ILLINOIS INSTITUTE OF TECHNOLOGY

Final Project

Lake Park Secondary School App

- Phase 2 -

ITMD 510
Object Oriented App Development

Fall 2020

Student: NAROA AMEZAGA
A20473682

INDEX

INTRODUCTION 3
DESCRIPTION 3
FUNCTIONAL REQUIREMENTS OF THE PROJECT 3
USE CASES 3
TECHNOLOGIES USED 5
PROJECT STRUCTURE 5
STARTUP INFORMATION AND LOGIN CREDENTIALS 6
SNAPSHOTS 8
EXTRA CREDIT 24
SOURCE CODE 26

INTRODUCTION

The purpose of this document is to complete the phase 2 of the final project corresponding to course ITMD 510 called "Object Oriented App Development". The objective of the final project is to implement an application using JavaFX and MySQL, by creating a Graphical User Interface to accomplish such objective. In this second phase, the application idea initiated in phase 1 will be implemented using JavaFX.

DESCRIPTION

Due to the persistent demand of students and teachers, and as many other schools around have also done so, Lake Park Secondary School has decided to modernize itself not to fall behind. Therefore, an application will be designed with the purpose of managing the student-teacher relationship and making the student routine way easier. Such application will be completely designed on my own.

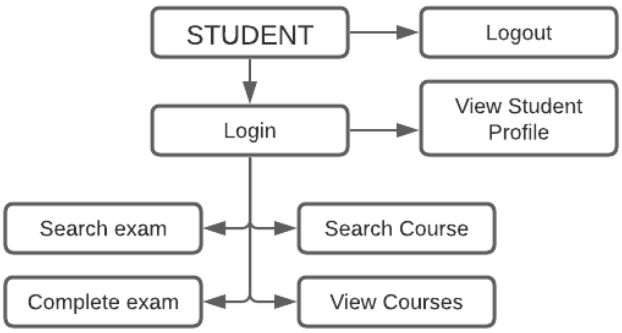
FUNCTIONAL REQUIREMENTS OF THE PROJECT

1. ***Log in***: students and professors will need to be authenticated using their user and password to enter their profiles.
2. ***Courses***: students will be able to search for a course and view its information. The professors will be able to update the information of the course.
3. ***Exams***: students will be able to search and complete the exams and the professors to add, delete or grade them.
4. ***Enrollment***: students will be able to enroll in a course, drop it or show the cost of it.
5. ***Log Out***: students and professors can sign out at any time.

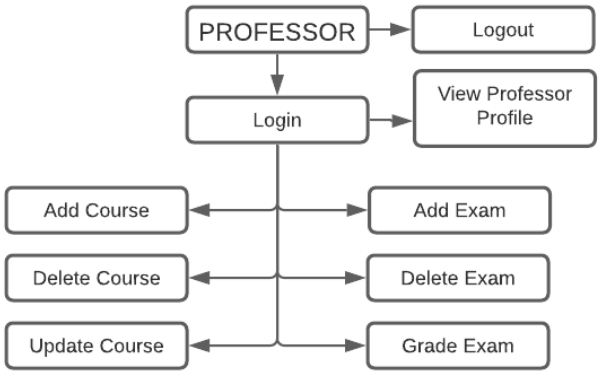
USE CASES

In summary, this project will simulate what a school management application looks like. It will be accessed by two type of users (student and professor) and each of them will have its own abilities:

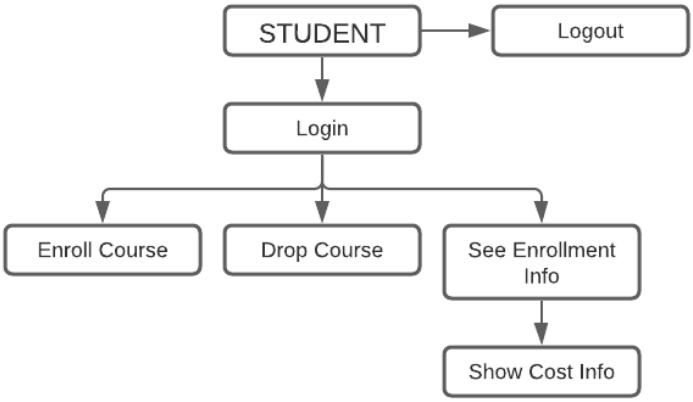
1. The first use case can refer to the student logging in/out the platform, being able to view its own profile, searching for a course, viewing its information, viewing all the courses, and also searching and completing an exam.



2. The second use case refers to the professors logging in/out the platform, being able to view its own profile, adding, deleting, or updating a course and finally, adding, deleting or grading an exam.



3. The last case refers to the students being able to enroll or drop a course and to view the information about the cost of the enrolled courses.



TECHNOLOGIES USED

- Front End:
  - JavaFX: A software platform for creating and delivering desktop applications.
- Back End:
  - Microsoft SQL Server Management Studio Technical
- Requirements:
  - Java
  - JavaFX
  - MySQL
- Applications:
  - Java Eclipse
  - MySQL Workbench
  - Microsoft Word
  - GitHub repository
  - Google Drive

PROJECT STRUCTURE

- application:
  - Main.java
- controllers:
  - AllCoursesController.java
  - LoginController.java
  - ProfessorController.java
  - ProfessorCoursesController.java
  - ProfessorExamsController.java
  - ProfessorProfileController.java
  - StudentController.java
  - StudentCoursesController.java
  - StudentEnrollmentController.java
  - StudentExamsController.java
  - StudentInfoController.java
  - StudentProfileController.java
- dao:
  - DBConnect.java
- models:
  - Course.java
  - CourseDB.java
  - Exam.java
  - ExamDB.java
  - LoginModel.java

- views:
    - AddCourseView.java
    - AddExamView.java
    - DeleteCourseView.java
    - DeleteExamView.java
    - GradeExamView.java
    - LoginView.java
    - ProfessorCoursesView.java
    - ProfessorExamView.java
    - ProfessorProfileView.java
    - ProfessorView.java
    - ShowAllCoursesView.java
    - StudentCourseView.java
    - StudentEnrollmentView.java
    - StudentExamsView.java
    - StudentInfoView.java
    - StudentProfileView.java
    - StudentView.java
    - UpdateCourseView.java
    - WelcomeView.java
- Database tables:
    - lakepark_users
    - lakepark_courses
    - lakepark_exams

STARTUP INFORMATION AND LOGIN CREDENTIALS

Startup

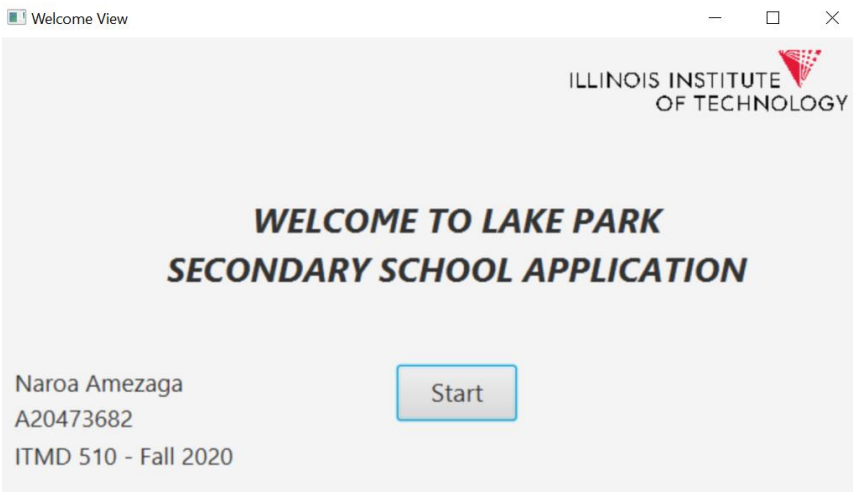
The file to execute at startup is the following: “Main.java” in the “application” package.

Credentials

The project will be accessed by two type of users:

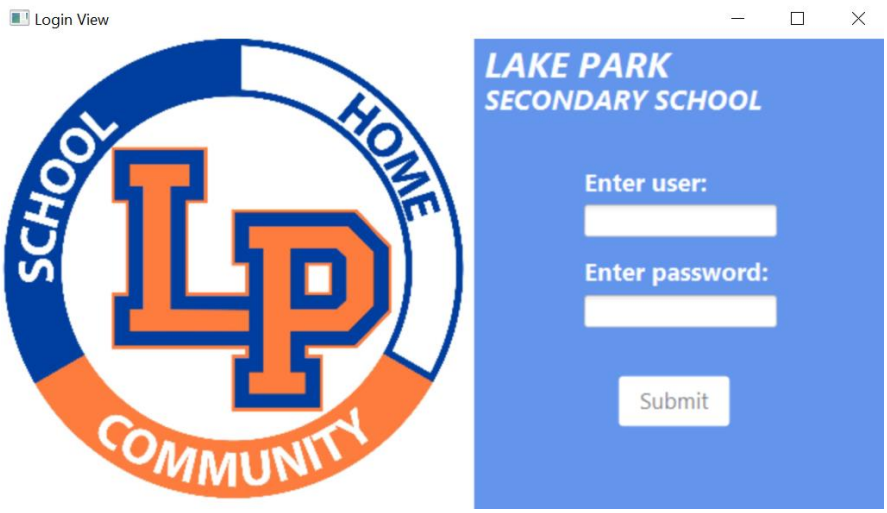
- **Student**
    - Username: student_lakepark
    - Password: student
- **Professor**
    - Username: professor_lakepark
    - Password: professor

The application starts with the following welcome scene:



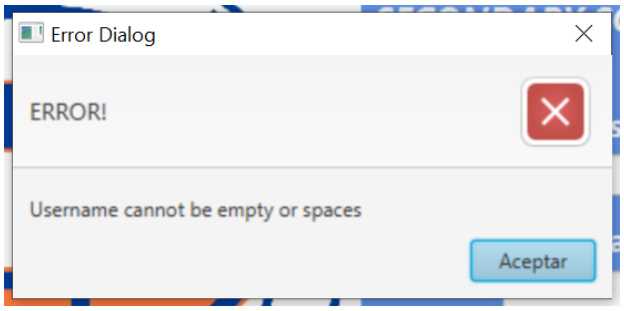
Snapshot 1 – Welcome view

When clicking in the "Start" button the application starts with the Login View in order to enter the credentials:

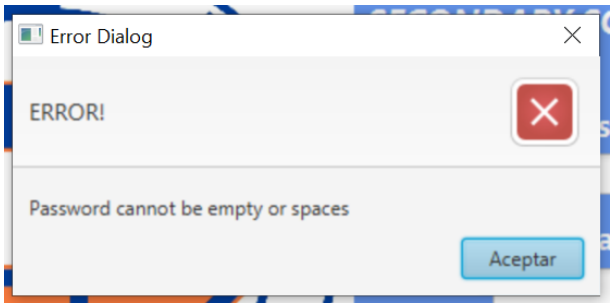


Snapshot 2 – Login view

If some of the fields (username or password) is left empty an error dialog appears:

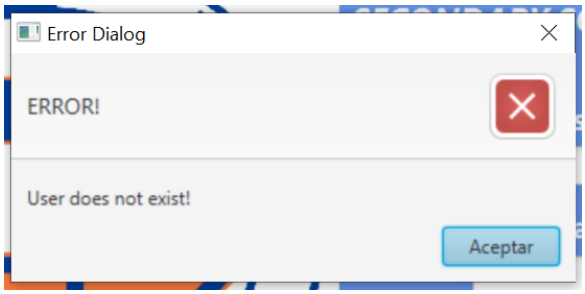


Snapshot 3 – Username empty error



Snapshot 4 – Password empty error

An error message appears also if the username or password are incorrect and the user does not exist in the database:

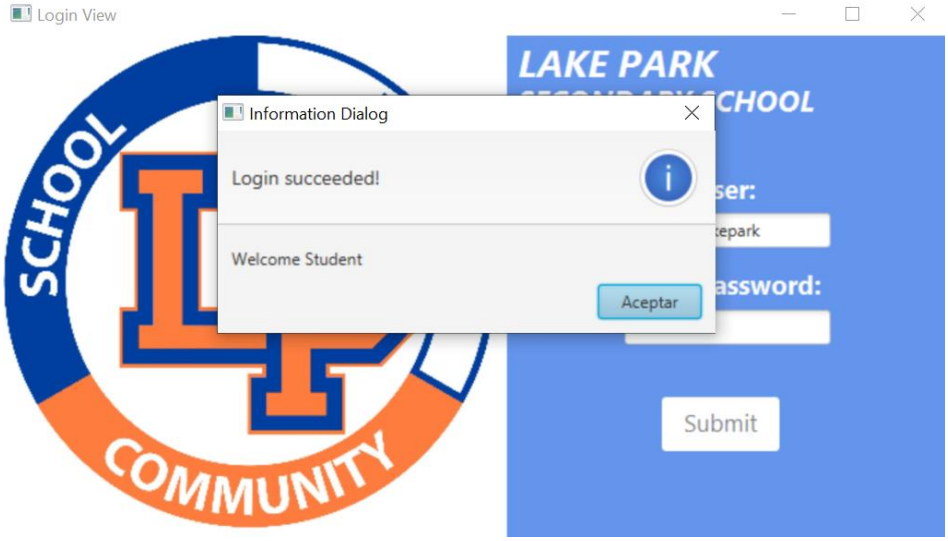


Snapshot 5 – User invalid error

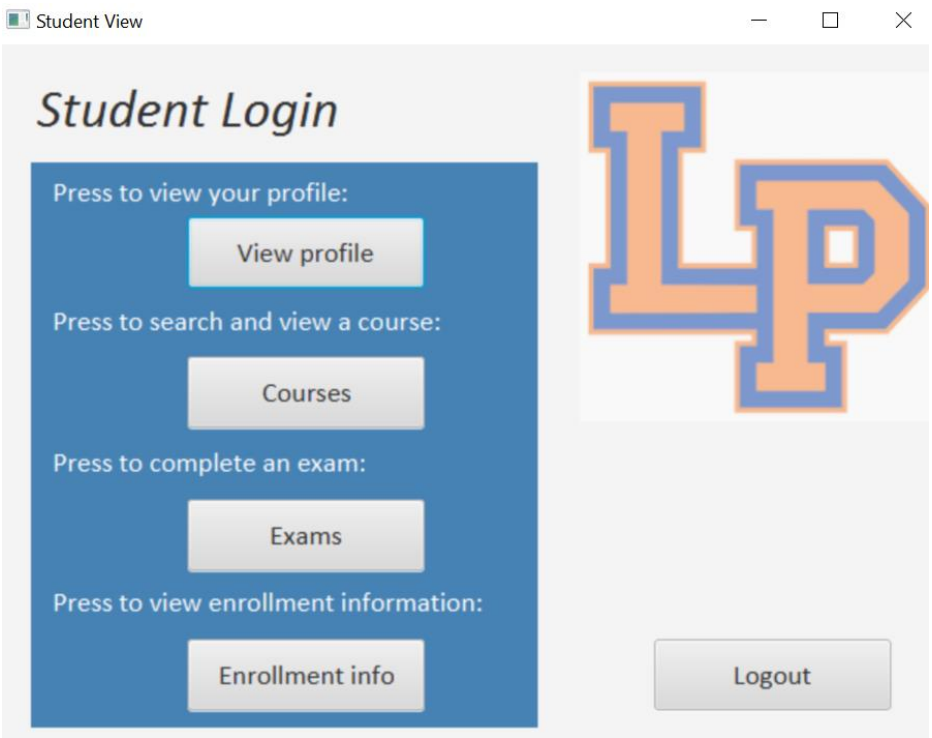
SNAPSHOTS

1. Showing a successful login for a regular user (student)

Using the student credentials mentioned above we login successfully and the main menu for the students appear.



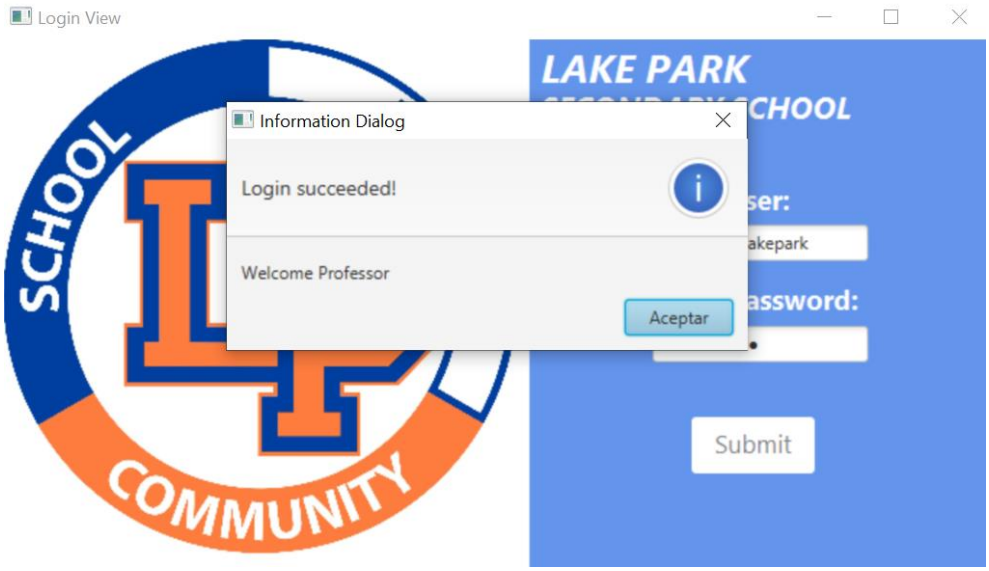
Snapshot 6 – Successful login for a regular user



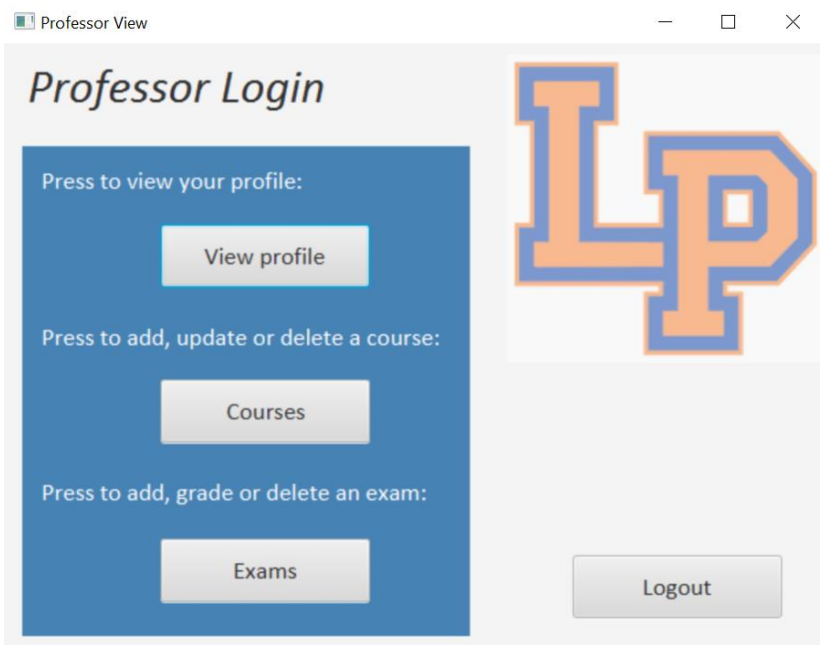
Snapshot 7 – Student main menu

2. Showing a successful login for an admin (professor)

Using the professor credentials mentioned above we login successfully and the main menu for the professors appear.



Snapshot 8 – Successful login for an admin



Snapshot 9 – Professor main menu

3. Show a result of all 10 records added to your database from the database itself

Apart from the users table, I have created two tables: lakepark_courses (table with courses) and lakepark_exams (table with exams).

