



**UTM**  
UNIVERSITI TEKNOLOGI MALAYSIA

SECD2523 - DATABASE







SEMESTER 1/20232024



**DATABASE LOGICAL DESIGN**

**LECTURER: DR.NOOR HIDAYAH BINTI ZAKARIA**



NAME	MATRIC NO.
	
	
	

# Table of Contents

<b>1.0 Introduction</b>	<b>4</b>
<b>2.0 Overview of project</b>	<b>5</b>
<b>3.0 Database conceptual design</b>	<b>6</b>
3.1 Updated business rule	6
3.2 Conceptual ERD	7
<b>4.0 DB logical design</b>	<b>8</b>
4.1 Logical ERD	8
4.2 Updated Data Dictionary	9
4.3 Normalization	14
<b>5.0 Relational DB Schemas (after normalization)</b>	<b>17</b>
<b>6.0 SQL Statements (DDL &amp; DML)</b>	<b>20</b>
<b>6.1 DDL:</b>	<b>20</b>
6.2 DML:	23
6.3 Test Query	41
6.3.1 View User Table	41
6.3.2 View Patient Table	41
6.3.3 View Dentist Table	42
6.3.4 View Receptionist Table	42
6.3.5 View Appointment Table	43
6.3.6 View Treatment Table	43
6.3.7 View Check Up Table	44
6.3.8 View Schedule Table	44
6.3.9 View Dental Record Table	44
6.3.10 View Payment Table	45
6.3.11 View Online Transfer Table	45
6.3.12 View Cash Table	46
<b>7.0 Interface</b>	<b>47</b>

7.1 User Login	47
7.2 User Create Account and Profile	47
7.3 Profile of each user (Patient, Dentist, Receptionist)	48
7.4 User Sign In	48
7.5 Patient View (Home Page)	49
7.6 Searching Dentist (Patient View)	49
7.7 Booking Appointment	50
7.8 Booking Appointment (Success / Fail)	51
7.9 Viewing Appointment	51
7.10 Canceling Appointment	52
7.11 Rescheduling Appointment	52
7.12 Making Payment	53
7.13 Making Online Transfer Payment	54
7.14 Making Cash Payment	54
7.15 Canceling Payment	55
7.15 Changing Payment Method	56
7.15 Dentist View	56
7.16 Dental Record for each dentist	57
7.17 Insert Dental Record	58
7.18 Viewing Dental Record	59
7.19 Approving Appointment	59
7.20 Viewing Schedule	60
<b>8.0 Summary</b>	<b>61</b>

## **1.0 Introduction**

In today's era, dental clinics have become a location that is frequented by modern people from time to time. In this fast-paced era, busy people prefer efficient appointment management systems that will help them save more time for other things. Traditional methods of appointment scheduling often lead to inefficiencies, inaccuracies, and extreme challenges in maintaining the optimal balance between the availability of dental professionals and the unique needs of patients.

Effective appointment management is the cornerstone of providing quality patient care and maintaining the operational efficiency of a dental office. The sophistication of the appointment management system will affect the patient's experience. In addition, an efficient appointment system can streamline the daily operations of a dental office and reduce the workload of dentists and nurses. Therefore, the development of a dental appointment rescheduling system is an innovative solution. We hope to improve the scheduling process, minimize wait times and enhance the overall patient experience. Indirectly, this will help dental clinics to reduce the burden on staff and increase the number of customers.

## **2.0 Overview of project**

In the phase of database logical design, we will convert the conceptual Entity Relationship Diagrams produced in Phase 2 into logical ERD. In the logical ERS, modifications will be made to address the complexities of dental appointment management. This will manage the stage for building a more structured and efficient database. We will also convert each entity in the logical ERD to a relational schema, define attributes and create primary keys for each table. This step is particularly relevant for dental appointment systems, where entities such as patient, appointment, dentist, receptionist and payment need to be clearly defined and interrelated.

Normalization of our system will also take place at this stage by eliminating redundancies and dependencies in the derived relational schema. In a dental appointment system, this ensures that patient information is not stored in duplicate, appointments are correctly associated with patients and dentists, and data integrity is maintained. At the same time, the data dictionary is updated as necessary to reflect the changes brought about by normalization.

Finally, we will propose SQL statements to validate logical ERDs with the system's transaction requirements using interface design. This is very important for ensuring the database structure supports basic operations such as appointment, rescheduling, and payment processing can be carried out smoothly . In the field of dental appointment management, the responsiveness and reliability of the system affects patient care, so this validation process is crucial.

### **3.0 Database conceptual design**

#### **3.1 Updated business rule**

Dentist:

- View and update the patient e-profiles through the system
- Reviews patient x-rays digitally and updates findings in system
- Digitally creates and updates treatment plans for patients
- E-prescribes medications and treatments for patients
- Adjusts digital appointment calendar to accommodate changes

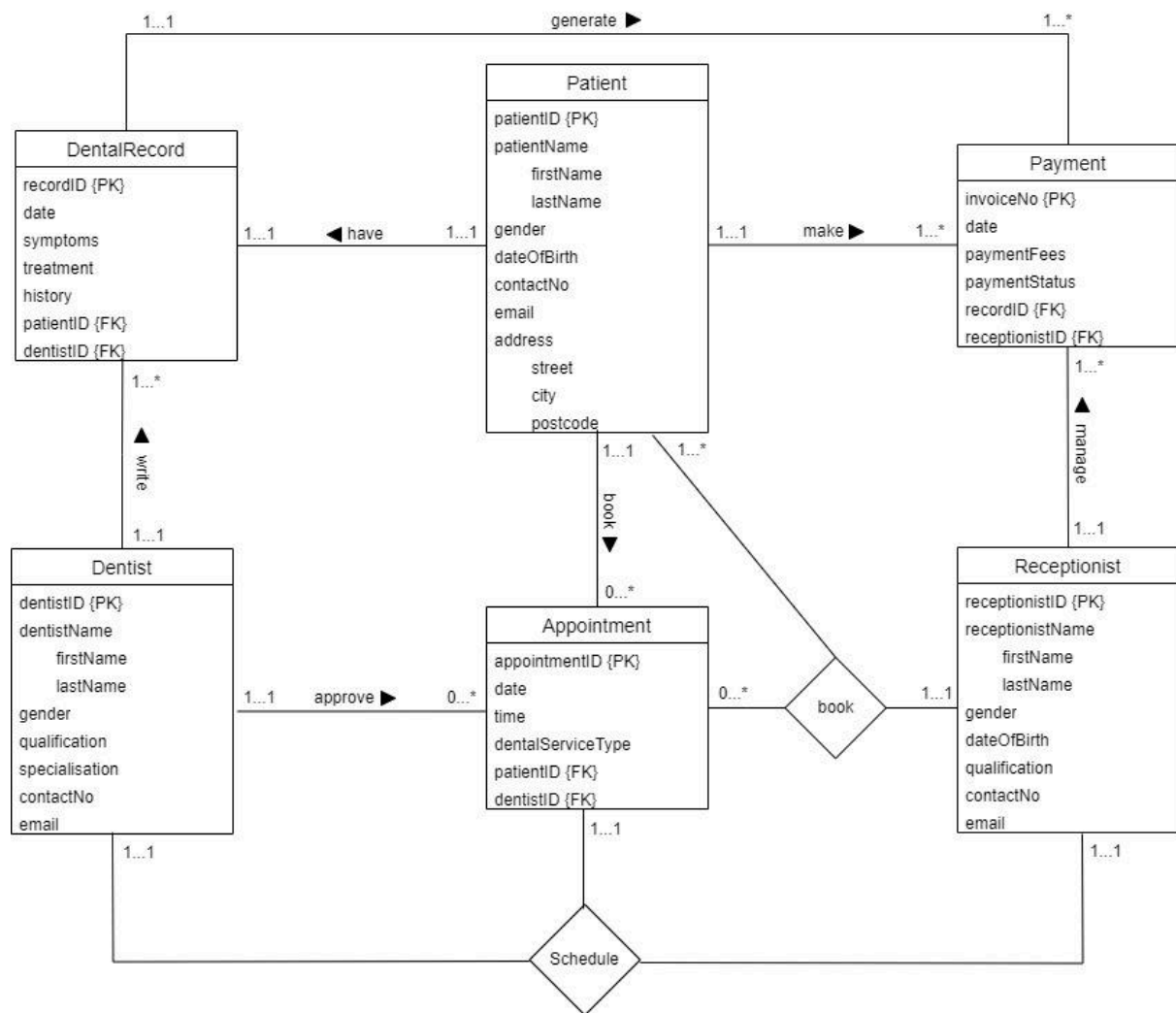
Receptionist:

- Book patient appointments through digital calendar system
- Enters new patient details directly into the system
- System automatically sends appointment reminders to patients
- Reschedule appointments digitally based on requests

Patient:

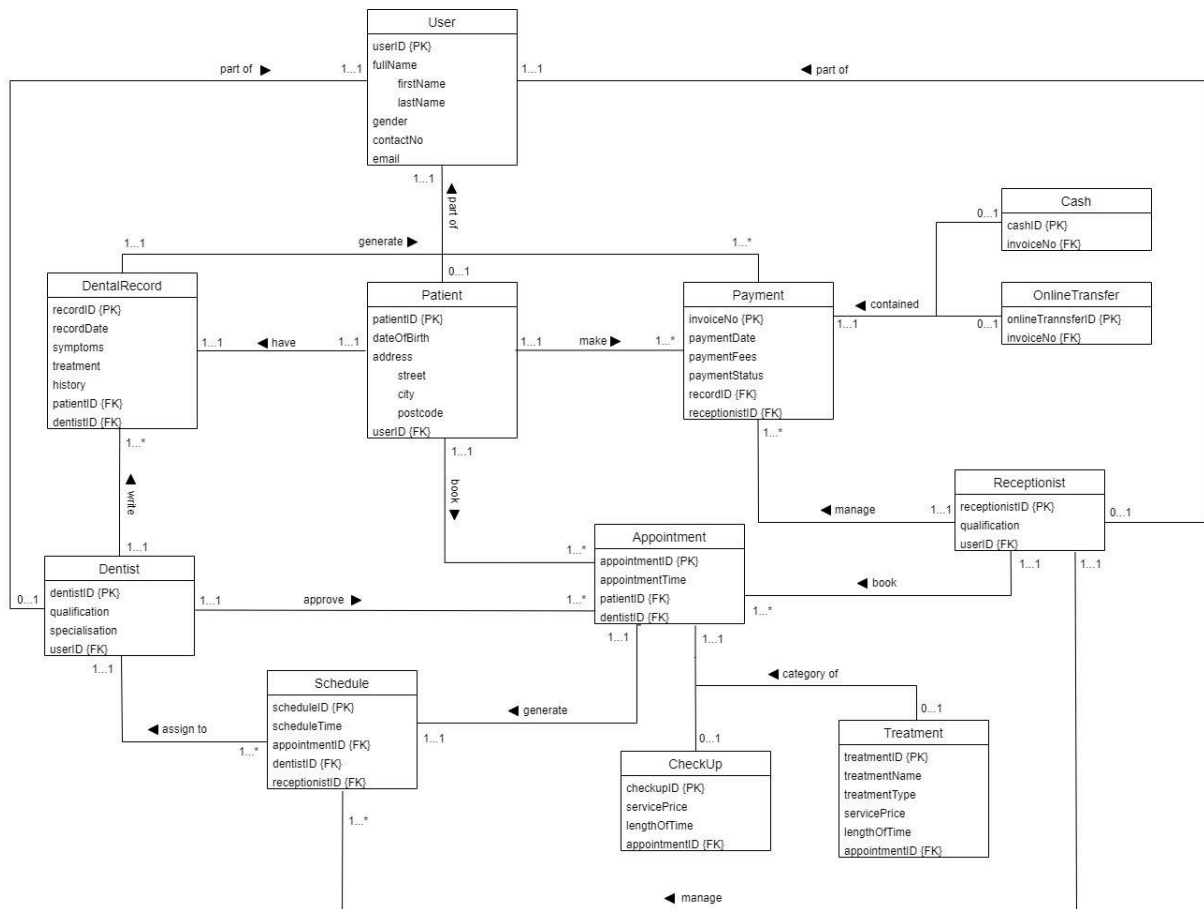
- Book initial and follow-up appointments through online portal
- Receives automatic appointment reminders
- Do an online payment for dental services
- Access dental records and treatment plans via patient portal

### 3.2 Conceptual ERD



## 4.0 DB logical design

### 4.1 Logical ERD





## 4.2 Updated Data Dictionary

### Description of Entity

Entity	Description	Occurrence
User	Superclass for Patient, Dentist, and Receptionist	User can be associated with either Patient, Dentist, or Receptionist, but not more than one at a time.
Patient	Hold patient's information	Patients make appointments then record in dental records and make payment based on the dental record.
Appointment	Hold appointment's information	Appointment booked by receptionist and approved by dentist.
CheckUp	Hold check up appointment's information	One of the options available to patients for booking appointments.
Treatment	Hold treatment appointment's information	One of the options available to patients for booking appointments.
Schedule	Hold schedule's information based on appointment	Schedule will be generated by appointment and managed by the receptionist and it will be assigned to the dentist.
DentalRecord	Hold dental record's information	Dental records written by dentists which include patient information and generate payment.
Dentist	Hold dentist's information	Dentists approve appointments and write dental records.
Receptionist	Hold receptionist's information	Receptionists book appointments and manage payment.
Payment	Hold payment information	Payment generated by dental record and made by patient.
Cash	Hold payment information when paying by cash	One of the selections available to patients when making payment.
OnlineTransfer	Hold payment information when paying by online transfer	One of the selections available to patients when making payment.

