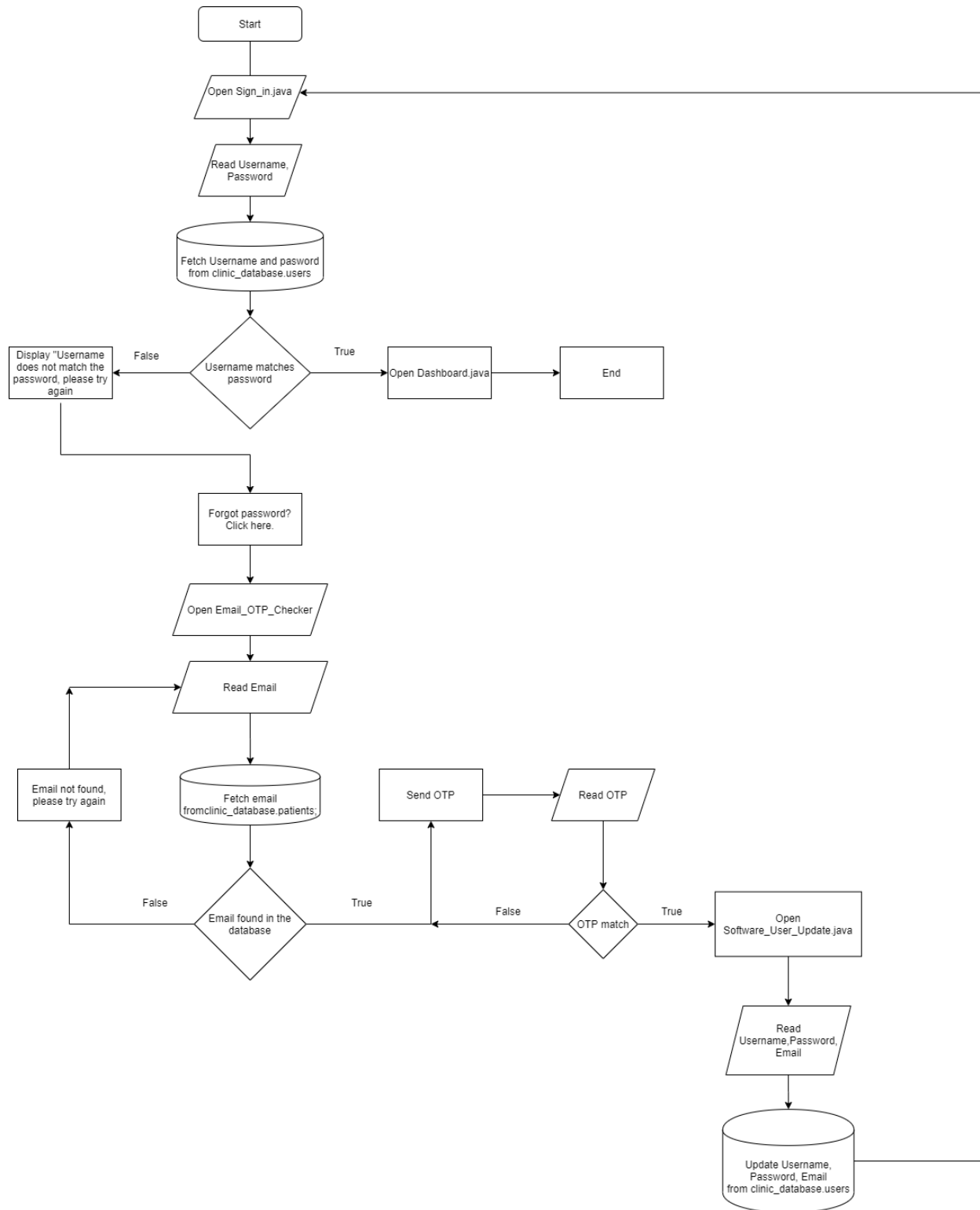
