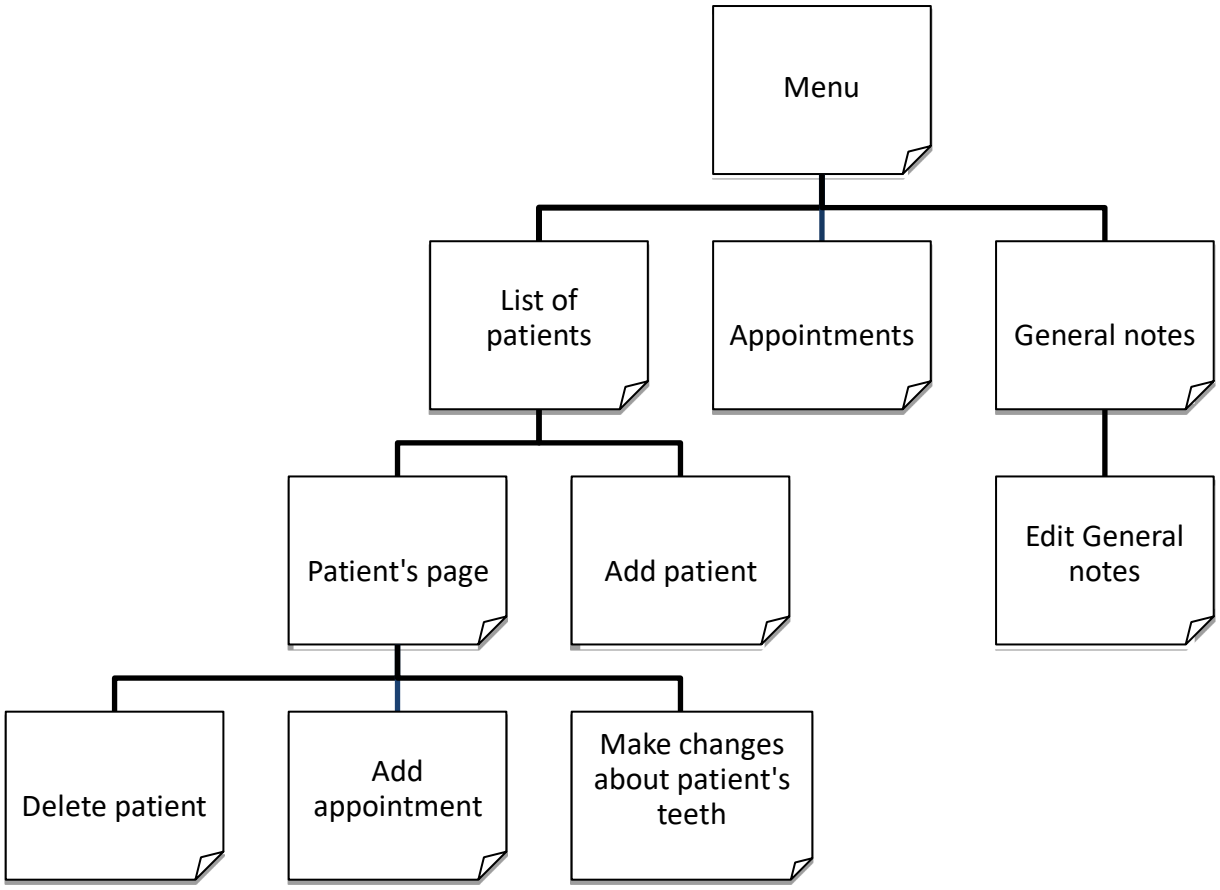
Criterion B: Design

Project Structure:



Databases created manually before writing the program.

I created a database called “dentist” locally with 3 tables.

Column	Type
id	int
name	varchar(20)
surname	varchar(20)
birth	date
gender	varchar(8)
phone	varchar(13)
email	varchar(30)
appointmentdate	date
appointmenttime	varchar(5)
appointmentduration	int
appointmentnote	varchar(100)

Table 1: patients

Column	Type
text	varchar(10000)

Table 2: notes

Column	Type		
id	int		
t1	varchar(350)	t17	varchar(350)
t2	varchar(350)	t18	varchar(350)
t3	varchar(350)	t19	varchar(350)
t4	varchar(350)	t20	varchar(350)
t5	varchar(350)	t21	varchar(350)
t6	varchar(350)	t22	varchar(350)
t7	varchar(350)	t23	varchar(350)
t8	varchar(350)	t24	varchar(350)
t9	varchar(350)	t25	varchar(350)
t10	varchar(350)	t26	varchar(350)
t11	varchar(350)	t27	varchar(350)
t12	varchar(350)	t28	varchar(350)
t13	varchar(350)	t29	varchar(350)
t14	varchar(350)	t30	varchar(350)
t15	varchar(350)	t31	varchar(350)
t16	varchar(350)	t32	varchar(350)

Table 3: teeth

Graphical visualization:

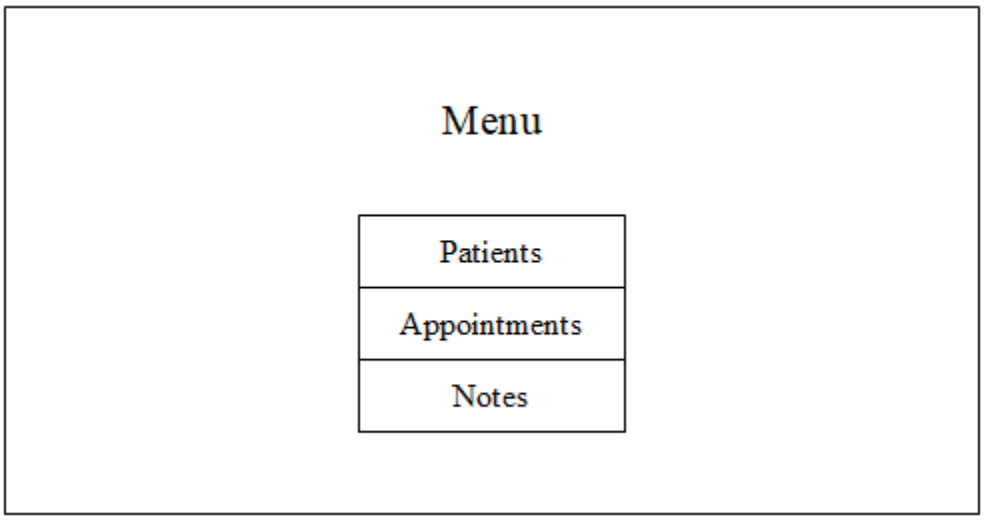


Figure 1: Main Menu.

- Click “Patients” to open the list of patients
- Click “Appointments” to open the list of appointments
- Click “Notes” to open general notes

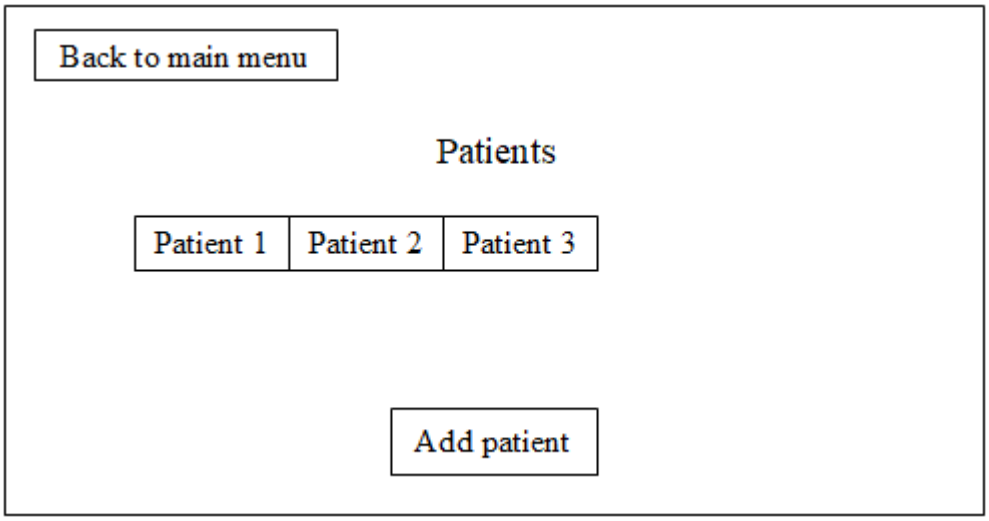


Figure 2: List of patients.

- Click “Back to main menu” to go back to main menu
- Click on the name of a patient (e.g. “Patient 1”) to open their page
- Click “Add Patient” to add a new patient to the list

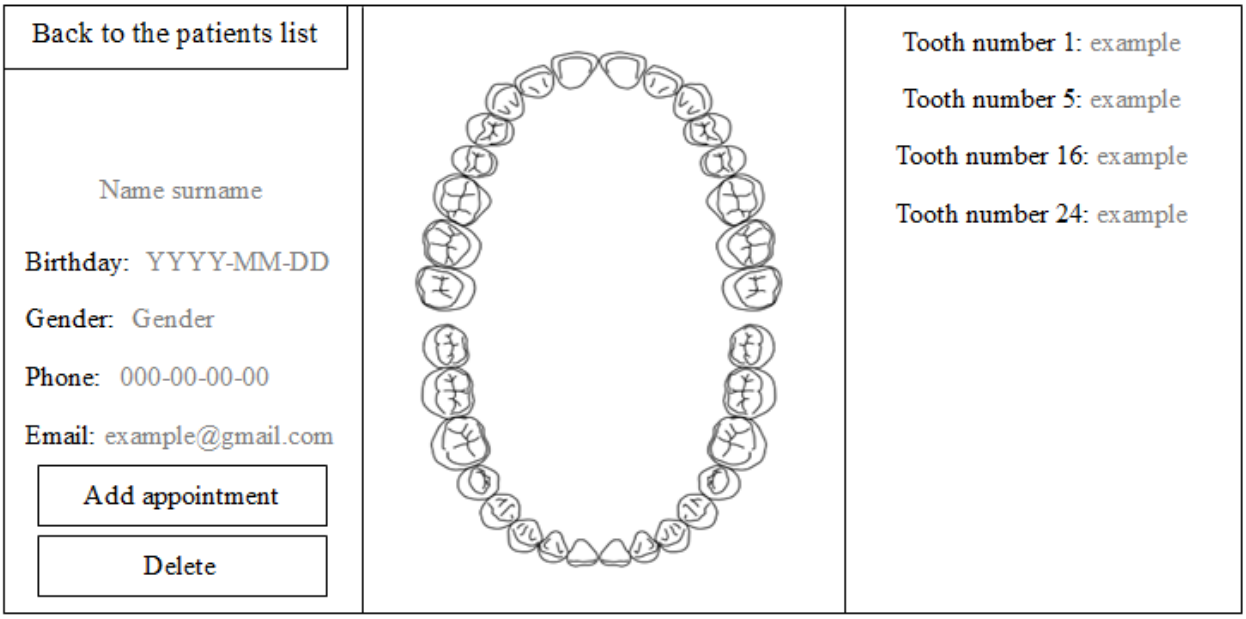


Figure 3: Patient's page.