### Description of Relationship

Entity	Multiplicity	Relationship	Multiplicity	Entity
Patient	1...1	Book	1...*	Appointment
	1...1	Have	1...1	DentalRecord
	1...1	Make	1...*	Payment
	0...1	Part of	1...1	User
Appointment	1...1	Generate	1...1	Schedule
CheckUp	0...1	Category of	1...1	Appointment
Treatment	0...1	Category of	1...1	Appointment
Schedule	1...*	Assign to	1...1	Dentist
DentalRecord	1...1	Generate	1...*	Payment
Dentist	1...1	Approve	1...*	Appointment
	1...1	Write	1...*	DentalRecord
	0...1	Part of	1...1	User
Receptionist	1...1	Book	1...*	Appointment
	1...1	Manage	1...*	Payment
	0...1	Manage	1...*	Schedule
	0...1	Part of	1...1	User
Cash	0...1	Contained	1...1	Payment
OnlineTransfer	0...1	Contained	1...1	Payment

### Description Attributes

Entity	Attributes	Description	Data Type	Null	Multi-Valued
User	userID	Uniquely identify a user(PK)	VARCHAR2(10)	No	No
	fullName	Full name of user	VARCHAR2(50)	No	No
	firstName	First name of user	VARCHAR2(30)	No	No
	lastName	Last name of user	VARCHAR2(30)	No	No
	gender	Gender of user	VARCHAR2(6)	No	No
	contactNo	Contact number of user	VARCHAR2(13)	No	Yes
	email	Email address of user	VARCHAR2(50)	Yes	No
Patient	patientID	Uniquely identify a patient (PK)	VARCHAR2(10)	No	No
	dateOfBirth	Date of birth of patient	DATE	No	No
	address	Address of patient	VARCHAR2(45)	Yes	No
	street	Street address of patient	VARCHAR2(30)	Yes	No
	city	City address of patient	VARCHAR2(15)	Yes	No
	postcode	Postcode address of patient	VARCHAR2(5)	Yes	No
	userID	Foreign key of User which uniquely identify a user(FK)	VARCHAR2(10)	No	No
Appointment	appointmentID	Uniquely identify an appointment (PK)	VARCHAR2(10)	No	No
	appointmentTime	Time of appointment	TIMESTAMP	No	No
	dentalServiceType	Type of dental service	VARCHAR2(20)	No	No
	patientID	Foreign key of Patient which uniquely identify a patient (FK)	VARCHAR2(10)	No	No
	dentistID	Foreign key of Dentist which uniquely identify a dentist (FK)	VARCHAR2(10)	No	No
CheckUp	checkupID	Uniquely identify a Check Up appointment (PK)	VARCHAR2(10)	No	No
	servicePrice	Price of check up service	FLOAT(8)	No	No
	lengthOfTime	Period of check up service	VARCHAR2(10)	No	No

	appointmentID	Foreign key of Appointment which uniquely identify a dentist (FK)	VARCHAR2(10)	No	No
Treatment	treatmentID	Uniquely identify Treatment appointment (PK)	VARCHAR2(10)	No	No
	treatmentName	Name of treatment	VARCHAR2(30)	No	No
	treatmentType	Type of treatment	VARCHAR2(20)	No	No
	servicePrice	Price of treatment service	FLOAT(8)	No	No
	lengthOfTime	Period of treatment service	VARCHAR2(10)	No	No
	appointmentID	Foreign key of Appointment which uniquely identify a dentist (FK)	VARCHAR2(10)	No	No
Schedule	scheduleID	Uniquely identify a Schedule (PK)	VARCHAR2(10)	No	No
	scheduleTime	Time of schedule	TIMESTAMP	No	No
	appointmentID	Foreign key of Appointment which uniquely identify a dentist (FK)	VARCHAR2(10)	No	No
	dentistID	Foreign key of Dentist which uniquely identify a dentist (FK)	VARCHAR2(10)	No	No
	receptionistID	Foreign key of Receptionist which uniquely identify a receptionist (FK)	VARCHAR2(10)	No	No
DentalRecord	recordID	Uniquely identify a dental record (PK)	VARCHAR2(10)	No	No
	recordDate	Date of dental record	DATE	No	No
	symptoms	symptoms of patient	VARCHAR2(20)	Yes	Yes
	treatment	treatment of patient	VARCHAR2(20)	Yes	Yes
	history	history dental record	VARCHAR2(50)	Yes	Yes
	patientID	Foreign key of Patient which uniquely identify a patient (FK)	VARCHAR2(10)	No	No
	dentistID	Foreign key of Dentist which uniquely identify a dentist (FK)	VARCHAR2(10)	No	No

Dentist	dentistID	Uniquely identify a dentist (PK)	VARCHAR2(10)	No	No
	qualification	Qualification of dentist	VARCHAR2(10)	No	No
	specialization	Specialization of dentist	VARCHAR2(20)	No	No
	userID	Foreign key of User which uniquely identify a user(FK)	VARCHAR2(10)	No	No
Receptionist	receptionistID	Uniquely identify a receptionist (PK)	VARCHAR2(10)	No	No
	qualification	Qualification of receptionist	VARCHAR2(25)	Yes	No
	userID	Foreign key of User which uniquely identify a user(FK)	VARCHAR2(10)	No	No
Payment	invoiceNo	Uniquely identify once payment (PK)	VARCHAR2(10)	No	No
	paymentDate	Date of payment done	DATE	No	No
	paymentFees	Total fee of payment	FLOAT(8)	No	No
	paymentStatus	Status of payment	VARCHAR2(20)	No	No
	recordID	Foreign key of DentalRecord which uniquely identify a dental record (FK)	VARCHAR2(10)	No	No
	receptionistID	Foreign key of Receptionist which uniquely identify a receptionist (FK)	VARCHAR2(10)	No	No
Cash	cashID	Uniquely identify the cash payment method (PK)	VARCHAR2(15)	No	No
	invoiceNo	Foreign key of Payment which uniquely identify a payment (FK)	VARCHAR2(10)	No	No
OnlineTransfer	onlineTransferID	Uniquely identify the online transfer method (PK)	VARCHAR2(15)	No	No
	invoiceNo	Foreign key of Payment which uniquely identify a payment (FK)	VARCHAR2(10)	No	No

### 4.3 Normalization

1. USER (userID, firstName, lastName, gender, contactNo, email)  
**fd1:** userID → firstName, lastName, gender, contactNo, email  
**1NF&2NF&3NF&BCNF:**  
USER (userID, firstName, lastName, gender, contactNo, email)
2. PATIENT (patientID, dateOfBirth, street, city, postcode, userID)  
**fd1:** patientID → dateOfBirth, street, city, postcode, userID  
**1NF&2NF&3NF&BCNF:**  
PATIENT (patientID, dateOfBirth, street, city, postcode, userID)
3. RECEPTIONIST (receptionistID, qualification, userID)  
**Fd1:** receptionistID → qualification, userID  
**1NF&2NF&3NF&BCNF:**  
RECEPTIONIST (receptionistID, qualification, userID)
4. DENTIST (dentistID, qualification, specialization, userID)  
**fd1:** dentistID → qualification, specialization, userID  
**1NF&2NF&3NF&BCNF:**  
DENTIST (dentistID, qualification, specialization, userID)
5. APPOINTMENT (appointmentID, appointmentTime, patientID, dentistID)  
**fd1:** appointmentID → appointmentTime, patientID, dentistID  
**1NF&2NF&3NF&BCNF:**  
APPOINTMENT (appointmentID, appointmentTime, patientID, dentistID)
6. CHECKUP (checkupID, servicePrice, lengthOfTime, appointmentID)  
**Fd1:** checkupID → servicePrice, lengthOfTime, appointmentID  
**1NF&2NF&3NF&BCNF:**  
CHECKUP (checkupID, servicePrice, lengthOfTime, appointmentID)
7. TREATMENT (treatmentID, treatmentName, treatmentType, servicePrice, lengthOfTime, appointmentID)

**fd1:** treatmentID → treatmentName, treatmentType, servicePrice, lengthOfTime, appointmentID

**1NF&2NF&3NF&BCNF:**

TREATMENT (treatmentID, treatmentName, treatmentType, servicePrice, lengthOfTime, appointmentID)

8. SCHEDULE (scheduleID, scheduleTime, appointmentID, dentistID, receptionistID)

**fd1:** scheduleID → scheduleTime, appointmentID, dentistID, receptionistID

**1NF&2NF&3NF&BCNF:**

SCHEDULE (scheduleID, scheduleTime, appointmentID, dentistID, receptionistID)

9. DENTALRECORD (recordID, recordDate, symptoms, treatment, history, patientID, dentistID)

**fd1:** recordID → recordDate, symptoms, treatment, history, patientID, dentistID

**1NF&2NF&3NF&BCNF:**

DENTALRECORD (recordID, recordDate, symptoms, treatment, history, patientID, dentistID)

10. PAYMENT (invoiceNo, paymentDate, paymentFees, paymentStatus, recordID, receptionistID)

**fd1:** invoiceNo → paymentDate, paymentFees, paymentStatus, recordID, receptionistID

**1NF&2NF&3NF&BCNF:**

PAYMENT (invoiceNo, paymentDate, paymentFees, paymentStatus, recordID, receptionistID)

11. CASH (cashNo, invoiceNo)

**fd1:** cashNo → invoiceNo

**1NF&2NF&3NF&BCNF:**

CASH (cashNo, invoiceNo)

12. ONLINETRANSFER (onlineTransferID, invoiceNo)

**fd1:** onlineTransferID → invoiceNo

**1NF&2NF&3NF&BCNF:**

ONLINE\_TRANSFER (onlineTransferID, invoiceNo)

\*Remark: Underline word is primary key.



## 5.0 Relational DB Schemas (after normalization)

The relational database schema for Dental Appointment and Rescheduling System database is a set of relation schema which consist of;

User	( <u>userID</u> , firstName, lastName, gender, contactNo, email)
Patient	( <u>patientID</u> , dateOfBirth, street, city, postcode, <i>userID</i> )
Receptionist	( <u>receptionistID</u> , qualification, <i>userID</i> )
Dentist	( <u>dentistID</u> , qualification, specialization, <i>userID</i> )
Appointment	( <u>appointmentID</u> , appointmentTime, <i>patientID</i> , <i>dentistID</i> )
CheckUp	( <u>checkupID</u> , servicePrice, lengthOfTime, <i>appointmentID</i> )
Treatment	( <u>treatmentID</u> , treatmentName, treatmentType, servicePrice, lengthOfTime, <i>appointmentID</i> )
Schedule	( <u>scheduleID</u> , scheduleTime, <i>appointmentID</i> , <i>dentistID</i> , <i>receptionistID</i> )
DentalRecord	( <u>recordID</u> , recordDate, symptoms, treatment, history, <i>patientID</i> , <i>dentistID</i> )
Payment	( <u>invoiceNo</u> , paymentDate, paymentFees, paymentStatus, <i>recordID</i> , <i>receptionistID</i> )
Cash	( <u>cashNo</u> , <i>invoiceNo</i> )
OnlineTransfer	( <u>onlineTransferID</u> , <i>invoiceNo</i> )

\*Remark: Underline word is primary key.

**User**

userID	firstName	lastName	gender	contactNo	email
--------	-----------	----------	--------	-----------	-------

**Patient**

patientID	dateOfBirth	street	city	postcode	userID
-----------	-------------	--------	------	----------	--------

**Receptionist**

receptionistID	qualification	userID
----------------	---------------	--------

**Dentist**

dentistID	qualification	specialisation	userID
-----------	---------------	----------------	--------

**Appointment**

appointmentID	appointmentTime	patientID	dentistID
---------------	-----------------	-----------	-----------

**CheckUp**

checkupID	servicePrice	lengthOfTime	appointmentID
-----------	--------------	--------------	---------------

**Treatment**

treatmentID	treatmentName	treatmentType	servicePrice	lengthOfTime	appointmentID
-------------	---------------	---------------	--------------	--------------	---------------

**Schedule**

scheduleID	scheduleTime	appointmentID	dentistID	receptionistID
------------	--------------	---------------	-----------	----------------

**DentalRecord**

recordID	recordDate	symptoms	treatment	history	patientID	dentistID
----------	------------	----------	-----------	---------	-----------	-----------

**Payment**

invoiceNo	paymentDate	paymentFees	paymentStatus	recordID	receptionistID
-----------	-------------	-------------	---------------	----------	----------------

**Cash**

CashNo	invoiceNo
--------	-----------

**OnlineTransfer**

onlineTransferID	invoiceNo
------------------	-----------

## 6.0 SQL Statements (DDL & DML)

### 6.1 DDL:

/\*Create table for each entity\*/

```
CREATE TABLE "User"(  
    userID VARCHAR2(10) PRIMARY KEY,  
    firstName VARCHAR2(30) NOT NULL,  
    lastName VARCHAR2(30) NOT NULL,  
    gender VARCHAR2(6) NOT NULL,  
    contactNo VARCHAR2(13) NOT NULL,  
    email VARCHAR2(50)  
);
```

```
CREATE TABLE "Patient"(  
    patientID VARCHAR2(10) PRIMARY KEY,  
    dateOfBirth DATE DEFAULT SYSDATE,  
    street VARCHAR2(15),  
    city VARCHAR2(15),  
    postcode VARCHAR2(5),  
    userID VARCHAR2(10),  
    FOREIGN KEY(userID) REFERENCES "User"(userID)  
);
```

```
CREATE TABLE "Dentist"(  
    dentistID VARCHAR2(10) PRIMARY KEY,  
    qualification VARCHAR2(10) NOT NULL,  
    specialization VARCHAR2(10),  
    userID VARCHAR2(10),  
    FOREIGN KEY (userID) REFERENCES "User"(userID)  
);
```

```
CREATE TABLE "Receptionist"(  
    receptionistID VARCHAR2(10) PRIMARY KEY,  
    dateOfBirth DATE DEFAULT SYSDATE,  
    qualification VARCHAR2(10),  
    userID VARCHAR2(10),  
    FOREIGN KEY (userID) REFERENCES "User"(userID)  
);
```

```
CREATE TABLE "Appointment"(  

```

```

        appointmentID VARCHAR2(10) PRIMARY KEY,
        appointmentTime TIMESTAMP,
        dentalServiceType VARCHAR2(20) NOT NULL,
        patientID VARCHAR2(10),
        dentistID VARCHAR2(10),
        FOREIGN KEY(patientID) REFERENCES "Patient"(patientID),
        FOREIGN KEY(dentistID) REFERENCES "Dentist"(dentistID)
    );

```

```

CREATE TABLE TREATMENT(
    treatmentID VARCHAR2(10) PRIMARY KEY,
    treatmentName VARCHAR2(30) NOT NULL,
    treatmentType VARCHAR2(20) NOT NULL,
    servicePrice FLOAT(8) NOT NULL,
    lengthOfTime VARCHAR2(5) NOT NULL,
    appointmentID VARCHAR2(10),
    FOREIGN KEY (appointmentID) REFERENCES "Appointment"
    (appointmentID)
);

```

```

CREATE TABLE "CheckUp"(
    checkupID VARCHAR2(10) PRIMARY KEY,
    servicePrice FLOAT(8) NOT NULL,
    lengthOfTime VARCHAR2(5) NOT NULL,
    appointmentID VARCHAR2(10),
    FOREIGN KEY (appointmentID) REFERENCES
    "Appointment"(appointmentID)
);

```

```

CREATE TABLE "DentalRecord"(
    recordID VARCHAR2(10) PRIMARY KEY,
    recordDate TIMESTAMP,
    symptoms VARCHAR2(20),
    treatment VARCHAR2(20),
    history VARCHAR2(50),
    patientID VARCHAR2(10),
    dentistID VARCHAR2(10),
    FOREIGN KEY(patientID) REFERENCES "Patient"(patientID),
    FOREIGN KEY(dentistID) REFERENCES "Dentist"(dentistID)
);