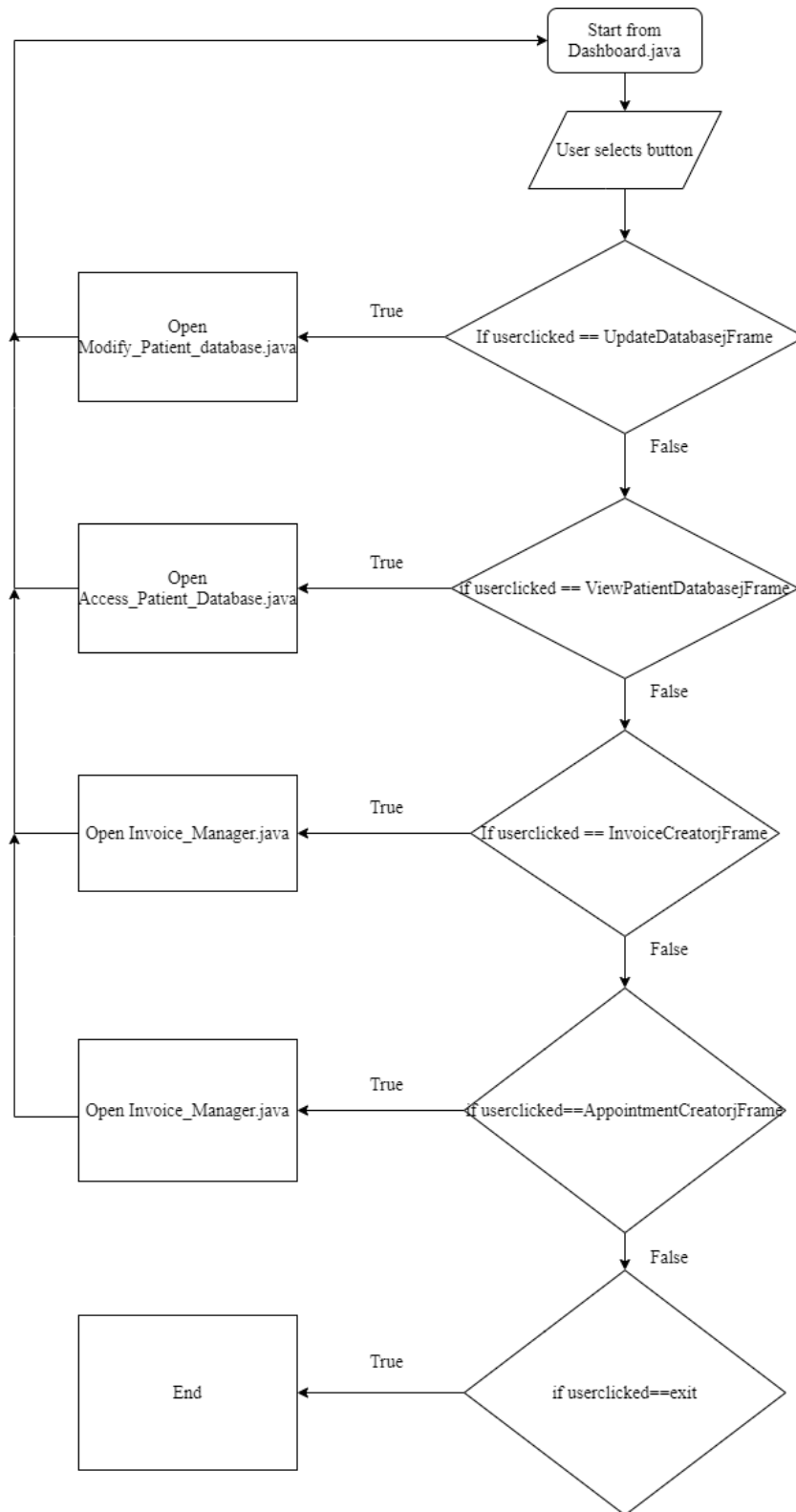


