# Application Requirements Specification

For

**Course Registration System**

Prepared By:

Mengyun Xie

# Table of Contents

# 1. Purpose and Scope Statement

The purpose of this project is to provide users with a service for course registration. The target users of the course registration system are all instructors and students. The scope of this project includes enrolling in courses, dropping courses, creating courses, and processing student requests to register for courses. In the future we may consider adding features such as visualization of student timetables and forms validation to this project.

# 2. Requirements Narrative

In this project, a course registration system will be created. Users will be able to create a new account and then log into the system using their username and password. There are two roles in this system, one is the instructor and the other is the student. After entering the system, depending on the user's role, the user will have the following different options:

When an instructor logs into the system:

1) View the profile of this instructor
2) Update the profile of this instructor
3) View all courses belonging to this instructor
4) View the information of a course
5) Update the information of a course
6) Create a new course belonging to this instructor
7) View all students belonging to a course
8) Approve the application of students for courses registration
9) Decline the application of students for courses registration

When a student logs into the system:

1) View the profile of this student
2) Update the profile of this student
3) View all course information and the corresponding course status. The course status includes pending and enrolled.
4) View the information of a course
5) Register for a course
6) Drop a course

# 3. Objectives

According to different roles, users have different permissions. For example, instructors can create new courses, but students can register courses created by instructors. In this section, we will storyboard the two roles separately.