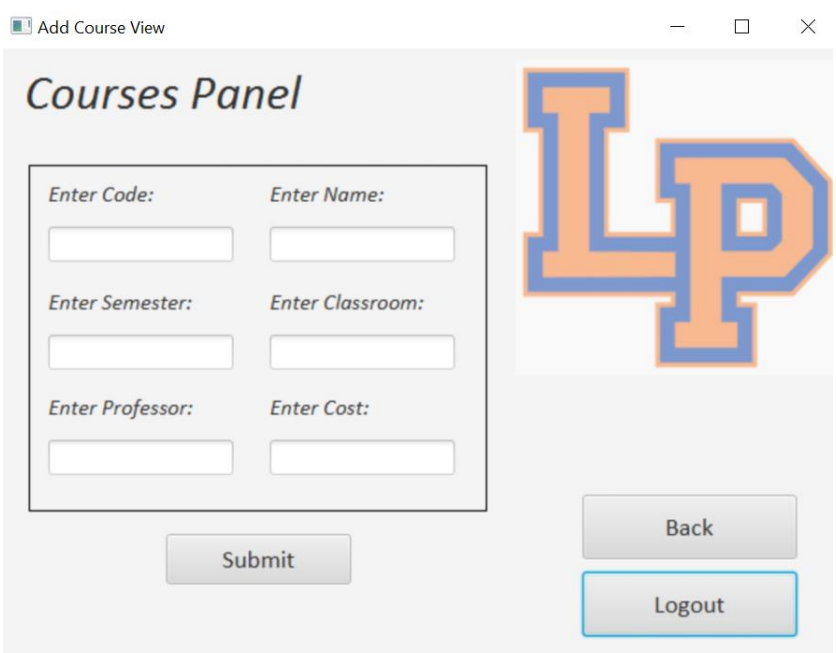
id	code	name	semester	classroom	professor	cost
1	bio	Biology	1	22	Anna Parker	102.67
2	che	Chemistry	2	8	Thomas Smith	98.87
3	geo	Geography	2	14	Eddie Tang	103
4	mat	Mathematics	1	2	Alex Sanchez	124.98
5	lat	Latin	2	67	Eva Hill	78.5
6	eng	English Culture	1	32	Sarah Delvey	99.67
7	bus	Business Economics	2	6	Mark Ford	104.78
8	his	History	1	35	Julia Blunt	96.12
9	phy	Physics	2	40	Tom Ross	120
10	mus	Music	1	18	Millie Danes	76.44

Snapshot 10 – lakepark_courses table

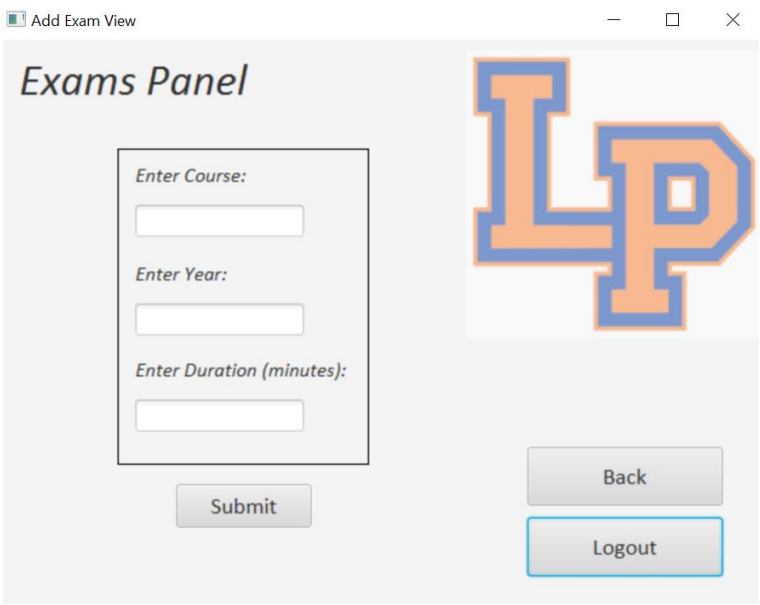
id	course	year	duration	completed	grade
1	geo	2020	90	YES	89
2	che	2020	120	NO	NULL
3	mus	2020	60	YES	75
4	mat	2020	115	YES	NULL
5	phy	2020	180	NO	NULL
6	lat	2020	60	YES	87
7	bus	2020	75	YES	NULL
8	bio	2020	90	YES	NULL
9	his	2020	120	YES	100
10	eng	2020	75	NO	NULL

Snapshot 11 – lakepark_exams table

These records can be added from a professor account by entering the information of the new course or the new exam to be added:



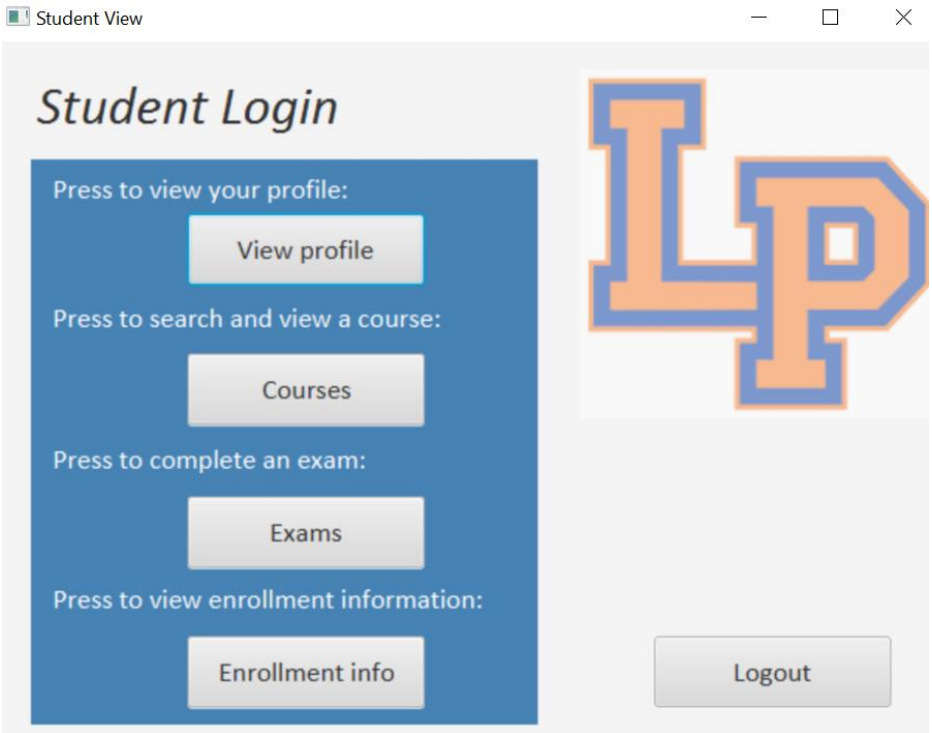
Snapshot 12 – Adding a course



Snapshot 13 – Adding a exam

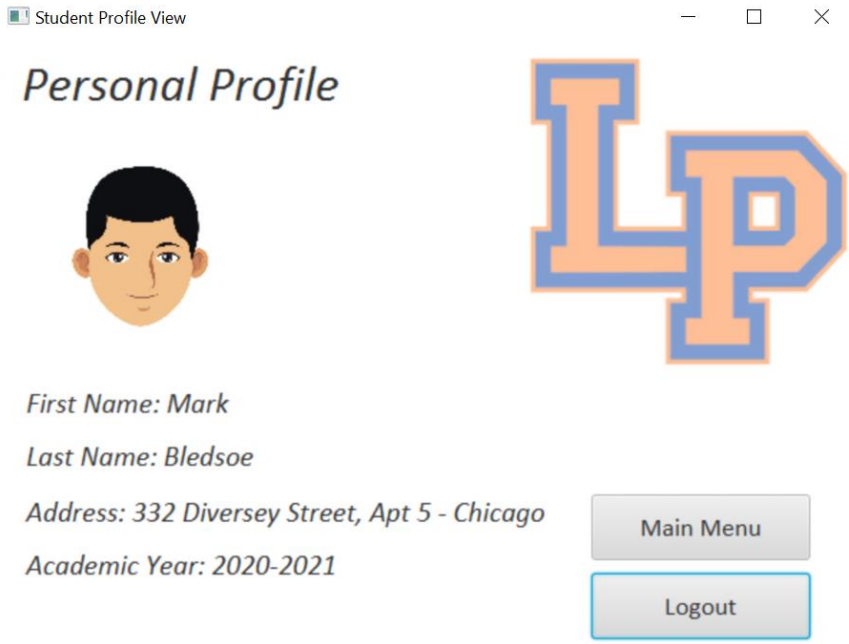
4. Demonstrate sufficient functionality for the regular user

For a student, the main menu is the following:



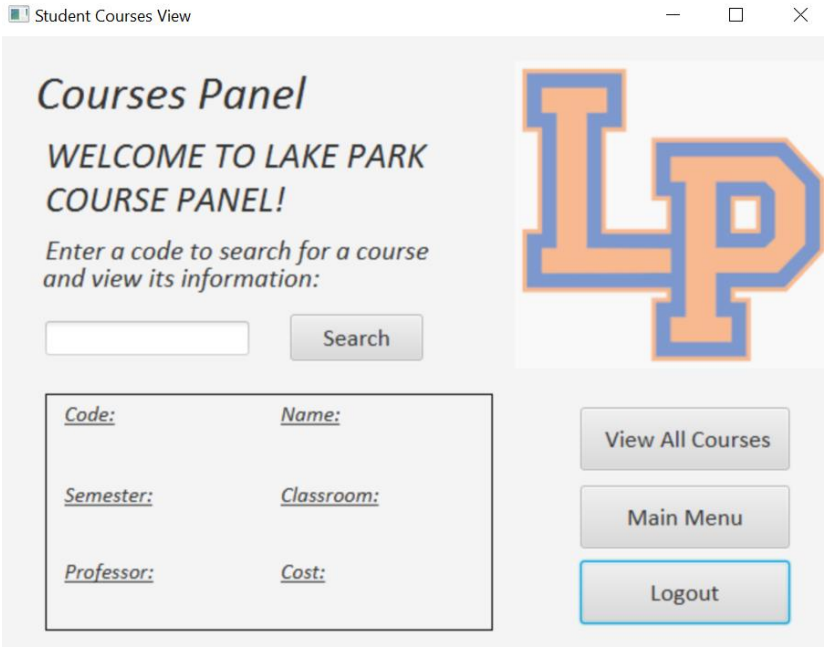
Snapshot 14 – Student main menu

- **View Profile:** the personal profile of the student can be seen.



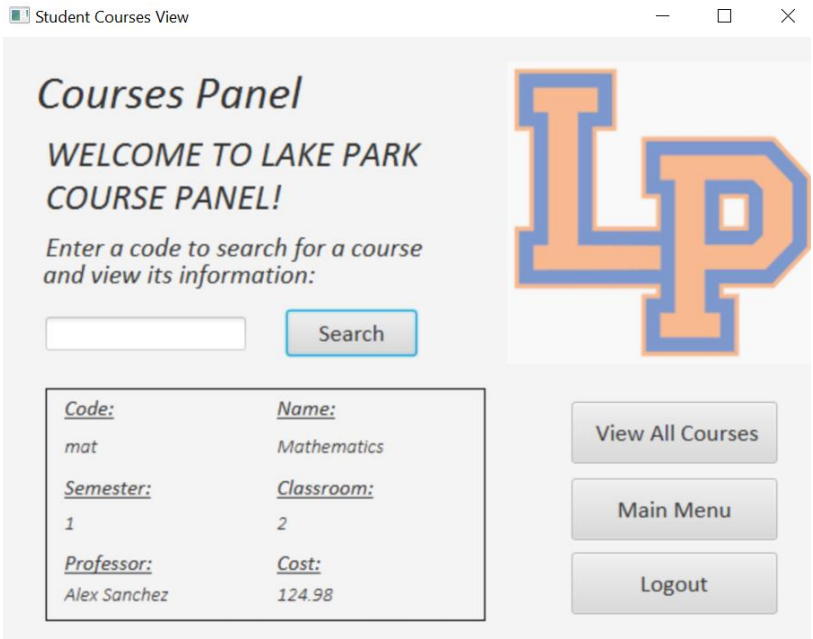
Snapshot 15 – Student Profile View

- **Courses:** the student can search for a course by its code (if a non-existing code is entered an error dialog box appears) and can also view all the courses in a table format.



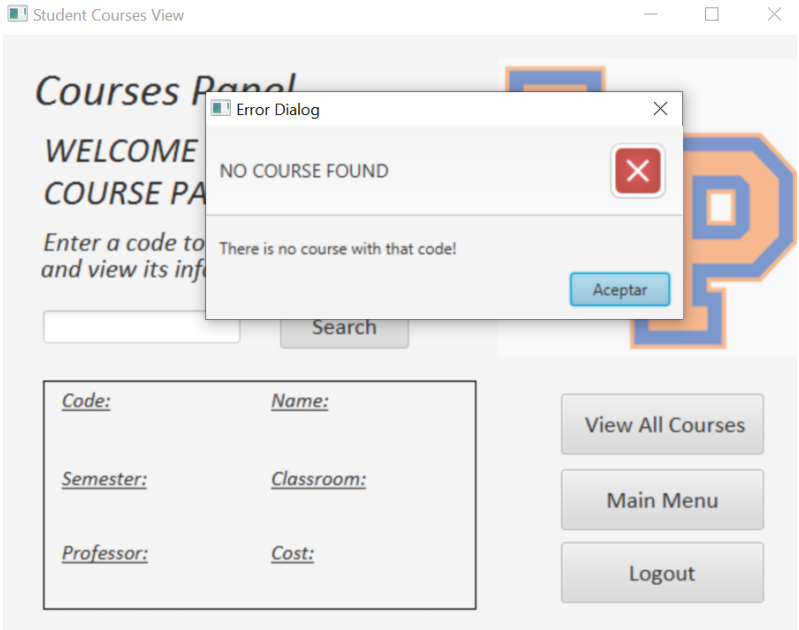
Snapshot 16 – Student Courses View

For example, if we enter the code “mat” the information related to it (name, semester, classroom, professor and cost) appears:



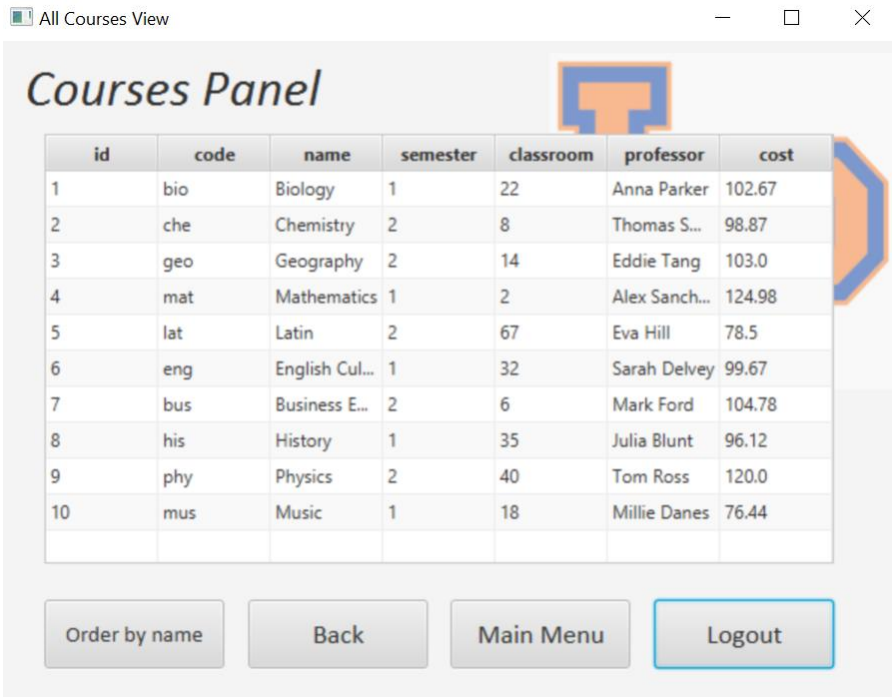
Snapshot 17 – Searching the course code “mat”

If we enter a code that is not in the database, an error dialog box appears stating that there is no course with that code:



Snapshot 17 – Error dialog box (course not found)

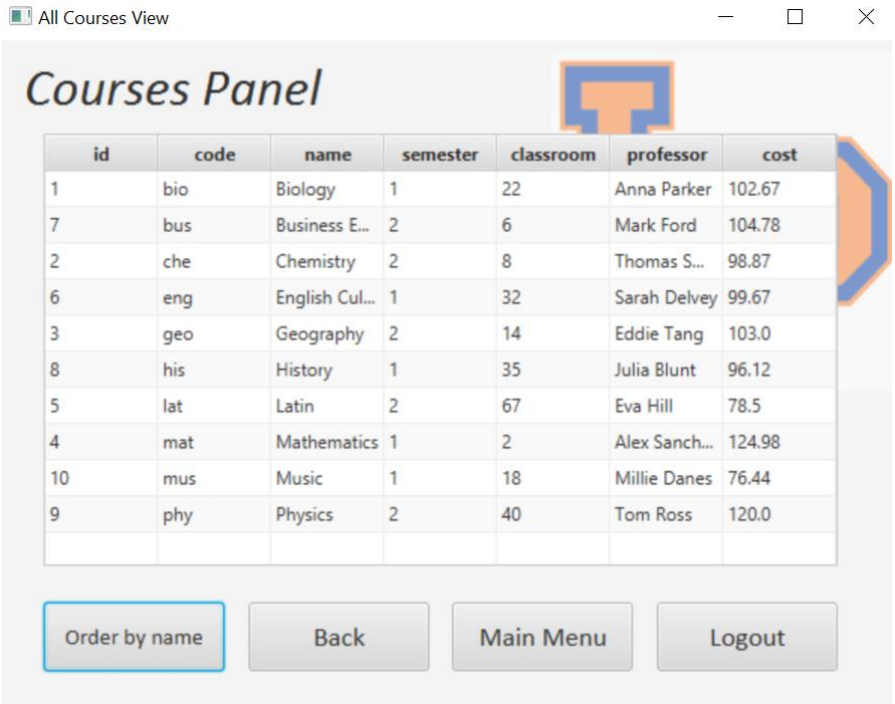
If we click on “View All Courses” button, all the courses in the table can be seen in a table:



id	code	name	semester	classroom	professor	cost
1	bio	Biology	1	22	Anna Parker	102.67
2	che	Chemistry	2	8	Thomas S...	98.87
3	geo	Geography	2	14	Eddie Tang	103.0
4	mat	Mathematics	1	2	Alex Sanch...	124.98
5	lat	Latin	2	67	Eva Hill	78.5
6	eng	English Cul...	1	32	Sarah Delvey	99.67
7	bus	Business E...	2	6	Mark Ford	104.78
8	his	History	1	35	Julia Blunt	96.12
9	phy	Physics	2	40	Tom Ross	120.0
10	mus	Music	1	18	Millie Danes	76.44

Snapshot 18 – All the courses in a table

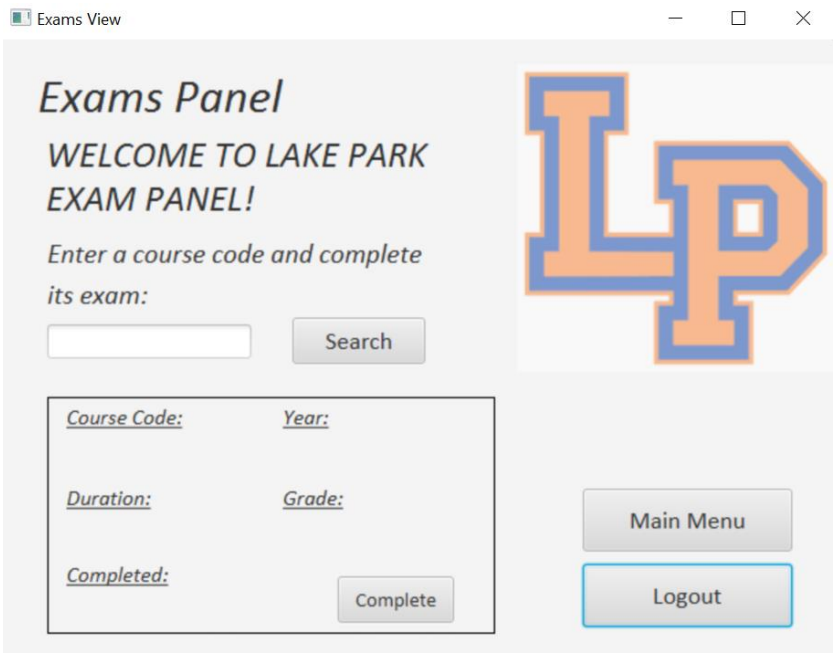
If we click on “Order by name” button, the courses are ordered by the “name” column:



id	code	name	semester	classroom	professor	cost
1	bio	Biology	1	22	Anna Parker	102.67
7	bus	Business E...	2	6	Mark Ford	104.78
2	che	Chemistry	2	8	Thomas S...	98.87
6	eng	English Cul...	1	32	Sarah Delvey	99.67
3	geo	Geography	2	14	Eddie Tang	103.0
8	his	History	1	35	Julia Blunt	96.12
5	lat	Latin	2	67	Eva Hill	78.5
4	mat	Mathematics	1	2	Alex Sanch...	124.98
10	mus	Music	1	18	Millie Danes	76.44
9	phy	Physics	2	40	Tom Ross	120.0

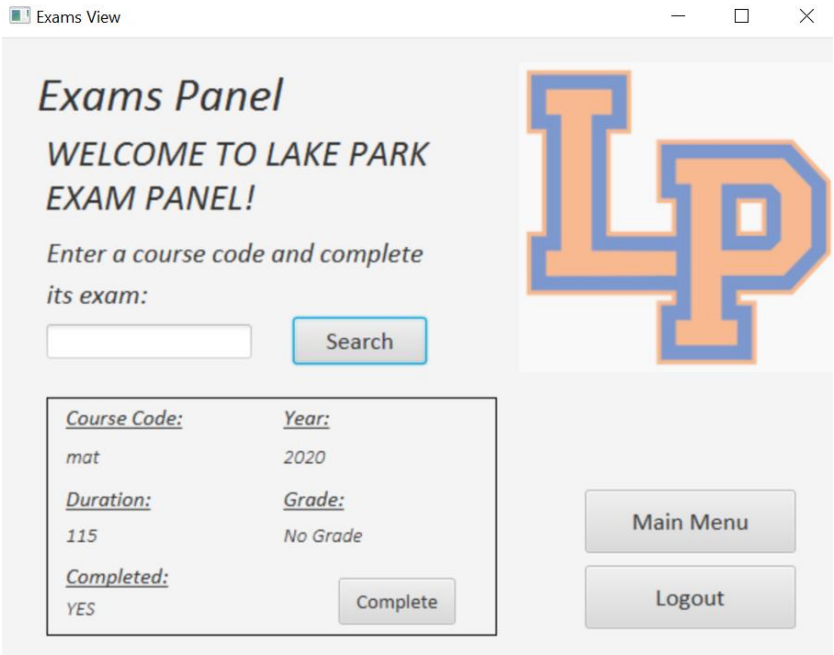
Snapshot 19 – All the courses in a table ordered by name

- **Exams:** the student can search for an exam by the course code (if a non-existing code is entered an error dialog box appears) and can also complete an exam if it is not completed (in case an exam is already completed and the "Complete" button is clicked, an error dialog box appears).