```

```

CREATE TABLE "Schedule"(
    scheduleID VARCHAR2(10) PRIMARY KEY,
    scheduleTime TIMESTAMP,
    patientID VARCHAR2(10),
    dentistID VARCHAR2(10),
    receptionistID VARCHAR2(10),
    FOREIGN KEY (dentistID) REFERENCES "Dentist"(dentistID),
    FOREIGN KEY (receptionistID) REFERENCES
"Receptionist"(receptionistID)
);

```

```

CREATE TABLE "Payment"(
    invoiceNo VARCHAR2(10) PRIMARY KEY,
    paymentDate TIMESTAMP,
    paymentFees FLOAT(8) NOT NULL,
    paymentStatus VARCHAR2(20) NOT NULL,
    recordID VARCHAR2(10),
    receptionistID VARCHAR2(10),
    FOREIGN KEY(recordID) REFERENCES "DentalRecord"(recordID),
    FOREIGN KEY(receptionistID) REFERENCES
"Receptionist"(receptionistID)
);

```

```

CREATE TABLE "Cash"(
    cashID VARCHAR2(15) PRIMARY KEY,
    invoiceNo VARCHAR2(10) NOT NULL,
    FOREIGN KEY (invoiceNo) REFERENCES "Payment"(invoiceNo)
);

```

```

CREATE TABLE "Online Transfer"(
    OnlineTransferID VARCHAR2(15) PRIMARY KEY,
    invoiceNo VARCHAR2(10) NOT NULL,
    FOREIGN KEY (invoiceNo) REFERENCES "Payment"(invoiceNo)
);

```

```

ALTER TABLE "Schedule"
DROP (patientID);

```

```

ALTER TABLE "Schedule"
ADD appointmentID VARCHAR2(10);

```

```
ALTER TABLE "Schedule"  
ADD CONSTRAINT fk_appointment_ID FOREIGN KEY (appointmentID)  
REFERENCES "Appointment"(appointmentID);
```

```
ALTER TABLE "Receptionist"  
MODIFY qualification VARCHAR2(25) NOT NULL;
```

```
ALTER TABLE "Dentist"  
MODIFY specialization VARCHAR2(20) NOT NULL;
```

```
ALTER TABLE "TREATMENT" RENAME TO "Treatment";
```

```
ALTER TABLE "Patient"  
MODIFY street VARCHAR2(30);
```

```
ALTER TABLE "Receptionist"  
DROP (dateOfBirth);
```

```
ALTER TABLE "Treatment"  
MODIFY lengthOfTime VARCHAR2(10);
```

```
ALTER TABLE "CheckUp"  
MODIFY lengthOfTime VARCHAR2(10);
```

## 6.2 DML:

```
--User (Patient)
```

```
INSERT INTO  
"User"(userID,firstName,lastName,gender,contactNo,email)  
VALUES('U001','Denies','Wong','Female','01139762466','denies05  
16@gmail.com');
```

```
INSERT INTO  
"User"(userID,firstName,lastName,gender,contactNo,email)  
VALUES('U002','Joyce','Lee','Female','0167710851','joyce031019  
@yahoo.com');
```

```
INSERT INTO
"User"(userID,firstName,lastName,gender,contactNo,email)
VALUES('U003','Titanic','Tan','Female','01111413231','ttf0406@
yahoo.com');
```

```
INSERT INTO
"User"(userID,firstName,lastName,gender,contactNo,email)
VALUES('U004','Nancy','Lim','Female','01137707837','enting0601
@yahoo.com');
```

```
INSERT INTO
"User"(userID,firstName,lastName,gender,contactNo,email)
VALUES('U005','Christina','Tang','Female','0127376115','zyiibu
sytoeat@gmail.com');
```

```
INSERT INTO
"User"(userID,firstName,lastName,gender,contactNo,email)
VALUES('U006','Koh','Jing
Yi','Male','01110741160','jy030531@gmail.com');
```

#### -- User (Dentist)

```
INSERT INTO
"User"(userID,firstName,lastName,gender,contactNo,email)
VALUES('U007','John','Doe','Male','01169382348',
'john.doe@hotmail.com');
```

```
INSERT INTO
"User"(userID,firstName,lastName,gender,contactNo,email)
VALUES('U008','Alice','Smith','Female','01128962522',
'alice.smith@gmail.com');
```

```
INSERT INTO
"User"(userID,firstName,lastName,gender,contactNo,email)
VALUES('U009','Bob','Johnson','Male','0124141691',
'bob.johnson@yahoo.com');
```

#### -- User (Receptionist)

```
INSERT INTO "User" (userID, firstName, lastName, gender,
contactNo, email)
VALUES('U010','Emily','Johnson','Female','0145678901',
'emily.johnson@yahoo.com');
```

```
INSERT INTO "User" (userID, firstName, lastName, gender,
contactNo, email)
```



```
VALUES('U011', 'David', 'Miller', 'Male', '0156789012',  
'david.miller@hotmail.com');
```

#### **-- Patient**

```
INSERT INTO  
"Patient"(patientID,dateOfBirth,street,city,postcode,userID)  
VALUES('P001',TO_DATE('2003-05-16', 'YYYY-MM-DD'),'20,jalan  
lawa 16','johor','81800','U001');
```

```
INSERT INTO  
"Patient"(patientID,dateOfBirth,street,city,postcode,userID)  
VALUES('P002',TO_DATE('2003-10-19', 'YYYY-MM-DD'),'60,jalan  
pesona 16','johor','81800','U002');
```

```
INSERT INTO  
"Patient"(patientID,dateOfBirth,street,city,postcode,userID)  
VALUES('P003',TO_DATE('2003-04-06', 'YYYY-MM-DD'),'12,jalan  
danau 27','johor','81800','U003');
```

```
INSERT INTO  
"Patient"(patientID,dateOfBirth,street,city,postcode,userID)  
VALUES('P004',TO_DATE('2003-06-01', 'YYYY-MM-DD'),'blok  
L,jalan gaya 11','johor','81800','U004');
```

```
INSERT INTO  
"Patient"(patientID,dateOfBirth,street,city,postcode,userID)  
VALUES('P005',TO_DATE('2003-09-05', 'YYYY-MM-DD'),'31,jalan  
harmonium 30/5','johor','81800','U005');
```

```
INSERT INTO  
"Patient"(patientID,dateOfBirth,street,city,postcode,userID)  
VALUES('P006',TO_DATE('2003-05-31', 'YYYY-MM-DD'),'25,jalan  
cemerlang 6','johor','81800','U006');
```

#### **-- Dentist**

```
INSERT INTO  
"Dentist"(dentistID,qualification,specialization,userID)  
VALUES('D001','DDS','Orthodontics','U007');
```

```
INSERT INTO  
"Dentist"(dentistID,qualification,specialization,userID)  
VALUES('D002','DMD','Endodontics','U008');
```

```

INSERT INTO
"Dentist"(dentistID,qualification,specialization,userID)
VALUES('D003','BDS','General Dentistry','U009');

-- Receptionist

INSERT INTO
"Receptionist"(receptionistID,qualification,userID)
VALUES('R001','Reception Management','U010');

INSERT INTO
"Receptionist"(receptionistID,qualification,userID)
VALUES('R002','Office Administration','U011');

-- Concatenate the name and address

SELECT
    userID,
    firstName || ' ' || lastName AS fullName
FROM "User";

SELECT
    patientID,
    firstName || ' ' || lastName AS fullName,
    street || ', ' || postcode || ' ' || city AS fullAddress
FROM "Patient"
INNER JOIN "User" ON "Patient".userID = "User".userID;

SELECT
    dentistID,
    firstName || ' ' || lastName AS fullName
FROM "Dentist"
INNER JOIN "User" ON "Dentist".userID = "User".userID;

SELECT
    receptionistID,
    firstName || ' ' || lastName AS fullName
FROM "Receptionist"
INNER JOIN "User" ON "Receptionist".userID = "User".userID;

```

USERID	FULLNAME
U001	Denies Wong
U002	Joyce Lee
U003	Titanic Tan
U004	Nancy Lim
U005	Christina Tang
U006	Koh Jing Yi
U007	John Doe
U008	Alice Smith
U009	Bob Johnson
U010	Emily Johnson
U011	David Miller

**Diagram 6.2.1: Output for User Table**

PATIENTID	FULLNAME	FULLADDRESS
P001	Denies Wong	20,jalan lawa 16, 81800 johor
P002	Joyce Lee	60,jalan pesona 16, 81800 johor
P003	Titanic Tan	12,jalan danau 27, 81800 johor
P004	Nancy Lim	blok L,jalan gaya 11, 81800 johor
P005	Christina Tang	31,jalan harmonium 30/5, 81800 johor
P006	Koh Jing Yi	25,jalan cemerlang 6, 81800 johor

**Diagram 6.2.2: Output for Patient Table**

DENTISTID	FULLNAME
D001	John Doe
D002	Alice Smith
D003	Bob Johnson

**Diagram 6.2.3: Output for Dentist Table**

RECEPTIONISTID	FULLNAME
R001	Emily Johnson
R002	David Miller

**Diagram 6.2.4: Output for Receptionist Table**

#### **-- Booking the appointment**

```

INSERT INTO "Appointment"(appointmentID, appointmentTime,
dentalServiceType, patientID, dentistID)
VALUES ('A001', TO_TIMESTAMP('2024-02-01 11:00:00',
'YYYY-MM-DD HH24:MI:SS'), 'CheckUp', 'P001', 'D002');

INSERT INTO "Appointment"(appointmentID, appointmentTime,
dentalServiceType, patientID, dentistID)
VALUES ('A002',TO_TIMESTAMP('2024-02-01 15:00:00', 'YYYY-MM-DD
HH24:MI:SS'), 'Treatment', 'P002', 'D001');

INSERT INTO "Appointment"(appointmentID, appointmentTime,
dentalServiceType, patientID, dentistID)
VALUES ('A003',TO_TIMESTAMP('2024-02-02 09:00:00', 'YYYY-MM-DD
HH24:MI:SS'), 'Treatment', 'P003', 'D001');

INSERT INTO "Appointment"(appointmentID, appointmentTime,
dentalServiceType, patientID, dentistID)
VALUES ('A004',TO_TIMESTAMP('2024-02-03 09:00:00', 'YYYY-MM-DD
HH24:MI:SS'), 'CheckUp', 'P004', 'D003');

INSERT INTO "Appointment"(appointmentID, appointmentTime,
dentalServiceType, patientID, dentistID)

```

```
VALUES ('A005',TO_TIMESTAMP('2024-02-05 13:00:00', 'YYYY-MM-DD
HH24:MI:SS'), 'Treatment', 'P005', 'D002');
```

```
INSERT INTO "Appointment"(appointmentID, appointmentTime,
dentalServiceType, patientID, dentistID)
VALUES ('A006',TO_TIMESTAMP('2024-02-06 11:00:00', 'YYYY-MM-DD
HH24:MI:SS'), 'Treatment', 'P006', 'D002');
```

```
-- View the Appointment Table
```

```
SELECT * FROM "Appointment";
```

APPOINTMENTID	APPOINTMENTTIME	DENTALSERVICETYPE	PATIENTID	DENTISTID
A001	01-FEB-24 11.00.00.000000 AM	CheckUp	P001	D002
A002	01-FEB-24 03.00.00.000000 PM	Treatment	P002	D001
A003	02-FEB-24 09.00.00.000000 AM	Treatment	P003	D001
A004	03-FEB-24 09.00.00.000000 AM	CheckUp	P004	D003
A005	05-FEB-24 01.00.00.000000 PM	Treatment	P005	D002
A006	06-FEB-24 11.00.00.000000 AM	Treatment	P006	D002

**Diagram 6.2.5: Output for Appointment Table**

```
-- Cancelling the appointment
```

```
DELETE FROM "Appointment"
```

```
WHERE appointmentID = 'A006';
```

```
-- View the Appointment Table After Update
```

```
SELECT * FROM "Appointment";
```

APPOINTMENTID	APPOINTMENTTIME	DENTALSERVICETYPE	PATIENTID	DENTISTID
A001	01-FEB-24 11.00.00.000000 AM	CheckUp	P001	D002
A002	01-FEB-24 03.00.00.000000 PM	Treatment	P002	D001
A003	02-FEB-24 09.00.00.000000 AM	Treatment	P003	D001
A004	03-FEB-24 09.00.00.000000 AM	CheckUp	P004	D003
A005	05-FEB-24 01.00.00.000000 PM	Treatment	P005	D002

**Diagram 6.2.6: Output for Updated Appointment Table**

**-- Appointment Type 1 - Treatment**

```
INSERT INTO "Treatment" (treatmentID, treatmentName,
treatmentType, servicePrice, lengthOfTime, appointmentID)
VALUES('T001', 'Teeth Cleaning', 'Preventive', 100, '60 mins',
'A002');
```

```
INSERT INTO "Treatment"(treatmentID, treatmentName,
treatmentType, servicePrice, lengthOfTime, appointmentID)
VALUES('T002', 'Crown Placement', 'Restorative', 350, '90
mins', 'A003');
```

```
INSERT INTO "Treatment"(treatmentID, treatmentName,
treatmentType, servicePrice, lengthOfTime, appointmentID)
VALUES('T003', 'Teeth Whitening', 'Cosmetic', 250, '60 mins',
'A005');
```

**-- View the Treatment Table**

```
SELECT * FROM "Treatment";
```

TREATMENTID	TREATMENTNAME	TREATMENTTYPE	SERVICEPRICE	LENGTHOFTIME	APPOINTMENTID
T001	Teeth Cleaning	Preventive	100	60 mins	A002
T002	Crown Placement	Restorative	350	90 mins	A003
T003	Teeth Whitening	Cosmetic	250	60 mins	A005

**Diagram 6.2.7: Output for Treatment Table**

```
-- Appointment Type 2 - CheckUp
INSERT INTO "CheckUp"(checkupID, servicePrice, lengthOfTime,
appointmentID)
VALUES ('C001', 80, '30 mins', 'A001');

INSERT INTO "CheckUp"(checkupID, servicePrice, lengthOfTime,
appointmentID)
VALUES('C002', 100, '45 mins', 'A004');

-- View the CheckUp Table
SELECT * FROM "CheckUp";
```