- Information about the patient is shown on the left side
- Click "Back to the patients' list" to go back to the list of patients
- Click "Add appointment" to create or change an appointment for the patient
- Click "Delete" to delete the patient from the list of patients
- Click on a tooth to make notes about that tooth
- If a new note is added about a tooth it appears on the right side

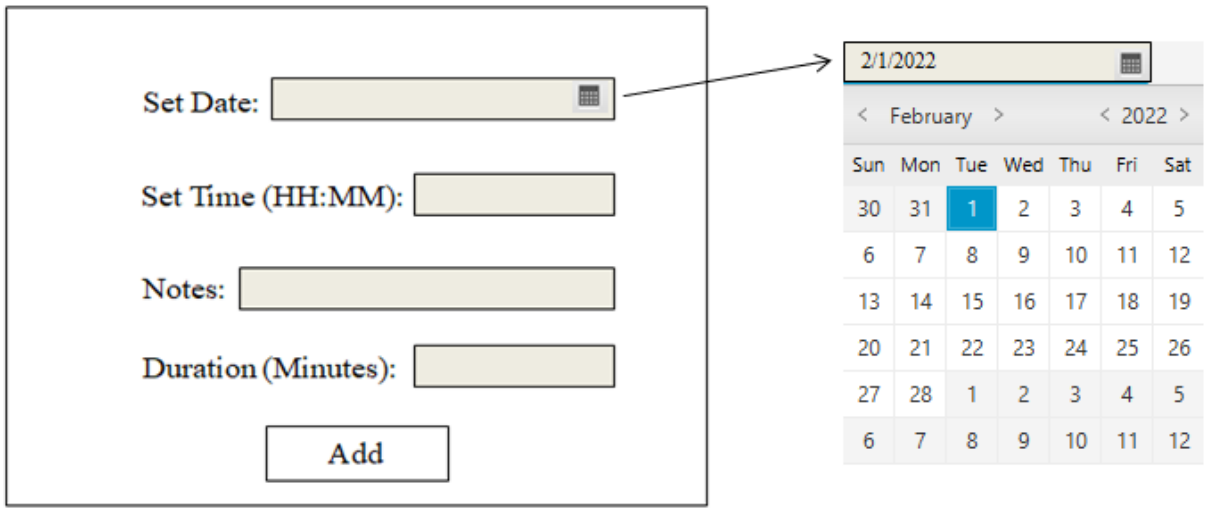


Figure 4: A window opens to add an appointment, fill out the form and click "Add" to save the appointment. Click on the "▦" to pick a date from calendar

Figure 5: Warning message appears if the entered information is not valid or if the appointment overlaps with another appointment

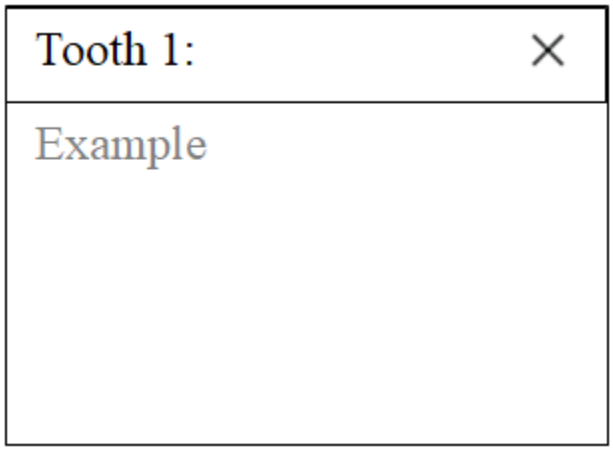


Figure 6: A window opens when a tooth is clicked. Make notes and click "X" to save the changes