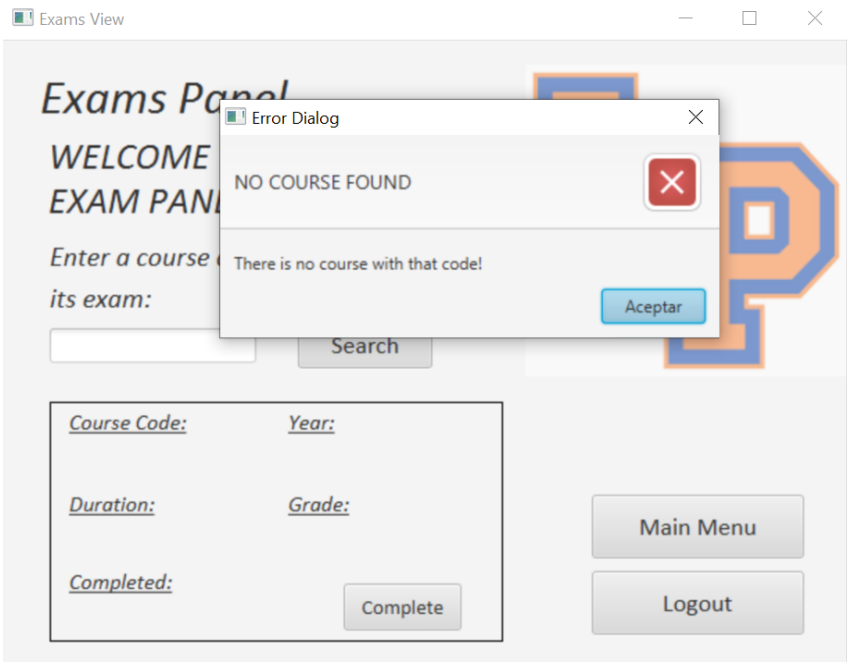
Snapshot 20 – Student Exams View

For example. if we enter the code "mat" the exam related to it appears:



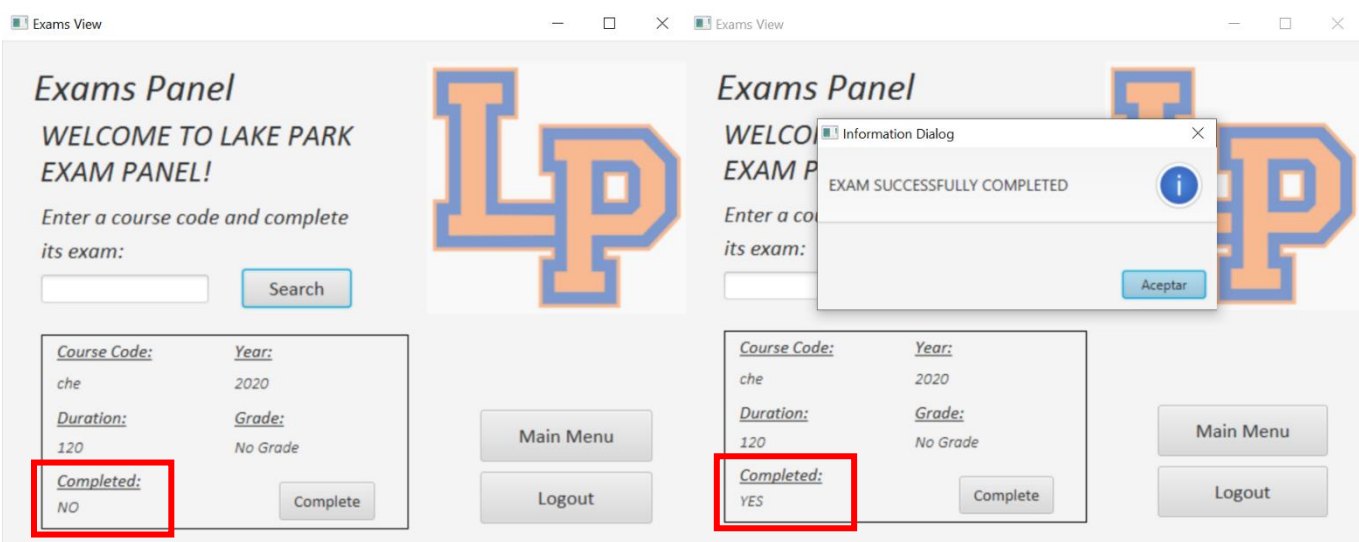
Snapshot 21 – Searching the exam with code "mat"

If we enter a code that is not in the database, an error dialog box appears stating that there is no course with that code:



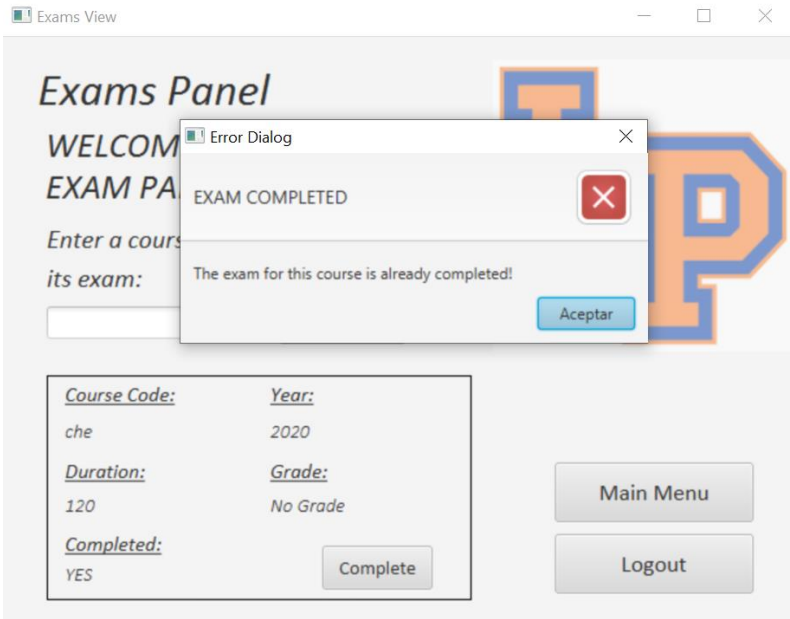
Snapshot 22 – Error dialog box (course not found)

To complete an exam, "Completed" field needs to be "NO". If it is successfully completed an information dialog box pops up. For example, in course with course code "che":



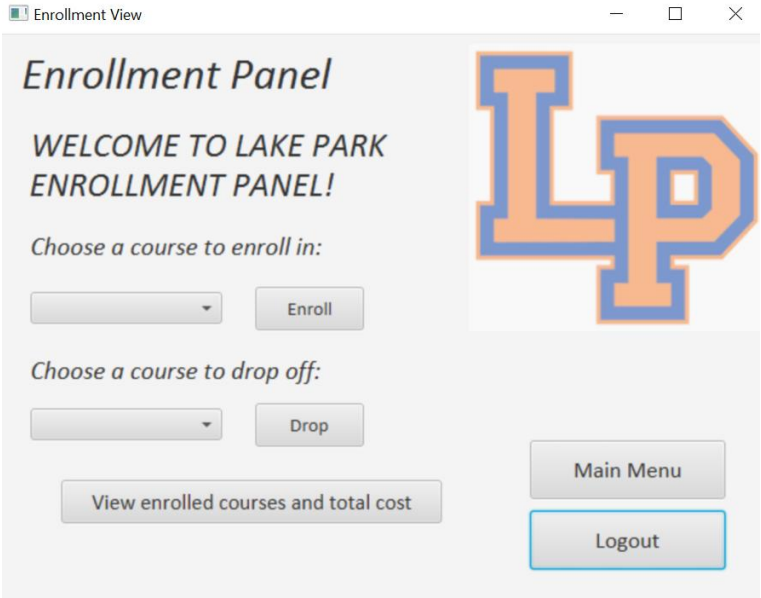
Snapshot 23 – Completing an exam successfully

If we attempt to complete an exam that it is already completed, an error dialog box will appear:



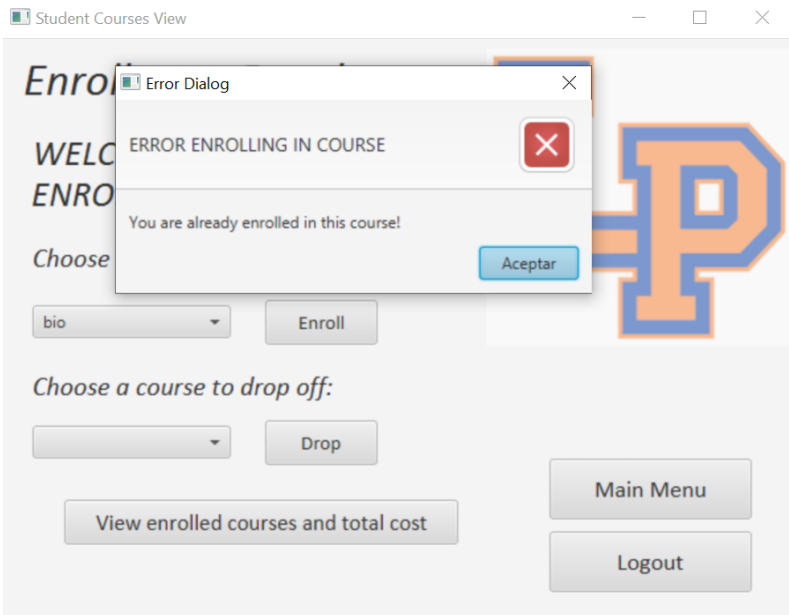
Snapshot 24 – Error dialog box (exam already completed)

- **Enrollment Info:** the student is able to enroll in courses choosing them from a combo box and then dropping off those courses he/she is enrolled in. Apart from that, the student can also view the total cost of the courses he is enrolled in.



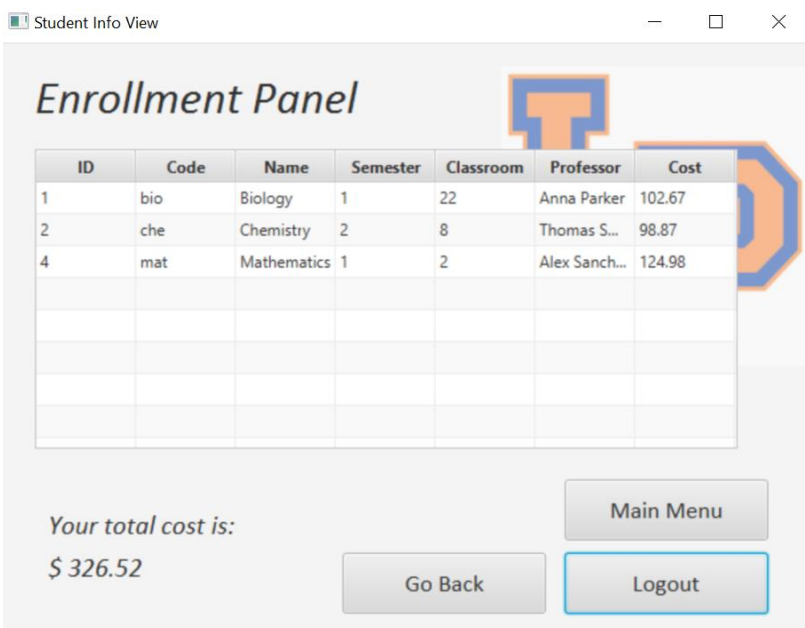
Snapshot 25 – Student Enrollment View

If the student enrolls for example in Biology, Chemistry and Mathematics, those options will appear in the drop off combo box. If the student attempts to enroll twice in a course an error dialog box will appear:



Snapshot 26 – Error dialog box (already enrolled in course)

Finally, if we click on "View enrolled courses and total cost" button, we will see the enrolled courses in a table format and the total cost (the sum of the costs of the enrolled courses):



Snapshot 27 – Student Info View

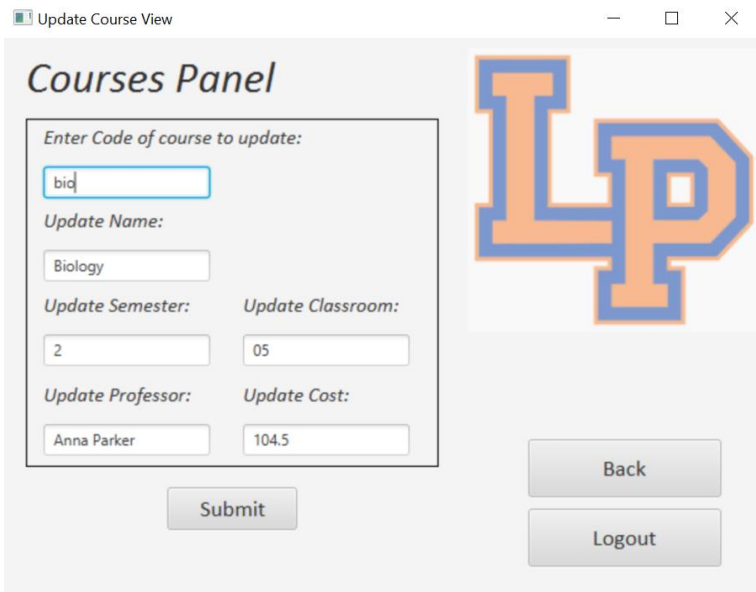
5. Show some update performed by the admin for the first record entered into the db

The first record in lakepark_courses table is the following:

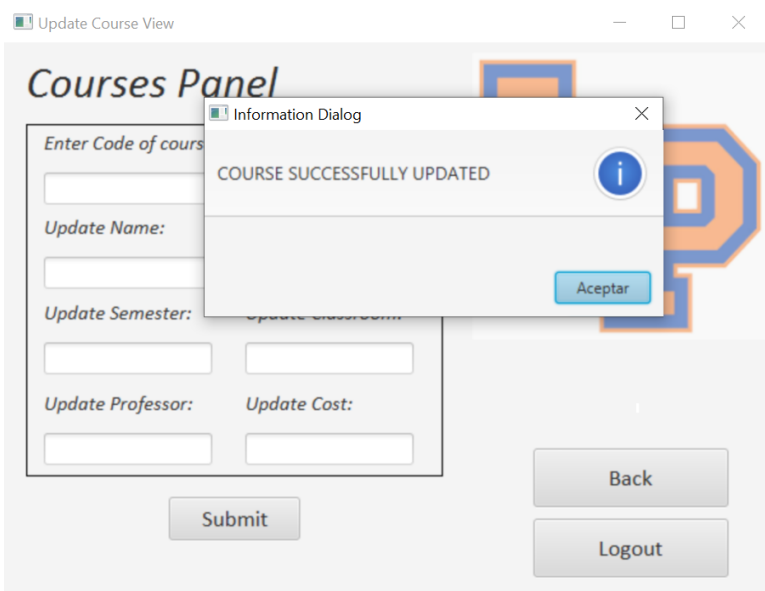
id	code	name	semester	classroom	professor	cost
1	bio	Biology	1	22	Anna Parker	102.67

Snapshot 28 – Original first record

For example, I will update the semester, the classroom, and the cost of the course:



Snapshot 29 – First record update



Snapshot 30 – Successful update dialog box

id	code	name	semester	classroom	professor	cost
1	bio	Biology	2	5	Anna Parker	104.5

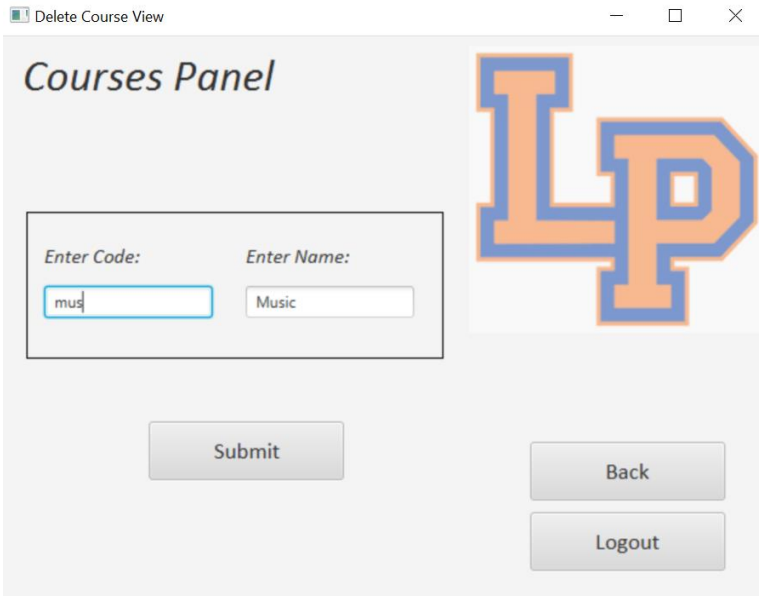
Snapshot 31 – Updated first record

6. Show the deletion of the last record by the admin entered in the db

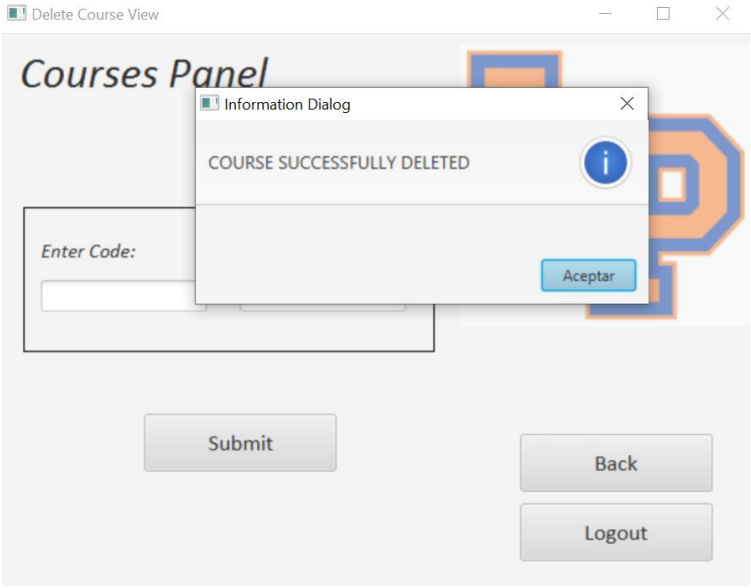
The last record in lakepark_courses table is the following:

10	mus	Music	1	18	Millie Danes	76.44

Snapshot 32 – Original last record



Snapshot 33 – Last record deletion



Snapshot 34 – Successful deletion dialog box

+ Options

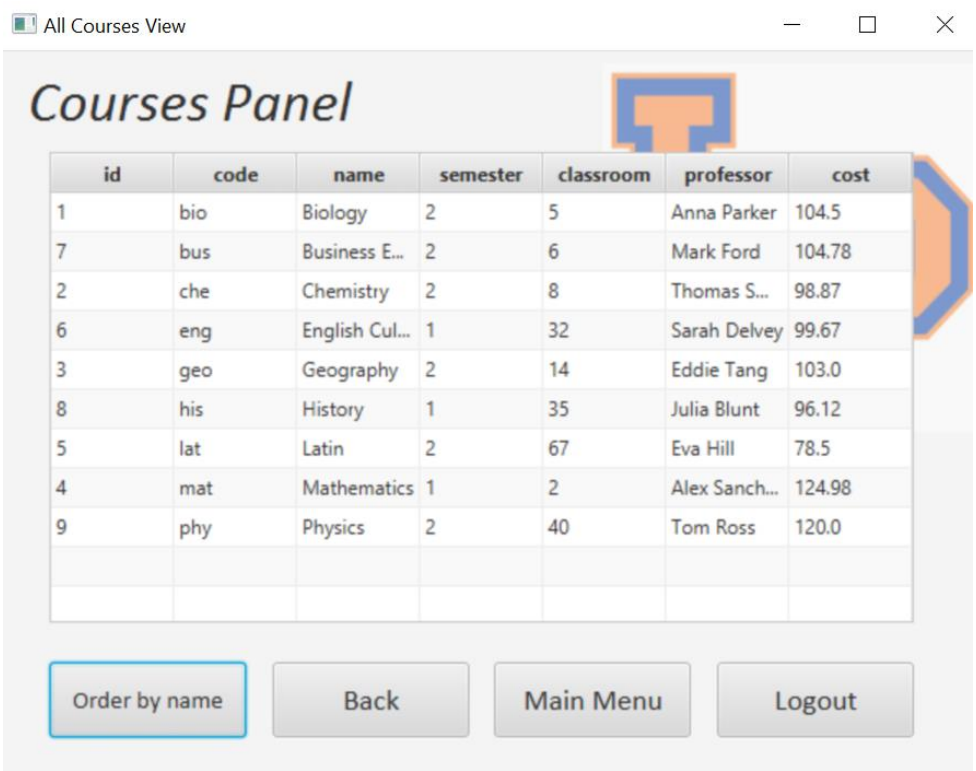
	id	code	name	semester	classroom	professor	cost
Edit Copy Delete	1	bio	Biology	2	5	Anna Parker	104.5
Edit Copy Delete	2	che	Chemistry	2	8	Thomas Smith	98.87
Edit Copy Delete	3	geo	Geography	2	14	Eddie Tang	103
Edit Copy Delete	4	mat	Mathematics	1	2	Alex Sanchez	124.98
Edit Copy Delete	5	lat	Latin	2	67	Eva Hill	78.5
Edit Copy Delete	6	eng	English Culture	1	32	Sarah Delvey	99.67
Edit Copy Delete	7	bus	Business Economics	2	6	Mark Ford	104.78
Edit Copy Delete	8	his	History	1	35	Julia Blunt	96.12
Edit Copy Delete	9	phy	Physics	2	40	Tom Ross	120

Check all *With selected:* Edit Copy Delete Export

Snapshot 35 – Table with deleted last record

7. Show a snapshot of your remaining records in some columnar view

The following snapshot shows the remaining course records ordered by name:



id	code	name	semester	classroom	professor	cost
1	bio	Biology	2	5	Anna Parker	104.5
7	bus	Business E...	2	6	Mark Ford	104.78
2	che	Chemistry	2	8	Thomas S...	98.87
6	eng	English Cul...	1	32	Sarah Delvey	99.67
3	geo	Geography	2	14	Eddie Tang	103.0
8	his	History	1	35	Julia Blunt	96.12
5	lat	Latin	2	67	Eva Hill	78.5
4	mat	Mathematics	1	2	Alex Sanch...	124.98
9	phy	Physics	2	40	Tom Ross	120.0

Snapshot 36 – Course records ordered by name

8. Show a snapshot of your user table from the database itself

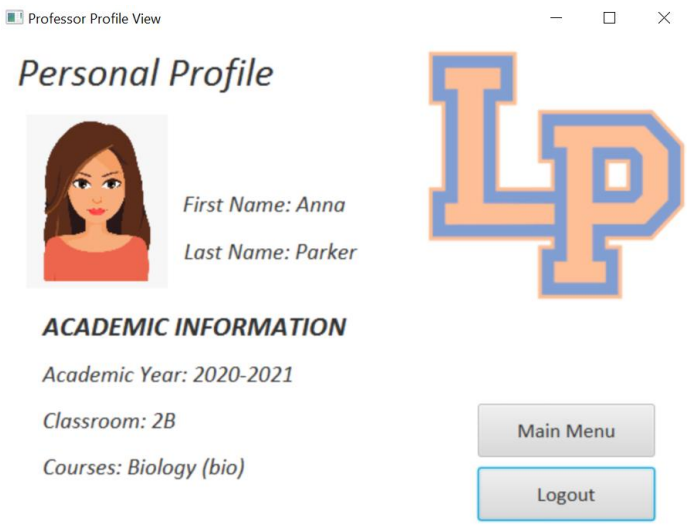
id	uname	passwd	role
1	professor_lakepark	professor	1
2	student_lakepark	student	0

Snapshot 37 – lakepark_users database table

9. Extra snapshots

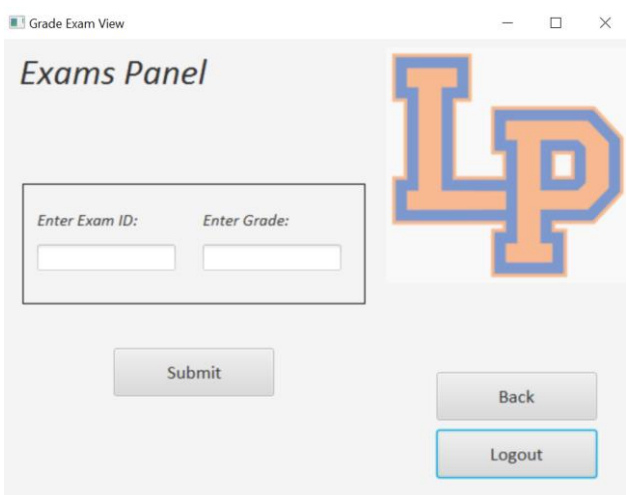
In a professor account, apart from adding, deleting, or updating a course and adding or deleting an exam, we can also do the following:

- **View its own personal profile:**



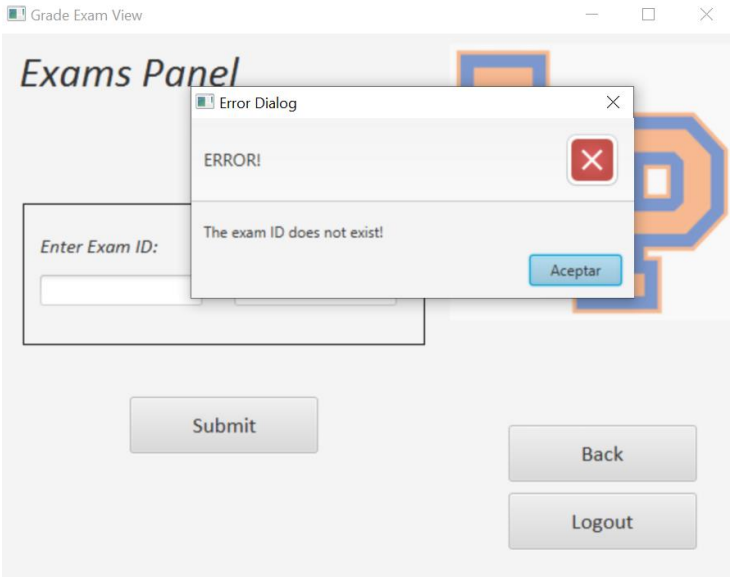
Snapshot 38 – Professor Profile View

- **Grade an exam:**



Snapshot 39 – Grade Exam View

If a non-existing Exam ID is entered, an error dialog box appears:



Snapshot 40 – Error dialog box (exam not found)

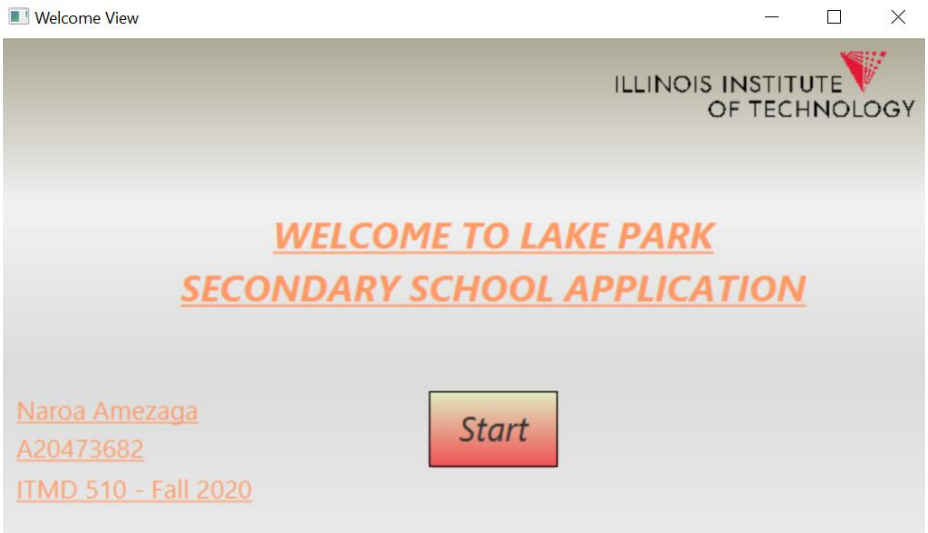
EXTRA CREDIT

Outstansing GUI

By using the following "styles.css" file in the main.java class we can change the appearance of the Welcome View scene. As an example, I have changed the background, the "Start" button, and the labels. The code in "styles.css" is the following:

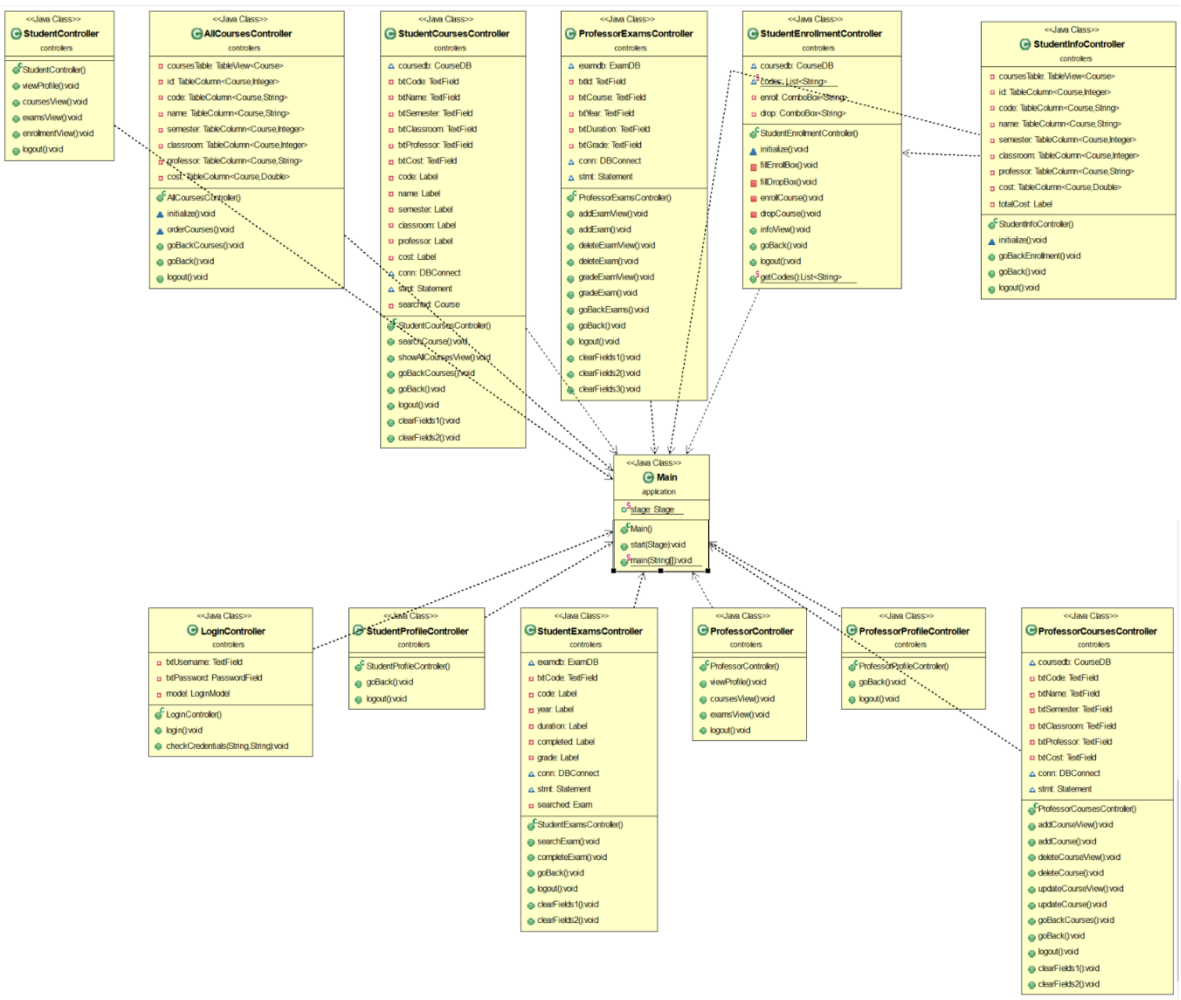
```
.root{
 -fx-background-color: linear-gradient(#ADA996, #F2F2F2,#DBDBDB,#EAEAEA);
 }

 .button{
-fx-background-color: linear-gradient(#e1eec3, #f05053);
-fx-border-color: black;
-fx-text-alignment: center;
-fx-font-size: 25 px;
-fx-font-style: italic;
 }

 .label{
 -fx-text-fill: #ff9966;
 -fx-underline: true;
 }
```

After applying it, the appearance of Welcome View is shown in the next snapshot:



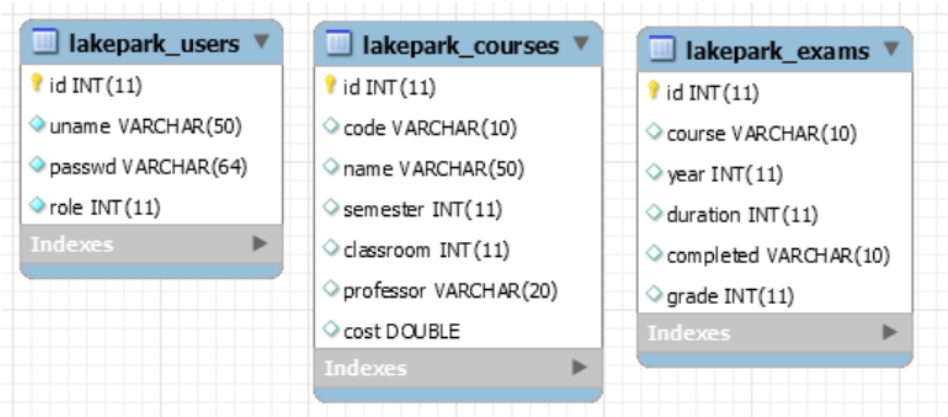
Snapshot 41 – Welcome View with styles.css

Class Diagram



Snapshot 42 – Class Diagram

Entity Relational Diagram



Snapshot 43 – Entity Relational Diagram

SOURCE CODE

APPLICATION

1. Main.java

```
package application;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.layout.AnchorPane;

public class Main extends Application {

	public static Stage stage;
	@Override
	public void start(Stage primaryStage) {

		try {

			//Set starting view: WelcomeView
			stage = primaryStage;
			AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/WelcomeView.fxml"));
			Scene scene = new Scene(root);
			 //Take comment off to see the effect of styles.css

	//scene.getStylesheets().add(getClass().getResource("styles.css").toExternalForm());
```

```
                stage.setTitle("Welcome View");
                stage.setScene(scene);
                stage.show();

            } catch(Exception e) {
                System.out.println("Error occured while inflating view: " + e);
            }
        }

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

CONTROLLERS

1. AllCoursesController.java

```
package controllers;

import application.Main;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.layout.AnchorPane;
import models.Course;
import models.CourseDB;

public class AllCoursesController {

        //fx:id in ShowAllCoursesView
        @FXML
        private TableView<Course> coursesTable;
        @FXML
        private TableColumn<Course, Integer> id;
        @FXML
        private TableColumn<Course, String> code;
        @FXML
        private TableColumn<Course, String> name;
        @FXML
        private TableColumn<Course, Integer> semester;
        @FXML
        private TableColumn<Course, Integer> classroom;
```

```
        @FXML
        private TableColumn<Course, String> professor;
        @FXML
        private TableColumn<Course, Double> cost;

        //initializing method to show all courses in a table format
        @FXML
        void initialize() {
                CourseDB dao = new CourseDB();

                id.setCellValueFactory(new PropertyValueFactory<Course, Integer>("id"));
                code.setCellValueFactory(new PropertyValueFactory<Course, String>("code"));
                name.setCellValueFactory(new PropertyValueFactory<Course, String>("name"));
                semester.setCellValueFactory(new PropertyValueFactory<Course,
Integer>("semester"));
                classroom.setCellValueFactory(new PropertyValueFactory<Course,
Integer>("classroom"));
                professor.setCellValueFactory(new PropertyValueFactory<Course,
String>("professor"));
                cost.setCellValueFactory(new PropertyValueFactory<Course, Double>("cost"));

                coursesTable.getItems().setAll(dao.listCourses());

    }

        //method to show all courses ordered by name in a table format
        @FXML
        void orderCourses() {
                CourseDB dao = new CourseDB();

                id.setCellValueFactory(new PropertyValueFactory<Course, Integer>("id"));
                code.setCellValueFactory(new PropertyValueFactory<Course, String>("code"));
                name.setCellValueFactory(new PropertyValueFactory<Course, String>("name"));
                semester.setCellValueFactory(new PropertyValueFactory<Course,
Integer>("semester"));
                classroom.setCellValueFactory(new PropertyValueFactory<Course,
Integer>("classroom"));
                professor.setCellValueFactory(new PropertyValueFactory<Course,
String>("professor"));
                cost.setCellValueFactory(new PropertyValueFactory<Course, Double>("cost"));

                coursesTable.getItems().setAll(dao.listCoursesByName());

    }
```

```
	//method to go back to StudentCoursesView
	public void goBackCourses() {
		try {
			AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/StudentCoursesView.fxml"));
			Scene scene = new Scene(root);
			Main.stage.setScene(scene);
			Main.stage.setTitle("Student Courses View");
		} catch (Exception e) {
			System.out.println("Error occured while inflating view: " + e);
		}
	}

	//method to go back to the main menu
	public void goBack() {
		try {
			AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/StudentView.fxml"));
			Scene scene = new Scene(root);
			Main.stage.setScene(scene);
			Main.stage.setTitle("Student View");
		} catch (Exception e) {
			System.out.println("Error occured while inflating view: " + e);
		}
	}

	//method to logout
	public void logout() {
		try {
			AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/LoginView.fxml"));
			Scene scene = new Scene(root);
			Main.stage.setScene(scene);
			Main.stage.setTitle("Login View");
		} catch (Exception e) {
			System.out.println("Error occured while inflating view: " + e);
		}
	}

}
```

2. LoginController.java

```
package controllers;

import application.Main;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.PasswordField;
import javafx.scene.control.TextField;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.layout.AnchorPane;
import models.LoginModel;

public class LoginController {

	//fx:id in LoginView
	@FXML
	private TextField txtUsername;

	@FXML
	private PasswordField txtPassword;

	private LoginModel model;



	public LoginController() {
		model = new LoginModel();
	}

	//method to start application by going to LoginView
	public void start() {
		try {
			AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/LoginView.fxml"));
			Scene scene = new Scene(root);
			Main.stage.setScene(scene);
			Main.stage.setTitle("Login View");
		} catch (Exception e) {
			System.out.println("Error occured while inflating view: " + e);
		}
	}
```

```
        //method to login
        public void login() {

                //get values in text fields
                String username = this.txtUsername.getText();
                String password = this.txtPassword.getText();

                //validations
                if (username == null || username.trim().equals("") && (password == null ||
password.trim().equals"")) {

                        //pop up error dialog box if username or password empty
                        Alert alert = new Alert(AlertType.ERROR);
                alert.setTitle("Error Dialog");
                alert.setHeaderText("ERROR!");
                alert.setContentText("Username / Password cannot be empty or spaces");
                alert.showAndWait();

                        return;
                }
                if (username == null || username.trim().equals("")) {

                        //pop up error dialog box if username empty
                        Alert alert = new Alert(AlertType.ERROR);
                alert.setTitle("Error Dialog");
                alert.setHeaderText("ERROR!");
                alert.setContentText("Username cannot be empty or spaces");
                alert.showAndWait();

                        return;
                }
                if (password == null || password.trim().equals("")) {

                        //pop up error dialog box if password empty
                        Alert alert = new Alert(AlertType.ERROR);
                alert.setTitle("Error Dialog");
                alert.setHeaderText("ERROR!");
                alert.setContentText("Password cannot be empty or spaces");
                alert.showAndWait();

                        return;
                }
```

```
            //authentication check
            checkCredentials(username, password);

        }

        //method to check the entered credentials
        public void checkCredentials(String username, String password) {

            Boolean isValid = model.getCredentials(username, password);

            if (!isValid) {

                //pop up error dialog box if user does not exist
                Alert alert = new Alert(AlertType.ERROR);
            alert.setTitle("Error Dialog");
            alert.setHeaderText("ERROR!");
            alert.setContentText("User does not exist!");
            alert.showAndWait();

                return;
            }

            try {
                AnchorPane root;
                if (model.isProfessor() && isValid) {

                    //pop up error dialog box if valid for professor
                    Alert alert = new Alert(AlertType.INFORMATION);
                alert.setTitle("Information Dialog");
                alert.setHeaderText("Login succeeded!");
                alert.setContentText("Welcome Professor");
                alert.showAndWait();

                    //if user is a professor, inflate professor view
                    root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/ProfessorView.fxml"));
                    Main.stage.setTitle("Professor View");

                } else {

                    //pop up error dialog box if valid for student
                    Alert alert = new Alert(AlertType.INFORMATION);
                alert.setTitle("Information Dialog");
                alert.setHeaderText("Login succeeded!");
```

```
                        alert.setContentText("Welcome Student");
                        alert.showAndWait();

                                // If user is student, inflate student view
                                root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/StudentView.fxml"));
                                Main.stage.setTitle("Student View");

                        }

                        Scene scene = new Scene(root);
                        Main.stage.setScene(scene);

                } catch (Exception e) {
                        System.out.println("Error occured while inflating view: " + e);
                        e.printStackTrace();
                }

        }
}
```

3. ProfessorController.java

```
package controllers;

import application.Main;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.scene.layout.AnchorPane;

public class ProfessorController {

        //method to set up ProfessorProfileView
        public void viewProfile() {
                try {
                        AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/ProfessorProfileView.fxml"));
                        Scene scene = new Scene(root);
                        Main.stage.setScene(scene);
                        Main.stage.setTitle("Professor Profile View");
                } catch (Exception e) {
                        System.out.println("Error occured while inflating view: " + e);
                }

        }
```

```
        //method to set up ProfessorCoursesView
        public void coursesView() {
                try {
                        AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/ProfessorCoursesView.fxml"));
                        Scene scene = new Scene(root);
                        Main.stage.setScene(scene);
                        Main.stage.setTitle("Courses View");
                } catch (Exception e) {
                        System.out.println("Error occured while inflating view: " + e);
                }

        }

        //method to set up ProfessorExamsView
        public void examsView() {
                try {
                        AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/ProfessorExamsView.fxml"));
                        Scene scene = new Scene(root);
                        Main.stage.setScene(scene);
                        Main.stage.setTitle("Exams View");
                } catch (Exception e) {
                        System.out.println("Error occured while inflating view: " + e);
                }

        }

        //method to logout
        public void logout() {
                try {
                        AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/LoginView.fxml"));
                        Scene scene = new Scene(root);
                        Main.stage.setScene(scene);
                        Main.stage.setTitle("Login View");
                } catch (Exception e) {
                        System.out.println("Error occured while inflating view: " + e);
                }
        }

}
```

4. ProfessorCoursesController.java

```
package controllers;

import java.sql.Statement;

import dao.DBConnect;
import application.Main;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.TextField;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.layout.AnchorPane;
import models.Course;
import models.CourseDB;

public class ProfessorCoursesController {


	CourseDB coursedb = new CourseDB();

	//fx:id in AddCourseView, DeleteCourseView and UpdateCourseView
	@FXML
	private TextField txtCode;
	@FXML
	private TextField txtName;
	@FXML
	private TextField txtSemester;
	@FXML
	private TextField txtClassroom;
	@FXML
	private TextField txtProfessor;
	@FXML
	private TextField txtCost;

	// Declare DB objects
	DBConnect conn = null;
	Statement stmt = null;

	public ProfessorCoursesController() {
		conn = new DBConnect();
	}
```

```
        //method to set up AddCourseView
        public void addCourseView() {

                try {
                        AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/AddCourseView.fxml"));
                        Scene scene = new Scene(root);
                        Main.stage.setScene(scene);
                        Main.stage.setTitle("Add Course View");
                } catch (Exception e) {
                        System.out.println("Error occured while inflating view: " + e);
                }

        }

        //method to add a course
        @FXML
        public void addCourse() {

                // ADD COURSE INTO COURSE TABLE
                Course course = new Course();
                course.setCode(txtCode.getText().toString());
                course.setName(txtName.getText().toString());
                course.setSemester(Integer.parseInt(txtSemester.getText().toString()));
                course.setClassroom(Integer.parseInt(txtClassroom.getText().toString()));
                course.setProfessor(txtProfessor.getText().toString());
                course.setCost(Double.parseDouble(txtCost.getText().toString()));
                clearFields1();
                coursedb.addCourse(course);

                //pop up information box when added correctly
                Alert alert = new Alert(AlertType.INFORMATION);
                alert.setTitle("Information Dialog");
                alert.setHeaderText("COURSE SUCCESSFULLY ADDED");
                alert.showAndWait();
        }


        //method to set up DeleteCourseView
        public void deleteCourseView() {

                try {
                        AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/DeleteCourseView.fxml"));
```

```
                Scene scene = new Scene(root);
                Main.stage.setScene(scene);
                Main.stage.setTitle("Delete Course View");
            } catch (Exception e) {
                System.out.println("Error occured while inflating view: " + e);
            }

        }

        //method to delete a course
        @FXML
        public void deleteCourse() {

            int ok = 0;

            //DELETE COURSE FROM COURSE TABLE
            Course course = new Course();
            course.setCode(txtCode.getText().toString());
            course.setName(txtName.getText().toString());
            clearFields2();
            ok = coursedb.deleteCourse(course);
            if(ok==1) {

                //pop up information box when deleted correctly
                Alert alert = new Alert(AlertType.INFORMATION);
                alert.setTitle("Information Dialog");
                alert.setHeaderText("COURSE SUCCESSFULLY DELETED");
                alert.showAndWait();

            }

        }

        //method to set up UpdateCourseView
        public void updateCourseView() {

            try {
                AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/UpdateCourseView.fxml"));
                Scene scene = new Scene(root);
                Main.stage.setScene(scene);
                Main.stage.setTitle("Update Course View");
            } catch (Exception e) {
                System.out.println("Error occured while inflating view: " + e);
```

```
		}

	}

	//method to update a course
	@FXML
	public void updateCourse() {

		int ok = 0;

		//UPDATE COURSE IN COURSE TABLE
		Course course = new Course();
		course.setCode(txtCode.getText().toString());
		course.setName(txtName.getText().toString());
		course.setSemester(Integer.parseInt(txtSemester.getText().toString()));
		course.setClassroom(Integer.parseInt(txtClassroom.getText().toString()));
		course.setProfessor(txtProfessor.getText().toString());
		course.setCost(Double.parseDouble(txtCost.getText().toString()));
		clearFields1();
		ok = coursedb.updateCourse(course);
		if(ok==1) {

			//pop up information box when updated correctly
			Alert alert = new Alert(AlertType.INFORMATION);
			alert.setTitle("Information Dialog");
			alert.setHeaderText("COURSE SUCCESSFULLY UPDATED");
			alert.showAndWait();

		}

	}

	//method to go back to ProfessorCoursesView
	public void goBackCourses() {
		try {
			AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/ProfessorCoursesView.fxml"));
			Scene scene = new Scene(root);
			Main.stage.setScene(scene);
			Main.stage.setTitle("Professor View");
		} catch (Exception e) {
			System.out.println("Error occured while inflating view: " + e);
		}
	}
```