# Flowcharts

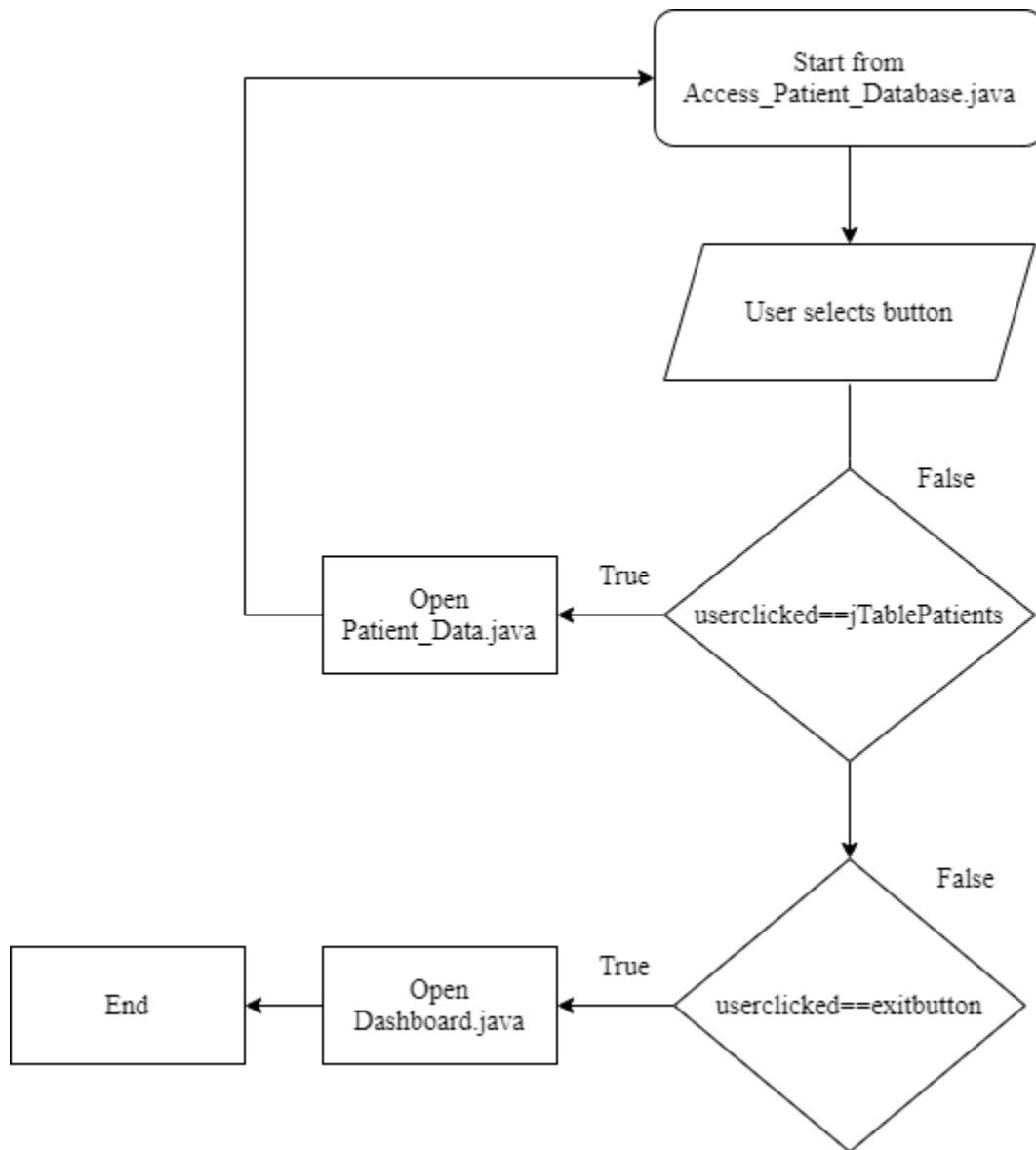
(Figure 1.) Starting the programme



**(Figure 2.)** Dashboard navigation



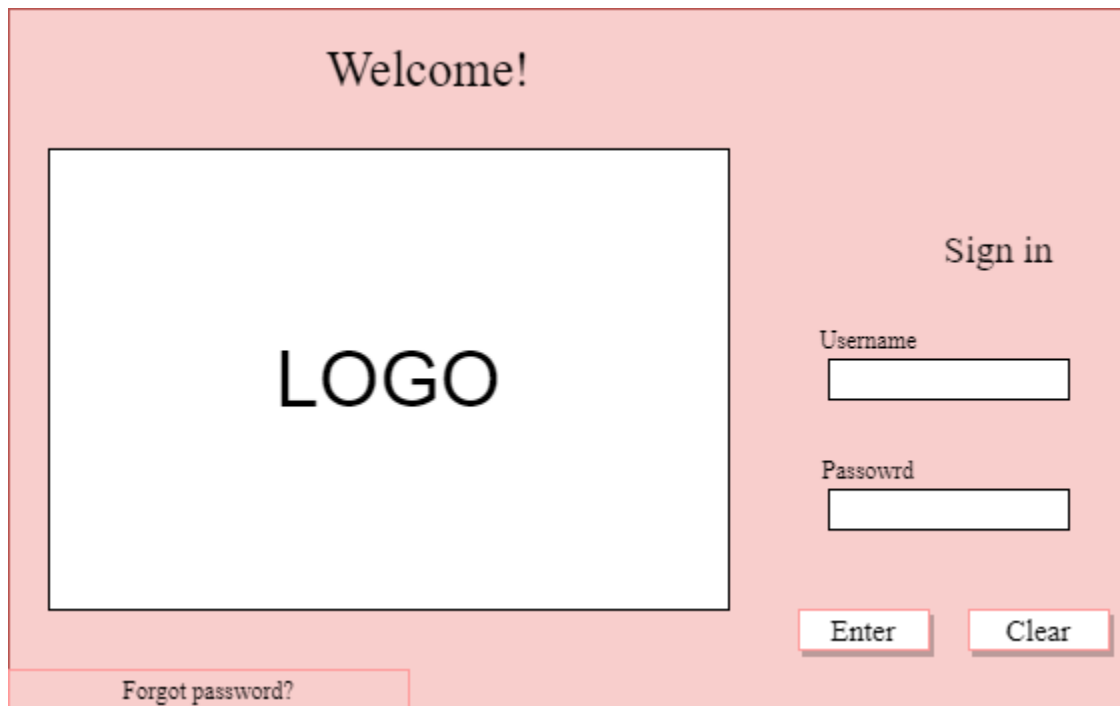
(Figure 3.) Individual patient view



## User Interface Design

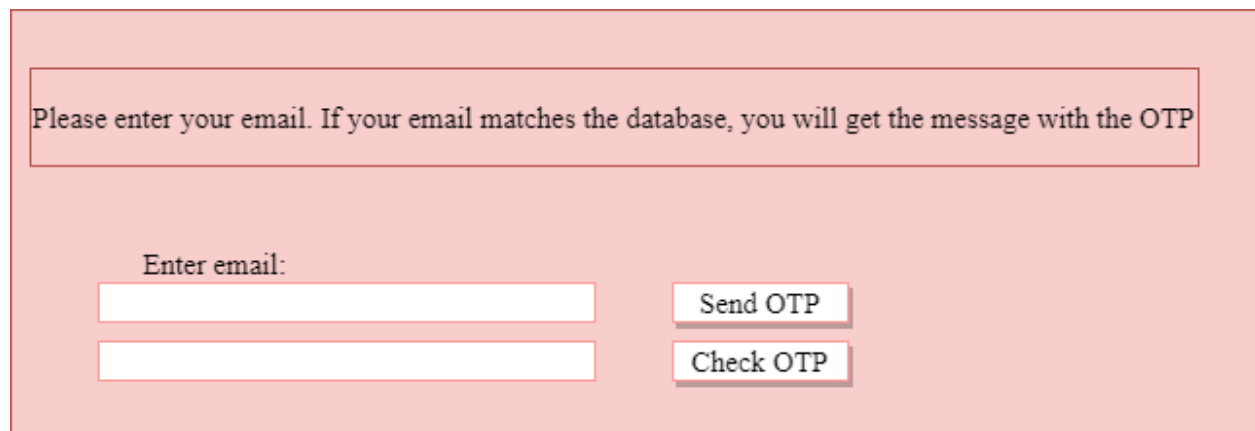
The following is the program's user interface. The color palette and the logo on the login page were both reviewed and accepted by the client and are a close match of the actual design. Every input field is a jTextField (blank white rectangles), which were incorporated with jButton (white shadowed rectangles) in order to manipulate data from jTextField. In addition, jTable was used to represent data from the MySQL database. In order to avoid clashes and inconsistencies of values which have a small range (Doctor, Sex, etc.), JComboBox and jCheckbox were used.

(Figure 4.) Login screen