## 3.1 The storyboard for the instructor.

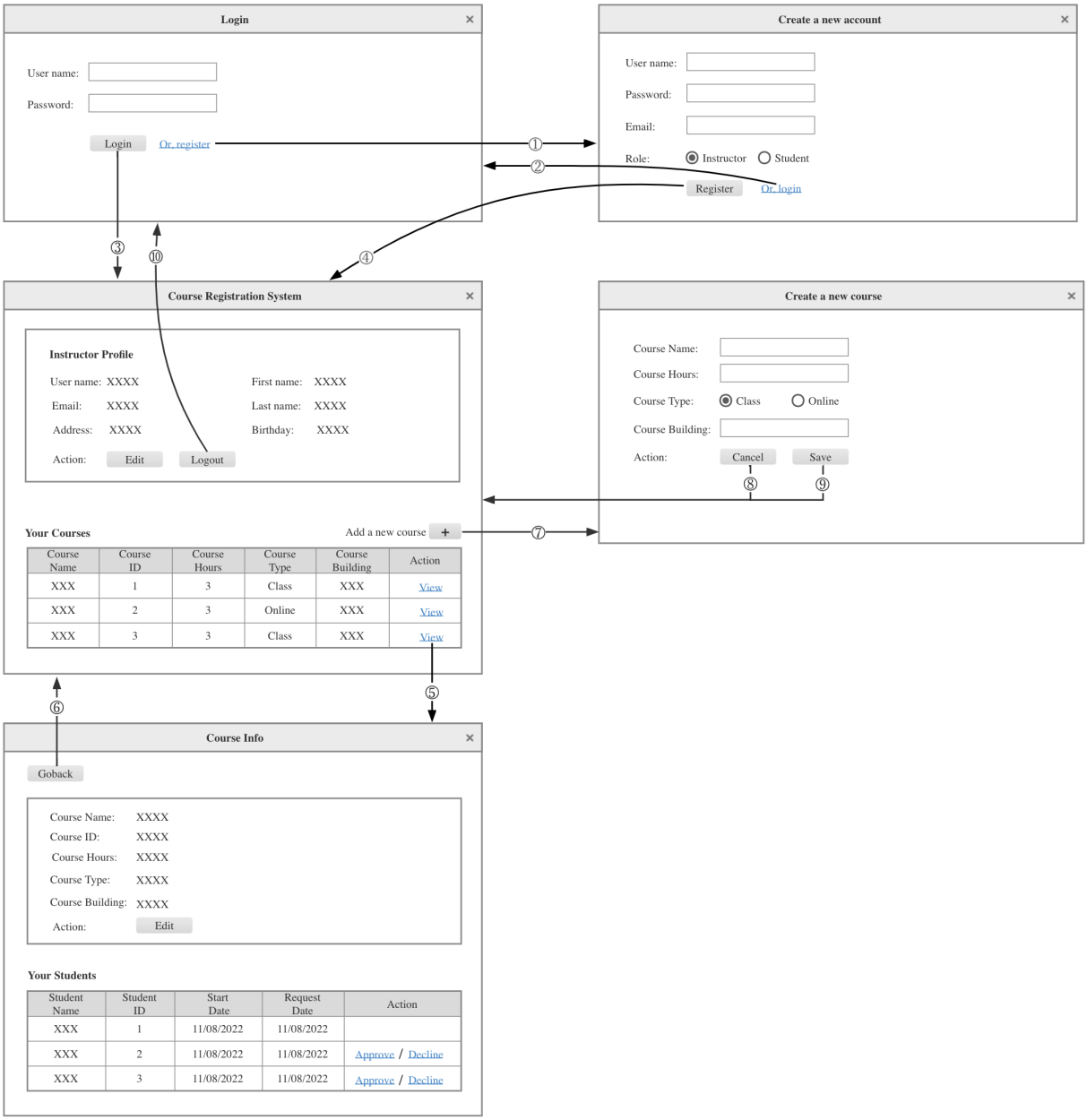


Figure 1: The storyboard of the instructor.

### 3.1.1 Use Case 1: User Registration

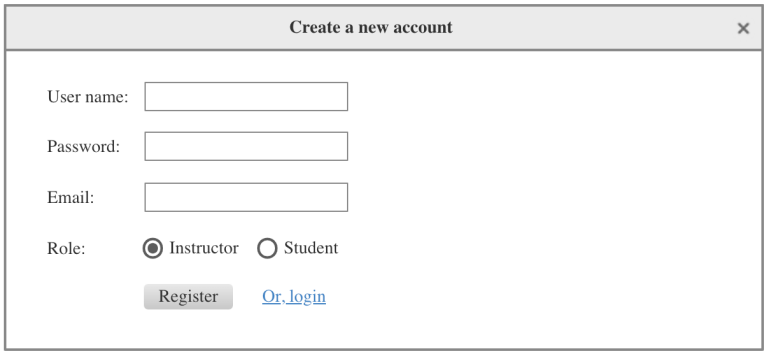


Figure 2: User register page.

Screen of user register page consists of:

- Window title
- Text fields:
  - Username
  - Password
  - Email
  - Role
- Button:
  - “Register” button
  - “Or, login” button

Actions:

When the user first enters the system, the user can fill in these text fields to create a new account. Click the “Register” button to enter the home page, as shown in Figure 1, Step 4. Click the “Or, login” button to enter the login page, as shown in Figure 1, Step 2.

### 3.1.2 Use Case 2: User Login


Figure 3: User login page.

Screen of login page consists of:

- Window title
- Text fields:
  - Username
  - Password
- Button:
  - “Or, register” button
  - “Login” button

Actions:

The user can fill in these text fields to enter course registration system. Click the “Login” button to enter the home page, as shown in Figure 1, Step 3. Click the “Or, register” button to enter the user registration page, as shown in Figure 1, Step 1.

### 3.1.3 Use Case 3: An instructor login to the system

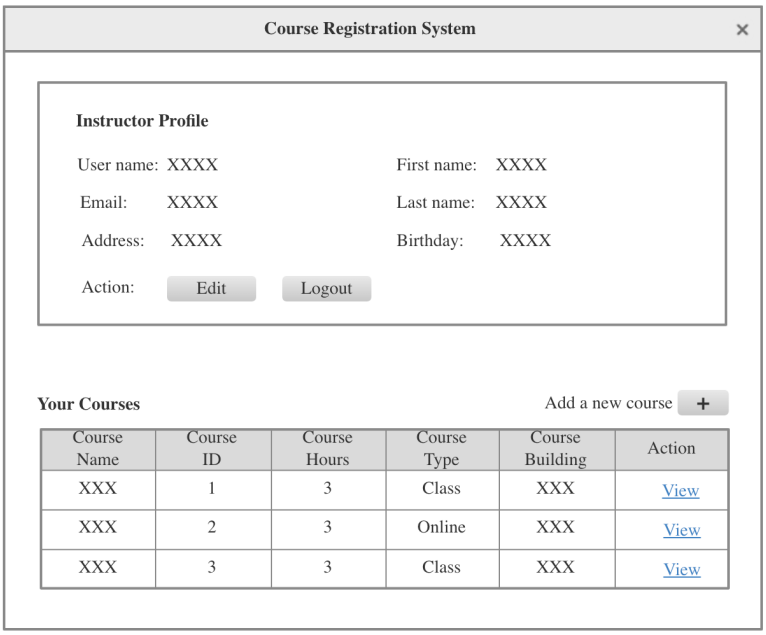


Figure 4: Home page of the instructor.