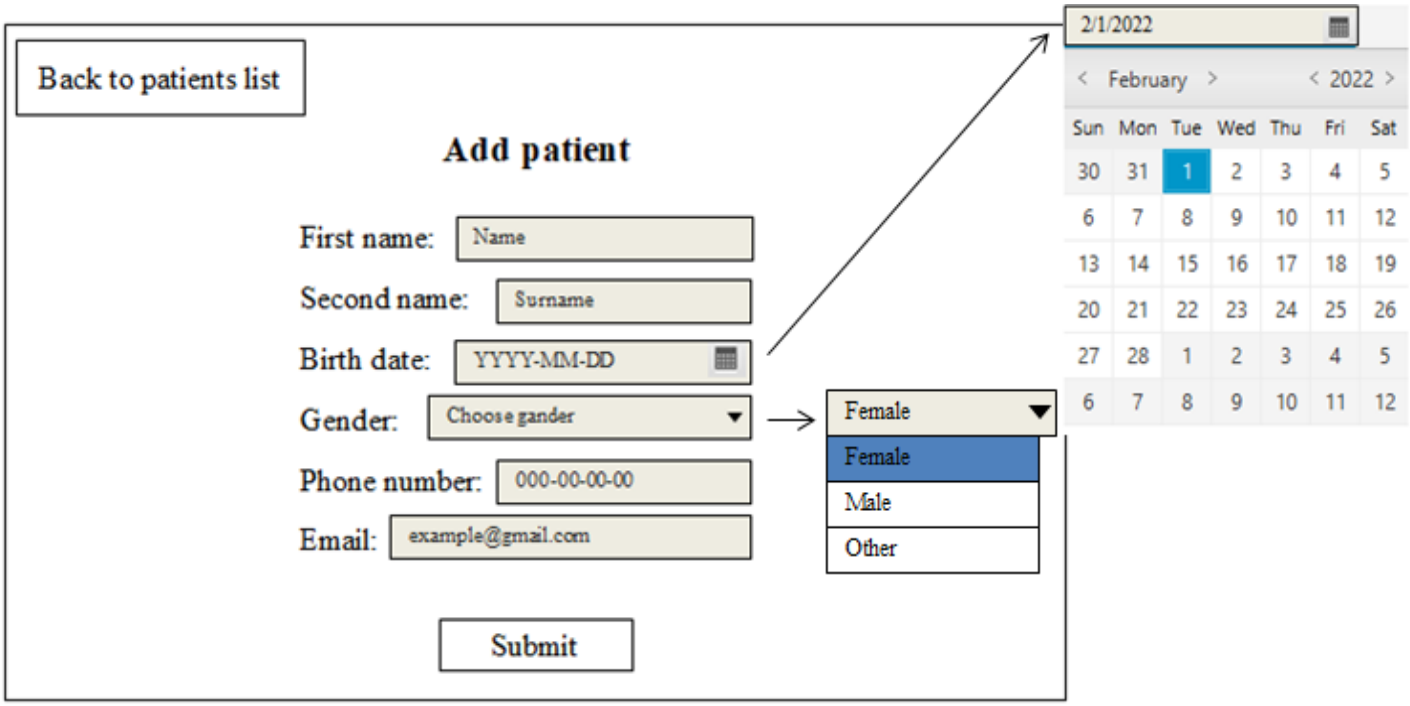


Figure 7: A window opens to add a patient. Fill out the form and click "Submit" to add a new patient. Click the "▦" to pick a date from calendar. Click the "▼" to choose from dropdown menu.

Figure 8: Warning message appears if the entered information is not valid

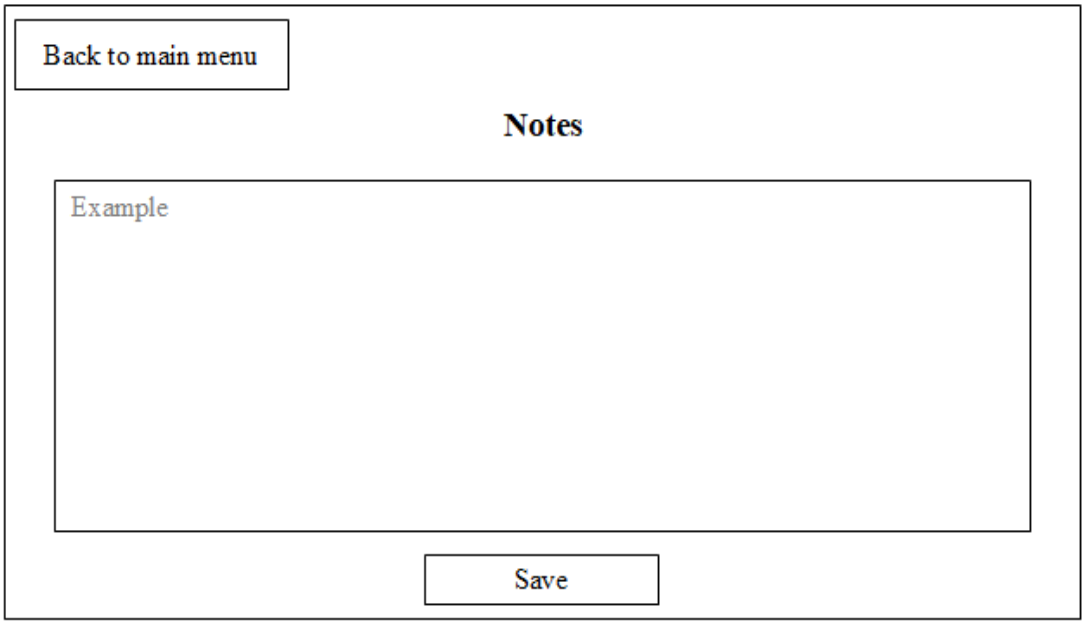


Figure 9: Appointments. Registered appointments appear sorted by their date and time. Click “Back to main menu” to go back to the main menu.