The login screen has a light red background. At the top center, it says "Welcome!". Below this, on the left, is a large white rectangle labeled "LOGO". To the right of the logo, the text "Sign in" is displayed. Below "Sign in" are two input fields: "Username" and "Passowrd" (note the typo). Below these fields are two buttons: "Enter" and "Clear". At the bottom left, there is a link that says "Forgot password?".

(Figure 5.) OTP request panel



The OTP request panel has a light red background. At the top, there is a message box that says "Please enter your email. If your email matches the database, you will get the message with the OTP". Below this message box, there is a label "Enter email:" followed by two input fields. To the right of the input fields are two buttons: "Send OTP" and "Check OTP".

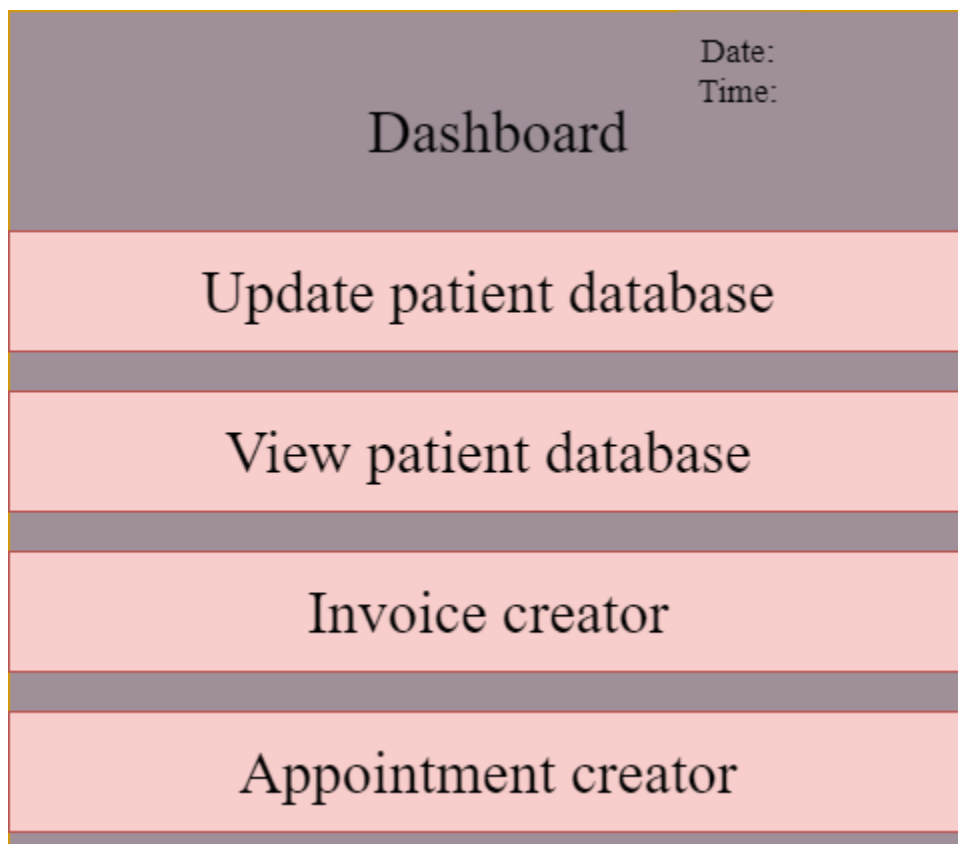
(Figure 6.) Admin data update page



A form for updating admin data, enclosed in a light red rectangular box. It contains three input fields: 'Username' and 'Email' at the top, and 'Password' below 'Username'. To the right of the 'Password' field is an 'Update' button. All text and the button are in black.