Screen of home page consists of:

- Window title
- User profile:
  - Username
  - Email
  - Address
  - First name
  - Last name

    - Birthday
    - “Edit” button
    - “Logout” button
- “Add a new Course” button
- Courses table:
    - “View” button

Actions:

In the home page, the user can view their profile and courses. Click the “Edit” button to update the user’s information. Click the “Logout” button to quit this system and jump to the login page, as shown in Figure 1, Step 10. Click the “Add a new course” button to enter the course create page, as shown in Figure 1, Step 7. Click the “view” button of the courses table to enter the course information page, as shown in Figure 1, Step 5.

### 3.1.4 Use Case 4: An instructor enters the course information page

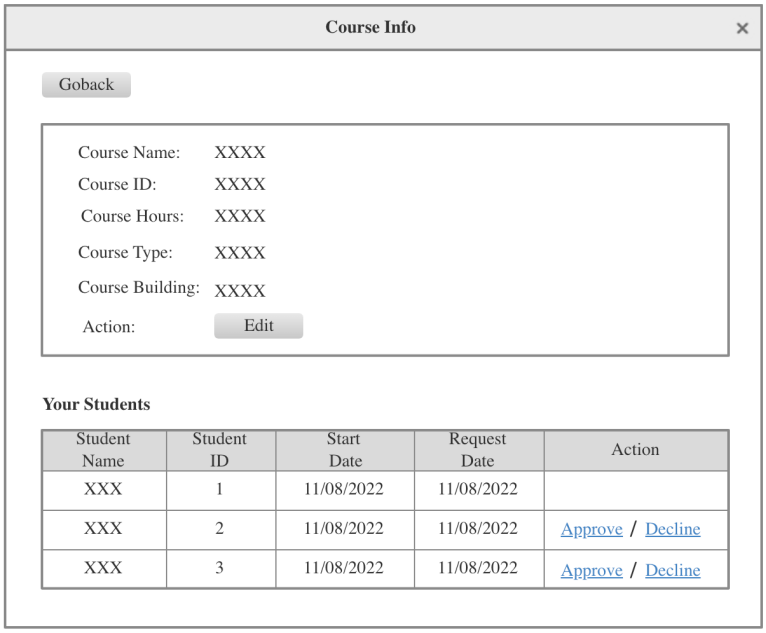


Figure 5: Course information page of the instructor.

Screen of course information page consists of:

- Window title
- “Goback” button
- Course information:
    - Course name
    - Course ID
    - Course Hours
    - Course Type
    - Course Building

    - “Edit” button
- Students table:
    - “Approve” button
    - “Decline” button

Actions:

In the course information page, the instructor can view the detail information of one course and their students in this course. Click the “Edit” button to update the course’s information. Click the “Approve” button of the students table to approve the course registration request for students. Click the “Decline” button of the students table to reject the course registration request for students. Click the “Goback” button to go back the home page, as shown in Figure 1, Step 6.

### 3.1.5 Use Case 5: An instructor creates a new course

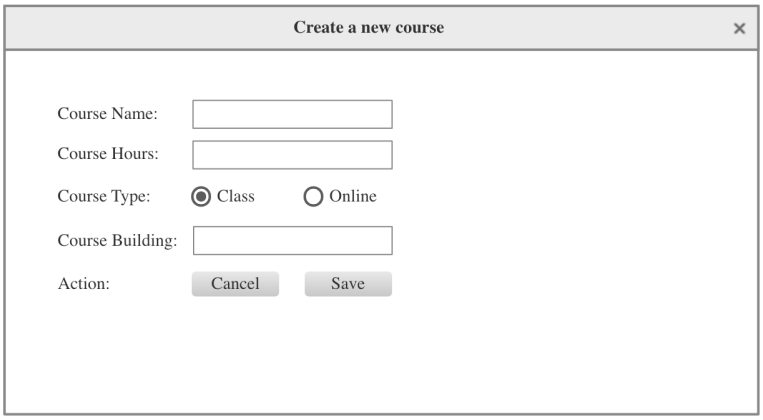


Figure 6: Course create page.

