Jason Mejia
Professor Auda
CSC221

11/14/2019

Final Project Report

Objective:

To create a database of students, the courses and the classes. There is more information within these sections. Using any database management system.

Solution:

I have decided to use SQLite for my project. I have started with the connection being established. We would have three tables which are students, classes and courses. Buttons have been created for each of them which are interactive. But unfortunately the data input from the user is my issue in the project. The queries weren't being displayed in the chart. It would go to the next prompt but then the data will not be saved and be put into columns. I couldn't figure out the method for the project.

Code:

DrawDesign.java

```
package DataProj;

import javafx.application.Application;
import javafx.beans.property.SimpleStringProperty;
import javafx.beans.value.ObservableValue;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.scene.Scene;
import javafx.scene.control.* ;
import javafx.scene.layout.BorderPane;
import javafx.scene.layout.HBox;
import javafx.scene.layout.VBox;
import javafx.scene.paint.Color;
import javafx.scene.text.Font;
import javafx.scene.text.FontPosture;
import javafx.scene.text.Text;
import javafx.stage.Stage;
import javafx.util.Callback;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.DriverManager;
import java.sql.Connection;
import java.sql.SQLException;
import java.util.Optional;
```

```
	public class DrawDesign extends Application {
		public Connection db; //Database connection variable
		public ComboBox comboBox; //Dropdown menu
		public ResultSet result; // Holds the output of a query
		public TextField textField; // Query line
		public TableView table; //Actual db table
		public Button S, Co, Cl, D, Process; //All buttons
		public Statement stmt; //Query statement variable
		public double CommRate;
		public String query; // query string variable
		public ObservableList < ObservableList > input = FXCollections.observableArrayList();
//List to add to table
		public String[] all = {
				"courses",
				"classes ",
				"students"
		};

		String url = "jdbc:sqlite:C:\\Users\\good_\\Desktop\\Students.db";

		@Override
		public void start(Stage primaryStage) throws Exception {
			//DATABASE Connection
			try {
				Class.forName("org.sqlite.JDBC");
				db = DriverManager.getConnection(url); //Establishing connection
toDatabase
				System.out.println("Connected to Tunngle"); //Successful message
			}
			catch(SQLException e) {
				System.err.println("Connection Failed"); //Failure to connect
				e.printStackTrace(); //Traces what went wrong in program
				System.exit(1); //Exit program
			}
			//All Options for the combo box (drop down menu)
			ObservableList < String > dropdown = FXCollections.observableArrayList("All
Students", "Students with grade A",
					"Students with grade B", "Students with grade C", "Students with
grade D", "Students with grade F",
					"All Male Students", "All Female Students");
			//Query field
			textField = new TextField();
			textField.setPrefWidth(500);
			//Buttons and actions handled
			S = new Button("Add Student");
			S.setOnAction(e
					->{
				String EMPL = dataInput("Students EMPL ID", "Insert ID", "12345678");
				String First = dataInput("Adding Student", "Enter First Name", "First");
				String Last = dataInput("Adding them still", "Enter Last Name", "Last");
				String sex = dataInput("Almost there", "Enter students sex", "M/F?");
				query = "INSERT INTO students VALUES ( '" + EMPL + "', '" + First + "',
'" + Last + "', '" + sex + "')";
				try {
					execute(query);
```

```
                        }
                        catch(SQLException ea) {
                        }
                });
                Co = new Button("Add Course");
                Co.setOnAction(e ->{
                        String EMPL = dataInput("Students EMPL ID", "Insert ID", "12345678");
                        String courseID = dataInput("Course ID", "Insert ID", "CSC220");
                        String title = dataInput("Course Title", "Enter Course", "Title");
                        String dep = dataInput("Course Department", "Enter Department",
"Department");
                        query = "INSERT INTO courses VALUES ( '" + EMPL + "', '" + courseID +
"', '" + title + "', '" + dep + "')";
                        try {
                                execute(query);
                        }
                        catch(SQLException ea) {
                        }
                });
                Cl = new Button("Add Classes");
                Cl.setOnAction(e -> {
                        String EMPL = dataInput("Students EMPL ID", "Insert ID", "12345678");
                        String code = dataInput("Class Code", "Insert Code", "AB12345");
                        String courseID = dataInput("Course ID", "Insert ID", "CSC220");
                        String year = dataInput("What Year", "Insert year", "2019");
                        String semester = dataInput("Which Semester", "Insert semester",
"Fall?");
                        String gpa = dataInput("Students GPA", "Enter GPA", "A");
                        query = "INSERT INTO classes VALUES ( '" + EMPL + "', '" + code + "', '"
+ courseID + "', '" + year + "', '" + semester + "', '" + gpa + "')";
                        try {
                                execute(query);
                        }
                        catch(SQLException ea) {
                        }
                });
                D = new Button("Delete Student");
                D.setOnAction(e ->{
                        String EMPL = dataInput("Students EMPL ID", "Insert ID", "12345678");
                        for (int i = 0; i < all.length; i++) {
                                query = "DELETE FROM " + all[i] + " WHERE EMPL = ( '" + EMPL +
"')";
                                try {
                                        execute(query);
                                } catch (SQLException eo) {
                                }
                        }
                });

                Process = new Button("Process");
                HBox buttons = new HBox(30);