Username	Email
<input type="text"/>	<input type="text"/>
Password	Update
<input type="text"/>	

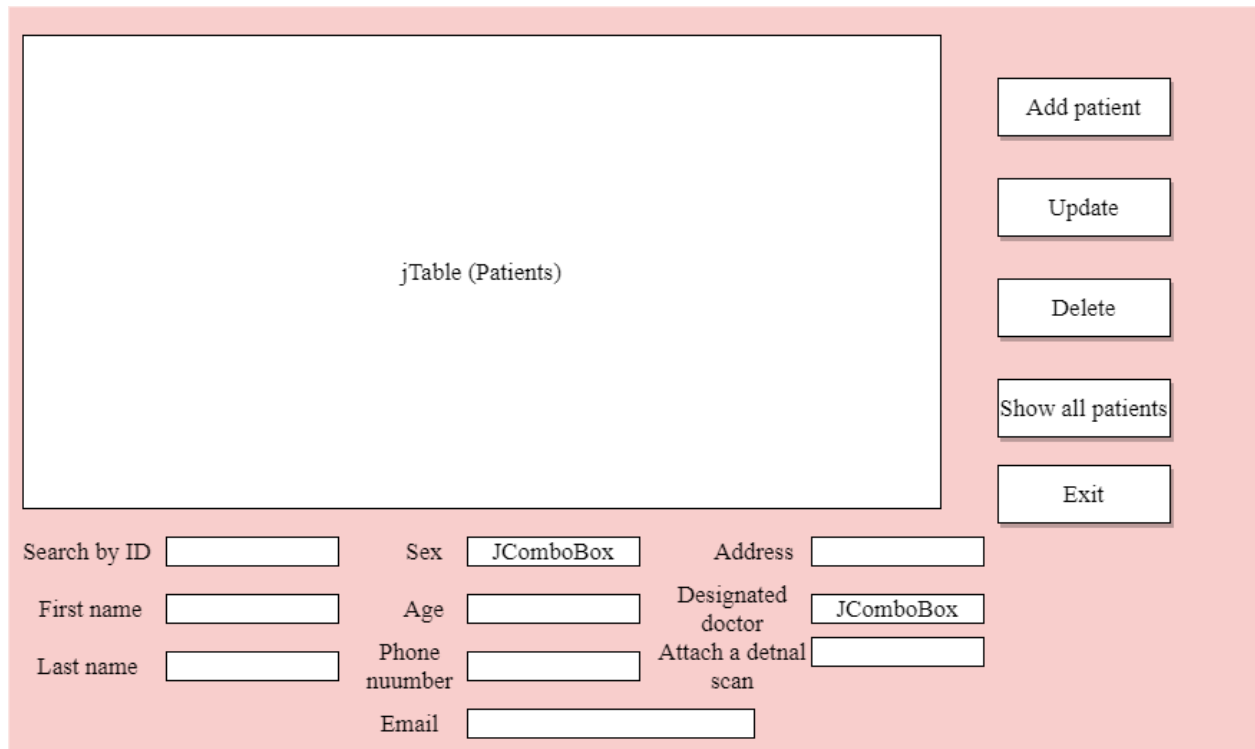
(Figure 7.) Non-edit access to the database



A dashboard interface for non-edit access to the database. It features a dark grey header bar with the word 'Dashboard' in the center and 'Date:' and 'Time:' on the right. Below the header are five alternating light red and dark grey horizontal bars, each containing a menu item: 'Update patient database', 'View patient database', 'Invoice creator', and 'Appointment creator'. The interface is enclosed in a thin yellow border.

Dashboard	Date: Time:
Update patient database	
View patient database	
Invoice creator	
Appointment creator	

(Figure 8.) Individual user data



The panel features a large white rectangular area labeled " jTable (Patients) " in the center. To the right of this area is a vertical stack of five buttons: "Add patient", "Update", "Delete", "Show all patients", and "Exit". Below the main area, there are several input fields and labels arranged in three rows. The first row contains "Search by ID" followed by a text box, "Sex" followed by a "JComboBox", and "Address" followed by a text box. The second row contains "First name" followed by a text box, "Age" followed by a text box, and "Designated doctor" followed by a "JComboBox". The third row contains "Last name" followed by a text box, "Phone nuumber" followed by a text box, and "Attach a detnal scan" followed by a text box. Below these, the label "Email" is followed by a text box.

jTable (Patients)