Screen of course create page consists of:

- Window title
- Text fields:
    - Course name
    - Course Hours
    - Course Type
    - Course Building
- Button:
    - “Cancel” button
    - “Save” button

Actions:

When the instructor enters this page, the instructor can fill in these text fields to create a new course. Click the “Cancel” button will do nothing just go back the home page, as shown in Figure 1, Step 8. Click the “Save” button will create a new course and go back the home page, as shown in Figure 1, Step 9.

## 3.2 The storyboard for the student.

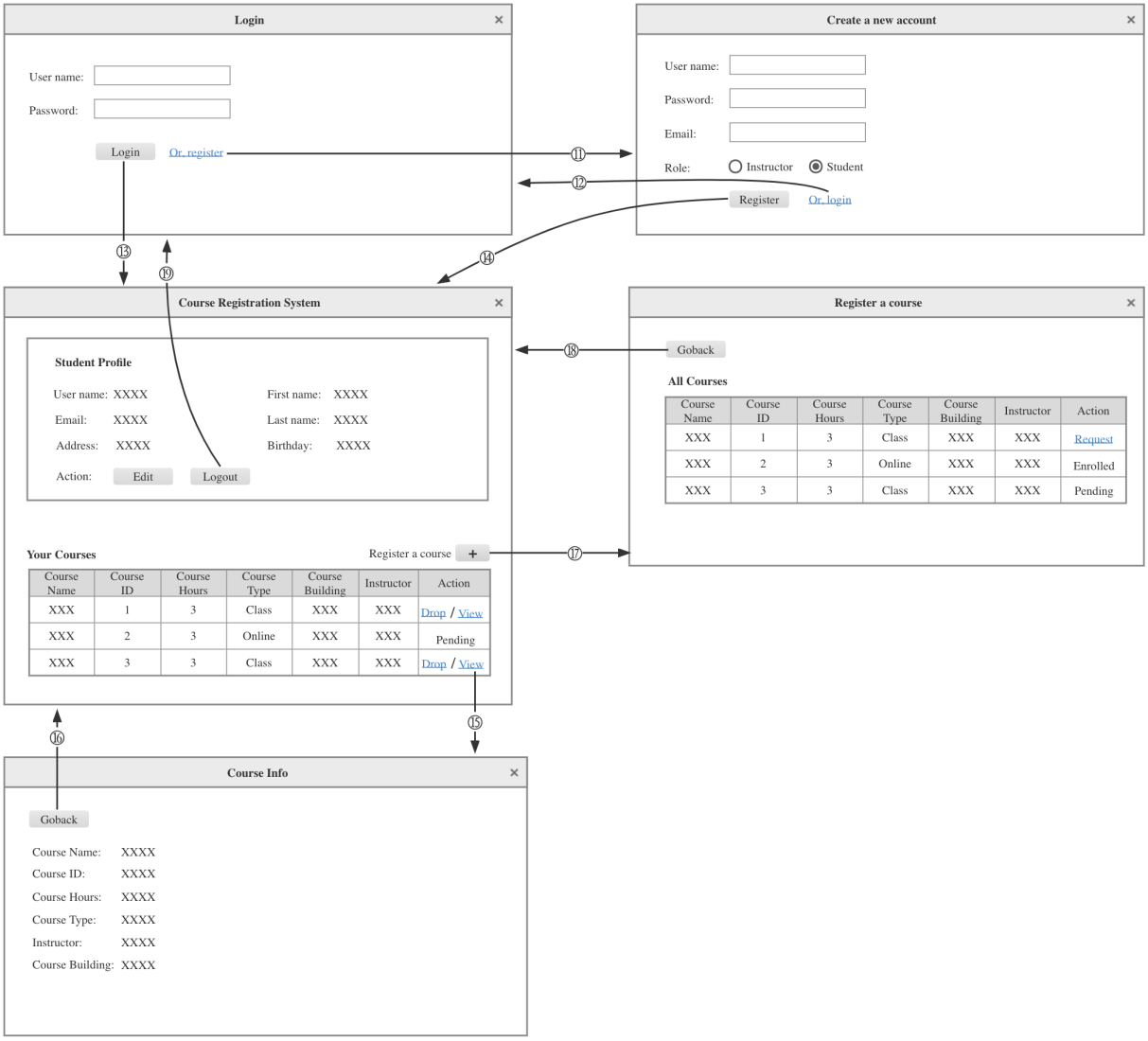


Figure 7: The Storyboard of the student.