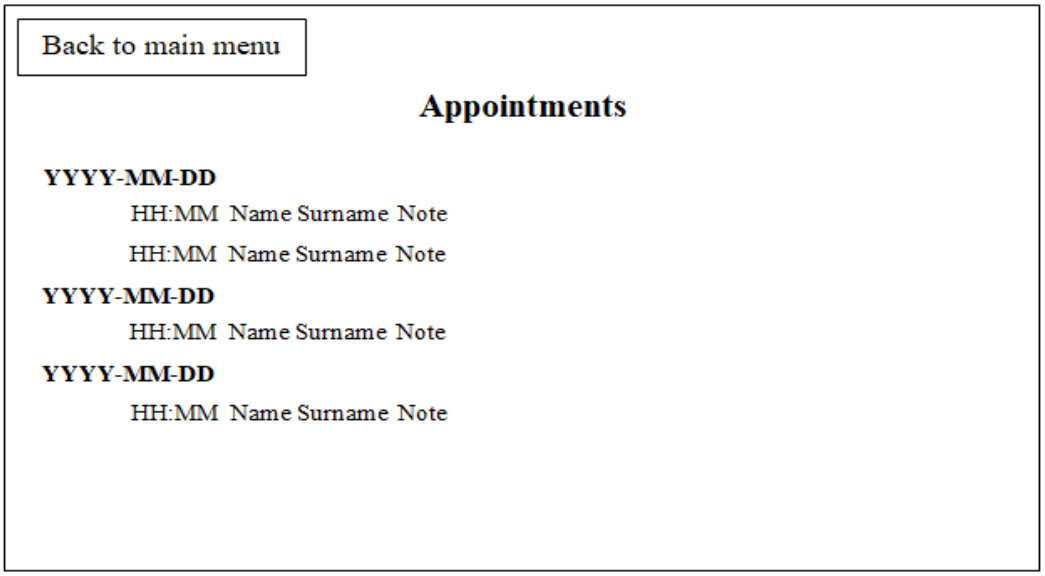
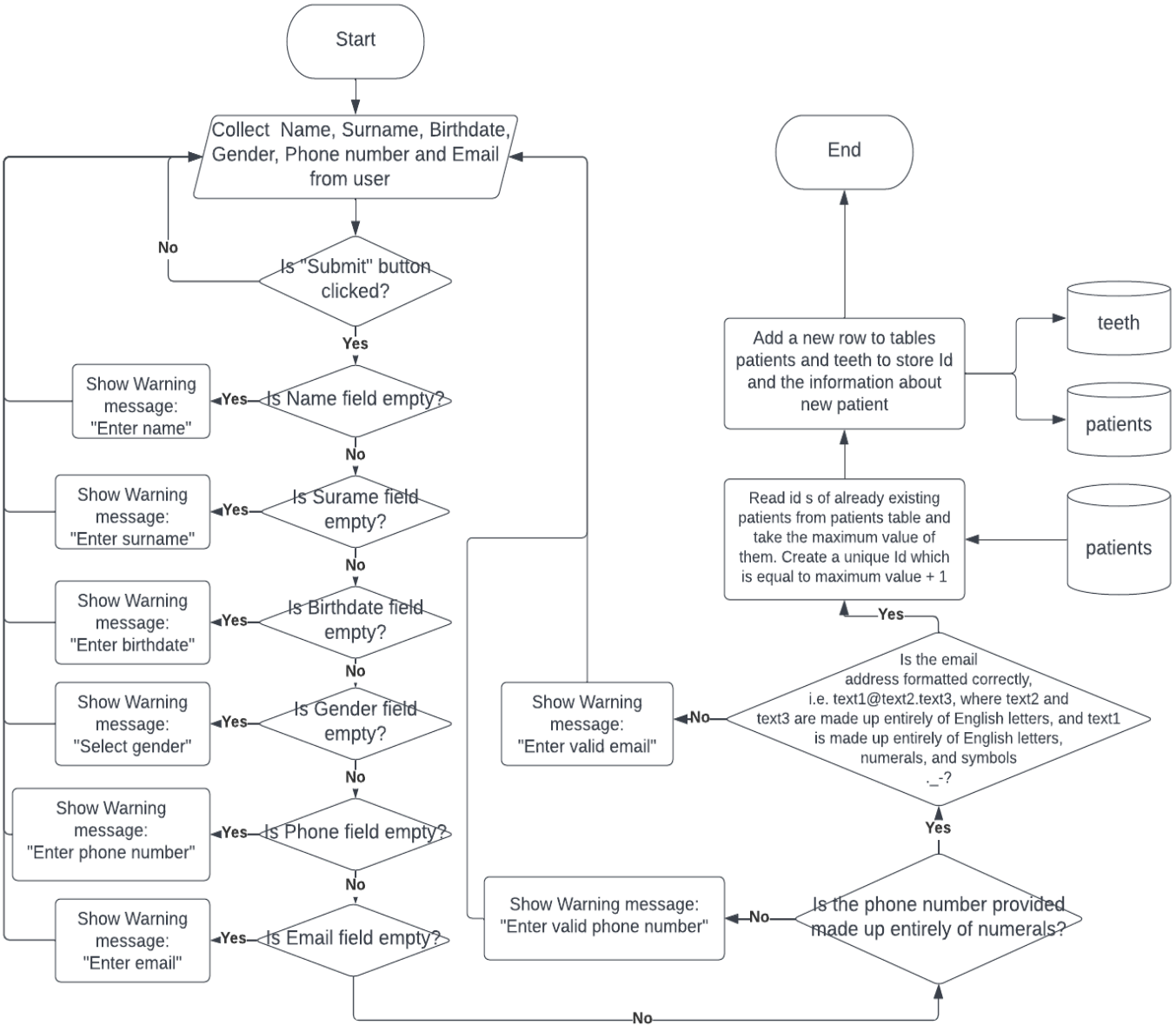
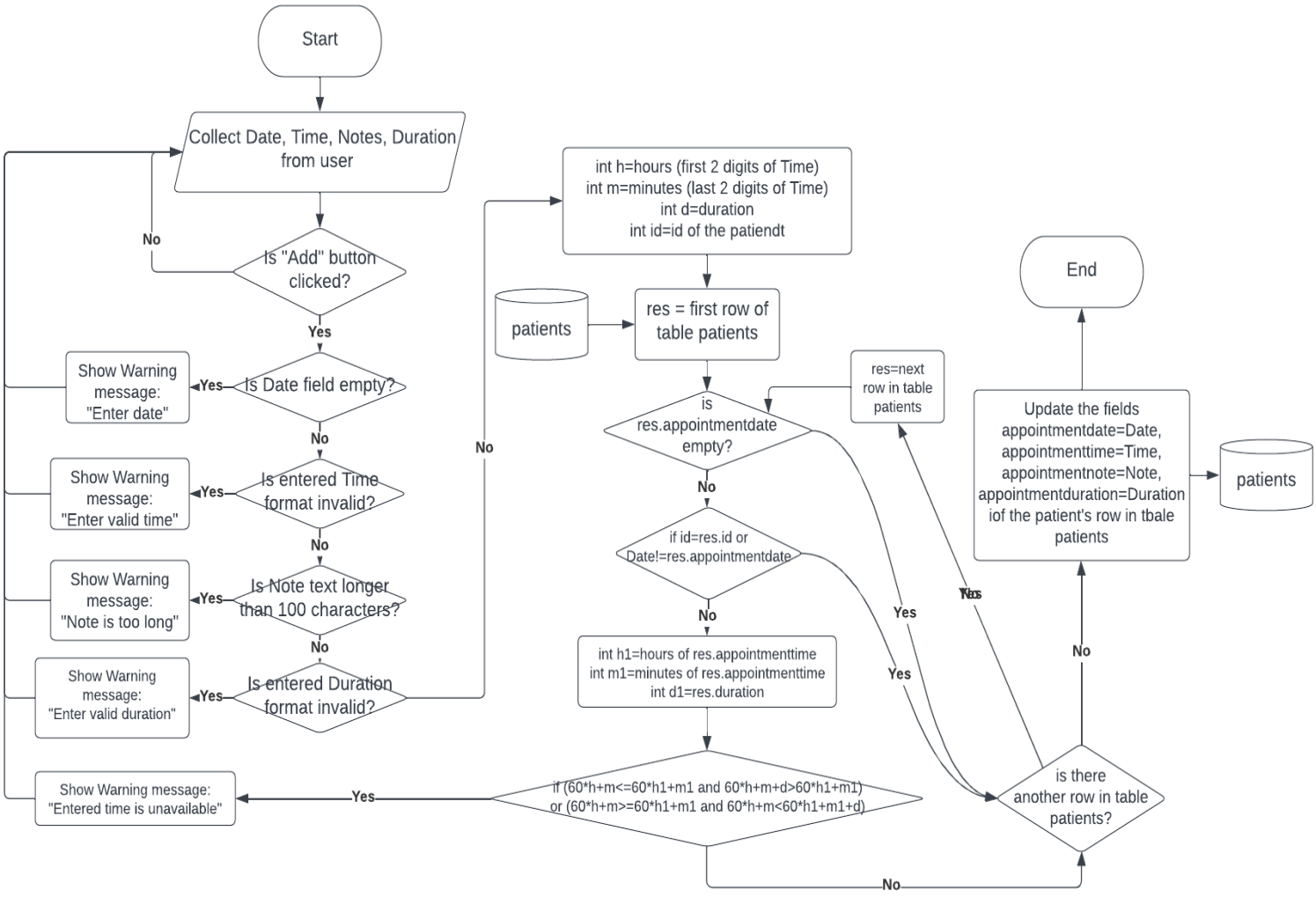