Add patient

Update

Delete

Show all patients

Exit

Search by ID

Sex JComboBox

Address

First name

Age

Designated doctor JComboBox

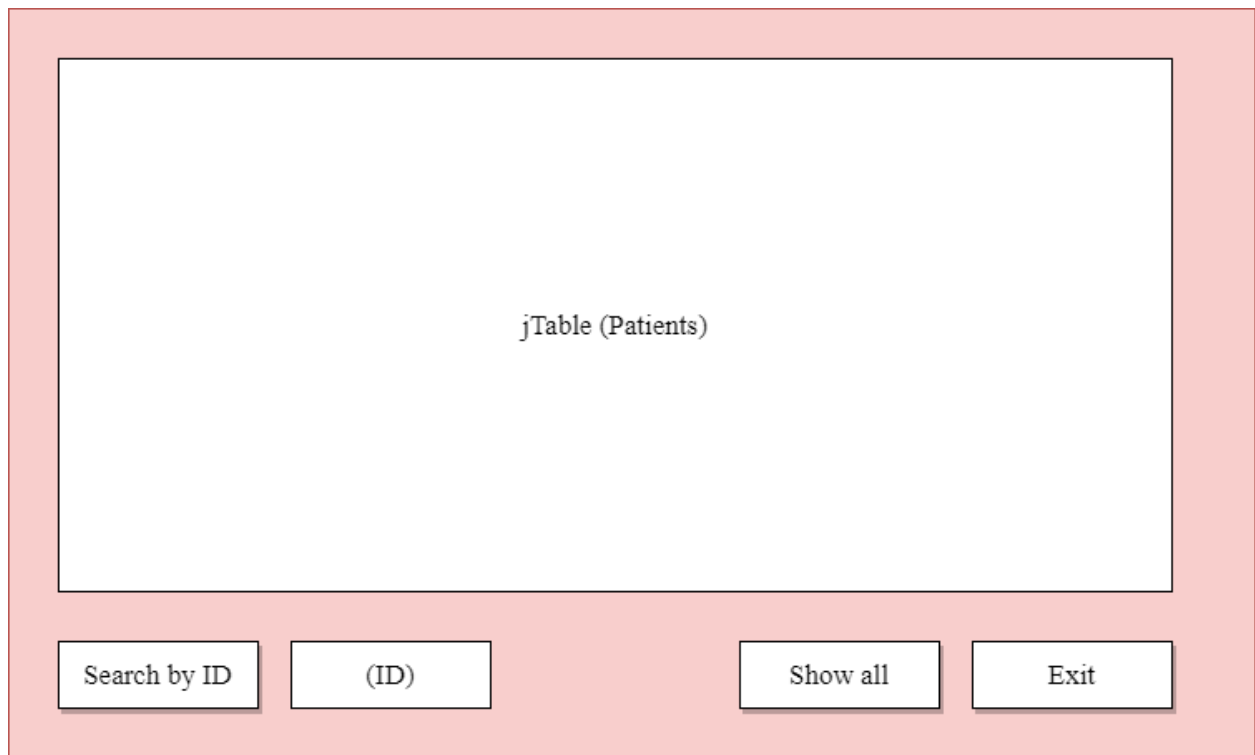
Last name

Phone nuumber

Attach a detnal scan

Email

(Figure 9.) Invoice manager (Create invoice panel)



The panel features a large white rectangular area labeled " jTable (Patients) " in the center. Below this area, there are four buttons arranged in a row: "Search by ID", "(ID)", "Show all", and "Exit".

jTable (Patients)

Search by ID

(ID)

Show all

Exit

(Figure 10.) Invoice manager (View invoices panel)

The screenshot shows a software interface for viewing invoices. On the left, there are input fields for patient information: ID, First name, Last name, Sex, Age, Phone number, Address, and Designated doctor. To the right of these fields is a large rectangular area labeled "jTable (Appointments/Invoices)". Below the input fields are three buttons: "Exit", "Show dental scan", and "Print patient data". To the right of the "Designated doctor" field are two buttons: "Check appointments" and "Check invoices".

(Figure 11.) Appointment maker

The screenshot displays the "Appointment maker" interface. On the left is a sidebar with four options: "Options", "Create an invoice", "View invoices", and "Exit". The "View invoices" option is currently selected. The main area is titled "Invoice manager" and contains a form for creating an invoice. The form includes input fields for ID, First name, Last name, and Age. To the right of these fields is a large text area labeled "jTextArea". Below the input fields are two columns of checkboxes for dental procedures: "Examination", "Tartar removal", "Helio fill", and "Dental treatment". Below these checkboxes are two input fields labeled "Procedure" and "Pricing". At the bottom of the form is a button labeled "Generate invoice". On the right side of the form is a button labeled "Print invoice".

(Figure 12.) Invoice Manager

The screenshot shows a Java Swing window titled "Invoice manager". On the left is a vertical sidebar with four buttons: "Options", "Create an invoice", "View invoices", and "Exit". The "View invoices" button is highlighted. The main area of the window contains a large rectangular box labeled "jTable (Invoices)". Below this box, at the bottom right of the main area, is a button labeled "Print".

(Figure 13.) Appointment creator

The screenshot shows a Java Swing window titled "Appointment creator". On the left is a large rectangular box labeled "jTable (Appointments)". To the right of this box is a control panel with several buttons and input fields. The buttons are: "Search by ID", "Book Appointment", "Print today's schedule", "Cancel appointment", "Print specific schedule", "Send email", and "Exit". There are also four empty rectangular input fields arranged in two pairs, one pair above the "Book Appointment" button and one pair above the "Print specific schedule" button.