### 3.2.1 Use Case 6: A Student login to the system

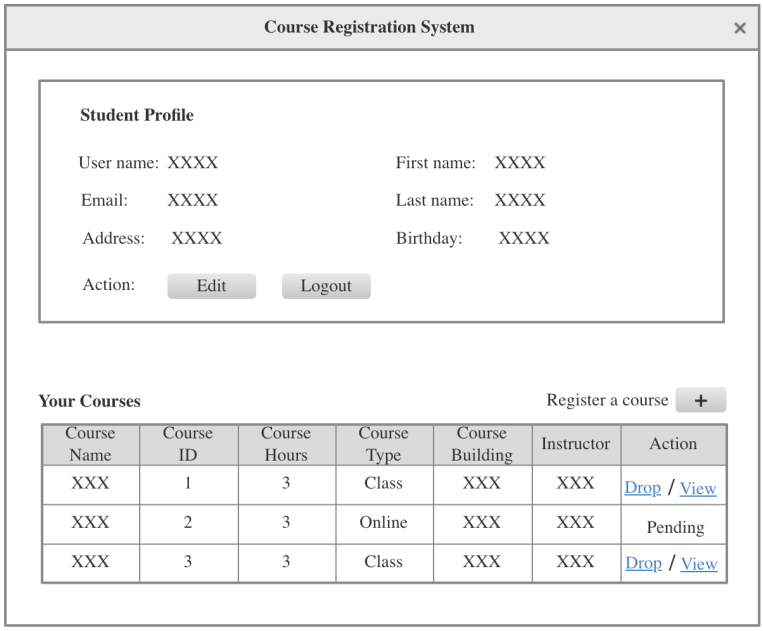


Figure 8: Home page of the student.

Screen of home page consists of:

- Window title
- User profile:
  - Username
  - Email
  - Address
  - First name
  - Last name
  - Birthday
  - Edit button
  - Logout button
- "Register a course" button
- Courses table:
  - "View" button
  - "Drop" button

Actions:

When a student enters the home page, they can view their profile and courses. Click the "Edit" button to update their information. Click the "Logout" button to quit this system and jump to the login page. Click the "Register a course" button to enter the course registration page, as shown in Figure 7, Step 17. Click the "view" button of the courses table to enter the course information page, as shown in Figure 7, Step 15. Click the "Drop" button of the courses table to drop this course.

### 3.2.2 Use Case 7: A student registers a course

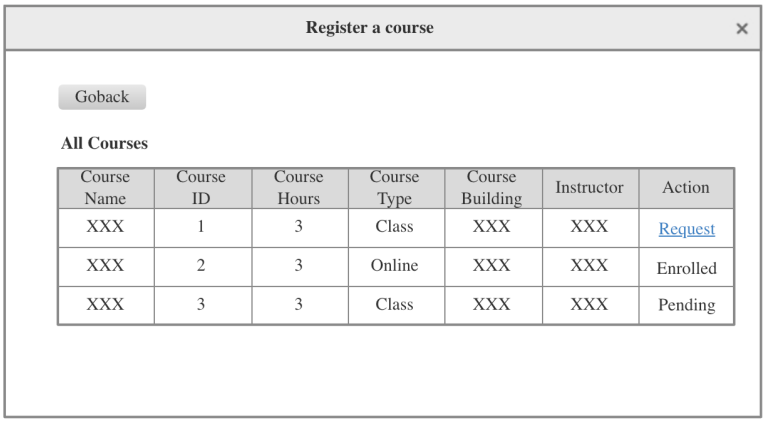


Figure 9: Course register page.

Screen of course register page consists of:

- Window title
- Go back button
- Courses table:
    - “Request” button

Actions:

When the student enters this page, the student can register courses of all courses in the system. Click the “Goback” button to go back the home page, as shown in Figure 7, Step 18. Click the “Request” button will send a request of course registration to this course’s instructor.

# 4. Functional Specification

In this section, we will discuss all functions of the course registration system and corresponding classes, Java libraries and databases.