CHECKUPID	SERVICEPRICE	LENGTHOFTIME	APPOINTMENTID
C001	80	30 mins	A001
C002	100	45 mins	A004

Diagram 6.2.8: Output for Check Up Table

```
-- Scheduling the appointment
INSERT INTO "Schedule" (scheduleID, scheduleTime,
appointmentID, dentistID, receptionistID)
VALUES('S001',TO_TIMESTAMP('2024-02-01 11:00:00','YYYY-MM-DD
HH24:MI:SS')) , 'A001', 'D002', 'R001');

INSERT INTO "Schedule"(scheduleID, scheduleTime,
appointmentID, dentistID, receptionistID)
VALUES('S002',TO_TIMESTAMP('2024-02-01 15:00:00','YYYY-MM-DD
HH24:MI:SS')) , 'A002', 'D001', 'R002');

INSERT INTO "Schedule"(scheduleID, scheduleTime,
appointmentID, dentistID, receptionistID)
VALUES('S003',TO_TIMESTAMP('2024-02-02 09:00:00','YYYY-MM-DD
HH24:MI:SS')) , 'A003', 'D001', 'R001');

INSERT INTO "Schedule"(scheduleID, scheduleTime,
appointmentID, dentistID, receptionistID)
VALUES('S004', TO_TIMESTAMP('2024-02-03 09:00:00','YYYY-MM-DD
HH24:MI:SS')), 'A004', 'D003', 'R002');
```

```
INSERT INTO "Schedule"(scheduleID, scheduleTime,
appointmentID, dentistID, receptionistID)
VALUES('S005',TO_TIMESTAMP('2024-02-05 13:00:00','YYYY-MM-DD
HH24:MI:SS'), 'A005', 'D002', 'R001');
```

```
-- View the Schedule Table
```

```
SELECT * FROM "Schedule";
```

SCHEDULEID	SCHEDULETIME	DENTISTID	RECEPTIONISTID	APPOINTMENTID
S001	01-FEB-24 11.00.00.000000 AM	D002	R001	A001
S002	01-FEB-24 03.00.00.000000 PM	D001	R002	A002
S003	02-FEB-24 09.00.00.000000 AM	D001	R001	A003
S004	03-FEB-24 09.00.00.000000 AM	D003	R002	A004
S005	05-FEB-24 01.00.00.000000 PM	D002	R001	A005

**Diagram 6.2.9: Output for Schedule Table**

```
-- Updating the appointment schedule
```

```
-- If modify date
```

```
UPDATE "Schedule"
```

```
SET scheduleTime = TO_TIMESTAMP('2024-02-06
13:00:00','YYYY-MM-DD HH24:MI:SS')
```

```
WHERE scheduleID = 'S005';
```

```
-- View the Schedule Table After Update
```

```
SELECT * FROM "Schedule";
```

SCHEDULEID	SCHEDULETIME	DENTISTID	RECEPTIONISTID	APPOINTMENTID
S001	01-FEB-24 11.00.00.000000 AM	D002	R001	A001
S002	01-FEB-24 03.00.00.000000 PM	D001	R002	A002
S003	02-FEB-24 09.00.00.000000 AM	D001	R001	A003
S004	03-FEB-24 09.00.00.000000 AM	D003	R002	A004
S005	06-FEB-24 01.00.00.000000 PM	D002	R001	A005

**Diagram 6.2.10: Output for Updated Schedule Table**



```

-- Do dental record
INSERT INTO "DentalRecord" (recordID, recordDate, symptoms,
treatment, history, patientID, dentistID)
VALUES ('DR001', TO_TIMESTAMP('2024-02-01 12:00:00',
'YYYY-MM-DD HH24:MI:SS'), 'None', 'Cleaning', 'No major dental
issues', 'P001', 'D002');

INSERT INTO "DentalRecord" (recordID, recordDate, symptoms,
treatment, history, patientID, dentistID)
VALUES ('DR002', TO_TIMESTAMP('2024-02-01 17:00:00',
'YYYY-MM-DD HH24:MI:SS'), 'Sensitive teeth', 'Cleaning',
'Previous cavities', 'P002', 'D001');

INSERT INTO "DentalRecord" (recordID, recordDate, symptoms,
treatment, history, patientID, dentistID)
VALUES ('DR003', TO_TIMESTAMP('2024-02-02 11:00:00',
'YYYY-MM-DD HH24:MI:SS'), 'Toothache', 'Crown Placement',
'Previous root canal', 'P003', 'D001');

INSERT INTO "DentalRecord" (recordID, recordDate, symptoms,
treatment, history, patientID, dentistID)
VALUES ('DR004', TO_TIMESTAMP('2024-02-03 10:10:00',
'YYYY-MM-DD HH24:MI:SS'), 'None', 'Routine exam', 'No issues',
'P004', 'D003');

INSERT INTO "DentalRecord" (recordID, recordDate, symptoms,
treatment, history, patientID, dentistID)
VALUES ('DR005', TO_TIMESTAMP('2024-02-06 14:20:00',
'YYYY-MM-DD HH24:MI:SS'), 'Dull teeth', 'Teeth whitening',
'Previous whitening treatment', 'P005', 'D002');

-- View the DentalRecord Table
SELECT * FROM "DentalRecord";

```

RECORDID	RECORDDATE	SYMPTOMS	TREATMENT	HISTORY	PATIENTID	DENTISTID
DR001	01-FEB-24 12.00.00.000000 PM	None	Cleaning	No major dental issues	P001	D002
DR002	01-FEB-24 05.00.00.000000 PM	Sensitive teeth	Cleaning	Previous cavities	P002	D001
DR003	02-FEB-24 11.00.00.000000 AM	Toothache	Crown Placement	Previous root canal	P003	D001
DR004	03-FEB-24 10.10.00.000000 AM	None	Routine exam	No issues	P004	D003
DR005	06-FEB-24 02.20.00.000000 PM	Dull teeth	Teeth whitening	Previous whitening treatment	P005	D002

**Diagram 6.2.11: Output for Dental Record Table**

-- Make payment

```
INSERT INTO "Payment"
(invoiceNo, paymentDate, paymentFees, paymentStatus, recordID,
receptionistID)
VALUES ('I001',TO_TIMESTAMP('2024-02-01 11:50:00','YYYY-MM-DD
HH24:MI:SS')) , 80, 'Paid', 'DR001', 'R001');
```

```
INSERT INTO "Payment"
(invoiceNo, paymentDate, paymentFees, paymentStatus, recordID,
receptionistID)
VALUES ('I002',TO_TIMESTAMP('2024-02-01 16:50:00','YYYY-MM-DD
HH24:MI:SS')) , 100, 'Paid', 'DR002', 'R002');
```

```
INSERT INTO "Payment"
(invoiceNo, paymentDate, paymentFees, paymentStatus, recordID,
receptionistID)
VALUES ('I003',TO_TIMESTAMP('2024-02-02 10:50:00','YYYY-MM-DD
HH24:MI:SS')) , 350, 'Paid', 'DR003', 'R001');
```

```
INSERT INTO "Payment"
(invoiceNo, paymentDate, paymentFees, paymentStatus, recordID,
receptionistID)
VALUES ('I004',TO_TIMESTAMP('2024-02-03 10:00:00','YYYY-MM-DD
HH24:MI:SS')) , 100, 'Paid', 'DR004', 'R002');
```

```
INSERT INTO "Payment"
(invoiceNo, paymentDate, paymentFees, paymentStatus, recordID,
receptionistID)
VALUES ('I005',TO_TIMESTAMP('2024-02-06 14:10:00','YYYY-MM-DD
HH24:MI:SS')) , 250, 'Paid', 'DR005', 'R001');
```

-- View the Payment Table

```
SELECT * FROM "Payment";
```

INVOICENO	PAYMENTDATE	PAYMENTFEES	PAYMENTSTATUS	RECORDID	RECEPTIONISTID
I001	01-FEB-24 11.50.00.000000 AM	80	Paid	DR001	R001
I002	01-FEB-24 04.50.00.000000 PM	100	Paid	DR002	R002
I003	02-FEB-24 10.50.00.000000 AM	350	Paid	DR003	R001
I004	03-FEB-24 10.00.00.000000 AM	100	Paid	DR004	R002
I005	06-FEB-24 02.10.00.000000 PM	250	Paid	DR005	R001

**Diagram 6.2.12: Output for Payment Table**

**-- Cancel the payment**

**UPDATE "Payment"**

**SET paymentStatus = 'Cancelled'**

**WHERE invoiceNo = 'I002';**

**-- View the Payment Table After Update**

**SELECT \* FROM "Payment";**

INVOICENO	PAYMENTDATE	PAYMENTFEES	PAYMENTSTATUS	RECORDID	RECEPTIONISTID
I001	01-FEB-24 11.50.00.000000 AM	80	Paid	DR001	R001
I002	01-FEB-24 04.50.00.000000 PM	100	Cancelled	DR002	R002
I003	02-FEB-24 10.50.00.000000 AM	350	Paid	DR003	R001
I004	03-FEB-24 10.00.00.000000 AM	100	Paid	DR004	R002
I005	06-FEB-24 02.10.00.000000 PM	250	Paid	DR005	R001

**Diagram 6.2.12: Output for Updated Payment Table**

**DELETE FROM "Payment"**

**WHERE invoiceNo = 'I002';**

**-- View the Payment Table After Delete**

**SELECT \* FROM "Payment";**

INVOICENO	PAYMENTDATE	PAYMENTFEES	PAYMENTSTATUS	RECORDID	RECEPTIONISTID
I001	01-FEB-24 11.50.00.000000 AM	80	Paid	DR001	R001
I003	02-FEB-24 10.50.00.000000 AM	350	Paid	DR003	R001
I004	03-FEB-24 10.00.00.000000 AM	100	Paid	DR004	R002
I005	06-FEB-24 02.10.00.000000 PM	250	Paid	DR005	R001

**Diagram 6.2.13: Output for Updated Payment Table**

**-- Online Transfer Payment**

```
INSERT INTO "Online Transfer"(onlineTransferID, invoiceNo)
VALUES ('T001', 'I001');
```

```
INSERT INTO "Online Transfer"(onlineTransferID, invoiceNo)
VALUES ('T002', 'I003');
```

**-- View the Online Transfer Payment Table**

```
SELECT * FROM "Online Transfer";
```

ONLINETRANSFERID	INVOICENO
T001	I001
T002	I003

**Diagram 6.2.14: Output for Online Transfer Table**

**-- Cash Payment**

```
INSERT INTO "Cash"(cashID, invoiceNo)
VALUES('CH001', 'I004');
```

```
INSERT INTO "Cash"(cashID, invoiceNo)
VALUES('CH002', 'I005');
```

**-- View the Cash Payment Table**

```
SELECT * FROM "Cash";
```

CASHID	INVOICENO
CH001	I004
CH002	I005

**Diagram 6.2.15: Output for Cash Table**

```
-- Change Cash to Online Transfer
-- Delete the existing cash payment
DELETE FROM "Cash"
where cashID = 'CH002';

-- Insert a new record into Online Transfer for same invoice
INSERT INTO "Online Transfer"(onlineTransferID,invoiceNo)
VALUES('T003','I005');
-- View the Cash and Online Transfer Payment Table
SELECT * FROM "Cash";
SELECT * FROM "Online Transfer";
```

CASHID	INVOICENO
CH001	I004

ONLINETRANSFERID	INVOICENO
T001	I001
T002	I003
T003	I005

**Diagram 6.2.16: Output for Cash and Online Transfer Table**

```

-- Update Patient Name
UPDATE "User"
SET firstName = 'Vivian', lastName = 'Tang'
WHERE userID = 'U005';

-- View the User Table
SELECT * FROM "User";

```

USERID	FIRSTNAME	LASTNAME	GENDER	CONTACTNO	EMAIL
U001	Denies	Wong	Female	01139762466	denies0516@gmail.com
U002	Joyce	Lee	Female	0167710851	joyce031019@yahoo.com
U003	Titanic	Tan	Female	01111413231	ttf0406@yahoo.com
U004	Nancy	Lim	Female	01137707837	enting0601@yahoo.com
U005	Vivian	Tang	Female	0127376115	zyiibusytoeat@gmail.com
U006	Koh	Jing Yi	Male	01110741160	jy030531@gmail.com
U007	John	Doe	Male	01169382348	john.doe@hotmail.com
U008	Alice	Smith	Female	01128962522	alice.smith@gmail.com
U009	Bob	Johnson	Male	0124141691	bob.johnson@yahoo.com
U010	Emily	Johnson	Female	0145678901	emily.johnson@yahoo.com
U011	David	Miller	Male	0156789012	david.miller@hotmail.com

**Diagram 6.2.17: Output for Updated User Table**

```

-- Concatenate the dentalrecord with dentist name
SELECT
    dr.recordID,
    dr.recordDate,
    dr.symptoms,
    dr.treatment,
    dr.history,
    dr.patientID,
    u.firstName || ' ' || u.lastName AS dentistName
FROM "DentalRecord" dr
INNER JOIN "Dentist" d ON dr.dentistID = d.dentistID
INNER JOIN "User" u ON d.userID = u.userID;

```

RECORDID	RECORDDATE	SYMPTOMS	TREATMENT	HISTORY	PATIENTID	DENTISTNAME
DR001	01-FEB-24 12.00.00.000000 PM	None	Cleaning	No major dental issues	P001	Alice Smith
DR002	01-FEB-24 05.00.00.000000 PM	Sensitive teeth	Cleaning	Previous cavities	P002	John Doe
DR003	02-FEB-24 11.00.00.000000 AM	Toothache	Crown Placement	Previous root canal	P003	John Doe
DR004	03-FEB-24 10.10.00.000000 AM	None	Routine exam	No issues	P004	Bob Johnson
DR005	06-FEB-24 02.20.00.000000 PM	Dull teeth	Teeth whitening	Previous whitening treatment	P005	Alice Smith

**Diagram 6.2.18: Output for Dental Record Joining User Table**

```
-- View the dental record with dentist John Doe
SELECT
    dr.recordID,
    dr.recordDate,
    dr.symptoms,
    dr.treatment,
    dr.history,
    dr.patientID,
    u.firstName || ' ' || u.lastName AS dentistName
FROM "DentalRecord" dr
INNER JOIN "Dentist" d ON dr.dentistID = d.dentistID
INNER JOIN "User" u ON d.userID = u.userID
WHERE u.firstName || ' ' || u.lastName = 'John Doe';
```

RECORDID	RECORDDATE	SYMPTOMS	TREATMENT	HISTORY	PATIENTID	DENTISTNAME
DR002	01-FEB-24 05.00.00.000000 PM	Sensitive teeth	Cleaning	Previous cavities	P002	John Doe
DR003	02-FEB-24 11.00.00.000000 AM	Toothache	Crown Placement	Previous root canal	P003	John Doe

**Diagram 6.2.19: Output for Dental Record Joining User Table (John Doe)**

```
-- View the dental record with dentist Alice Smith
SELECT
    dr.recordID,
    dr.recordDate,
    dr.symptoms,
    dr.treatment,
    dr.history,
    dr.patientID,
    u.firstName || ' ' || u.lastName AS dentistName
```

```

FROM "DentalRecord" dr
INNER JOIN "Dentist" d ON dr.dentistID = d.dentistID
INNER JOIN "User" u ON d.userID = u.userID
WHERE u.firstName || ' ' || u.lastName = 'Alice Smith';