```
        //method to go back to the main menu
        public void goBack() {
                try {
                        AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/ProfessorView.fxml"));
                        Scene scene = new Scene(root);
                        Main.stage.setScene(scene);
                        Main.stage.setTitle("Professor View");
                } catch (Exception e) {
                        System.out.println("Error occured while inflating view: " + e);
                }
        }

        //method to logout
        public void logout() {
                try {
                        AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/LoginView.fxml"));
                        Scene scene = new Scene(root);
                        Main.stage.setScene(scene);
                        Main.stage.setTitle("Login View");
                } catch (Exception e) {
                        System.out.println("Error occured while inflating view: " + e);
                }
        }

        //methods to clear values in fields
        public void clearFields1() {
                txtCode.clear();
                txtName.clear();
                txtSemester.clear();
                txtClassroom.clear();
                txtProfessor.clear();
                txtCost.clear();
        }
        public void clearFields2() {

                txtCode.clear();
                txtName.clear();

        }

}
```

5. ProfessorExamsController.java

```
package controllers;

import java.sql.Statement;

import application.Main;
import dao.DBConnect;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.TextField;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.layout.AnchorPane;
import models.Exam;
import models.ExamDB;

public class ProfessorExamsController {

	ExamDB examdb = new ExamDB();

	//fx:id in AddExamView, DeleteExamView and GradeExamView
	@FXML
	private TextField txtId;
	@FXML
	private TextField txtCourse;
	@FXML
	private TextField txtYear;
	@FXML
	private TextField txtDuration;
	@FXML
	private TextField txtGrade;

	// Declare DB objects
	DBConnect conn = null;
	Statement stmt = null;

	public ProfessorExamsController() {
		conn = new DBConnect();
	}

	//method to set up AddExamView
	public void addExamView() {
```

```
		try {
			AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/AddExamView.fxml"));
			Scene scene = new Scene(root);
			Main.stage.setScene(scene);
			Main.stage.setTitle("Add Exam View");
		} catch (Exception e) {
			System.out.println("Error occured while inflating view: " + e);
		}

	}

	//method to add exam
	@FXML
	public void addExam() {

		// ADD EXAM INTO EXAM TABLE
		Exam exam = new Exam();
		exam.setCourseCode(txtCourse.getText().toString());
		exam.setYear(Integer.parseInt(txtYear.getText().toString()));
		exam.setDuration(Integer.parseInt(txtDuration.getText().toString()));
		clearFields1();
		examdb.addExam(exam);

		//pop up information box when added correctly
		Alert alert = new Alert(AlertType.INFORMATION);
		alert.setTitle("Information Dialog");
		alert.setHeaderText("EXAM SUCCESSFULLY ADDED");
		alert.showAndWait();
	}

	//method to set up DeleteExamView
	public void deleteExamView() {

		try {
			AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/DeleteExamView.fxml"));
			Scene scene = new Scene(root);
			Main.stage.setScene(scene);
			Main.stage.setTitle("Delete Exam View");
		} catch (Exception e) {
			System.out.println("Error occured while inflating view: " + e);
		}
```

```
        }

        //method to delete exam
        @FXML
        public void deleteExam() {

                int ok = 0;

                //DELETE EXAM FROM EXAM TABLE
                Exam exam = new Exam();
                exam.setId(Integer.parseInt(txtId.getText().toString()));
                clearFields2();
                ok = examdb.deleteExam(exam);
                if(ok==1) {

                        //pop up information box when deleted correctly
                        Alert alert = new Alert(AlertType.INFORMATION);
                        alert.setTitle("Information Dialog");
                        alert.setHeaderText("EXAM SUCCESSFULLY DELETED");
                        alert.showAndWait();

                }

        }

        //method to set up GradeExamView
        public void gradeExamView() {

                try {
                        AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/GradeExamView.fxml"));
                        Scene scene = new Scene(root);
                        Main.stage.setScene(scene);
                        Main.stage.setTitle("Grade Exam View");
                } catch (Exception e) {
                        System.out.println("Error occured while inflating view: " + e);
                }

        }

        //method to grade exam
        @FXML
        public void gradeExam() {
```

```
            int ok = 0;

            //GRADE EXAM
            Exam exam = new Exam();
            exam.setId(Integer.parseInt(txtId.getText().toString()));
            exam.setGrade(Integer.parseInt(txtGrade.getText().toString()));
            clearFields3();
            ok=examdb.gradeExam(exam);
            if(ok==1) {

                //pop up information box when graded correctly
                Alert alert = new Alert(AlertType.INFORMATION);
                alert.setTitle("Information Dialog");
                alert.setHeaderText("EXAM SUCCESSFULLY UPDATED");
                alert.showAndWait();

            }

        }

        //method to go back to ProfessorExamsView
        public void goBackExams() {
            try {
                AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/ProfessorExamsView.fxml"));
                Scene scene = new Scene(root);
                Main.stage.setScene(scene);
                Main.stage.setTitle("Professor View");
            } catch (Exception e) {
                System.out.println("Error occured while inflating view: " + e);
            }
        }

        //method to go back to main menu
        public void goBack() {
            try {
                AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/ProfessorView.fxml"));
                Scene scene = new Scene(root);
                Main.stage.setScene(scene);
                Main.stage.setTitle("Professor View");
            } catch (Exception e) {
                System.out.println("Error occured while inflating view: " + e);
            }
```

```
        }

        //method to logout
        public void logout() {
                try {
                        AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/LoginView.fxml"));
                        Scene scene = new Scene(root);
                        Main.stage.setScene(scene);
                        Main.stage.setTitle("Login View");
                } catch (Exception e) {
                        System.out.println("Error occured while inflating view: " + e);
                }
        }

        //methods to clear values in fields

        public void clearFields1() {

                txtCourse.clear();
                txtYear.clear();
                txtDuration.clear();

        }

        public void clearFields2() {

                txtId.clear();

        }

        public void clearFields3() {

                txtId.clear();
                txtGrade.clear();
        }

}
```

6. ProfessorProfileController.java

```
package controllers;

import application.Main;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.scene.layout.AnchorPane;

public class ProfessorProfileController {

	//method to go back to the main menu
	public void goBack() {
		try {
			AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/ProfessorView.fxml"));
			Scene scene = new Scene(root);
			Main.stage.setScene(scene);
			Main.stage.setTitle("Professor View");
		} catch (Exception e) {
			System.out.println("Error occured while inflating view: " + e);
		}
	}

	//method to logout
	public void logout() {
		try {
			AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/LoginView.fxml"));
			Scene scene = new Scene(root);
			Main.stage.setScene(scene);
			Main.stage.setTitle("Login View");
		} catch (Exception e) {
			System.out.println("Error occured while inflating view: " + e);
		}
	}

}
```

7. StudentController.java

```
package controllers;

import application.Main;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.scene.layout.AnchorPane;

public class StudentController {

	//method to set up StudentProfileView
	public void viewProfile() {
		try {
			AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/StudentProfileView.fxml"));
			Scene scene = new Scene(root);
			Main.stage.setScene(scene);
			Main.stage.setTitle("Student Profile View");
		} catch (Exception e) {
			System.out.println("Error occured while inflating view: " + e);
		}

	}

	//method to set up StudentCoursesView
	public void coursesView() {
		try {
			AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/StudentCoursesView.fxml"));
			Scene scene = new Scene(root);
			Main.stage.setScene(scene);
			Main.stage.setTitle("Courses View");
		} catch (Exception e) {
			e.printStackTrace();
		}

	}

	//method to set up StudentExamsView
	public void examsView() {
		try {
			AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/StudentExamsView.fxml"));
			Scene scene = new Scene(root);
```

```
                Main.stage.setScene(scene);
                Main.stage.setTitle("Exams View");
            } catch (Exception e) {
                e.printStackTrace();
            }

        }

        //method to set up StudentEnrollmentView
        public void enrollmentView() {
            try {
                AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/StudentEnrollmentView.fxml"));
                Scene scene = new Scene(root);
                Main.stage.setScene(scene);
                Main.stage.setTitle("Enrollment View");
            } catch (Exception e) {
                e.printStackTrace();
            }

        }

        //method to logout
        public void logout() {
            try {
                AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/LoginView.fxml"));
                Scene scene = new Scene(root);
                Main.stage.setScene(scene);
                Main.stage.setTitle("Login View");
            } catch (Exception e) {
                System.out.println("Error occured while inflating view: " + e);
            }
        }

}
```

8. StudentCoursesController.java

```
package controllers;

import java.sql.Statement;

import application.Main;
import dao.DBConnect;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.layout.AnchorPane;
import models.Course;
import models.CourseDB;

public class StudentCoursesController {

	CourseDB coursedb = new CourseDB();

	//fx:id in StudentCoursesView
	@FXML
	private TextField txtCode;
	@FXML
	private TextField txtName;
	@FXML
	private TextField txtSemester;
	@FXML
	private TextField txtClassroom;
	@FXML
	private TextField txtProfessor;
	@FXML
	private TextField txtCost;

	@FXML
	private Label code;
	@FXML
	private Label name;
	@FXML
	private Label semester;
	@FXML
	private Label classroom;
```

```
@FXML
private Label professor;
@FXML
private Label cost;


// Declare DB objects
DBConnect conn = null;
Statement stmt = null;

private Course searched;

public StudentCoursesController() {
        conn = new DBConnect();
}

//method to search course when entering code
@FXML
public void searchCourse() {

        // SEARCH COURSE IN COURSE TABLE
        Course course = new Course();
        course.setCode(txtCode.getText().toString());
        clearFields1();

        searched = coursedb.searchCourse(course);

        if(searched.getSemester()==0) {

                clearFields2();

                //pop up error box when no course found
                Alert alert = new Alert(AlertType.ERROR);
                alert.setTitle("Error Dialog");
                alert.setHeaderText("NO COURSE FOUND");
                alert.setContentText("There is no course with that code!");
                alert.showAndWait();


        }else {

                code.setText(searched.getCode());
                name.setText(searched.getName());
                semester.setText(Integer.toString(searched.getSemester()));
```

```
                        classroom.setText(Integer.toString(searched.getClassroom()));
                        professor.setText(searched.getProfessor());
                        cost.setText(Double.toString(searched.getCost()));

                }

        }

        //method to set up ShowAllCoursesView
        public void showAllCoursesView() {

                try {
                        AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/ShowAllCoursesView.fxml"));
                        Scene scene = new Scene(root);
                        Main.stage.setScene(scene);
                        Main.stage.setTitle("All Courses View");
                } catch (Exception e) {
                        e.printStackTrace();
                }

        }

        //method to go back to StudentCoursesView
        public void goBackCourses() {
                try {
                        AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/StudentCoursesView.fxml"));
                        Scene scene = new Scene(root);
                        Main.stage.setScene(scene);
                        Main.stage.setTitle("Student Courses View");
                } catch (Exception e) {
                        System.out.println("Error occured while inflating view: " + e);
                }
        }

        //method to go back to the main menu
        public void goBack() {
                try {
                        AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/StudentView.fxml"));
                        Scene scene = new Scene(root);
                        Main.stage.setScene(scene);
                        Main.stage.setTitle("Student View");
```

```
		} catch (Exception e) {
			System.out.println("Error occured while inflating view: " + e);
		}
	}

	//method to logout
	public void logout() {
		try {
			AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/LoginView.fxml"));
			Scene scene = new Scene(root);
			Main.stage.setScene(scene);
			Main.stage.setTitle("Login View");
		} catch (Exception e) {
			System.out.println("Error occured while inflating view: " + e);
		}
	}

	//methods to clear values in fields

	public void clearFields1() {
		txtCode.clear();
	}

	public void clearFields2() {

		code.setText("");
		name.setText("");
		semester.setText("");
		classroom.setText("");
		professor.setText("");
		cost.setText("");

	}

}
```

9. StudentEnrollmentController.java

```
package controllers;

import java.util.ArrayList;
import java.util.List;

import application.Main;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.ComboBox;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.layout.AnchorPane;
import models.CourseDB;

public class StudentEnrollmentController {

	CourseDB coursedb = new CourseDB();
	static List<String> codes = new ArrayList<String>();

	//fx:id in StudentEnrollmentView
  @FXML
  private ComboBox<String> enroll;
  @FXML
  private ComboBox<String> drop;

	//initializing method
	@FXML
  void initialize() {
		fillEnrollBox();
		fillDropBox();
  }

	//method to fill the enrolling combo box
  private void fillEnrollBox() {
		enroll.getItems().clear();
		enroll.getItems().addAll(coursedb.listCourseCodes());
	}

  //method to fill the drop combo box
  private void fillDropBox() {
		drop.getItems().clear();
		drop.getItems().addAll(codes);
```

```
        }

    //method to enroll in a course
    @FXML
    private void enrollCourse(){

        boolean out = true;

        if(codes.isEmpty()) {

            codes.add(enroll.getValue());

                //pop up information box when an enrollment is successful
            Alert alert = new Alert(AlertType.INFORMATION);
            alert.setTitle("Information Dialog");
            alert.setHeaderText("SUCCESSFUL ENROLLMENT IN COURSE");
            alert.setContentText("You have successfully enrolled in the course!");
            alert.showAndWait();

        }else {

            for (int i = 0; i < codes.size() && out; i++) {
                if(codes.get(i).equals(enroll.getValue())) {
                    out=false;

                    //pop up error box when you are already enrolled in a course
                Alert alert = new Alert(AlertType.ERROR);
                alert.setTitle("Error Dialog");
                alert.setHeaderText("ERROR ENROLLING IN COURSE");
                alert.setContentText("You are already enrolled in this course!");
                alert.showAndWait();

                }
            }
            if (out==true) {
                codes.add(enroll.getValue());

                //pop up information box when an enrollment is successful
                Alert alert = new Alert(AlertType.INFORMATION);
                alert.setTitle("Information Dialog");
                alert.setHeaderText("SUCCESSFUL ENROLLMENT IN COURSE");
                alert.setContentText("You have successfully enrolled in the course!");
                alert.showAndWait();
```

```
                }

            }
        fillDropBox();
        enroll.getSelectionModel().clearSelection();

    }

    //method to drop a course
    @FXML
    private void dropCourse(){

        codes.remove(drop.getValue());

                //pop up information box when a drop is successful
        Alert alert = new Alert(AlertType.INFORMATION);
                alert.setTitle("Information Dialog");
                alert.setHeaderText("SUCCESSFUL DROP IN COURSE");
                alert.setContentText("You have successfully dropped off the course!");
                alert.showAndWait();

        drop.getSelectionModel().clearSelection();
        fillDropBox();

    }

            //method to set up StudentInfoView
            @FXML
            public void infoView() {
                    try {
                            AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/StudentInfoView.fxml"));
                            Scene scene = new Scene(root);
                            Main.stage.setScene(scene);
                            Main.stage.setTitle("Student Info View");
                    } catch (Exception e) {
                            e.printStackTrace();

                    }
            }

            //method to go back to main menu
            public void goBack() {
                    try {
```

```
                        AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/StudentView.fxml"));
                        Scene scene = new Scene(root);
                        Main.stage.setScene(scene);
                        Main.stage.setTitle("Student View");
                } catch (Exception e) {
                        System.out.println("Error occured while inflating view: " + e);
                }
        }

        //method to logout
        public void logout() {
                try {
                        AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/LoginView.fxml"));
                        Scene scene = new Scene(root);
                        Main.stage.setScene(scene);
                        Main.stage.setTitle("Login View");
                } catch (Exception e) {
                        System.out.println("Error occured while inflating view: " + e);
                }
        }

        //codes getter
        public static List<String> getCodes() {
                return codes;
        }

}
```

10. StudentExamsController.java

```
package controllers;

import java.sql.Statement;

import application.Main;
import dao.DBConnect;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.control.Alert.AlertType;
```

```
import javafx.scene.layout.AnchorPane;
import models.Exam;
import models.ExamDB;

public class StudentExamsController {

	ExamDB examdb = new ExamDB();

	//fx:id in StudentExamsView
	@FXML
	private TextField txtCode;

	@FXML
	private Label code;
	@FXML
	private Label year;
	@FXML
	private Label duration;
	@FXML
	private Label completed;
	@FXML
	private Label grade;

	// Declare DB objects
	DBConnect conn = null;
	Statement stmt = null;

	private Exam searched;

	public StudentExamsController() {
		conn = new DBConnect();
	}

	//method to search exam when entering code
	@FXML
	public void searchExam() {

		// SEARCH EXAM IN COURSE TABLE
		Exam exam = new Exam();
		exam.setCourseCode(txtCode.getText().toString());
		clearFields1();

		searched = examdb.searchExam(exam);
```

```
        if(searched.getDuration()==0) {

                clearFields2();

                //pop up error box when no course found
        Alert alert = new Alert(AlertType.ERROR);
        alert.setTitle("Error Dialog");
        alert.setHeaderText("NO COURSE FOUND");
        alert.setContentText("There is no course with that code!");
        alert.showAndWait();

        }else {

                code.setText(searched.getCourseCode());
                year.setText(Integer.toString(searched.getYear()));
                duration.setText(Integer.toString(searched.getDuration()));

                //if completed is NULL, print a NO in screen
                if(searched.getCompleted()== null) {
                        completed.setText("NO");
                }else {
                        completed.setText(searched.getCompleted());
                }
                //if grade is NULL, print No Grade in screen
                if(searched.getGrade()==0) {
                        grade.setText("No Grade");
                }else {
                        grade.setText(Integer.toString(searched.getGrade()));
                }
        }
}

//method to complete exam when entering code
@FXML
public void completeExam() {

        int ok=0;
        Exam exam = new Exam();

        if(code.getText()=="") {

                //pop up error box when no course found
                Alert alert = new Alert(AlertType.ERROR);
        alert.setTitle("Error Dialog");
```

```
                alert.setHeaderText("CANNOT COMPLETE EXAM");
                alert.setContentText("No course selected!");
                alert.showAndWait();

                }else {

                        exam.setCourseCode(code.getText());
                        ok=examdb.completeExam(exam);
                        if (ok == 1) {
                                completed.setText("YES");

                                //pop up information box when exam completed successfully
                                Alert alert = new Alert(AlertType.INFORMATION);
                                alert.setTitle("Information Dialog");
                                alert.setHeaderText("EXAM SUCCESSFULLY COMPLETED");
                                alert.showAndWait();
                        }


                }


        }

        //method to go back to the main menu
        public void goBack() {
                try {
                        AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/StudentView.fxml"));
                        Scene scene = new Scene(root);
                        Main.stage.setScene(scene);
                        Main.stage.setTitle("Student View");
                } catch (Exception e) {
                        System.out.println("Error occured while inflating view: " + e);
                }
        }

        //method to logout
        public void logout() {
                try {
                        AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/LoginView.fxml"));
                        Scene scene = new Scene(root);
                        Main.stage.setScene(scene);
```

```
                Main.stage.setTitle("Login View");
            } catch (Exception e) {
                System.out.println("Error occured while inflating view: " + e);
            }
        }

        //methods to clear values in fields

        public void clearFields1() {
            txtCode.clear();
        }

        public void clearFields2() {

            code.setText("");
            year.setText("");
            duration.setText("");
            completed.setText("");
            grade.setText("");

        }

}
```

11. StudentInfoController.java

```
package controllers;

import java.util.ArrayList;
import java.util.List;
import java.util.Locale;

import application.Main;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.Label;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.layout.AnchorPane;
import models.Course;
import models.CourseDB;
```

```
public class StudentInfoController {

	//fx:id in StudentInfoView
	@FXML
	private TableView<Course> coursesTable;
	@FXML
	private TableColumn<Course, Integer> id;
	@FXML
	private TableColumn<Course, String> code;
	@FXML
	private TableColumn<Course, String> name;
	@FXML
	private TableColumn<Course, Integer> semester;
	@FXML
	private TableColumn<Course, Integer> classroom;
	@FXML
	private TableColumn<Course, String> professor;
	@FXML
	private TableColumn<Course, Double> cost;

	@FXML
	private Label totalCost;

	//method to initialize by showing the enrolled courses in a table and the total cost
	@FXML
	void initialize() {

		List<String> codes = new ArrayList<String>();
		List<Course> enrolledCourses = new ArrayList<Course>();
		Course course = new Course();
	CourseDB dao = new CourseDB();
	double total=0.00;

		id.setCellValueFactory(new PropertyValueFactory<Course, Integer>("id"));
		code.setCellValueFactory(new PropertyValueFactory<Course, String>("code"));
		name.setCellValueFactory(new PropertyValueFactory<Course, String>("name"));
		semester.setCellValueFactory(new PropertyValueFactory<Course,
Integer>("semester"));
		classroom.setCellValueFactory(new PropertyValueFactory<Course,
Integer>("classroom"));
	professor.setCellValueFactory(new PropertyValueFactory<Course, String>("professor"));
	cost.setCellValueFactory(new PropertyValueFactory<Course, Double>("cost"));
```

```
    codes = StudentEnrollmentController.getCodes();
    for(int i = 0; i < codes.size(); i++) {
        course = dao.getEnrolledCourse(codes.get(i));
        enrolledCourses.add(course);
    }

    if(!enrolledCourses.isEmpty()) {
        coursesTable.getItems().setAll(enrolledCourses);
      for(int i=0; i < enrolledCourses.size(); i++) {
                        //calculate total cost of enrolled courses
        total+=enrolledCourses.get(i).getCost();
      }
    }
    else {
                        //pop up error box when no courses enrolled found
                Alert alert = new Alert(AlertType.WARNING);
                alert.setTitle("Warning Dialog");
                alert.setHeaderText("NO COURSES FOUND");
                alert.setContentText("You are not enrolled in any course!");
                alert.showAndWait();

    }

    String tot = String.format(Locale.US,"%.2f", total);
    totalCost.setText("$ "+tot);
        }

        //method to go back to StudentEnrollmentView
        public void goBackEnrollment() {

                try {
                        AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/StudentEnrollmentView.fxml"));
                        Scene scene = new Scene(root);
                        Main.stage.setScene(scene);
                        Main.stage.setTitle("Student Courses View");
                } catch (Exception e) {
                        System.out.println("Error occured while inflating view: " + e);
                }
        }

        //method to go back to the main menu
        public void goBack() {
```

```
		try {
			AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/StudentView.fxml"));
			Scene scene = new Scene(root);
			Main.stage.setScene(scene);
			Main.stage.setTitle("Student View");
		} catch (Exception e) {
			System.out.println("Error occured while inflating view: " + e);
		}
	}

	//method to logout
	public void logout() {
		try {
			AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/LoginView.fxml"));
			Scene scene = new Scene(root);
			Main.stage.setScene(scene);
			Main.stage.setTitle("Login View");
		} catch (Exception e) {
			System.out.println("Error occured while inflating view: " + e);
		}
	}