## 4.1 The structure of the user interface

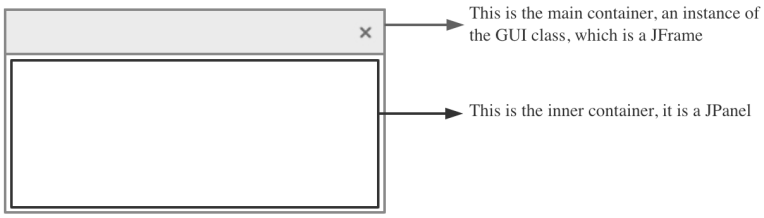


Figure 10: The structure of the user interface.

In this system, the main container is an instance of the GUI class, which is a JFrame. The inner container is an instance of corresponding JPanel class, such as LoginJPanel class, RegisterJPanel class, HomeJPanel class and so on. These JPanel instance serves as different pages, like login page, register page, home page and so on.

There are the various classes used to build the user interface for this system.

| Class Name | Function |
|---|---|
| GUI | The main container to contain different pages, it's a JFrame. |
| LoginJPanel | A JPanel serves as a login page. |
| RegisterJPanel | A JPanel serves as a user register page. |
| HomeJPanel | A JPanel serves as a home page. |
| CourseJPanel | A JPanel serves as a course information page. |
| CourseCreateJPanel | A JPanel serves as a course create page. |
| CourseRegisterJPanel | A JPanel serves as a course register page. |

## 4.2 Database

### 4.2.1 The entity relationship

In the course registration system, we use the SQLite library to build the database functionality of the system. To implement the function of this system, we created two kinds of entities, user entity and course entity. The user is divided into two roles, instructor and student. In this system, a course belongs to only one instructor, an instructor can have multiple courses, and a student can have multiple courses.

The entity relationship of this system is as shown in the figure below.

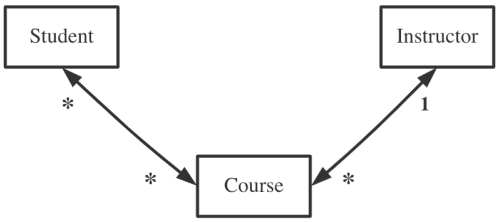


Figure 11: The entity relationship of this system.

### 4.2.2 The courseRegistrationSystem database

Therefore, we need to build four database tables to satisfy these relationships. The follow figure is the detail of these database tables: User table, Course table, Requested table and Enrolled table.

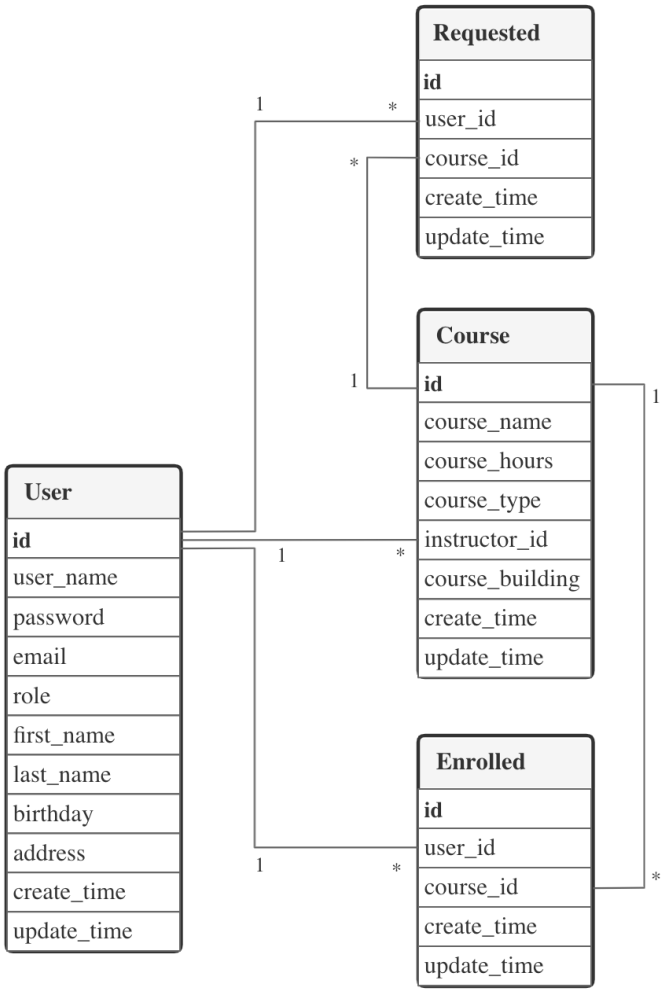


Figure 12: The courseRegistrationSystem database.