```

RECORDID	RECORDDATE	SYMPTOMS	TREATMENT	HISTORY	PATIENTID	DENTISTNAME
DR001	01-FEB-24 12.00.00.000000 PM	None	Cleaning	No major dental issues	P001	Alice Smith
DR005	06-FEB-24 02.20.00.000000 PM	Dull teeth	Teeth whitening	Previous whitening treatment	P005	Alice Smith

**Diagram 6.2.19: Output for Dental Record Joining User Table (Alice Smith)**

```

-- View the dental record with dentist Bob Johnson
SELECT
  dr.recordID,
  dr.recordDate,
  dr.symptoms,
  dr.treatment,
  dr.history,
  dr.patientID,
  u.firstName || ' ' || u.lastName AS dentistName
FROM "DentalRecord" dr
INNER JOIN "Dentist" d ON dr.dentistID = d.dentistID
INNER JOIN "User" u ON d.userID = u.userID
WHERE u.firstName || ' ' || u.lastName = 'Bob Johnson';

```

RECORDID	RECORDDATE	SYMPTOMS	TREATMENT	HISTORY	PATIENTID	DENTISTNAME
DR004	03-FEB-24 10.10.00.000000 AM	None	Routine exam	No issues	P004	Bob Johnson

**Diagram 6.2.20: Output for Dental Record Joining User Table (Bob Johnson)**



## 6.3 Test Query

### 6.3.1 View User Table

**-- View User Table**

```
SELECT *  
FROM "User";
```

USERID	FIRSTNAME	LASTNAME	GENDER	CONTACTNO	EMAIL
U001	Denies	Wong	Female	01139762466	denies0516@gmail.com
U002	Joyce	Lee	Female	0167710851	joyce031019@yahoo.com
U003	Titanic	Tan	Female	01111413231	ttf0406@yahoo.com
U004	Nancy	Lim	Female	01137707837	enting0601@yahoo.com
U005	Vivian	Tang	Female	0127376115	zyiibusytoeat@gmail.com
U006	Koh	Jing Yi	Male	01110741160	jy030531@gmail.com
U007	John	Doe	Male	01169382348	john.doe@hotmail.com
U008	Alice	Smith	Female	01128962522	alice.smith@gmail.com
U009	Bob	Johnson	Male	0124141691	bob.johnson@yahoo.com
U010	Emily	Johnson	Female	0145678901	emily.johnson@yahoo.com
U011	David	Miller	Male	0156789012	david.miller@hotmail.com

### 6.3.2 View Patient Table

**-- View Patient Table**

```
SELECT *  
FROM "Patient";
```

PATIENTID	DATEOFBIRTH	STREET	CITY	POSTCODE	USERID
P001	16-MAY-03	20,jalan lawa 16	johor	81800	U001
P002	19-OCT-03	60,jalan pesona 16	johor	81800	U002
P003	06-APR-03	12,jalan danau 27	johor	81800	U003
P004	01-JUN-03	blok L,jalan gaya 11	johor	81800	U004
P005	05-SEP-03	31,jalan harmonium 30/5	johor	81800	U005
P006	31-MAY-03	25,jalan cemerlang 6	johor	81800	U006

### 6.3.3 View Dentist Table

-- View Dentist Table

```
SELECT *
FROM "Dentist";
```

DENTISTID	QUALIFICATION	SPECIALIZATION	USERID
D001	DDS	Orthodontics	U007
D002	DMD	Endodontics	U008
D003	BDS	General Dentistry	U009

### 6.3.4 View Receptionist Table

-- View Receptionist Table

```
SELECT *
FROM "Receptionist";
```

RECEPTIONISTID	QUALIFICATION	USERID
R001	Reception Management	U010
R002	Office Administration	U011

### 6.3.5 View Appointment Table

-- View Appointment Table

```
SELECT *
FROM "Appointment";
```

APPOINTMENTID	APPOINTMENTTIME	DENTALSERVICETYPE	PATIENTID	DENTISTID
A001	01-FEB-24 11.00.00.000000 AM	CheckUp	P001	D002
A002	01-FEB-24 03.00.00.000000 PM	Treatment	P002	D001
A003	02-FEB-24 09.00.00.000000 AM	Treatment	P003	D001
A004	03-FEB-24 09.00.00.000000 AM	CheckUp	P004	D003
A005	05-FEB-24 01.00.00.000000 PM	Treatment	P005	D002

### 6.3.6 View Treatment Table

-- View Treatment Table

```
SELECT *
FROM "Treatment";
```

TREATMENTID	TREATMENTNAME	TREATMENTTYPE	SERVICEPRICE	LENGTHOFTIME	APPOINTMENTID
T001	Teeth Cleaning	Preventive	100	60 mins	A002
T002	Crown Placement	Restorative	350	90 mins	A003
T003	Teeth Whitening	Cosmetic	250	60 mins	A005

### 6.3.7 View Check Up Table

-- View Check Up Table

```
SELECT *  
FROM "CheckUp";
```

CHECKUPID	SERVICEPRICE	LENGTHOFTIME	APPOINTMENTID
C001	80	30 mins	A001
C002	100	45 mins	A004

### 6.3.8 View Schedule Table

-- View Schedule Table

```
SELECT *  
FROM "Schedule";
```

SCHEDULEID	SCHEDULETIME	DENTISTID	RECEPTIONISTID	APPOINTMENTID
S001	01-FEB-24 11.00.00.000000 AM	D002	R001	A001
S002	01-FEB-24 03.00.00.000000 PM	D001	R002	A002
S003	02-FEB-24 09.00.00.000000 AM	D001	R001	A003
S004	03-FEB-24 09.00.00.000000 AM	D003	R002	A004
S005	06-FEB-24 01.00.00.000000 PM	D002	R001	A005

### 6.3.9 View Dental Record Table

-- View Dental Record Table

```
SELECT *  
FROM "DentalRecord";
```

RECORDID	RECORDDATE	SYMPTOMS	TREATMENT	HISTORY	PATIENTID	DENTISTID
DR001	01-FEB-24 12.00.00.000000 PM	None	Cleaning	No major dental issues	P001	D002
DR002	01-FEB-24 05.00.00.000000 PM	Sensitive teeth	Cleaning	Previous cavities	P002	D001
DR003	02-FEB-24 11.00.00.000000 AM	Toothache	Crown Placement	Previous root canal	P003	D001
DR004	03-FEB-24 10.10.00.000000 AM	None	Routine exam	No issues	P004	D003
DR005	06-FEB-24 02.20.00.000000 PM	Dull teeth	Teeth whitening	Previous whitening treatment	P005	D002

### 6.3.10 View Payment Table

-- View Payment Table

```
SELECT *
FROM "Payment";
```

INVOICENO	PAYMENTDATE	PAYMENTFEES	PAYMENTSTATUS	RECORDID	RECEPTIONISTID
I001	01-FEB-24 11.50.00.000000 AM	80	Paid	DR001	R001
I003	02-FEB-24 10.50.00.000000 AM	350	Paid	DR003	R001
I004	03-FEB-24 10.00.00.000000 AM	100	Paid	DR004	R002
I005	06-FEB-24 02.10.00.000000 PM	250	Paid	DR005	R001

### 6.3.11 View Online Transfer Table

-- View Online Transfer Table

```
SELECT *
FROM "Online Transfer";
```

ONLINETRANSFERID	INVOICENO
T001	I001
T002	I003
T003	I005

### 6.3.12 View Cash Table

-- View Cash Table

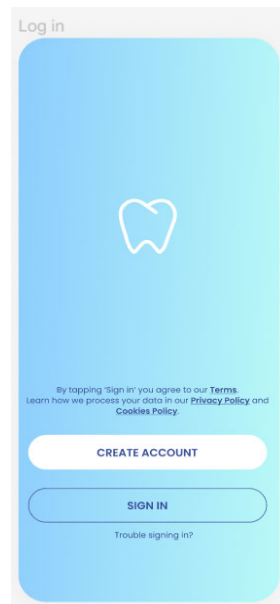
SELECT \*

FROM "Cash";

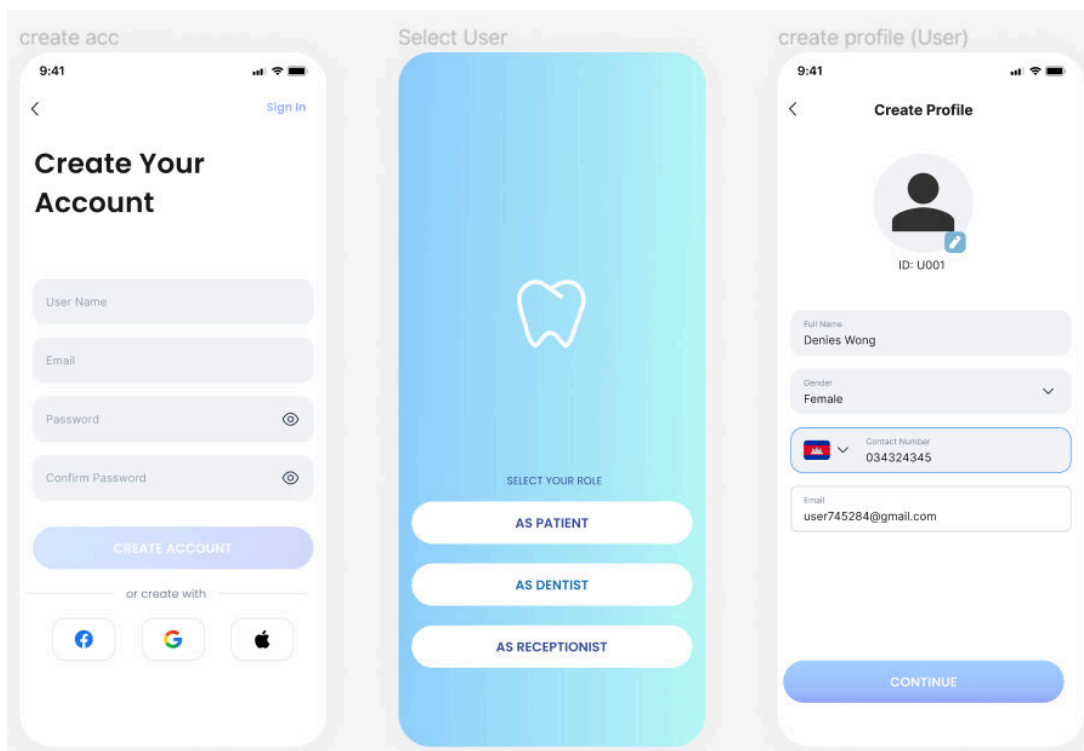
CASHID	INVOICENO
CH001	I004

## 7.0 Interface

### 7.1 User Login



### 7.2 User Create Account and Profile



### 7.3 Profile of each user (Patient, Dentist, Receptionist)

The image displays three mobile app screens, each titled 'Edit Profile' for a different user role: Patient, Dentist, and Receptionist. All screens show a time of 9:41 and a battery level of 100%.

- Patient profile:** ID: U001 (P001). Fields include Full Name (Denies Wong), Gender (Female), Contact Number (034324345), Email (denieswong@gmail.com), Date of Birth (16/May/2003), and Address (street | city | postcode). A blue 'CONTINUE' button is at the bottom.
- Dentist profile:** ID: U007 (D001). Fields include Full Name (John Doe), Contact Number (058435928), Email (john DOE@gmail.com), Specialization (Select level of specialization), and Qualification (Drop here to attach or upload, Max size: 50B). A blue 'CONTINUE' button is at the bottom.
- Receptionist profile:** ID: U010 (R001). Fields include Full Name (Emily Johnson), Gender (Female), Contact Number (034324345), Email (emilyjohnson@gmail.com), and Qualification (Drop here to attach or upload, Max size: 50B). A blue 'CONTINUE' button is at the bottom.

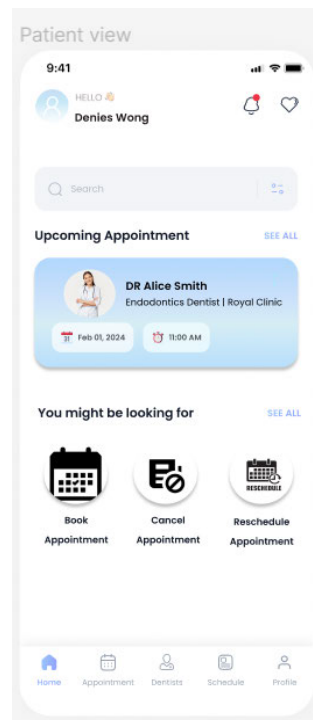
### 7.4 User Sign In

The image displays two mobile app screens related to user authentication.

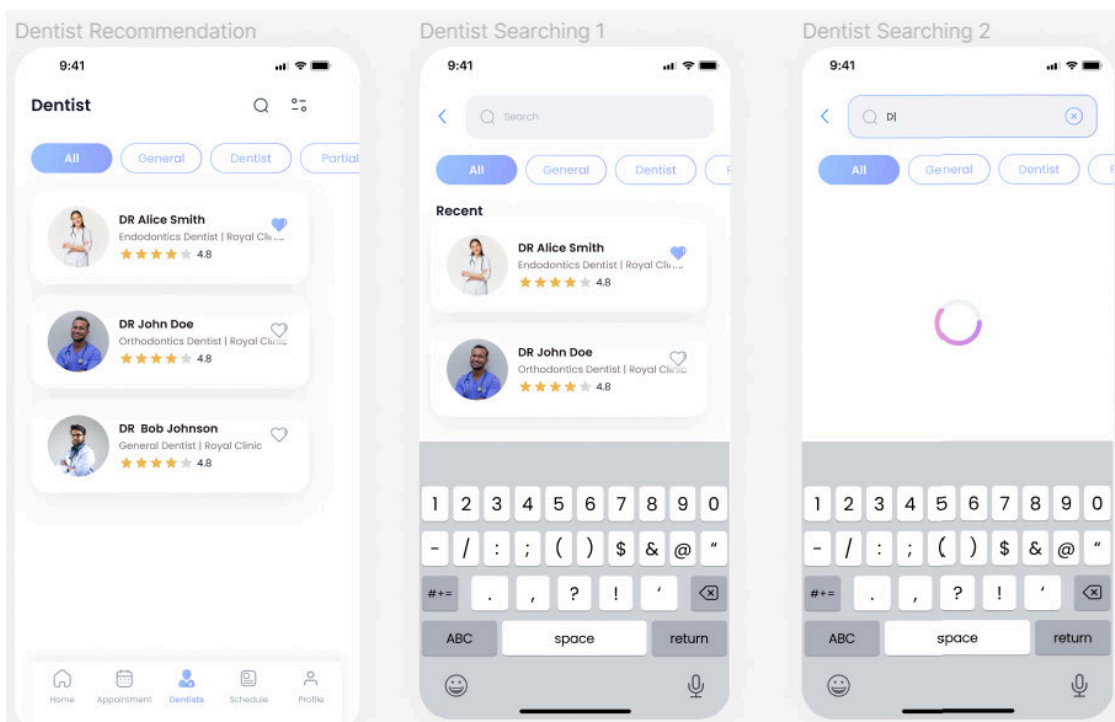
- Sign in:** Features a 'Create Account' link, a 'Sign In to Your Account' title, and input fields for Email and Password. A blue 'SIGN IN' button is present, along with a 'Forgot the password?' link and social login options for Facebook, Google, and Apple.
- Select User:** Features a large tooth icon and a 'SELECT YOUR ROLE' prompt. Three buttons are available: 'AS PATIENT', 'AS DENTIST', and 'AS RECEPTIONIST'.