}
```

12. StudentProfileController.java

```
package controllers;

import application.Main;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.scene.layout.AnchorPane;

public class StudentProfileController {

	//method to go back to the main menu
	public void goBack() {
		try {
			AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/StudentView.fxml"));
			Scene scene = new Scene(root);
			Main.stage.setScene(scene);
			Main.stage.setTitle("Student View");
```

```
		} catch (Exception e) {
			System.out.println("Error occured while inflating view: " + e);
		}
	}

	//method to logout
	public void logout() {
		try {
			AnchorPane root = (AnchorPane)
FXMLLoader.load(getClass().getResource("/views/LoginView.fxml"));
			Scene scene = new Scene(root);
			Main.stage.setScene(scene);
			Main.stage.setTitle("Login View");
		} catch (Exception e) {
			System.out.println("Error occured while inflating view: " + e);
		}
	}

}
```

DAO

1. DBConnect.java

```
package dao;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBConnect  {

	protected Connection connection;
	public Connection getConnection() {
		return connection;
	}

	private static String url =
"jdbc:mysql://www.papademas.net:3307/510fp?autoReconnect=true&useSSL=false";
	private static String username = "fp510";
	private static String password = "510";

	//connection with the database
	public DBConnect() {
		try {
```

```
                connection = DriverManager.getConnection(url, username, password);
            } catch (SQLException e) {
                System.out.println("Error creating connection to database: " + e);
                System.exit(-1);
            }
        }


}
```

MODELS

1. Course.java

```
package models;

public class Course {

        //data members of a course
        private int id;
        private String code;
        private String name;
        private int semester;
        private int classroom;
        private String professor;
        private double cost;

        //constructor
        public Course() {

        }

        //getters and setters
        public int getId() {
                return id;
        }
        public void setId(int id) {
                this.id = id;
        }

        public String getCode() {
                return code;
        }
        public void setCode(String code) {
                this.code = code;
```

```
        }

        public String getName() {
                return name;
        }
        public void setName(String name) {
                this.name = name;
        }

        public int getSemester() {
                return semester;
        }
        public void setSemester(int semester) {
                this.semester = semester;
        }

        public int getClassroom() {
                return classroom;
        }
        public void setClassroom(int classroom) {
                this.classroom = classroom;
        }

        public String getProfessor() {
                return professor;
        }
        public void setProfessor(String professor) {
                this.professor = professor;
        }

        public double getCost() {
                return cost;
        }
        public void setCost(double cost) {
                this.cost = cost;
        }
}
```

2. CourseDB.java

```
package models;

import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;


import dao.DBConnect;
import javafx.scene.control.Alert;
import javafx.scene.control.Alert.AlertType;

public class CourseDB {

	//Declare DB objects
	DBConnect conn = null;
	PreparedStatement  stm = null;

	public CourseDB() {//Create db object instance

		conn = new DBConnect();

	}

	//method to add a course to database
	public void addCourse(Course course) {

		try {

			stm = conn.getConnection().prepareStatement("INSERT INTO
lakepark_courses(code,name,semester,classroom,professor,cost) VALUES (?,?,?,?,?,?);");
			stm.setString(1, course.getCode());
			stm.setString(2, course.getName());
			stm.setInt(3, course.getSemester());
			stm.setInt(4, course.getClassroom());
			stm.setString(5, course.getProfessor());
			stm.setDouble(6, course.getCost());

			stm.executeUpdate();

			stm.close();
```

```
		} catch (SQLException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();

		}
	}

	//method to delete a course from database
	public int deleteCourse(Course course) {

		int ok = 0, found = 2;

		try {
			stm = conn.getConnection().prepareStatement("DELETE FROM
lakepark_courses WHERE code = ? AND name = ?;");
			stm.setString(1, course.getCode());
			stm.setString(2, course.getName());

			found=stm.executeUpdate();


			if(found==0) {

				//pop up error box when course not found
				Alert alert = new Alert(AlertType.ERROR);
			alert.setTitle("Error Dialog");
			alert.setHeaderText("ERROR!");
			alert.setContentText("That course does not exist!");
			alert.showAndWait();

			}
			else {
				ok = 1;
			}

			stm.close();

		} catch (SQLException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}

		return ok;
```

```
	}

	//method to update a course from database
	public int updateCourse(Course course) {

		int ok = 0, found = 2;

		try {
			stm = conn.getConnection().prepareStatement("UPDATE lakepark_courses
SET code= ?,name= ?,semester= ?,classroom= ?,professor= ?,cost= ? WHERE code= ?;");
			stm.setString(1, course.getCode());
			stm.setString(2, course.getName());
			stm.setInt(3, course.getSemester());
			stm.setInt(4, course.getClassroom());
			stm.setString(5, course.getProfessor());
			stm.setDouble(6, course.getCost());

			stm.setString(7, course.getCode());

			found=stm.executeUpdate();

			if(found==0) {

				//pop up error box when course not found
				Alert alert = new Alert(AlertType.ERROR);
			alert.setTitle("Error Dialog");
			alert.setHeaderText("ERROR!");
			alert.setContentText("That course code does not exist!");
			alert.showAndWait();

			}
			else {
				ok = 1;
			}

			stm.close();
		} catch (SQLException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}

		return ok;
	}
```

```
	//method to search course from database
	public Course searchCourse(Course course) {

		Course searched = new Course();
		ResultSet rs = null;
		try {
			stm = conn.getConnection().prepareStatement("SELECT * FROM
lakepark_courses WHERE code = ?;");
			stm.setString(1, course.getCode());
			rs = stm.executeQuery();

			if (rs.next() == false) { //ResultSet empty

		}
			else {

				do {

					searched.setCode(rs.getString("code"));
					searched.setName(rs.getString("name"));
					searched.setSemester(rs.getInt("semester"));
					searched.setClassroom(rs.getInt("classroom"));
					searched.setProfessor(rs.getString("professor"));
					searched.setCost(rs.getDouble("cost"));

				} while (rs.next());
			}

			rs.close();
			stm.close();

		}catch(Exception e){
			e.printStackTrace();
		}

		return searched;
	}

	//method to list all courses from database
	public List<Course> listCourses() {

		List<Course> courses = new ArrayList<Course>();
		ResultSet rs = null;
```

```
		try{
			stm = conn.getConnection().prepareStatement("SELECT * FROM
lakepark_courses;");
			rs = stm.executeQuery();

			if (rs.next() == false) { //ResultSet empty

			}
			else {

				do {
					Course course = new Course();

					course.setId(rs.getInt("id"));
					course.setCode(rs.getString("code"));
					course.setName(rs.getString("name"));
					course.setSemester(rs.getInt("semester"));
					course.setClassroom(rs.getInt("classroom"));
					course.setProfessor(rs.getString("professor"));
					course.setCost(rs.getDouble("cost"));

					courses.add(course);

				} while (rs.next());
			}

			rs.close();
			stm.close();

		}catch(Exception e){
			e.printStackTrace();
		}

		return courses;
	}

	//method to list all courses from database ordered by name
	public List<Course> listCoursesByName() {

		List<String> names = new ArrayList<String>();
		List<Course> courses = new ArrayList<Course>();
		ResultSet rs = null;
```

```
		try{

			names = listCourseNames();
			for(int i=0;i<names.size();i++) {
				stm = conn.getConnection().prepareStatement("SELECT * FROM
lakepark_courses WHERE name = ?;");
				stm.setString(1, names.get(i));
				rs = stm.executeQuery();

				if (rs.next() == false) { //ResultSet empty

				}
				else {

					do {
						Course course = new Course();

						course.setId(rs.getInt("id"));
						course.setCode(rs.getString("code"));
						course.setName(rs.getString("name"));
						course.setSemester(rs.getInt("semester"));
						course.setClassroom(rs.getInt("classroom"));
						course.setProfessor(rs.getString("professor"));
						course.setCost(rs.getDouble("cost"));

						courses.add(course);

					} while (rs.next());
				}

				rs.close();
				stm.close();

			}

		}catch(Exception e){
					e.printStackTrace();
		}

		return courses;
	}

	//method to list the names of the courses in order
	public List<String> listCourseNames(){
```

```
		List<String> names = new ArrayList<String>();
		List<Course> courses = listCourses();
		for (int i = 0; i < courses.size(); i++) {
			names.add(courses.get(i).getName());
		}
		Collections.sort(names);
		return names;
	}

	//method to list the codes of the courses
	public List<String> listCourseCodes(){

		List<String> codes = new ArrayList<String>();
		List<Course> courses = listCourses();
		for (int i = 0; i < courses.size(); i++) {
			codes.add(courses.get(i).getCode());
		}
		return codes;
	}

	//method to search a course by the code
	public Course getEnrolledCourse(String code) {

		Course searched = new Course();
		ResultSet rs = null;
		try {
			stm = conn.getConnection().prepareStatement("SELECT * FROM
lakepark_courses WHERE code = ?;");
			stm.setString(1, code);
			rs = stm.executeQuery();

			if (rs.next() == false) { //ResultSet empty

			}
			else {

				do {

					searched.setId(rs.getInt("id"));
					searched.setCode(rs.getString("code"));
					searched.setName(rs.getString("name"));
					searched.setSemester(rs.getInt("semester"));
					searched.setClassroom(rs.getInt("classroom"));
```

```
                                        searched.setProfessor(rs.getString("professor"));
                                        searched.setCost(rs.getDouble("cost"));

                                } while (rs.next());
                        }

                        rs.close();
                        stm.close();

                }catch(Exception e){
                        e.printStackTrace();
                }

                return searched;
        }

}
```

3. Exam.java

```
package models;

public class Exam {

        //data members of an exam
        private int id;
        private String courseCode;
        private int year;
        private int duration;
        private String completed;
        private int grade;

        //getters and setters
        public int getId() {
                return id;
        }
        public void setId(int id) {
                this.id = id;
        }

        public String getCourseCode() {
                return courseCode;
        }
        public void setCourseCode(String courseCode) {
                this.courseCode = courseCode;
```

```
        }

        public int getYear() {
                return year;
        }
        public void setYear(int year) {
                this.year = year;
        }

        public int getDuration() {
                return duration;
        }
        public void setDuration(int duration) {
                this.duration = duration;
        }

        public String getCompleted() {
                return completed;
        }
        public void setCompleted(String completed) {
                this.completed = completed;
        }

        public int getGrade() {
                return grade;
        }
        public void setGrade(int grade) {
                this.grade = grade;
        }

}
```

4. ExamDB.java

```
package models;

import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

import dao.DBConnect;
import javafx.scene.control.Alert;
import javafx.scene.control.Alert.AlertType;
```

```
public class ExamDB {

	//Declare DB objects
	DBConnect conn = null;
	PreparedStatement  stm = null;

	public ExamDB() {//Create db object instance

		conn = new DBConnect();

	}

	//method to add an exam to database
	public void addExam(Exam exam) {

		try {

			stm = conn.getConnection().prepareStatement("INSERT INTO
lakepark_exams(course,year,duration) VALUES (?,?,?);");
			stm.setString(1, exam.getCourseCode());
			stm.setInt(2, exam.getYear());
			stm.setInt(3, exam.getDuration());


			stm.executeUpdate();

			stm.close();
		} catch (SQLException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
	}

	//method to delete an exam from database
	public int deleteExam(Exam exam) {

		int ok = 0, found = 2;

		try {
			stm = conn.getConnection().prepareStatement("DELETE FROM
lakepark_exams WHERE id = ?;");
			stm.setInt(1, exam.getId());

			found=stm.executeUpdate();
```

```
                if(found==0) {

                        //pop up error box when exam not found
                        Alert alert = new Alert(AlertType.ERROR);
                        alert.setTitle("Error Dialog");
                        alert.setHeaderText("ERROR!");
                        alert.setContentText("The exam ID does not exist!");
                        alert.showAndWait();

                }else {
                        ok = 1;
                }

                stm.close();
        } catch (SQLException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
        }

        return ok;
    }

    //method to grade an exam from database
    public int gradeExam(Exam exam) {

        int ok = 0,found = 2;

        try {
                stm = conn.getConnection().prepareStatement("UPDATE lakepark_exams
SET grade= ? WHERE id= ?;");
                stm.setInt(1, exam.getGrade());

                stm.setInt(2, exam.getId());

                found=stm.executeUpdate();

                if(found==0) {

                        //pop up error box when exam not found
                        Alert alert = new Alert(AlertType.ERROR);
                        alert.setTitle("Error Dialog");
                        alert.setHeaderText("ERROR!");
                        alert.setContentText("The exam ID does not exist!");
```

```
                        alert.showAndWait();

                }else {
                        ok = 1;
                }

                stm.close();
        } catch (SQLException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
        }

        return ok;
    }

    //method to search exam from database
    public Exam searchExam(Exam exam) {

        Exam searched = new Exam();
        ResultSet rs = null;
        try {
                stm = conn.getConnection().prepareStatement("SELECT * FROM
lakepark_exams WHERE course = ?;");
                stm.setString(1, exam.getCourseCode());
                rs = stm.executeQuery();

                if (rs.next() == false) { //ResultSet empty

                }
                else {

                        do {

                                searched.setCourseCode(rs.getString("course"));
                                searched.setYear(rs.getInt("year"));
                                searched.setDuration(rs.getInt("duration"));
                                searched.setCompleted(rs.getString("completed"));
                                searched.setGrade(rs.getInt("grade"));

                        } while (rs.next());
                }


                rs.close();
```

```
                stm.close();

            }catch(Exception e){
                e.printStackTrace();
            }

            return searched;
        }

        //method to complete exam from database
        public int completeExam(Exam exam) {

            int ok = 0;
            ResultSet rs = null;
            try {
                stm = conn.getConnection().prepareStatement("SELECT * FROM
lakepark_exams WHERE course = ?;");
                stm.setString(1, exam.getCourseCode());
                rs = stm.executeQuery();

                if (rs.next() == false) { //ResultSet empty

                }
                else {

                    do {

                        exam.setCourseCode(rs.getString("course"));
                        exam.setYear(rs.getInt("year"));
                        exam.setDuration(rs.getInt("duration"));
                        exam.setCompleted(rs.getString("completed"));
                        exam.setGrade(rs.getInt("grade"));

                    } while (rs.next());
                }


                rs.close();
                stm.close();

            }catch(Exception e){
                e.printStackTrace();
            }
```

```
		try {

			if(exam.getCompleted()==null || exam.getCompleted().equals("NO")) {
				stm = conn.getConnection().prepareStatement("UPDATE
lakepark_exams SET completed= ? WHERE course= ?;");
				stm.setString(1, "YES");

				stm.setString(2, exam.getCourseCode());
				stm.executeUpdate();
				ok = 1;
			}
			else {

				//pop up error box when exam is already completed
				Alert alert = new Alert(AlertType.ERROR);
				alert.setTitle("Error Dialog");
				alert.setHeaderText("EXAM COMPLETED");
				alert.setContentText("The exam for this course is already
completed!");
				alert.showAndWait();

			}


			stm.close();

		} catch (SQLException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}

		return ok;
	}
}
```

5. LoginModel.java

```
package models;

import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

import dao.DBConnect;
```

```
public class LoginModel extends DBConnect {

        //data members
        private Boolean professor;
        private int id;

        //getters and setters
        public int getId() {
                return id;
        }
        public void setId(int id) {
                this.id = id;
        }
        public Boolean isProfessor() {
                return professor;
        }
        public void setProfessor(Boolean professor) {
                this.professor = professor;
        }

        //check credentials in database
        public Boolean getCredentials(String username, String password){

        String query = "SELECT * FROM lakepark_users WHERE uname = ? and passwd = ?;";
       try(PreparedStatement stmt = connection.prepareStatement(query)) {
        stmt.setString(1, username);
        stmt.setString(2, password);
        ResultSet rs = stmt.executeQuery();
         if(rs.next()) {
                setId(rs.getInt("id"));
                setProfessor(rs.getBoolean("role"));
                return true;
                }
       }catch (SQLException e) {
        e.printStackTrace();
       }
                        return false;
   }

}
```

VIEWS

1. AddCourseView.java

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

<AnchorPane prefHeight="442.0" prefWidth="578.0" xmlns="http://javafx.com/javafx/8.0.241"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="controllers.ProfessorCoursesController">
  <children>
    <Button layoutX="416.0" layoutY="373.0" mnemonicParsing="false" onAction="#logout"
prefHeight="46.0" prefWidth="153.0" text="Logout">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
    <ImageView fitHeight="225.0" fitWidth="232.0" layoutX="368.0" layoutY="8.0" opacity="0.5"
pickOnBounds="true">
      <image>
        <Image url="@../../images/lplogo.png" />
      </image>
    </ImageView>
    <Label layoutX="18.0" layoutY="8.0" prefHeight="46.0" prefWidth="271.0" text="Courses
Panel">
      <font>
        <Font name="Calibri Italic" size="35.0" />
      </font>
    </Label>
    <AnchorPane layoutX="21.0" layoutY="83.0" prefHeight="247.0" prefWidth="327.0" style="-fx-
border-color: black;">
      <children>
        <Label layoutX="14.0" layoutY="12.0" text="Enter Code:">
          <font>
            <Font name="Calibri Italic" size="16.0" />
          </font>
        </Label>
        <TextField fx:id="txtCode" layoutX="14.0" layoutY="44.0" prefHeight="25.0"
prefWidth="132.0" />
```

```
        <Label layoutX="172.0" layoutY="12.0" text="Enter Name:">
          <font>
            <Font name="Calibri Italic" size="16.0" />
          </font>
        </Label>
        <TextField fx:id="txtName" layoutX="172.0" layoutY="44.0" prefHeight="25.0"
prefWidth="132.0" />
        <Label layoutX="14.0" layoutY="89.0" text="Enter Semester:">
          <font>
            <Font name="Calibri Italic" size="16.0" />
          </font>
        </Label>
        <TextField fx:id="txtSemester" layoutX="14.0" layoutY="121.0" prefHeight="25.0"
prefWidth="132.0" />
        <Label layoutX="172.0" layoutY="89.0" text="Enter Classroom:">
          <font>
            <Font name="Calibri Italic" size="16.0" />
          </font>
        </Label>
        <TextField fx:id="txtClassroom" layoutX="172.0" layoutY="121.0" prefHeight="25.0"
prefWidth="132.0" />
        <Label layoutX="14.0" layoutY="164.0" text="Enter Professor:">
          <font>
            <Font name="Calibri Italic" size="16.0" />
          </font>
        </Label>
        <TextField fx:id="txtProfessor" layoutX="14.0" layoutY="196.0" prefHeight="25.0"
prefWidth="132.0" />
        <Label layoutX="172.0" layoutY="164.0" text="Enter Cost:">
          <font>
            <Font name="Calibri Italic" size="16.0" />
          </font>
        </Label>
        <TextField fx:id="txtCost" layoutX="172.0" layoutY="196.0" prefHeight="25.0"
prefWidth="132.0" />
      </children>
    </AnchorPane>
    <Button layoutX="416.0" layoutY="318.0" mnemonicParsing="false"
onAction="#goBackCourses" prefHeight="46.0" prefWidth="153.0" text="Back">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
```

```
    <Button layoutX="119.0" layoutY="346.0" mnemonicParsing="false" onAction="#addCourse"
prefHeight="36.0" prefWidth="132.0" text="Submit">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
  </children>
</AnchorPane>
```

2. AddExamView.java

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

<AnchorPane prefHeight="442.0" prefWidth="578.0" xmlns="http://javafx.com/javafx/8.0.241"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="controllers.ProfessorExamsController">
  <children>
    <Button layoutX="416.0" layoutY="373.0" mnemonicParsing="false" onAction="#logout"
prefHeight="46.0" prefWidth="153.0" text="Logout">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
    <ImageView fitHeight="225.0" fitWidth="232.0" layoutX="368.0" layoutY="8.0" opacity="0.5"
pickOnBounds="true">
      <image>
        <Image url="@../../images/lplogo.png" />
      </image>
    </ImageView>
    <Label layoutX="18.0" layoutY="8.0" prefHeight="46.0" prefWidth="271.0" text="Exams
Panel">
      <font>
        <Font name="Calibri Italic" size="35.0" />
      </font>
    </Label>
    <AnchorPane layoutX="95.0" layoutY="85.0" prefHeight="247.0" prefWidth="197.0" style="-fx-
border-color: black;">
      <children>
```

```
        <Label layoutX="14.0" layoutY="12.0" text="Enter Course:">
          <font>
            <Font name="Calibri Italic" size="16.0" />
          </font>
        </Label>
        <TextField fx:id="txtCourse" layoutX="14.0" layoutY="44.0" prefHeight="25.0"
prefWidth="132.0" />
        <Label layoutX="14.0" layoutY="89.0" text="Enter Year:">
          <font>
            <Font name="Calibri Italic" size="16.0" />
          </font>
        </Label>
        <TextField fx:id="txtYear" layoutX="14.0" layoutY="121.0" prefHeight="25.0"
prefWidth="132.0" />
        <Label layoutX="14.0" layoutY="164.0" text="Enter Duration (minutes):">
          <font>
            <Font name="Calibri Italic" size="16.0" />
          </font>
        </Label>
        <TextField fx:id="txtDuration" layoutX="14.0" layoutY="196.0" prefHeight="25.0"
prefWidth="132.0" />
      </children>
    </AnchorPane>
    <Button layoutX="416.0" layoutY="318.0" mnemonicParsing="false"
onAction="#goBackExams" prefHeight="46.0" prefWidth="153.0" text="Back">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
    <Button layoutX="141.0" layoutY="347.0" mnemonicParsing="false" onAction="#addExam"
prefHeight="25.0" prefWidth="106.0" text="Submit">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
  </children>
</AnchorPane>
```

3. DeleteCourseView.java

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

<AnchorPane prefHeight="442.0" prefWidth="578.0" xmlns="http://javafx.com/javafx/8.0.241"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="controllers.ProfessorCoursesController">
  <children>
    <Button layoutX="416.0" layoutY="373.0" mnemonicParsing="false" onAction="#logout"
prefHeight="46.0" prefWidth="153.0" text="Logout">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
    <ImageView fitHeight="225.0" fitWidth="232.0" layoutX="368.0" layoutY="8.0" opacity="0.5"
pickOnBounds="true">
      <image>
        <Image url="@../../images/lplogo.png" />
      </image>
    </ImageView>
    <Label layoutX="18.0" layoutY="8.0" prefHeight="46.0" prefWidth="271.0" text="Courses
Panel">
      <font>
        <Font name="Calibri Italic" size="35.0" />
      </font>
    </Label>
    <AnchorPane layoutX="21.0" layoutY="138.0" prefHeight="115.0" prefWidth="327.0" style="-
fx-border-color: black;">
      <children>
        <Label layoutX="14.0" layoutY="26.0" text="Enter Code:">
          <font>
            <Font name="Calibri Italic" size="16.0" />
          </font>
        </Label>
        <TextField fx:id="txtCode" layoutX="14.0" layoutY="58.0" prefHeight="25.0"
prefWidth="132.0" />
        <Label layoutX="172.0" layoutY="26.0" text="Enter Name:">
          <font>
```

```
            <Font name="Calibri Italic" size="16.0" />
          </font>
        </Label>
        <TextField fx:id="txtName" layoutX="172.0" layoutY="58.0" prefHeight="25.0"
prefWidth="132.0" />
      </children>
    </AnchorPane>
    <Button layoutX="416.0" layoutY="318.0" mnemonicParsing="false"
onAction="#goBackCourses" prefHeight="46.0" prefWidth="153.0" text="Back">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
    <Button layoutX="117.0" layoutY="302.0" mnemonicParsing="false"
onAction="#deleteCourse" prefHeight="46.0" prefWidth="153.0" text="Submit">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
  </children>
</AnchorPane>
```