## Database tables

(Figure 12.) User table

[illegible]

(Figure 13.) Patient table

[illegible]

(Figure 15.) Invoice table

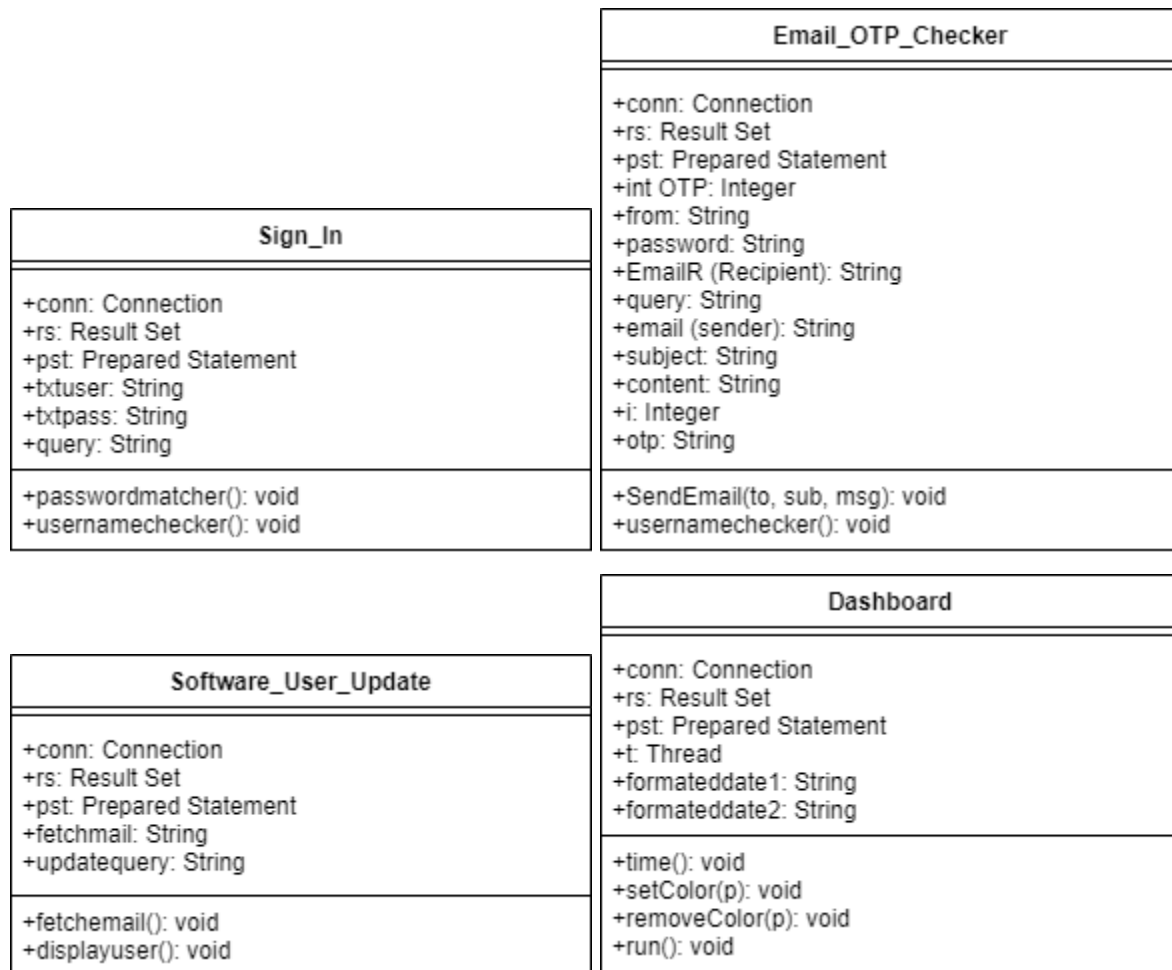
[illegible]