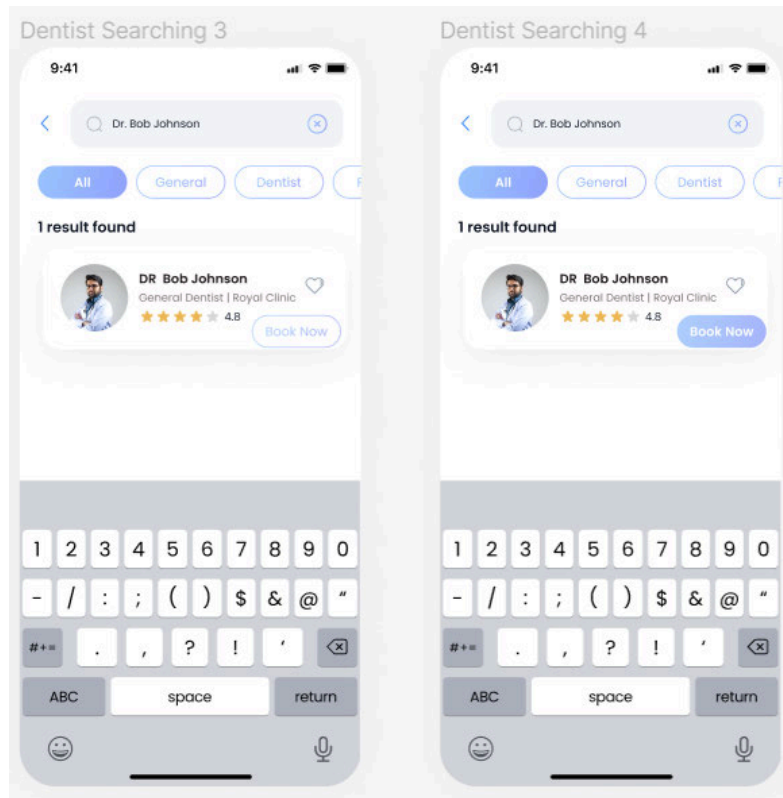


## 7.5 Patient View (Home Page)

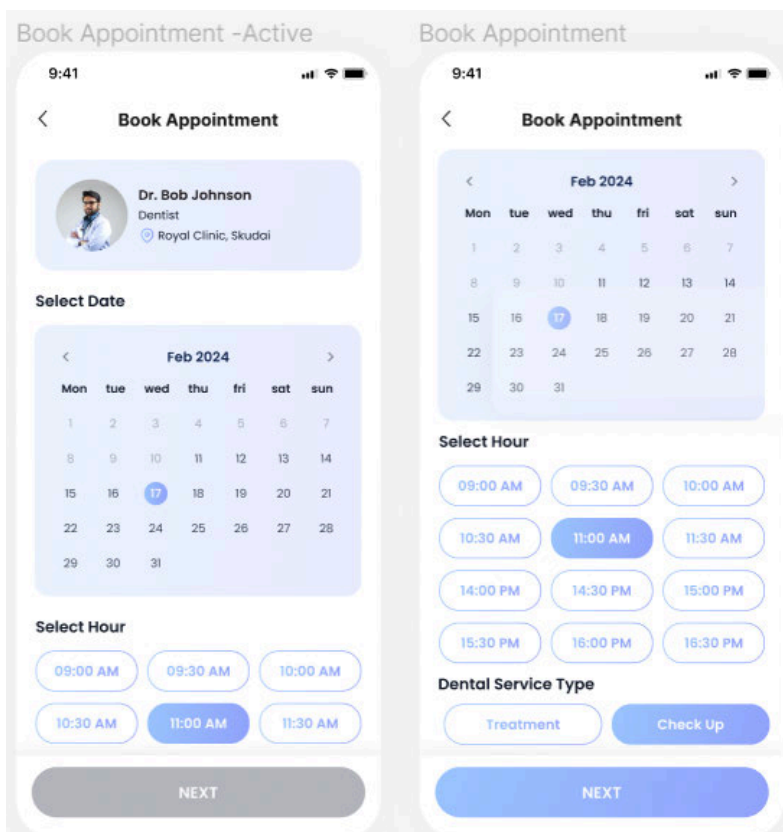


## 7.6 Searching Dentist (Patient View)

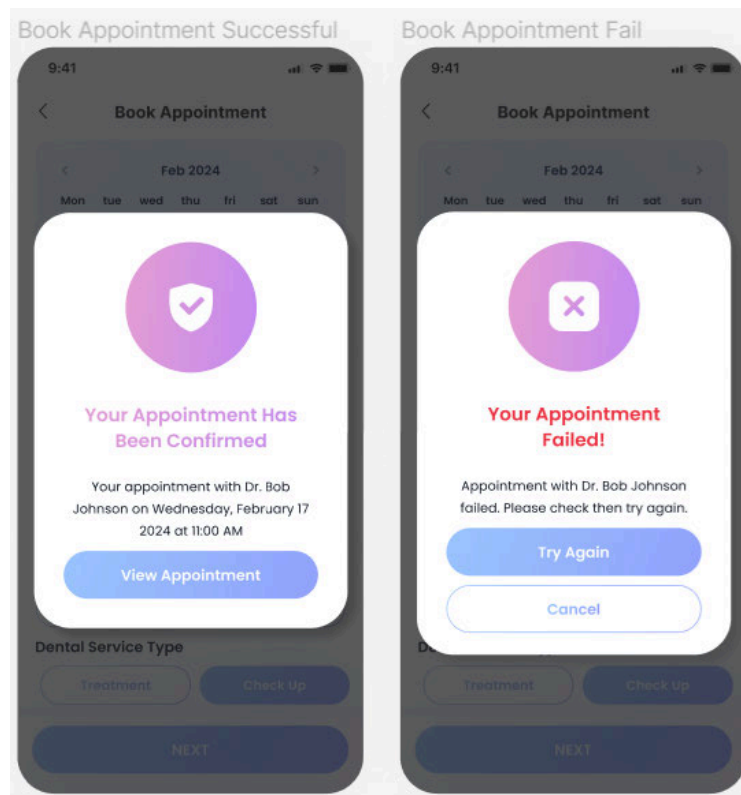




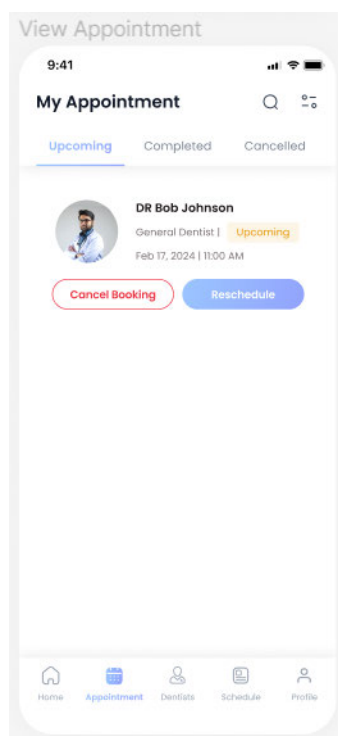
## 7.7 Booking Appointment



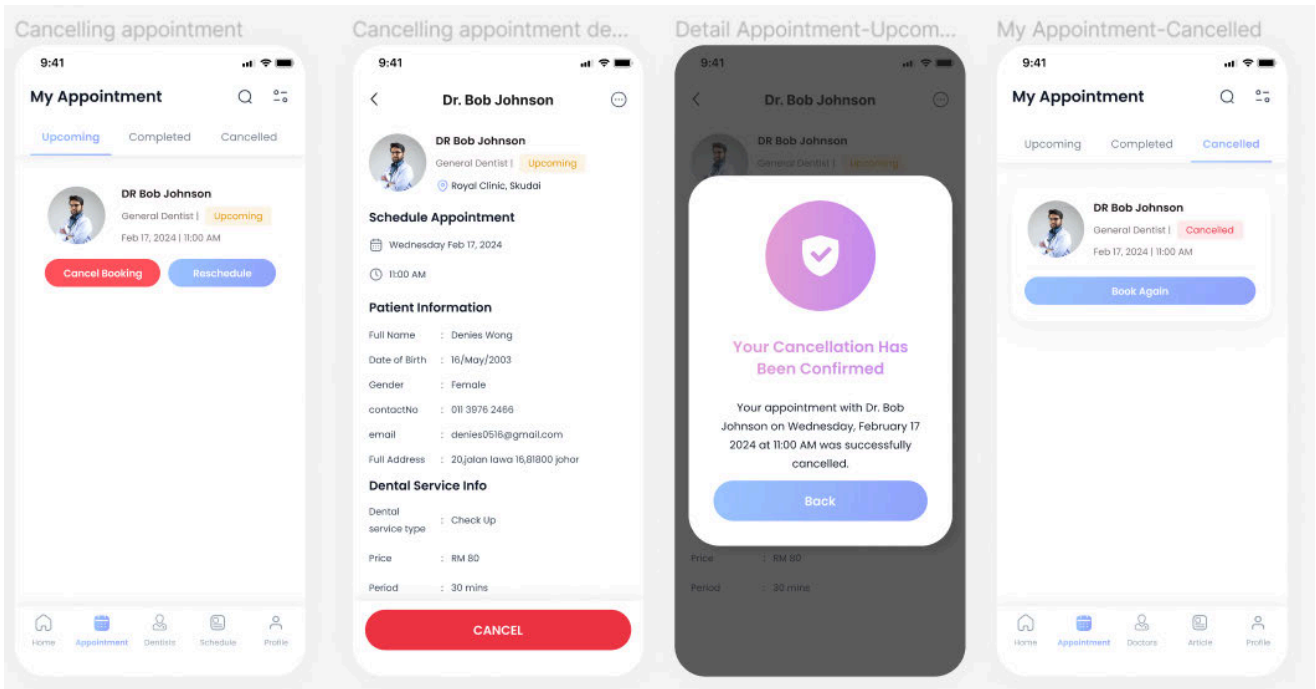
## 7.8 Booking Appointment (Success / Fail)