4. DeleteExamView.java

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

<AnchorPane prefHeight="442.0" prefWidth="578.0" xmlns="http://javafx.com/javafx/8.0.241"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="controllers.ProfessorExamsController">
  <children>
    <Button layoutX="416.0" layoutY="373.0" mnemonicParsing="false" onAction="#logout"
prefHeight="46.0" prefWidth="153.0" text="Logout">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
    <ImageView fitHeight="225.0" fitWidth="232.0" layoutX="368.0" layoutY="8.0" opacity="0.5"
pickOnBounds="true">
```

```
      <image>
        <Image url="@../../images/lplogo.png" />
      </image>
    </ImageView>
    <Label layoutX="18.0" layoutY="8.0" prefHeight="46.0" prefWidth="271.0" text="Exams
Panel">
      <font>
        <Font name="Calibri Italic" size="35.0" />
      </font>
    </Label>
    <AnchorPane layoutX="111.0" layoutY="139.0" prefHeight="115.0" prefWidth="165.0" style="-
fx-border-color: black;">
      <children>
        <Label layoutX="14.0" layoutY="26.0" text="Enter Exam ID:">
          <font>
            <Font name="Calibri Italic" size="16.0" />
          </font>
        </Label>
        <TextField fx:id="txtId" layoutX="14.0" layoutY="58.0" prefHeight="25.0"
prefWidth="132.0" />
      </children>
    </AnchorPane>
    <Button layoutX="416.0" layoutY="318.0" mnemonicParsing="false"
onAction="#goBackExams" prefHeight="46.0" prefWidth="153.0" text="Back">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
    <Button layoutX="117.0" layoutY="313.0" mnemonicParsing="false" onAction="#deleteExam"
prefHeight="46.0" prefWidth="153.0" text="Submit">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
  </children>
</AnchorPane>
```

5. GradeExamView.java

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

<AnchorPane prefHeight="442.0" prefWidth="578.0" xmlns="http://javafx.com/javafx/8.0.241"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="controllers.ProfessorExamsController">
  <children>
    <Button layoutX="416.0" layoutY="373.0" mnemonicParsing="false" onAction="#logout"
prefHeight="46.0" prefWidth="153.0" text="Logout">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
    <ImageView fitHeight="225.0" fitWidth="232.0" layoutX="368.0" layoutY="8.0" opacity="0.5"
pickOnBounds="true">
      <image>
        <Image url="@../../images/lplogo.png" />
      </image>
    </ImageView>
    <Label layoutX="18.0" layoutY="8.0" prefHeight="46.0" prefWidth="271.0" text="Exams
Panel">
      <font>
        <Font name="Calibri Italic" size="35.0" />
      </font>
    </Label>
    <AnchorPane layoutX="21.0" layoutY="138.0" prefHeight="115.0" prefWidth="327.0" style="-
fx-border-color: black;">
      <children>
        <Label layoutX="14.0" layoutY="26.0" text="Enter Exam ID:">
          <font>
            <Font name="Calibri Italic" size="16.0" />
          </font>
        </Label>
        <TextField fx:id="txtId" layoutX="14.0" layoutY="58.0" prefHeight="25.0"
prefWidth="132.0" />
        <Label layoutX="172.0" layoutY="26.0" text="Enter Grade:">
          <font>
```

```
            <Font name="Calibri Italic" size="16.0" />
          </font>
        </Label>
        <TextField fx:id="txtGrade" layoutX="172.0" layoutY="58.0" prefHeight="25.0"
prefWidth="132.0" />
      </children>
    </AnchorPane>
    <Button layoutX="416.0" layoutY="318.0" mnemonicParsing="false"
onAction="#goBackExams" prefHeight="46.0" prefWidth="153.0" text="Back">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
    <Button layoutX="108.0" layoutY="295.0" mnemonicParsing="false" onAction="#gradeExam"
prefHeight="46.0" prefWidth="153.0" text="Submit">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
  </children>
</AnchorPane>
```

6. LoginView.java

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.PasswordField?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

<AnchorPane prefHeight="368.0" prefWidth="687.0" style="-fx-background-color:
cornflowerblue;" xmlns="http://javafx.com/javafx/8.0.241" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="controllers.LoginController">
  <children>
    <ImageView fitHeight="385.0" fitWidth="367.0" pickOnBounds="true" preserveRatio="true">
      <image>
        <Image url="@../../images/LP.jpeg" />
      </image>
    </ImageView>
```

```
    <Button contentDisplay="CENTER" layoutX="479.0" layoutY="261.0" mnemonicParsing="false"
onAction="#login" prefHeight="38.0" prefWidth="85.0" style="-fx-background-color: white;"
text="Submit" textFill="GREY">
      <font>
        <Font size="17.0" />
      </font>
    </Button>
    <TextField fx:id="txtUsername" alignment="TOP_LEFT" layoutX="452.0" layoutY="128.0" />
    <PasswordField fx:id="txtPassword" alignment="TOP_LEFT" layoutX="452.0" layoutY="198.0"
/>
    <Label layoutX="452.0" layoutY="97.0" text="Enter user:" textFill="WHITE">
      <font>
        <Font name="System Bold" size="19.0" />
      </font>
    </Label>
    <Label layoutX="452.0" layoutY="166.0" text="Enter password:" textFill="WHITE">
      <font>
        <Font name="System Bold" size="19.0" />
      </font>
    </Label>
    <Label layoutX="375.0" layoutY="1.0" text="LAKE PARK " textFill="WHITE">
      <font>
        <Font name="System Bold Italic" size="27.0" />
      </font>
    </Label>
    <Label layoutX="375.0" layoutY="32.0" text="SECONDARY SCHOOL" textFill="WHITE">
      <font>
        <Font name="System Bold Italic" size="21.0" />
      </font>
    </Label>
  </children>
</AnchorPane>
```

7. ProfessorCoursesView.java

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>
```

```
<AnchorPane prefHeight="442.0" prefWidth="578.0" xmlns="http://javafx.com/javafx/8.0.241"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="controllers.ProfessorCoursesController">
  <children>
    <Button layoutX="416.0" layoutY="373.0" mnemonicParsing="false" onAction="#logout"
prefHeight="46.0" prefWidth="153.0" text="Logout">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
    <ImageView fitHeight="225.0" fitWidth="232.0" layoutX="368.0" layoutY="8.0" opacity="0.5"
pickOnBounds="true">
      <image>
        <Image url="@../../images/lplogo.png" />
      </image>
    </ImageView>
    <Label layoutX="18.0" layoutY="8.0" prefHeight="46.0" prefWidth="271.0" text="Courses
Panel">
      <font>
        <Font name="Calibri Italic" size="35.0" />
      </font>
    </Label>
    <AnchorPane layoutX="38.0" layoutY="83.0" prefHeight="200.0" prefWidth="301.0">
      <children>
        <Label layoutX="6.0" layoutY="21.0" text="WELCOME TO LAKE PARK">
          <font>
            <Font name="Calibri Italic" size="27.0" />
          </font>
        </Label>
        <Label layoutX="6.0" layoutY="53.0" text="COURSE PANEL!">
          <font>
            <Font name="Calibri Italic" size="27.0" />
          </font>
        </Label>
        <Label layoutX="11.0" layoutY="133.0" text="Choose one of the options below:">
          <font>
            <Font name="Calibri Italic" size="20.0" />
          </font>
        </Label>
      </children>
    </AnchorPane>
    <Button layoutX="416.0" layoutY="318.0" mnemonicParsing="false" onAction="#goBack"
prefHeight="46.0" prefWidth="153.0" text="Main Menu">
      <font>
        <Font name="Calibri" size="18.0" />
```

```
      </font>
    </Button>
    <Button layoutX="38.0" layoutY="264.0" mnemonicParsing="false"
onAction="#addCourseView" prefHeight="46.0" prefWidth="153.0" text="Add Course">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
    <Button layoutX="213.0" layoutY="264.0" mnemonicParsing="false"
onAction="#deleteCourseView" prefHeight="46.0" prefWidth="153.0" text="Delete Course">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
    <Button layoutX="127.0" layoutY="319.0" mnemonicParsing="false"
onAction="#updateCourseView" prefHeight="46.0" prefWidth="153.0" text="Update Course">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
  </children>
</AnchorPane>
```

8. ProfessorExamsView.java

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

<AnchorPane prefHeight="442.0" prefWidth="578.0" xmlns="http://javafx.com/javafx/8.0.241"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="controllers.ProfessorExamsController">
  <children>
    <Button layoutX="416.0" layoutY="373.0" mnemonicParsing="false" onAction="#logout"
prefHeight="46.0" prefWidth="153.0" text="Logout">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
    <ImageView fitHeight="225.0" fitWidth="232.0" layoutX="368.0" layoutY="8.0" opacity="0.5"
pickOnBounds="true">
```

```
      <image>
        <Image url="@../../images/lplogo.png" />
      </image>
    </ImageView>
    <Label layoutX="18.0" layoutY="8.0" prefHeight="46.0" prefWidth="271.0" text="Exams
Panel">
      <font>
        <Font name="Calibri Italic" size="35.0" />
      </font>
    </Label>
    <AnchorPane layoutX="38.0" layoutY="83.0" prefHeight="200.0" prefWidth="301.0">
      <children>
        <Label layoutX="6.0" layoutY="21.0" text="WELCOME TO LAKE PARK">
          <font>
            <Font name="Calibri Italic" size="27.0" />
          </font>
        </Label>
        <Label layoutX="6.0" layoutY="53.0" text="EXAMS PANEL!">
          <font>
            <Font name="Calibri Italic" size="27.0" />
          </font>
        </Label>
        <Label layoutX="11.0" layoutY="133.0" text="Choose one of the options below:">
          <font>
            <Font name="Calibri Italic" size="20.0" />
          </font>
        </Label>
      </children>
    </AnchorPane>
    <Button layoutX="416.0" layoutY="318.0" mnemonicParsing="false" onAction="#goBack"
prefHeight="46.0" prefWidth="153.0" text="Main Menu">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
    <Button layoutX="38.0" layoutY="264.0" mnemonicParsing="false" onAction="#addExamView"
prefHeight="46.0" prefWidth="153.0" text="Add Exam">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
    <Button layoutX="213.0" layoutY="264.0" mnemonicParsing="false"
onAction="#deleteExamView" prefHeight="46.0" prefWidth="153.0" text="Delete Exam">
      <font>
```

```
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
    <Button layoutX="127.0" layoutY="319.0" mnemonicParsing="false"
onAction="#gradeExamView" prefHeight="46.0" prefWidth="153.0" text="Grade Exam">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
  </children>
</AnchorPane>
```

9. ProfessorProfileView.java

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

<AnchorPane prefHeight="442.0" prefWidth="578.0" style="-fx-background-color: white;"
xmlns="http://javafx.com/javafx/8.0.241" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="controllers.ProfessorProfileController">
  <children>
    <Button layoutX="416.0" layoutY="373.0" mnemonicParsing="false" onAction="#logout"
prefHeight="46.0" prefWidth="153.0" text="Logout">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
    <ImageView fitHeight="225.0" fitWidth="232.0" layoutX="368.0" layoutY="8.0" opacity="0.5"
pickOnBounds="true">
      <image>
        <Image url="@../../images/lplogo.png" />
      </image>
    </ImageView>
    <Label layoutX="18.0" layoutY="8.0" prefHeight="46.0" prefWidth="271.0" text="Personal
Profile">
      <font>
        <Font name="Calibri Italic" size="35.0" />
      </font>
    </Label>
```

```
    <ImageView fitHeight="150.0" fitWidth="200.0" layoutX="26.0" layoutY="68.0"
pickOnBounds="true" preserveRatio="true">
      <image>
        <Image url="@../../images/90-906935_animated-faces-my-hero-design-clip-art-
woman.png" />
      </image>
    </ImageView>
    <AnchorPane layoutX="26.0" layoutY="228.0" prefHeight="179.0" prefWidth="429.0">
      <children>
        <Label layoutX="14.0" layoutY="53.0" text="Academic Year: 2020-2021">
          <font>
            <Font name="Calibri Italic" size="20.0" />
          </font>
        </Label>
        <Label layoutX="14.0" layoutY="93.0" text="Classroom: 2B">
          <font>
            <Font name="Calibri Italic" size="20.0" />
          </font>
        </Label>
        <Label layoutX="14.0" layoutY="133.0" text="Courses: Biology (bio)">
          <font>
            <Font name="Calibri Italic" size="20.0" />
          </font>
        </Label>
        <Label layoutX="14.0" layoutY="9.0" prefHeight="28.0" prefWidth="310.0"
text="ACADEMIC INFORMATION">
          <font>
            <Font name="Calibri Bold Italic" size="24.0" />
          </font>
        </Label>
      </children>
    </AnchorPane>
    <AnchorPane layoutX="150.0" layoutY="93.0" prefHeight="138.0" prefWidth="210.0">
      <children>
        <Label layoutX="11.0" layoutY="41.0" text="First Name: Anna">
          <font>
            <Font name="Calibri Italic" size="20.0" />
          </font>
        </Label>
        <Label layoutX="12.0" layoutY="82.0" text="Last Name: Parker">
          <font>
            <Font name="Calibri Italic" size="20.0" />
          </font>
        </Label>
```

```
      </children>
    </AnchorPane>
    <Button layoutX="416.0" layoutY="318.0" mnemonicParsing="false" onAction="#goBack"
prefHeight="46.0" prefWidth="153.0" text="Main Menu">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
  </children>
</AnchorPane>
```

10. ProfessorView.java

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

<AnchorPane prefHeight="442.0" prefWidth="578.0" xmlns="http://javafx.com/javafx/8.0.241"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="controllers.ProfessorController">
  <children>
    <AnchorPane layoutX="14.0" layoutY="75.0" prefHeight="357.0" prefWidth="327.0" style="-fx-
background-color: steelblue;">
      <children>
        <Label layoutX="14.0" layoutY="14.0" text="Press to view your profile:" textFill="WHITE">
          <font>
            <Font name="Calibri" size="18.0" />
          </font>
        </Label>
        <Label layoutX="14.0" layoutY="128.0" text="Press to add, update or delete a course:"
textFill="WHITE">
          <font>
            <Font name="Calibri" size="18.0" />
          </font>
        </Label>
        <Label layoutX="14.0" layoutY="241.0" text="Press to add, grade or delete an exam:"
textFill="WHITE">
          <font>
            <Font name="Calibri" size="18.0" />
          </font>
        </Label>
```

```
        <Button layoutX="101.0" layoutY="57.0" mnemonicParsing="false" onAction="#viewProfile"
prefHeight="46.0" prefWidth="153.0" text="View profile">
          <font>
            <Font name="Calibri" size="18.0" />
          </font>
        </Button>
        <Button layoutX="101.0" layoutY="170.0" mnemonicParsing="false"
onAction="#coursesView" prefHeight="46.0" prefWidth="153.0" text="Courses">
          <font>
            <Font name="Calibri" size="18.0" />
          </font>
        </Button>
        <Button layoutX="101.0" layoutY="286.0" mnemonicParsing="false"
onAction="#examsView" prefHeight="46.0" prefWidth="153.0" text="Exams">
          <font>
            <Font name="Calibri" size="18.0" />
          </font>
        </Button>
      </children>
    </AnchorPane>
    <Button layoutX="416.0" layoutY="373.0" mnemonicParsing="false" onAction="#logout"
prefHeight="46.0" prefWidth="153.0" text="Logout">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
    <ImageView fitHeight="225.0" fitWidth="232.0" layoutX="368.0" layoutY="8.0" opacity="0.5"
pickOnBounds="true">
      <image>
        <Image url="@../../images/lplogo.png" />
      </image>
    </ImageView>
    <Label layoutX="18.0" layoutY="8.0" prefHeight="46.0" prefWidth="271.0" text="Professor
Login">
      <font>
        <Font name="Calibri Italic" size="35.0" />
      </font>
    </Label>
  </children>
</AnchorPane>
```

11. ShowAllCoursesView.java

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

<AnchorPane prefHeight="442.0" prefWidth="578.0" xmlns="http://javafx.com/javafx/8.0.241"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="controllers.AllCoursesController">
  <children>
    <Button layoutX="438.0" layoutY="373.0" mnemonicParsing="false" onAction="#logout"
prefHeight="46.0" prefWidth="120.0" text="Logout">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
    <ImageView fitHeight="225.0" fitWidth="232.0" layoutX="368.0" layoutY="8.0" opacity="0.5"
pickOnBounds="true">
      <image>
        <Image url="@../../images/lplogo.png" />
      </image>
    </ImageView>
    <Label layoutX="18.0" layoutY="8.0" prefHeight="46.0" prefWidth="271.0" text="Courses
Panel">
      <font>
        <Font name="Calibri Italic" size="35.0" />
      </font>
    </Label>
    <Button layoutX="302.0" layoutY="373.0" mnemonicParsing="false" onAction="#goBack"
prefHeight="46.0" prefWidth="120.0" text="Main Menu">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
    <TableView fx:id="coursesTable" layoutX="31.0" layoutY="62.0" prefHeight="287.0"
prefWidth="527.0">
      <columns>
        <TableColumn fx:id="id" editable="false" prefWidth="75.0" text="id" />
        <TableColumn fx:id="code" editable="false" prefWidth="75.0" text="code" />
```

```
        <TableColumn fx:id="name" editable="false" prefWidth="75.0" text="name" />
        <TableColumn fx:id="semester" editable="false" prefWidth="75.0" text="semester" />
        <TableColumn fx:id="classroom" editable="false" prefWidth="75.0" text="classroom" />
        <TableColumn fx:id="professor" editable="false" prefWidth="75.0" text="professor" />
        <TableColumn fx:id="cost" editable="false" prefWidth="75.0" text="cost" />
      </columns>
    </TableView>
    <Button layoutX="167.0" layoutY="373.0" mnemonicParsing="false"
onAction="#goBackCourses" prefHeight="46.0" prefWidth="120.0" text="Back">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
    <Button layoutX="31.0" layoutY="373.0" mnemonicParsing="false" onAction="#orderCourses"
prefHeight="46.0" prefWidth="120.0" text="Order by name">
      <font>
        <Font name="Calibri" size="15.0" />
      </font>
    </Button>
  </children>
</AnchorPane>
```

12. StudentCourseView.java

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

<AnchorPane xmlns="http://javafx.com/javafx/8.0.241" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="controllers.StudentCoursesController">
  <children>
    <AnchorPane layoutX="10.0" layoutY="10.0" prefHeight="442.0" prefWidth="578.0">
      <children>
        <Button layoutX="416.0" layoutY="373.0" mnemonicParsing="false" onAction="#logout"
prefHeight="46.0" prefWidth="153.0" text="Logout">
          <font>
            <Font name="Calibri" size="18.0" />
          </font>
        </Button>
```

```
        <ImageView fitHeight="225.0" fitWidth="232.0" layoutX="368.0" layoutY="8.0"
opacity="0.5" pickOnBounds="true">
          <image>
            <Image url="@../../images/lplogo.png" />
          </image>
        </ImageView>
        <Label layoutX="18.0" layoutY="8.0" prefHeight="46.0" prefWidth="271.0" text="Courses
Panel">
          <font>
            <Font name="Calibri Italic" size="35.0" />
          </font>
        </Label>
        <AnchorPane layoutX="19.0" layoutY="41.0" prefHeight="98.0" prefWidth="301.0">
          <children>
            <Label layoutX="6.0" layoutY="21.0" text="WELCOME TO LAKE PARK">
              <font>
                <Font name="Calibri Italic" size="27.0" />
              </font>
            </Label>
            <Label layoutX="6.0" layoutY="53.0" text="COURSE PANEL!">
              <font>
                <Font name="Calibri Italic" size="27.0" />
              </font>
            </Label>
          </children>
        </AnchorPane>
        <Button layoutX="416.0" layoutY="318.0" mnemonicParsing="false" onAction="#goBack"
prefHeight="46.0" prefWidth="153.0" text="Main Menu">
          <font>
            <Font name="Calibri" size="18.0" />
          </font>
        </Button>
        <Button layoutX="416.0" layoutY="261.0" mnemonicParsing="false"
onAction="#showAllCoursesView" prefHeight="46.0" prefWidth="153.0" text=" View All Courses">
          <font>
            <Font name="Calibri" size="18.0" />
          </font>
        </Button>
        <Label layoutX="25.0" layoutY="135.0" text="Enter a code to search for a course">
          <font>
            <Font name="Calibri Italic" size="20.0" />
          </font>
        </Label>
        <TextField fx:id="txtCode" layoutX="25.0" layoutY="198.0" />
```

```
        <Button layoutX="204.0" layoutY="193.0" mnemonicParsing="false"
onAction="#searchCourse" prefHeight="23.0" prefWidth="97.0" text="Search">
          <font>
            <Font name="Calibri" size="18.0" />
          </font>
        </Button>
        <AnchorPane layoutX="25.0" layoutY="251.0" prefHeight="173.0" prefWidth="327.0"
style="-fx-border-color: black;">
          <children>
            <Label layoutX="14.0" layoutY="6.0" style="-fx-underline: true;" text="Code:">
              <font>
                <Font name="Calibri Italic" size="16.0" />
              </font>
            </Label>
            <Label layoutX="172.0" layoutY="6.0" style="-fx-underline: true;" text="Name:">
              <font>
                <Font name="Calibri Italic" size="16.0" />
              </font>
            </Label>
            <Label layoutX="14.0" layoutY="65.0" style="-fx-underline: true;" text="Semester:">
              <font>
                <Font name="Calibri Italic" size="16.0" />
              </font>
            </Label>
            <Label layoutX="172.0" layoutY="65.0" style="-fx-underline: true;" text="Classroom:">
              <font>
                <Font name="Calibri Italic" size="16.0" />
              </font>
            </Label>
            <Label layoutX="14.0" layoutY="121.0" style="-fx-underline: true;" text="Professor:">
              <font>
                <Font name="Calibri Italic" size="16.0" />
              </font>
            </Label>
            <Label layoutX="172.0" layoutY="121.0" style="-fx-underline: true;" text="Cost:">
              <font>
                <Font name="Calibri Italic" size="16.0" />
              </font>
            </Label>
            <Label fx:id="code" layoutX="14.0" layoutY="34.0">
              <font>
                <Font name="Calibri Light Italic" size="15.0" />
              </font>
            </Label>
```