Figure 10: General notes. Write in the text area and click “Save” to save the changes. Click “Back to main menu” to go back to main menu.

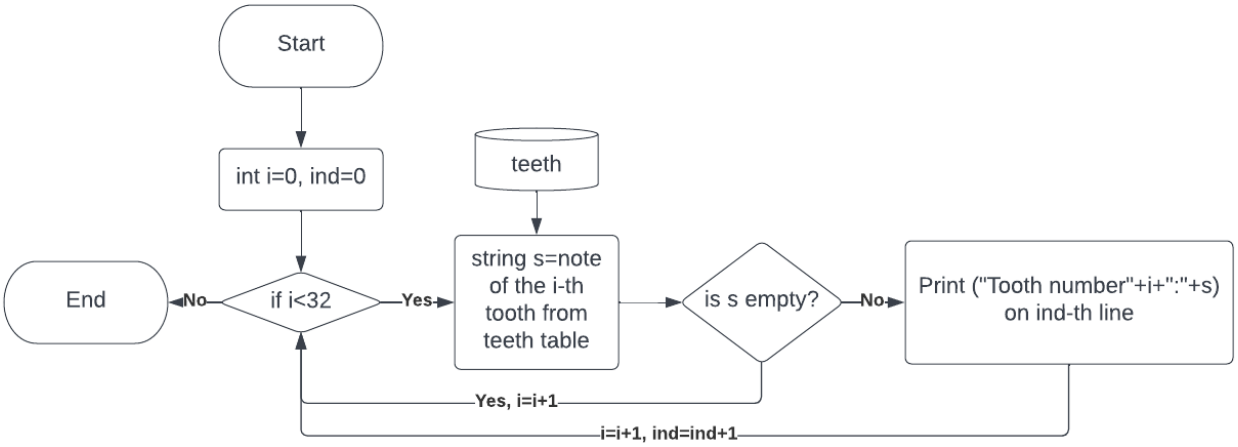
Important flowcharts:



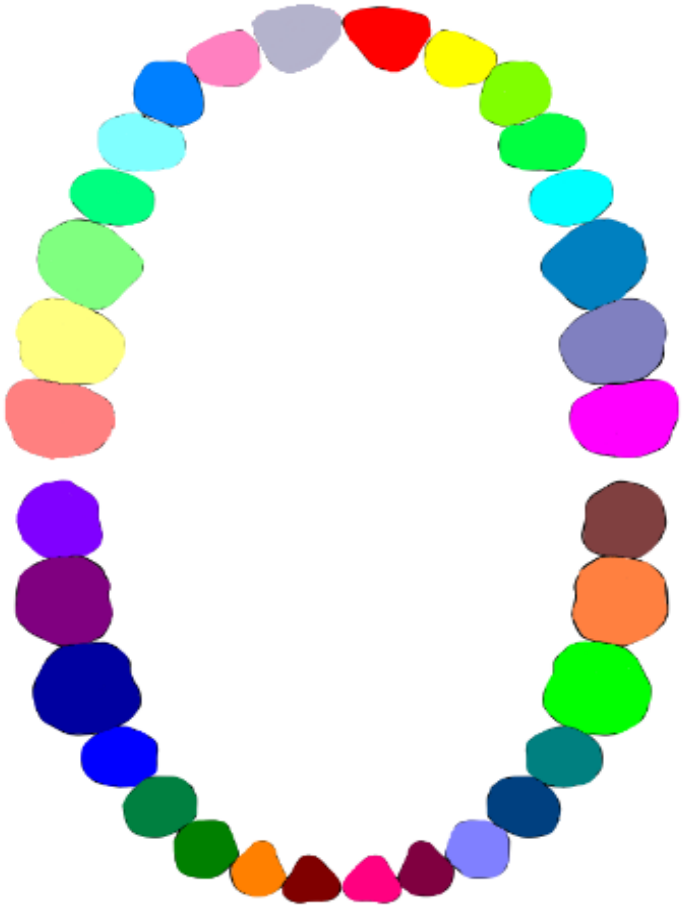
Flowchart 1: Adding a new patient.



