00  | Accessed |
| 5             | A5        | P2        | S2     | 2024-08-09 | 11:00  | Accessed |

## 8.15 Project Table

| ProjectID | Title  | Description                | DateOfPublication                     |
|-----------|--------|----------------------------|---------------------------------------|
| Filter    | Filter | Filter                     | Filter                                |
| 1         | PRO1   | New Vaccination Guidelines | Release of updated guidelines for ... |
| 2         | PRO2   | Mental Health Awareness    | Campaign to promote mental health ... |

## 8.16 Purchase Table

| POID   | SupplierID | ItemID | PricePerBox | QuantityInBox | PurchaseQuantityBox | TotalAmount |
|--------|------------|--------|-------------|---------------|---------------------|-------------|
| Filter | Filter     | Filter | Filter      | Filter        | Filter              | Filter      |
| 1      | PO1        | SU1    | I1          | 60.0          | 100                 | 8           |
| 2      | PO2        | SU2    | I1          | 50.0          | 100                 | 5           |

## 8.17 Supplier Table

| SupplierID | SupplierName | ContactNumber |
|------------|--------------|---------------|
| Filter     | Filter       | Filter        |
| 1          | SU1          | ABC Medical   |
| 2          | SU2          | YZ Supplier   |

## 8.18 Type of Service Table

| ServiceID |        |                     | TypeOfService                            | Description |  |
|-----------|--------|---------------------|------------------------------------------|-------------|--|
| Filter    | Filter | Filter              | Filter                                   |             |  |
| 1         | S1     | Laboratory Services | Diagnostic Services include blood and... |             |  |
| 2         | S2     | Vaccination         | Immunizations cover all ages             |             |  |
| 3         | S3     | Family Medicine     | Primary Care focuses on health, ...      |             |  |

## 8.19 User Table

| User   |                           |              |        |          |               |                         |                                                            |                |          |
|--------|---------------------------|--------------|--------|----------|---------------|-------------------------|------------------------------------------------------------|----------------|----------|
| UserID | Name                      | ICNO         | Gender | Position | ContactNumber | Email                   | Address                                                    | Username       | Password |
| P1     | Johnathan Alexander Doe   | 800101141239 | Male   | Patient  | 60162323245   | johndoe@example.com     | Majestic Maxim Jalan 9/142, Taman ...                      | John Doe       | ab7kd    |
| P2     | Jane Elizabeth Smith      | 850202105678 | Female | Patient  | 60145678901   | janesmith@example.com   | 7 Jalan Dataran Cheras 6, Balakong, ...                    | Jane Smith     | x9mn124  |
| P3     | Lilith Victoria Brown     | 880303089102 | Female | Patient  | 60171234567   | lilithbrown@example.com | Jalan Tasik Utama, Sungai Besi, ...                        | Alice Brown    | 1uz9x    |
| E1     | Johnathan Maxwell Smith   | 750123105671 | Male   | Doctor   | 60154489651   | john@example.com        | Jalan Setia Tropika U13/20, Setia Eco...<br>John Smith     | John Smith     | 156cd    |
| E2     | Matthew Harrison Wilson   | 820305891723 | Male   | Doctor   | 60166546545   | matthew@example.com     | Lebuhraya Sungai Besi BESRAYA, ...<br>Matthew Wilson       | Matthew Wilson | casca62  |
| E3     | Michael Christopher Brown | 910405141122 | Female | Admin    | 60141486554   | michael@example.com     | 1A Tingkat 1 Jalan S 10, 1, Jalan Buld...<br>Michael Brown | Michael Brown  | dsCg530  |

## 8.20 Vaccination Records Table

| VaccinationRecords |           |             |            |                    |
|--------------------|-----------|-------------|------------|--------------------|
| VaccinationID      | PatientID | VaccineType | DoseNumber | AdministrationDate |
| 1                  | V1        | P1          | AZ         | First dose         |
| 2                  | V2        | P3          | Pfizer     | First dose         |
| 3                  | V3        | P2          | Sinovac    | 3                  |
|                    |           |             |            | 2021-09-16         |
|                    |           |             |            | 2021-10-01         |
|                    |           |             |            | 2024-11-11         |

## 9.0 Conclusion

Creating a clinic scheduling system that efficiently manages appointments for patients, doctors, and administrators is a complex task that involves several advanced programming techniques and tools. The use of linked lists, hash functions, pointers, references, and an SQLite3 database for CRUD (Create, Read, Update, Delete) operations is a sophisticated approach that ensures both flexibility and performance.

Linked lists are a fundamental data structure in computer science that allows for dynamic memory allocation. In the context of a clinic scheduling system, linked lists are particularly useful for managing a potentially large and dynamic set of appointments. Each appointment can be represented as a node in the linked list, with pointers connecting each node to the next. This structure allows for efficient insertion and deletion of appointments, which is crucial in a medical environment where schedules are constantly changing. For instance, if a patient cancels an appointment, the corresponding node can be quickly removed from the list without needing to shift other elements, as would be the case with an array.