```
            <Label fx:id="name" layoutX="172.0" layoutY="34.0">
              <font>
                <Font name="Calibri Light Italic" size="15.0" />
              </font>
            </Label>
            <Label fx:id="semester" layoutX="14.0" layoutY="91.0">
              <font>
                <Font name="Calibri Light Italic" size="15.0" />
              </font>
            </Label>
            <Label fx:id="classroom" layoutX="172.0" layoutY="91.0">
              <font>
                <Font name="Calibri Light Italic" size="15.0" />
              </font>
            </Label>
            <Label fx:id="professor" layoutX="14.0" layoutY="144.0">
              <font>
                <Font name="Calibri Light Italic" size="15.0" />
              </font>
            </Label>
            <Label fx:id="cost" layoutX="172.0" layoutY="144.0">
              <font>
                <Font name="Calibri Light Italic" size="15.0" />
              </font>
            </Label>
          </children>
        </AnchorPane>
      </children>
    </AnchorPane>
    <Label layoutX="33.0" layoutY="165.0" text="and view its information:">
      <font>
        <Font name="Calibri Italic" size="20.0" />
      </font>
    </Label>
  </children>
</AnchorPane>
```

13. StudentEnrollmentView.java

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.ComboBox?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.image.Image?>
```

```
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

<AnchorPane prefHeight="442.0" prefWidth="578.0" xmlns="http://javafx.com/javafx/8.0.241"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="controllers.StudentEnrollmentController">
  <children>
    <Button layoutX="416.0" layoutY="373.0" mnemonicParsing="false" onAction="#logout"
prefHeight="46.0" prefWidth="153.0" text="Logout">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
    <ImageView fitHeight="225.0" fitWidth="232.0" layoutX="368.0" layoutY="8.0" opacity="0.5"
pickOnBounds="true">
      <image>
        <Image url="@../../images/lplogo.png" />
      </image>
    </ImageView>
    <Label layoutX="18.0" layoutY="8.0" prefHeight="46.0" prefWidth="271.0" text="Enrollment
Panel">
      <font>
        <Font name="Calibri Italic" size="35.0" />
      </font>
    </Label>
    <Button layoutX="416.0" layoutY="318.0" mnemonicParsing="false" onAction="#goBack"
prefHeight="46.0" prefWidth="153.0" text="Main Menu">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
    <ComboBox fx:id="enroll" layoutX="25.0" layoutY="202.0" prefWidth="150.0" />
    <Button layoutX="49.0" layoutY="349.0" mnemonicParsing="false" onAction="#infoView"
prefHeight="34.0" prefWidth="298.0" text="View enrolled courses and total cost">
      <font>
        <Font name="Calibri" size="17.0" />
      </font>
    </Button>
    <Label layoutX="25.0" layoutY="252.0" text="Choose a course to drop off:">
      <font>
        <Font name="Calibri Italic" size="20.0" />
      </font>
    </Label>
    <ComboBox fx:id="drop" layoutX="25.0" layoutY="293.0" prefWidth="150.0" />
```

```
    <Label layoutX="25.0" layoutY="155.0" text="Choose a course to enroll in:">
      <font>
        <Font name="Calibri Italic" size="20.0" />
      </font>
    </Label>
    <Label layoutX="24.0" layoutY="103.0" text="ENROLLMENT PANEL!">
      <font>
        <Font name="Calibri Italic" size="27.0" />
      </font>
    </Label>
    <Label layoutX="25.0" layoutY="72.0" text="WELCOME TO LAKE PARK">
      <font>
        <Font name="Calibri Italic" size="27.0" />
      </font>
    </Label>
    <Button layoutX="201.0" layoutY="198.0" mnemonicParsing="false" onAction="#enrollCourse"
prefHeight="34.0" prefWidth="85.0" text="Enroll">
      <font>
        <Font name="Calibri" size="15.0" />
      </font>
    </Button>
    <Button layoutX="201.0" layoutY="289.0" mnemonicParsing="false" onAction="#dropCourse"
prefHeight="34.0" prefWidth="85.0" text="Drop">
      <font>
        <Font name="Calibri" size="15.0" />
      </font>
    </Button>
  </children>
</AnchorPane>
```

14. StudentExamsView.java

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

<AnchorPane xmlns="http://javafx.com/javafx/8.0.241" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="controllers.StudentExamsController">
  <children>
```

```
    <AnchorPane layoutX="10.0" layoutY="10.0" prefHeight="442.0" prefWidth="578.0">
      <children>
        <Button layoutX="416.0" layoutY="373.0" mnemonicParsing="false" onAction="#logout"
prefHeight="46.0" prefWidth="153.0" text="Logout">
          <font>
            <Font name="Calibri" size="18.0" />
          </font>
        </Button>
        <ImageView fitHeight="225.0" fitWidth="232.0" layoutX="368.0" layoutY="8.0"
opacity="0.5" pickOnBounds="true">
          <image>
            <Image url="@../../images/lplogo.png" />
          </image>
        </ImageView>
        <Label layoutX="18.0" layoutY="8.0" prefHeight="46.0" prefWidth="271.0" text="Exams
Panel">
          <font>
            <Font name="Calibri Italic" size="35.0" />
          </font>
        </Label>
        <AnchorPane layoutX="38.0" layoutY="83.0" prefHeight="105.0" prefWidth="301.0" />
        <Button layoutX="416.0" layoutY="318.0" mnemonicParsing="false" onAction="#goBack"
prefHeight="46.0" prefWidth="153.0" text="Main Menu">
          <font>
            <Font name="Calibri" size="18.0" />
          </font>
        </Button>
      </children>
    </AnchorPane>
    <Label layoutX="34.0" layoutY="69.0" text="WELCOME TO LAKE PARK">
      <font>
        <Font name="Calibri Italic" size="27.0" />
      </font>
    </Label>
    <Label layoutX="34.0" layoutY="101.0" text="EXAM PANEL!">
      <font>
        <Font name="Calibri Italic" size="27.0" />
      </font>
    </Label>
    <Label layoutX="35.0" layoutY="145.0" text="Enter a course code and complete">
      <font>
        <Font name="Calibri Italic" size="20.0" />
      </font>
    </Label>
```

```
    <TextField fx:id="txtCode" layoutX="35.0" layoutY="208.0" />
    <Button layoutX="214.0" layoutY="203.0" mnemonicParsing="false" onAction="#searchExam"
prefHeight="23.0" prefWidth="97.0" text="Search">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
    <AnchorPane layoutX="35.0" layoutY="261.0" prefHeight="173.0" prefWidth="327.0" style="-
fx-border-color: black;">
      <children>
        <Label layoutX="14.0" layoutY="6.0" style="-fx-underline: true;" text="Course Code:">
          <font>
            <Font name="Calibri Italic" size="16.0" />
          </font>
        </Label>
        <Label layoutX="172.0" layoutY="6.0" style="-fx-underline: true;" text="Year:">
          <font>
            <Font name="Calibri Italic" size="16.0" />
          </font>
        </Label>
        <Label layoutX="14.0" layoutY="65.0" style="-fx-underline: true;" text="Duration:">
          <font>
            <Font name="Calibri Italic" size="16.0" />
          </font>
        </Label>
        <Label layoutX="172.0" layoutY="65.0" style="-fx-underline: true;" text="Grade:">
          <font>
            <Font name="Calibri Italic" size="16.0" />
          </font>
        </Label>
        <Label layoutX="14.0" layoutY="121.0" style="-fx-underline: true;" text="Completed:">
          <font>
            <Font name="Calibri Italic" size="16.0" />
          </font>
        </Label>
        <Label fx:id="code" layoutX="14.0" layoutY="34.0">
          <font>
            <Font name="Calibri Light Italic" size="15.0" />
          </font>
        </Label>
        <Label fx:id="year" layoutX="172.0" layoutY="34.0">
          <font>
            <Font name="Calibri Light Italic" size="15.0" />
          </font>
```

```
        </Label>
        <Label fx:id="duration" layoutX="14.0" layoutY="91.0">
          <font>
            <Font name="Calibri Light Italic" size="15.0" />
          </font>
        </Label>
        <Label fx:id="grade" layoutX="172.0" layoutY="91.0">
          <font>
            <Font name="Calibri Light Italic" size="15.0" />
          </font>
        </Label>
        <Label fx:id="completed" layoutX="14.0" layoutY="144.0">
          <font>
            <Font name="Calibri Light Italic" size="15.0" />
          </font>
        </Label>
        <Button layoutX="212.0" layoutY="131.0" mnemonicParsing="false"
onAction="#completeExam" prefHeight="34.0" prefWidth="85.0" text="Complete">
          <font>
            <Font name="Calibri" size="15.0" />
          </font>
        </Button>
      </children>
    </AnchorPane>
    <Label layoutX="35.0" layoutY="175.0" text="its exam:">
      <font>
        <Font name="Calibri Italic" size="20.0" />
      </font>
    </Label>
  </children>
</AnchorPane>
```

15. StudentInfoView.java

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>
```

```
<AnchorPane xmlns="http://javafx.com/javafx/8.0.241" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="controllers.StudentInfoController">
  <children>
    <AnchorPane layoutX="10.0" layoutY="10.0" prefHeight="442.0" prefWidth="578.0">
      <children>
        <Button layoutX="416.0" layoutY="373.0" mnemonicParsing="false" onAction="#logout"
prefHeight="46.0" prefWidth="153.0" text="Logout">
          <font>
            <Font name="Calibri" size="18.0" />
          </font>
        </Button>
        <ImageView fitHeight="225.0" fitWidth="232.0" layoutX="368.0" layoutY="8.0"
opacity="0.5" pickOnBounds="true">
          <image>
            <Image url="@../../images/lplogo.png" />
          </image>
        </ImageView>
        <Label layoutX="18.0" layoutY="8.0" prefHeight="46.0" prefWidth="271.0"
text="Enrollment Panel">
          <font>
            <Font name="Calibri Italic" size="35.0" />
          </font>
        </Label>
        <Button layoutX="416.0" layoutY="318.0" mnemonicParsing="false" onAction="#goBack"
prefHeight="46.0" prefWidth="153.0" text="Main Menu">
          <font>
            <Font name="Calibri" size="18.0" />
          </font>
        </Button>
        <Label layoutX="28.0" layoutY="341.0" text="Your total cost is:">
          <font>
            <Font name="Calibri Italic" size="20.0" />
          </font>
        </Label>
        <Label fx:id="totalCost" layoutX="28.0" layoutY="373.0">
          <font>
            <Font name="Calibri Italic" size="20.0" />
          </font>
        </Label>
        <Button layoutX="249.0" layoutY="373.0" mnemonicParsing="false"
onAction="#goBackEnrollment" prefHeight="46.0" prefWidth="153.0" text="Go Back">
          <font>
            <Font name="Calibri" size="18.0" />
          </font>
```

```
        </Button>
        <TableView fx:id="coursesTable" layoutX="18.0" layoutY="70.0" prefHeight="225.0"
prefWidth="528.0">
          <columns>
            <TableColumn fx:id="id" prefWidth="75.0" text="ID" />
            <TableColumn fx:id="code" prefWidth="75.0" text="Code" />
              <TableColumn fx:id="name" prefWidth="75.0" text="Name" />
              <TableColumn fx:id="semester" prefWidth="75.0" text="Semester" />
              <TableColumn fx:id="classroom" prefWidth="75.0" text="Classroom" />
              <TableColumn fx:id="professor" prefWidth="75.0" text="Professor" />
              <TableColumn fx:id="cost" prefWidth="75.0" text="Cost" />
          </columns>
        </TableView>
      </children>
    </AnchorPane>
  </children>
</AnchorPane>
```

16. StudentProfileView.java

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>


<AnchorPane prefHeight="442.0" prefWidth="578.0" style="-fx-background-color: white;"
xmlns="http://javafx.com/javafx/8.0.241" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="controllers.StudentProfileController">
  <children>
    <Button layoutX="416.0" layoutY="373.0" mnemonicParsing="false" onAction="#logout"
prefHeight="46.0" prefWidth="153.0" text="Logout">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
    <ImageView fitHeight="225.0" fitWidth="232.0" layoutX="368.0" layoutY="8.0" opacity="0.5"
pickOnBounds="true">
      <image>
        <Image url="@../../images/lplogo.png" />
      </image>
```

```
    </ImageView>
    <Label layoutX="18.0" layoutY="8.0" prefHeight="46.0" prefWidth="271.0" text="Personal
Profile">
      <font>
        <Font name="Calibri Italic" size="35.0" />
      </font>
    </Label>
    <AnchorPane layoutX="7.0" layoutY="229.0" prefHeight="179.0" prefWidth="429.0">
      <children>
        <Label layoutX="14.0" layoutY="127.0" text="Academic Year: 2020-2021">
          <font>
            <Font name="Calibri Italic" size="20.0" />
          </font>
        </Label>
        <Label layoutX="14.0" layoutY="90.0" text="Address: 332 Diversey Street, Apt 5 - Chicago
">
          <font>
            <Font name="Calibri Italic" size="20.0" />
          </font>
        </Label>
        <Label layoutX="14.0" layoutY="51.0" text="Last Name: Bledsoe">
          <font>
            <Font name="Calibri Italic" size="20.0" />
          </font>
        </Label>
        <Label layoutX="14.0" layoutY="14.0" text="First Name: Mark">
          <font>
            <Font name="Calibri Italic" size="20.0" />
          </font>
        </Label>
      </children>
    </AnchorPane>
    <ImageView fitHeight="129.0" fitWidth="153.0" layoutX="35.0" layoutY="79.0"
pickOnBounds="true" preserveRatio="true">
      <image>
        <Image url="@../../images/a.jpg" />
      </image>
    </ImageView>
    <Button layoutX="416.0" layoutY="318.0" mnemonicParsing="false" onAction="#goBack"
prefHeight="46.0" prefWidth="153.0" text="Main Menu">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
```

```
  </children>
</AnchorPane>
```

17. StudentView.java

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

<AnchorPane xmlns="http://javafx.com/javafx/8.0.241" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="controllers.StudentController">
  <children>
    <AnchorPane layoutX="10.0" layoutY="10.0" prefHeight="442.0" prefWidth="578.0">
      <children>
        <AnchorPane layoutX="14.0" layoutY="66.0" prefHeight="364.0" prefWidth="327.0"
style="-fx-background-color: steelblue;">
          <children>
            <Label layoutX="14.0" layoutY="8.0" text="Press to view your profile:" textFill="WHITE">
              <font>
                <Font name="Calibri" size="18.0" />
              </font>
            </Label>
            <Label layoutX="14.0" layoutY="91.0" text="Press to search and view a course:"
textFill="WHITE">
              <font>
                <Font name="Calibri" size="18.0" />
              </font>
            </Label>
            <Label layoutX="14.0" layoutY="182.0" text="Press to complete an exam:"
textFill="WHITE">
              <font>
                <Font name="Calibri" size="18.0" />
              </font>
            </Label>
            <Button layoutX="101.0" layoutY="35.0" mnemonicParsing="false"
onAction="#viewProfile" prefHeight="46.0" prefWidth="153.0" text="View profile">
              <font>
                <Font name="Calibri" size="18.0" />
              </font>
            </Button>
```

```
            <Button layoutX="101.0" layoutY="125.0" mnemonicParsing="false"
onAction="#coursesView" prefHeight="46.0" prefWidth="153.0" text="Courses">
              <font>
                <Font name="Calibri" size="18.0" />
              </font>
            </Button>
            <Button layoutX="101.0" layoutY="217.0" mnemonicParsing="false"
onAction="#examsView" prefHeight="46.0" prefWidth="153.0" text="Exams">
              <font>
                <Font name="Calibri" size="18.0" />
              </font>
            </Button>
            <Label layoutX="14.0" layoutY="272.0" text="Press to view enrollment information:"
textFill="WHITE">
              <font>
                <Font name="Calibri" size="18.0" />
              </font>
            </Label>
            <Button layoutX="101.0" layoutY="307.0" mnemonicParsing="false"
onAction="#enrollmentView" prefHeight="46.0" prefWidth="153.0" text="Enrollment info">
              <font>
                <Font name="Calibri" size="18.0" />
              </font>
            </Button>
          </children>
        </AnchorPane>
        <Button layoutX="416.0" layoutY="373.0" mnemonicParsing="false" onAction="#logout"
prefHeight="46.0" prefWidth="153.0" text="Logout">
          <font>
            <Font name="Calibri" size="18.0" />
          </font>
        </Button>
        <ImageView fitHeight="225.0" fitWidth="232.0" layoutX="368.0" layoutY="8.0"
opacity="0.5" pickOnBounds="true">
          <image>
            <Image url="@../../images/lplogo.png" />
          </image>
        </ImageView>
        <Label layoutX="18.0" layoutY="8.0" prefHeight="46.0" prefWidth="271.0" text="Student
Login">
          <font>
            <Font name="Calibri Italic" size="35.0" />
          </font>
        </Label>
```

```
      </children>
    </AnchorPane>
  </children>
</AnchorPane>
```

18. UpdateCourseView.java

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

<AnchorPane prefHeight="442.0" prefWidth="578.0" xmlns="http://javafx.com/javafx/8.0.241"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="controllers.ProfessorCoursesController">
  <children>
    <Button layoutX="416.0" layoutY="373.0" mnemonicParsing="false" onAction="#logout"
prefHeight="46.0" prefWidth="153.0" text="Logout">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
    <ImageView fitHeight="225.0" fitWidth="232.0" layoutX="368.0" layoutY="8.0" opacity="0.5"
pickOnBounds="true">
      <image>
        <Image url="@../../images/lplogo.png" />
      </image>
    </ImageView>
    <Label layoutX="18.0" layoutY="8.0" prefHeight="46.0" prefWidth="271.0" text="Courses
Panel">
      <font>
        <Font name="Calibri Italic" size="35.0" />
      </font>
    </Label>
    <AnchorPane layoutX="18.0" layoutY="86.0" prefHeight="276.0" prefWidth="327.0" style="-fx-
border-color: black;">
      <children>
        <Label layoutX="14.0" layoutY="6.0" text="Enter Code of course to update:">
          <font>
            <Font name="Calibri Italic" size="16.0" />
          </font>
```

```
        </Label>
        <TextField fx:id="txtCode" layoutX="14.0" layoutY="38.0" prefHeight="25.0"
prefWidth="132.0" />
        <Label layoutX="14.0" layoutY="72.0" text="Update Name:">
          <font>
            <Font name="Calibri Italic" size="16.0" />
          </font>
        </Label>
        <TextField fx:id="txtName" layoutX="14.0" layoutY="104.0" prefHeight="25.0"
prefWidth="132.0" />
        <Label layoutX="14.0" layoutY="139.0" text="Update Semester:">
          <font>
            <Font name="Calibri Italic" size="16.0" />
          </font>
        </Label>
        <TextField fx:id="txtSemester" layoutX="14.0" layoutY="171.0" prefHeight="25.0"
prefWidth="132.0" />
        <Label layoutX="172.0" layoutY="139.0" text="Update Classroom:">
          <font>
            <Font name="Calibri Italic" size="16.0" />
          </font>
        </Label>
        <TextField fx:id="txtClassroom" layoutX="172.0" layoutY="171.0" prefHeight="25.0"
prefWidth="132.0" />
        <Label layoutX="14.0" layoutY="210.0" text="Update Professor:">
          <font>
            <Font name="Calibri Italic" size="16.0" />
          </font>
        </Label>
        <TextField fx:id="txtProfessor" layoutX="14.0" layoutY="242.0" prefHeight="25.0"
prefWidth="132.0" />
        <Label layoutX="172.0" layoutY="210.0" text="Update Cost:">
          <font>
            <Font name="Calibri Italic" size="16.0" />
          </font>
        </Label>
        <TextField fx:id="txtCost" layoutX="172.0" layoutY="242.0" prefHeight="25.0"
prefWidth="132.0" />
      </children>
    </AnchorPane>
    <Button layoutX="416.0" layoutY="318.0" mnemonicParsing="false"
onAction="#goBackCourses" prefHeight="46.0" prefWidth="153.0" text="Back">
      <font>
        <Font name="Calibri" size="18.0" />
```

```
      </font>
    </Button>
    <Button layoutX="130.0" layoutY="378.0" mnemonicParsing="false"
onAction="#updateCourse" prefHeight="34.0" prefWidth="103.0" text="Submit">
      <font>
        <Font name="Calibri" size="18.0" />
      </font>
    </Button>
  </children>
</AnchorPane>
```

19. WelcomeView.java

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>


<AnchorPane prefHeight="368.0" prefWidth="687.0" xmlns="http://javafx.com/javafx/8.0.241"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="controllers.LoginController">
  <children>
    <Label layoutX="14.0" layoutY="260.0" text="Naroa Amezaga">
      <font>
        <Font size="19.0" />
      </font>
    </Label>
    <Label layoutX="203.0" layoutY="126.0" text="WELCOME TO LAKE PARK">
      <font>
        <Font name="System Bold Italic" size="27.0" />
      </font>
    </Label>
    <Label layoutX="135.0" layoutY="165.0" text="SECONDARY SCHOOL APPLICATION">
      <font>
        <Font name="System Bold Italic" size="27.0" />
      </font>
    </Label>
    <ImageView fitHeight="79.0" fitWidth="220.0" layoutX="456.0" layoutY="10.0"
pickOnBounds="true" preserveRatio="true">
      <image>
        <Image url="@../../images/it.png" />
```

```
      </image>
    </ImageView>
    <Label layoutX="14.0" layoutY="288.0" text="A20473682">
      <font>
        <Font size="19.0" />
      </font>
    </Label>
    <Label layoutX="14.0" layoutY="318.0" text="ITMD 510 - Fall 2020">
      <font>
        <Font size="19.0" />
      </font>
    </Label>
    <Button layoutX="318.0" layoutY="260.0" mnemonicParsing="false" onAction="#start"
prefHeight="39.0" prefWidth="95.0" text="Start">
      <font>
        <Font size="20.0" />
      </font>
    </Button>
  </children>
</AnchorPane>
```