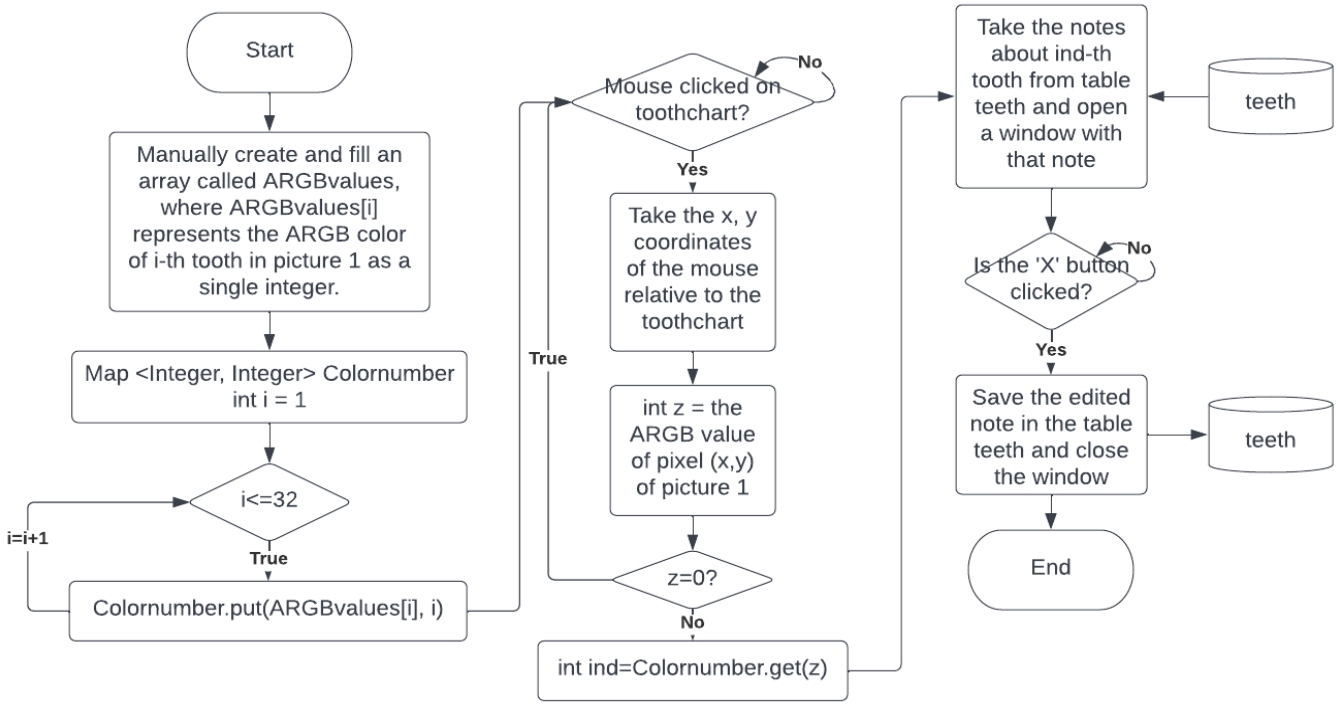
Flowchart 2: Adding an appointment



Flowchart 3: Showing the notes about teeth on patient's page

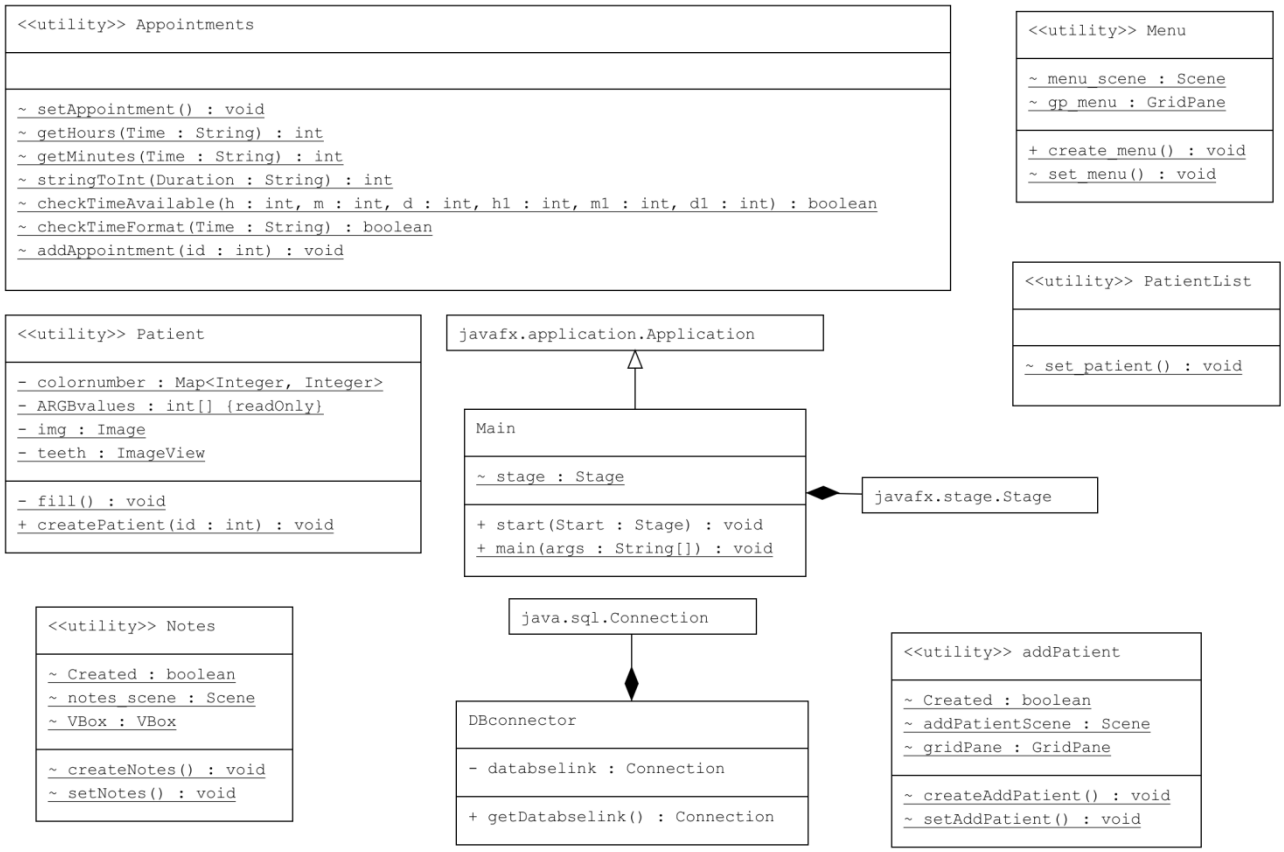


Picture 1: This picture is saved in project folder



Flowchart 4: Editing notes of teeth

UML diagram:


<<utility>> Appointments
~ setAppointment() : void
~ getHours(Time : String) : int
~ getMinutes(Time : String) : int
~ stringToInt(Duration : String) : int
~ checkTimeAvailable(h : int, m : int, d : int, h1 : int, m1 : int, d1 : int) : boolean
~ checkTimeFormat(Time : String) : boolean
~ addAppointment(id : int) : void
<<utility>> Menu
~ menu_scene : Scene
~ gp_menu : GridPane
+ create_menu() : void
~ set_menu() : void
<<utility>> PatientList
~ set_patient() : void
<<utility>> Patient
- colornumber : Map<Integer, Integer>
- ARGBvalues : int[] {readOnly}
- img : Image
- teeth : ImageView
- fill() : void
+ createPatient(id : int) : void
javafx.application.Application
Main
~ stage : Stage
+ start(Start : Stage) : void
+ main(args : String[]) : void
javafx.stage.Stage
<<utility>> Notes
~ Created : boolean
~ notes_scene : Scene
~ VBox : VBox
~ createNotes() : void
~ setNotes() : void
java.sql.Connection
DBconnector
- databselink : Connection
+ getDatabselink() : Connection
<<utility>> addPatient
~ Created : boolean
~ addPatientScene : Scene
~ gridPane : GridPane
~ createAddPatient() : void
~ setAddPatient() : void

Test plan:

Action to be tested	Test method
Page PatientsList is functioning correctly: names of patients are displayed correctly and buttons are functioning.	Manually input 10-15 rows to table patients. Make the program print the id of the patients that was clicked. Open page PatientsList and see if all the patients are displayed. Click a button and see if the right id is printed
Page Patient is functioning correctly: information of a patient is displayed correctly, tooth chart is visible and when clicked on a tooth a window opens, made notes are saved/displayed correctly	Manually input 3-4 patients to table patients. Go to their Patient pages and see if information is displayed correctly. Make notes on different sets of teeth for each patient and see if they are displayed correctly and are saved correctly in table teeth.