                buttons.getChildren().addAll(S, Co, Cl, D);
                comboBox = new ComboBox(dropdown);
                HBox others = new HBox(15);
                others.getChildren().addAll(comboBox, Process, textField);
```

```
		BorderPane pane = new BorderPane();
		pane.setTop(buttons);
		pane.setCenter(others);
		Text t = new Text(10, 20, "Student Grades");
		t.setFont(Font.font("Verdana", FontPosture.ITALIC, 20));
		t.setFill(Color.BLACK);
		Text creator = new Text(10, 10, "Jason Mejia CSC 221 Final Project");
		creator.setUnderline(true);
		creator.setFont(Font.font("Verdana", 20));
		table = new TableView();
		VBox vbox = new VBox();
		vbox.getChildren().addAll(t, table);
		BorderPane pane1 = new BorderPane();
		pane1.setBottom(creator);
		pane1.setCenter(pane);
		pane1.setTop(vbox);
		Scene scene = new Scene(pane1, 1000, 800);
		Stage stage = new Stage();
		primaryStage.setTitle("");
		stage.setScene(scene);
		stage.show();
		//When query is manually typed on the TextField
		textField.setOnAction(e -> {
			try {
				table.getItems().clear(); //clearing table
				table.getColumns().clear(); //clearing table
				query = textField.getText();
				stmt = db.createStatement();
				execute(query);
			}
			catch (SQLException ex) {
				System.err.println("Error");
				ex.printStackTrace();
			}
		});
		//Option selection
		Process.setOnAction(
			e ->{
				try{
					String choice =

comboBox.getSelectionModel().getSelectedItem().toString();
					if (choice == "All Students"){
						query = "SELECT * FROM students";
					}
					if (choice == "Students with grade A "){
						query = "SELECT * FROM classes WHERE gpa = 'A' ";
					}
					if (choice == "Students with grade B"){
						query = "SELECT * FROM classes WHERE gpa = 'B' ";
					}
					if (choice == "Students with grade C"){
						query = "SELECT * FROM classes WHERE gpa = 'C' ";
					}
					if (choice == "Students with grade D"){
```

```
                                        query = "SELECT * FROM classes WHERE gpa = 'D' ";
                                    }
                                    if (choice == "Students with grade F"){
                                        query = "SELECT * FROM classes WHERE gpa = 'F' ";
                                    }
                                    if (choice == "All Male Students"){
                                        query = "SELECT * FROM students WHERE sex = 'M' ";
                                    }
                                    if (choice == "All Female Students"){
                                        query = "SELECT * FROM students WHERE sex = 'F' ";
                                    }
                                    execute(query);
                                }
                                catch(SQLException ex){
            System.err.println("Error");
                                }


        table.setColumnResizePolicy(TableView.UNCONSTRAINED_RESIZE_POLICY);
                            });
            }

            public void execute(String query) throws SQLException {
                stmt = db.createStatement();
                result = stmt.executeQuery(query);
                dbDisplay(result);
            }

            public void dbDisplay(ResultSet result) throws SQLException {
                table.getItems().clear(); //Clears table
                table.getColumns().clear();//Clears table
                boolean data = result.next();
                if (!data) { //If there's no data return empty database
                    System.out.println("Empty Database");
                    return;
                }
                try {
                    int i;
                    for (i = 0; i < result.getMetaData().getColumnCount(); i++) {
                        int t = i;
                        TableColumn col = new
TableColumn(result.getMetaData().getColumnName(i+ 1));
                        col.setCellValueFactory((Callback <
TableColumn.CellDataFeatures<ObservableList,String> ,ObservableValue<String>>)
                                                param -> new

        SimpleStringProperty(param.getValue().get(t).toString()));
                        table.getColumns().addAll(col);
                    }
                    ObservableList < String > row = FXCollections.observableArrayList();
                    //Prints the first row
                    int l = 0;
                    while (l < result.getMetaData().getColumnCount()) {
                        row.add(result.getString(l + 1));
                        l++;
```

```
                        }
                        input.add(row);
                        while (result.next()) {
                                ObservableList < String > rows =
FXCollections.observableArrayList();
                                for (i = 0; i < result.getMetaData().getColumnCount(); i++) {
                                        rows.add(result.getString(i + 1));
                                }
                                input.add(rows);
                        }
                        table.setItems(input);
                }
                catch(SQLException ex) {
                        System.err.println("Error");
                        ex.printStackTrace();
                }
        }

        //Used TextInputDialog to get input from user, not ChoiceDialog or Alert
        public static String dataInput(String BoxTitle, String msgToUser, String indicator) {
                TextInputDialog textInputDialog = new TextInputDialog(indicator);
                textInputDialog.setTitle("Adding");
                textInputDialog.setHeaderText(BoxTitle);
                textInputDialog.setContentText(msgToUser);
                //textInputDialog.initStyle(StageStyle.UNIFIED);
                Optional < String > result = textInputDialog.showAndWait();
                if (result.isEmpty()) {
                        return null;
                }
                else {
                        return result.get();
                }
        }

        public static void main(String[] args) {
                launch(args); //launching program
        }
}
```