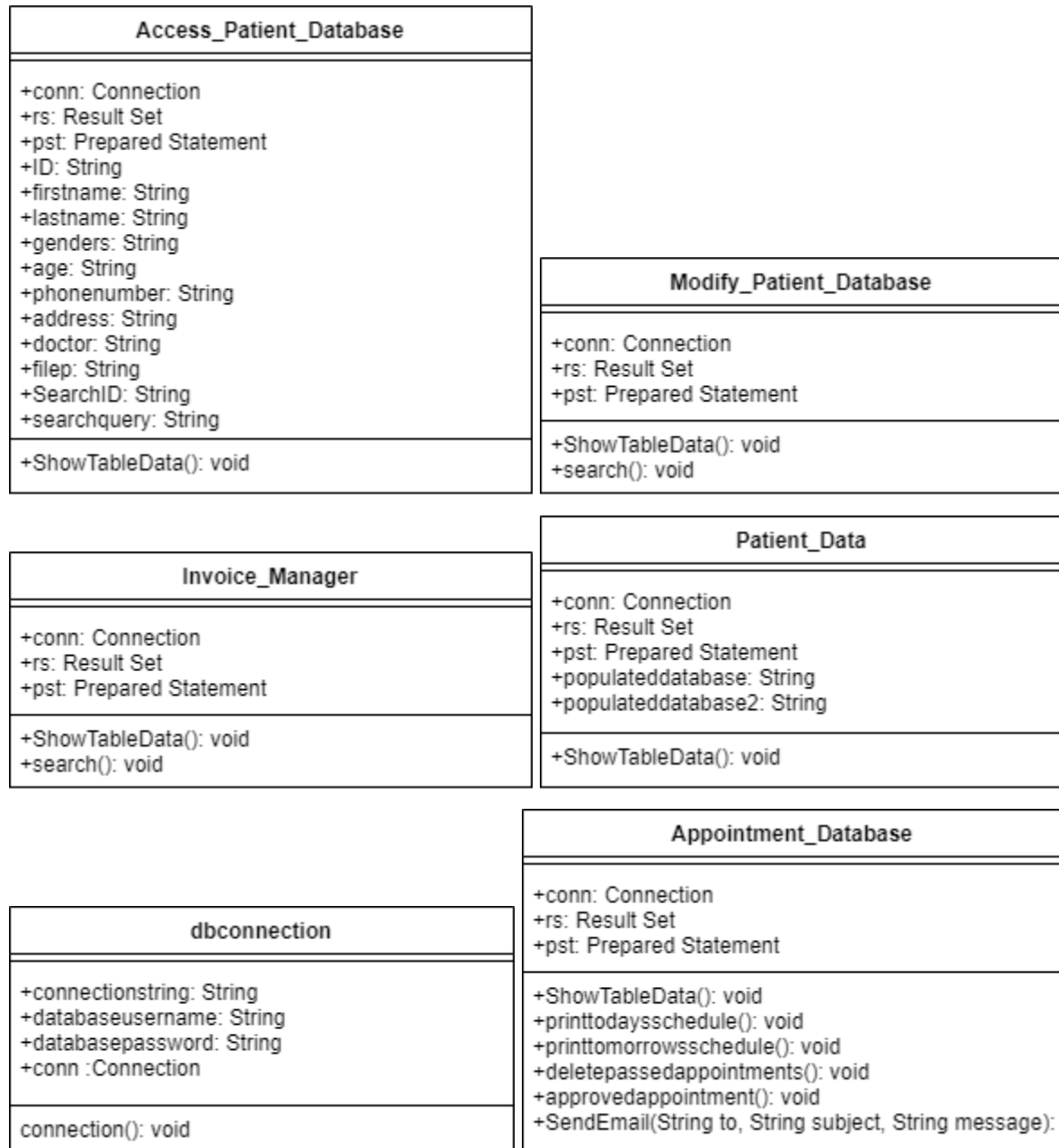
(Figure 16.) Appointment table

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
◇ First name	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
◇ Last name	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
◇ Patient ID	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
◇ Doctor	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
◇ Chair	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
◇ Time slot	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
◇ Date	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
📌 ID	VARCHAR(45)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

All of the column data types have been shown.

## UML Diagrams





Word count: 92

## Testing plan

Testing type	Test	Example
GUI Test	All of the GUI components run and display successfully	Class Modify_Database runs and displays all of the GUI components
GUI Navigation	The user can easily navigate between jFrames.	Class Dashboard is easily navigable and all of the classes are accessible on both ends (to and from)
Confirming that the authentication is accurate	The software starts with the login page. Once the user enters the login credentials, the programme should check whether the information matches. Appropriate responses should also be given	
Ensuring that the OTP system is working	In case the user wants to alter login credentials, the database checks whether the inserted email matches the database. If a match occurs, it should send an 6-digit random code to the user's email.	The class Email_OTP_Checker successfully communicates with the email server, allowing for an OTP security system
Database connection	Testing the communication of the program with the MySQL Database	Class dbconnection returns connection to respective Classes, enabling access to the database
SQL data insertion	All of the values from jTextField, JComboBox	Class Modify_Database successfully inserts and

	and jCheckbox should be entered and stored in the MySQL Database	data
SQL data deletion	User is able to remove data from the MySQL Database directly from the program	Appointment class gives the option to the user to cancel any appointments.
SQL data modification	The user is able to edit specific rows and data entries in case of an error or update.	Modify patient database displays selected row in appropriate JTextfields which the user can then update or modify
Individual user profiles	The user is able to access individual user profiles	Patient data class displays individual user information along with upcoming appointments and past invoices.
Invoice creator	Invoices are capable of being created from a template with predetermined prices.	Invoice manager class successfully creates invoices from a template, which could then be printed.
Invoice storage/access	The invoices should be stored in the database and should be accessible.	Invoice manager stores invoices along with the option of accessing and printing them.
Appointment clashes	User is able to successfully create an appointment and is able to prevent any clashes	The Appointments Class ensures that no two identical appointments can be made, prompting the user to change the appointment parameters if the programme finds anything clashing

Appointment chronology	Appointment schedule should be displayed in chronological order	Data in jTable from Appointment class is represented chronologically, both in dates and time
Printing data from MySQL database	Data extracted from MySQL is able to be printed	ViewOnly Class gives the user the ability to print Patient Data
Printing tables from MySQL database	jTables already populated from MySQL are printable	Receipt class gives the ability to print the receipt table.
Creating PDFs of both the data and the tables from MySQL	Data and jTable extracted from MySQL are able to be transformed into PDFs	Appointment class gives the ability to create a PDF of the schedule of an appointment in any given future date.
Sending reminder emails to patients regarding appointments	The ser is able to extract emails from database and send data to those emails directly from the app	The user can send reminder emails about schedule appointments