Page Appointments is functioning correctly: all the added appointments are displayed correctly and in chronological order	Manually input 5-10 patients and appointments to table patients. Go to page Appointments and see if appointments are displayed correctly.
Page Notes is functioning correctly: made notes are saved/displayed correctly	Go to page Notes, make some notes and click save button. See if it is saved correctly in table notes. Close and open page Notes again to see if notes are displayed correctly.
Go back buttons are functioning correctly	Run the program multiple times and check if back buttons are functioning correctly on every page
Page AddPatient is functioning correctly: it allows to enter the required information, throws a warning message if entered information is not valid, saves the entered data correctly	Go to page AddPatient and fill out the fields with different errors to see if warning messages are displayed correctly. Add 3-4 patients and check if the information is saved correctly in table patients.
Page AddAppointment is functioning correctly: it allows to enter the required information, throws a warning message if entered information is not valid or it overlaps with another appointment, saves the entered data correctly	Go to page AddAppointment and fill out the fields with different errors or with overlapping time to see if warning messages are displayed correctly. Add 3-4 appointments and check if the information is saved correctly in table patients.

Actions with the database are functioning correctly	Make the program to print error messages if it is unable to connect or interact with the database correctly. Run the program multiple times and do different actions with database. See if any error messages are printed.
All the components are functioning correctly	Run the program multiple times by myself to see if anything is wrong

Word count: 140