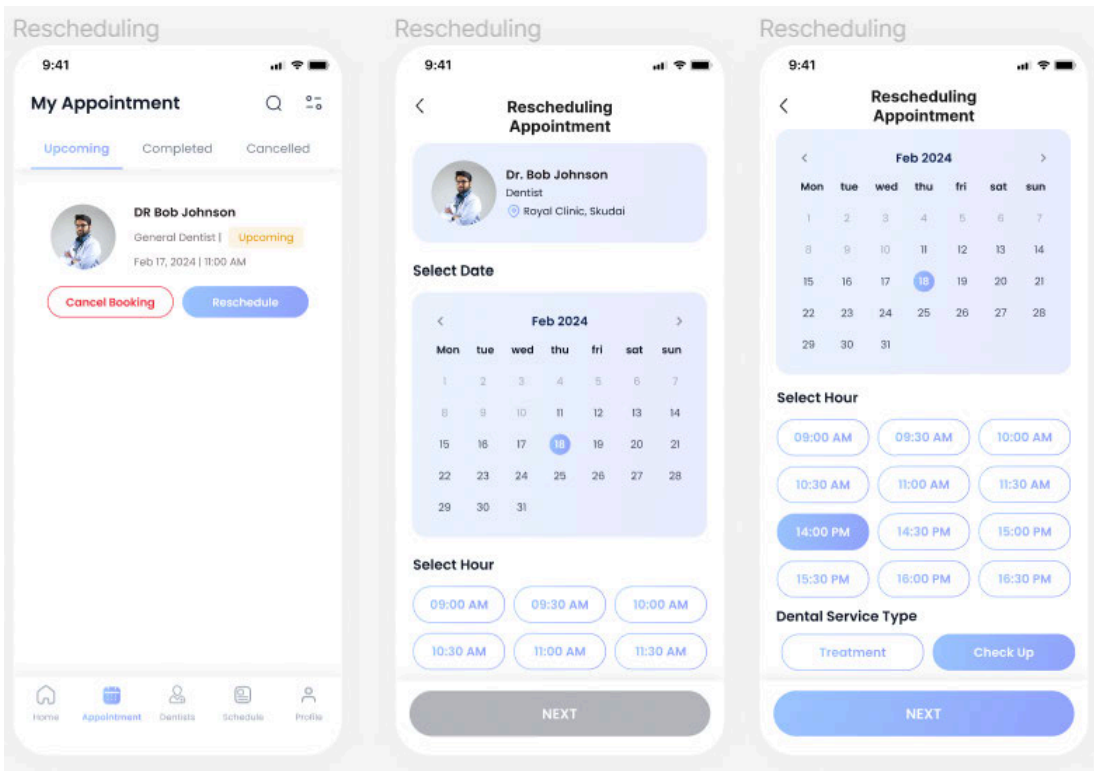
## 7.9 Viewing Appointment

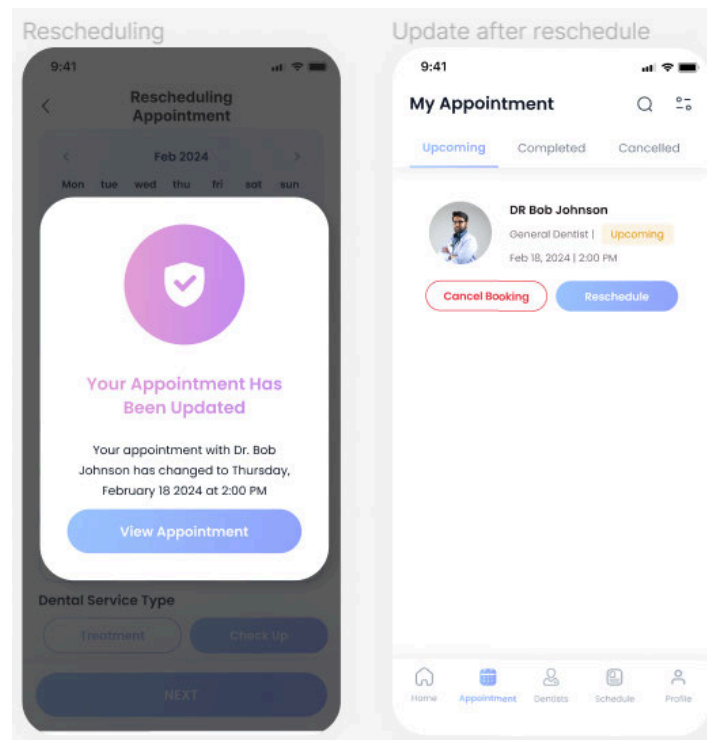


7.10 Canceling Appointment

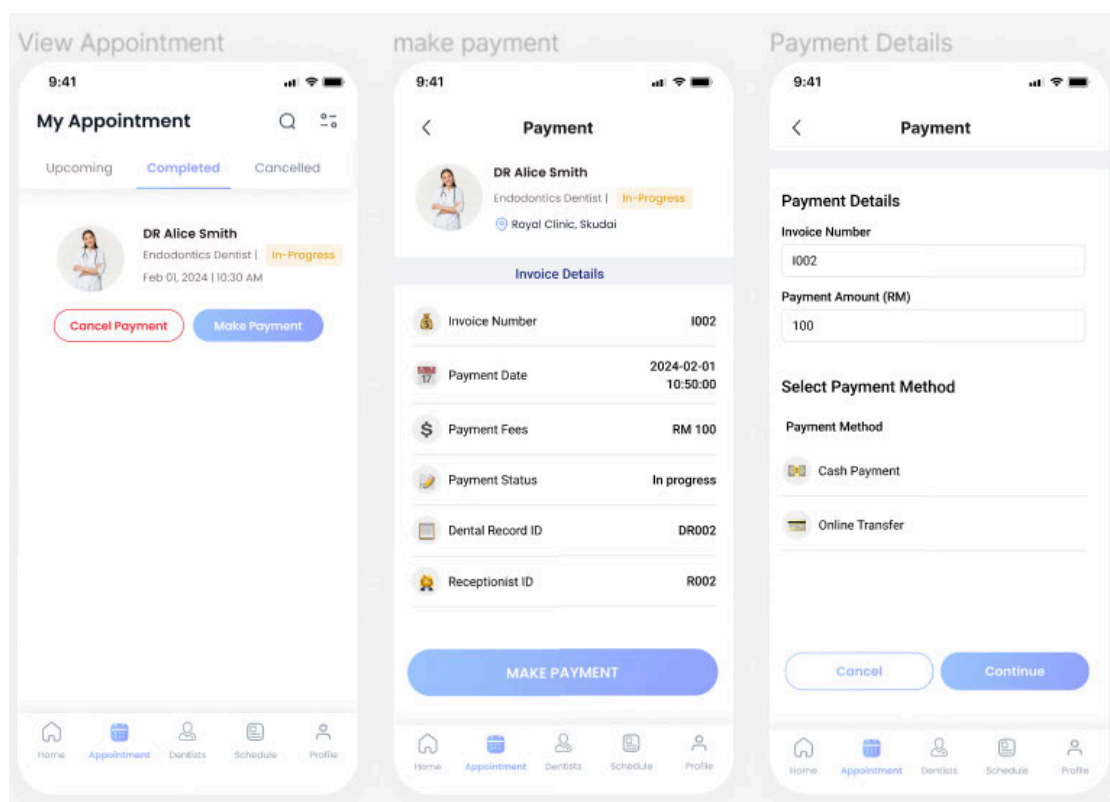


7.11 Rescheduling Appointment

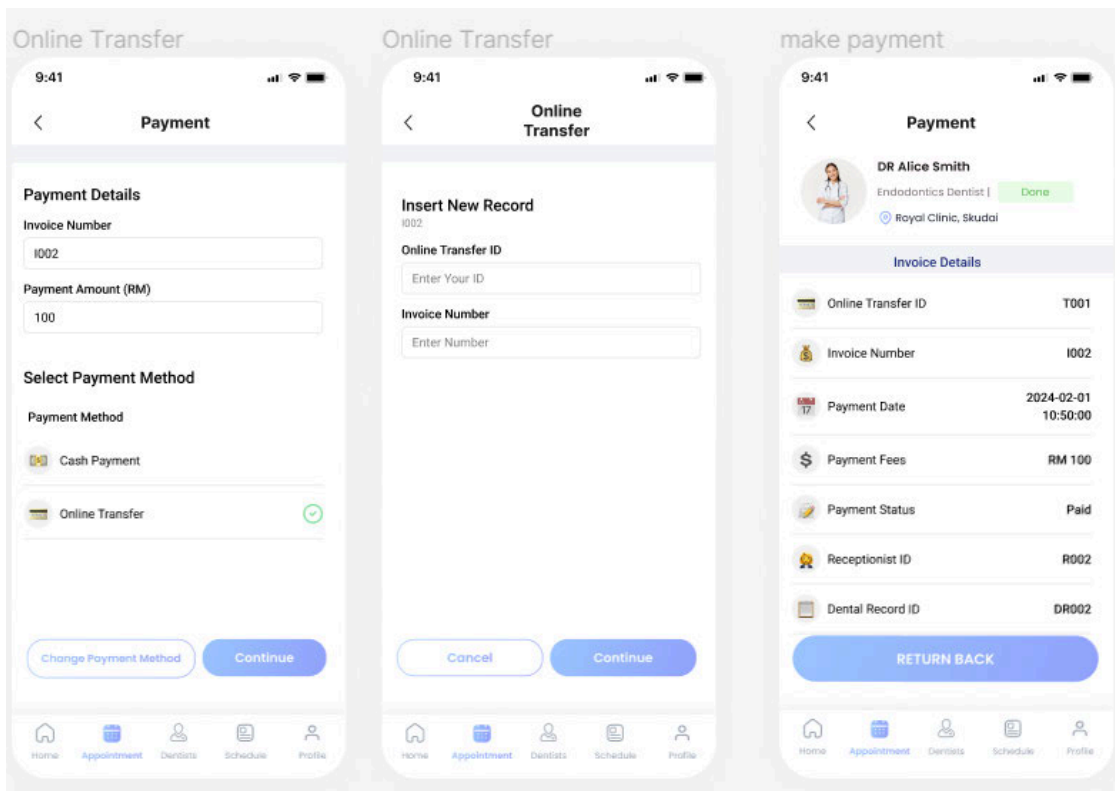




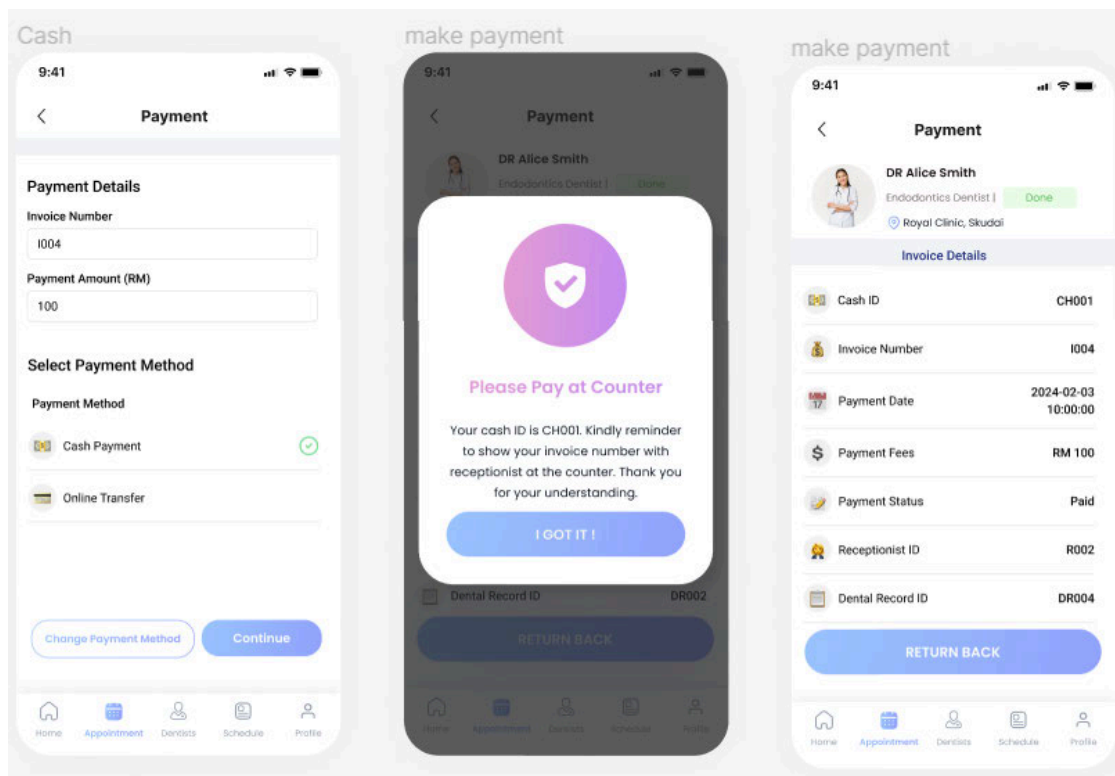
## 7.12 Making Payment



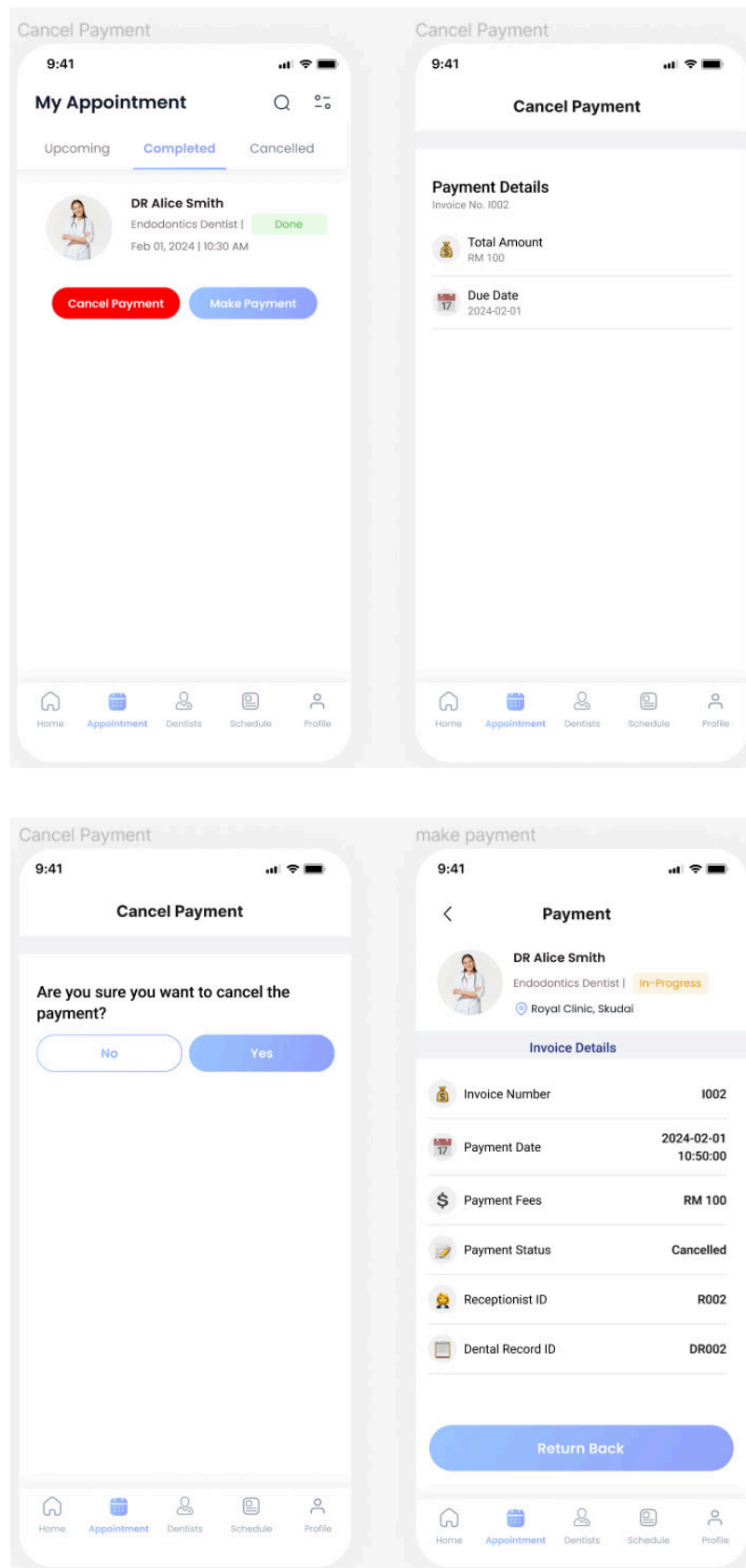
### 7.13 Making Online Transfer Payment



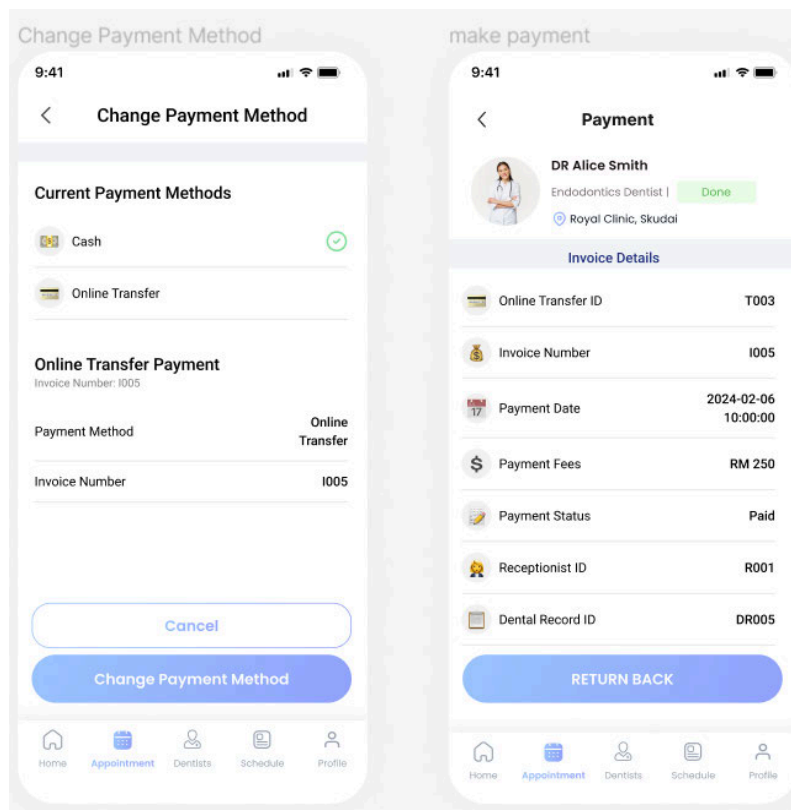
### 7.14 Making Cash Payment



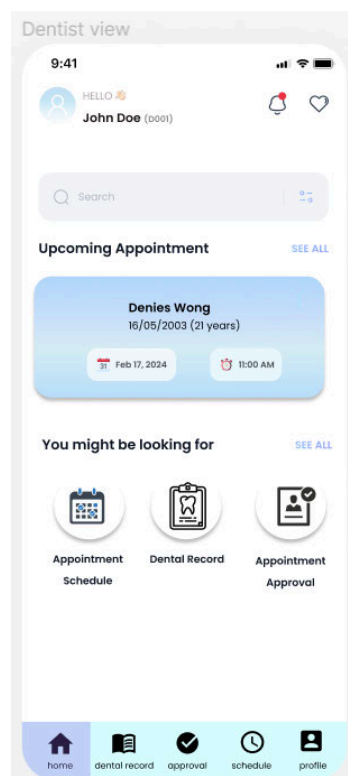
## 7.15 Canceling Payment



## 7.15 Changing Payment Method

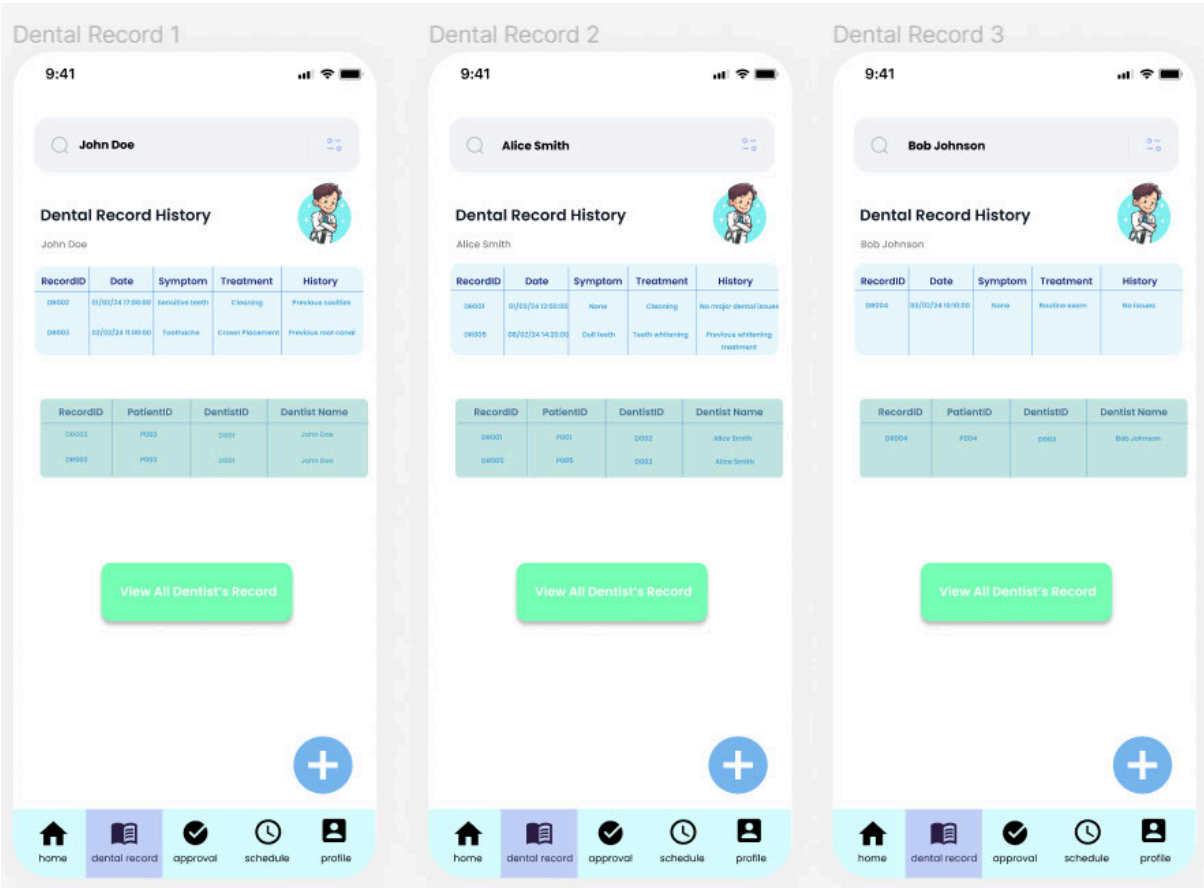


## 7.15 Dentist View

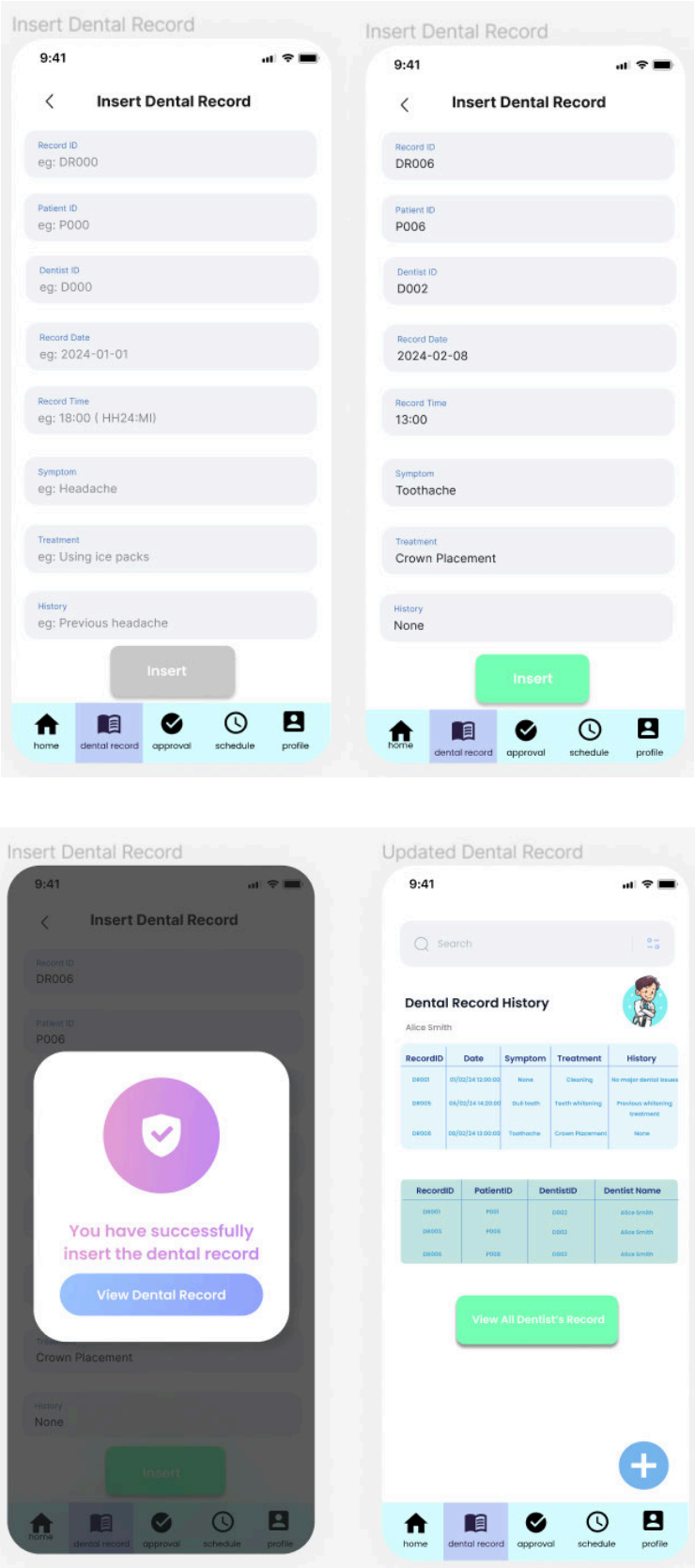




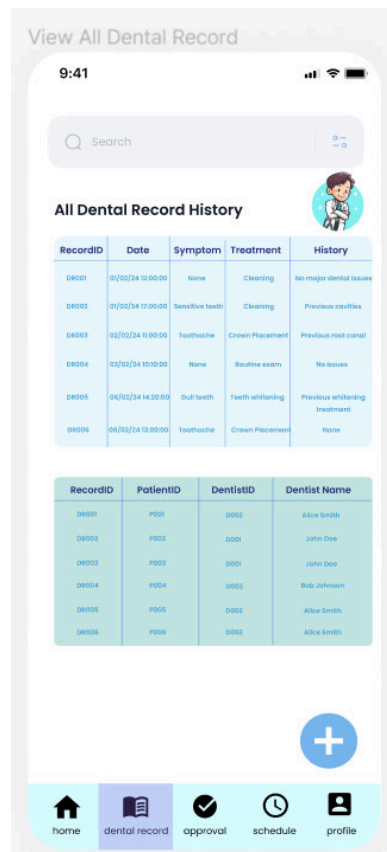
7.16 Dental Record for each dentist



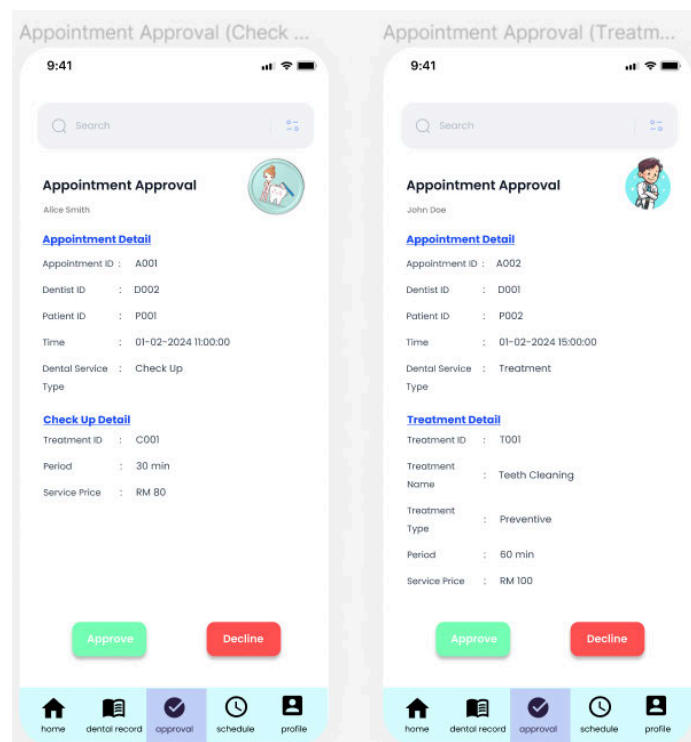
7.17 Insert Dental Record

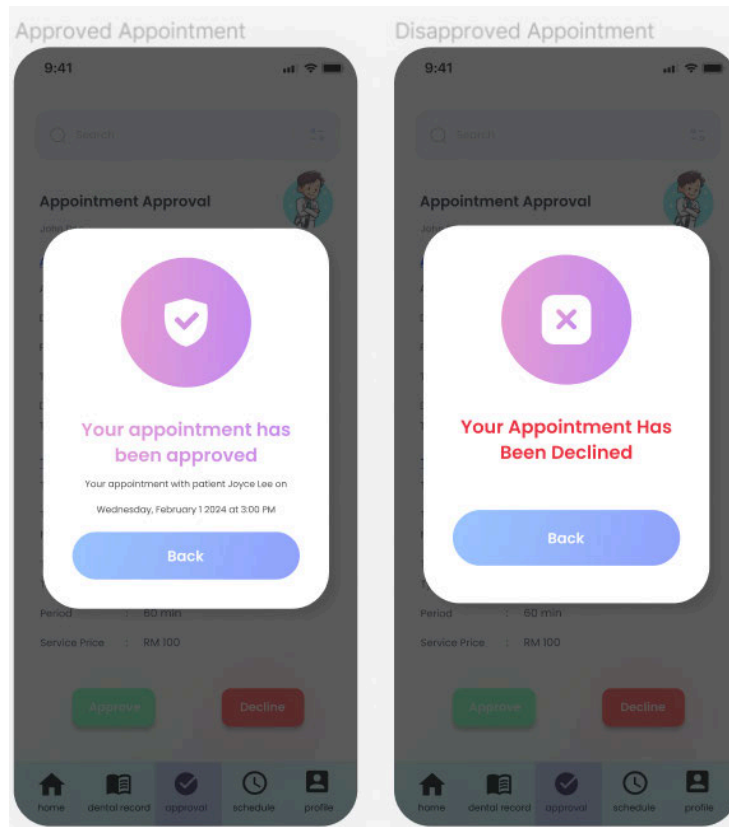


## 7.18 Viewing Dental Record

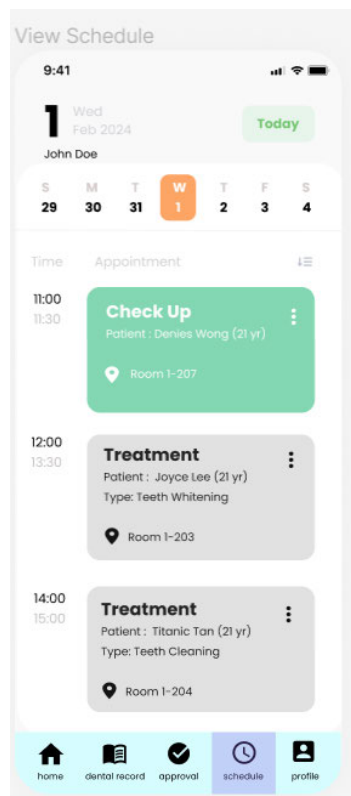


## 7.19 Approving Appointment





## 7.20 Viewing Schedule



## 8.0 Summary

During this phase, we meticulously revamped our Conceptual Entity Relationship Diagram (ERD) into a Logical ERD. We adapted it to the differences in efficient dental appointment management. This included removing non-relational features and refining relationships to ensure that our database adhered strictly to relational principles.

We also completed the relational schema, which transformed each entity such as patient, appointment, dentist, receptionist and payment into tables with explicit attributes and primary keys. With the premise of meeting the Boyce Codd Normative Format (BCNF), we perform normalization to eliminate redundancies and dependencies to ensure the integrity of the patient information, appointment details, and related transactions.

The creation of the final logical ERD is a visual proof of our refinement of the database structure. The ERD encapsulates the relationships between entities and provides a clear roadmap for efficient data retrieval and management. At the same time, our data dictionary was updated as necessary to align with the normalized structure and ensure consistency and accuracy of the stored information.

In conclusion, we learned to translate conceptual ideas into well-organized and normalized database structures during this phase. This lays the foundation for an efficient and reliable dental appointment management system. Our solution for the project would streamline the scheduling process, minimize wait times, and enhance the overall experience of dental clinic patients.