Hash functions play a critical role in optimizing the retrieval of records, such as patient or doctor information. In a scheduling system, the need to quickly access a patient's record or a doctor's schedule is paramount. By applying a hash function, you can convert data (like patient ID or doctor ID) into a fixed-size integer, which acts as an index in a hash table. This enables near-instantaneous access to the required data, reducing the time complexity of search operations from  $O(n)$  in a linked list to  $O(1)$  in the best-case scenario. This speed is crucial in a medical setting where time is of the essence.

Pointers and references are essential in managing memory efficiently, especially in a system that uses linked lists. Pointers allow the program to dynamically allocate memory for new appointments and manage the connections between nodes in the linked list. By using references, the system can pass large data structures without the overhead of copying them, which enhances performance. In a scheduling system, where many operations involve manipulating complex data structures like appointment records or patient data, pointers and references help in maintaining the integrity of the data while minimizing memory usage.

An SQLite3 database provides a robust backend for storing and managing the clinic's data. SQLite is a lightweight, file-based database that is well-suited for applications where simplicity and efficiency are key. In the clinic scheduling system, the database is used to store persistent information such as patient records, doctor schedules, and appointment histories. The CRUD operations (Create, Read, Update, Delete) are implemented to manage this data. For example, when a new appointment is created, a corresponding entry is added to the database. If an appointment is rescheduled or canceled, the database is updated accordingly.

By integrating SQLite3 with linked lists and hash functions, the scheduling system benefits from both the speed of in-memory operations and the reliability of persistent storage. The database serves as a backup for the linked list, ensuring that no data is lost even if the system is restarted.

In summary, the combination of linked lists, hash functions, pointers, references, and SQLite3 creates a highly efficient and flexible clinic scheduling system. Linked lists provide the dynamic structure needed for managing appointments, while hash functions optimize data retrieval. Pointers and references ensure efficient memory management, and the SQLite3 database offers a reliable backend for persistent storage. Together, these elements form a robust system capable of handling the complex and dynamic environment of a medical clinic, ensuring that patients receive timely care, doctors manage their schedules effectively, and administrators can oversee the system with ease. This integration of advanced programming techniques makes the system both powerful and reliable, ensuring its effectiveness in a real-world medical setting.

Thank you, teacher, for your unwavering dedication and the profound impact you've had on my learning journey. Your guidance, patience, and deep understanding have not only helped me grasp complex concepts but also inspired me to push beyond my limits. The time and effort you invest in each lesson, the care with which you address our questions, and the encouragement you provide have all contributed to my growth, both academically and personally. I am deeply grateful for your commitment to our education and for the positive influence you've had on my life. Your support and wisdom are truly appreciated, and I feel fortunate to have had the opportunity to learn from you.

## 10.0 Distribution

| No | Name          | ID          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|----|---------------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | Lim Feng Yao  | 2350102-DCS | <ul style="list-style-type: none"><li><b>1. Introduction</b></li><li><b>2. Data Structure System</b><ul style="list-style-type: none"><li>- Data Structure: Linked List</li><li>- Data Flow Diagram (DFD)</li></ul></li><li><b>3. Hash Function</b></li><li><b>4. Program Output</b><ul style="list-style-type: none"><li>- Register, login, and forgot password</li><li>- Doctor Menu<ul style="list-style-type: none"><li>· CRUD</li></ul></li></ul></li><li><b>5. Database Sqlite3</b></li><li><b>6. Example Coding</b><ul style="list-style-type: none"><li>- Doctor Menu<ul style="list-style-type: none"><li>· CRUD</li></ul></li></ul></li></ul> |
| 2  | Yap Hui Jun   | 2320020-DCS | <ul style="list-style-type: none"><li><b>1. Data Structure System</b><ul style="list-style-type: none"><li>- Data Flow Diagram (DFD)</li><li>- Data Table</li><li>- Data Type</li></ul></li><li><b>2. Pointers and References</b></li><li><b>3. Program Output</b><ul style="list-style-type: none"><li>- Patient Menu<ul style="list-style-type: none"><li>· CRUD</li></ul></li></ul></li><li><b>4. Example Coding</b><ul style="list-style-type: none"><li>- Patient Menu<ul style="list-style-type: none"><li>· CRUD</li></ul></li></ul></li></ul>                                                                                                   |
| 3  | Wong Jie Ying | 2350080-DCS | <ul style="list-style-type: none"><li><b>1. Flow of System (Flow Chart Diagram)</b></li><li><b>2. Database Table</b></li><li><b>3. Program Output</b><ul style="list-style-type: none"><li>- Admin Menu<ul style="list-style-type: none"><li>· CRUD</li></ul></li></ul></li><li><b>4. Example Coding</b></li></ul>                                                                                                                                                                                                                                                                                                                                      |

|  |  |  |              |
|--|--|--|--------------|
|  |  |  | - Admin Menu |
|  |  |  | · CRUD       |