### 4.2.3 The SqliteDatabase class

And the SqliteDatabase class acts as a bridge between the java program and the database. The SqliteDatabase class contains many methods that allow the program to communicate with the courseRegistrationSystem database, such as: getUser, getCourse, updateUser, updateCourse, dropCourse, and so on.

There are some important methods of the SqliteDatabase class.

| Method Name | Detail |
|---|---|
| createConnection | Create connection between the program and database. |
| createUser | Create a user data and insert to the User table. |
| getUser | Get a user data from the User table, return an instance of the UserEntry class. |
| getUsers | Get a list of user data from the User table, return a list of instances of the UserEntry class. |
| updateUser | Update a user data to the User table. |
| createCourse | Create a course data and insert to the Course table. |
| getCourse | Get a course data from the Course table, return an instance of the CourseEntry class. |
| getCourses | Get a list of course data from the Course table, return a list of instances of the CourseEntry class. |
| updateCourse | Update a course data to the Course table. |
| dropCourse | Delete a data from the Enrolled table. |
| registerCourse | Add a data to the Requested table. |
| approveCourseRequest | Move a data from the Requested table to the Enrolled table. |
| declineCourseRequest | Delete a data from the Requested table. |

## 4.3 Functions of this system

The following list is 14 functions in our system, we will describe the related information of each function separately.

### 4.3.1 Login function

The program of this system uses the LoginJPanel class to instantiate a login page, as shown in Figure 3. On the login page, the user enters the username and password to log in to the system. After clicking the “Login” button, the program will go to the User table of the courseRegistrationSystem database to check if this user exists. If it exists, go to the home page. Otherwise, the system will prompt an error message.

### 4.3.2 Register user function

The program of this system uses the RegisterJPanel class to instantiate a user register page, as shown in Figure 2. On this page, users can fill in these text fields: username, password, email, and role type to create a new account. Click the “Register” button, the program will call createUser method of the SqliteDatabase class to insert a new user data into the User table of courseRegistrationSystem database, and then enter the home page.

### 4.3.3 View user information function

When enter the home page, the program will query the information of the current user in the database and use this user’s data to create an instance of the UserEntry class, then use this instance object to communicate with the java program to display the user. The UserEntry class includes a user's basic attributes in the database and some relative set and get methods.

### 4.3.4 Edit user information function

The system allows the user to update some user’s information, such as first name, last name, email, address, birthday. When the request is submitted, the program will update this information to the User table.

### 4.3.5 Logout function

When the “Logout” button is clicked, the program will run the logout function, delete the CurrentUser object used to store the current user's information, and jump to the login page.

### 4.3.6 Create a new course function

The program of this system uses the CourseCreateJPanel class to instantiate a course create page, as shown in Figure 6. On this page, instructors can fill in these text fields: course name, course hours, course type, and course building to create a new course. Click the “Save” button and the

program will insert a new course data into the Course table and then return to the home page. Notice, only the instructor has permission to create this course.

### 4.3.7 Register a course function

When students login the system, they can register a course. Click the “Register a Course” button, and the program will jump to the course register page, which is instantiated by the CourseRegisterJPanel class, as shown in Figure 9. On this page, the program will use getCourses method of the SqliteDatabase class to get all the courses in the database, which will be a list of the CourseEntry class, and then display the list into the JTable. Since courses are set to three states in this system: unregistered, pending, enrolled, and only unregistered courses allow students to send course registration requests. After clicking the “Register a Course” button, the program will create a new data in the Requested table, waiting for the course owner to process this course registration request.

### 4.3.8 Drop a course function

When a student login the system, in the courses table of the home page, Students can click the “Drop” button to drop out of an Enrolled course. When the “Drop” button is clicked, the program will call the dropCourse method of the SqliteDatabase class to delete the current student's data from the Enrolled table of the courseRegistrationSystem database.

### 4.3.9 View list of courses

In the home page, the program will call getCourses method of the SqliteDatabase class to get all the courses of the current user, which will be a list of instances of the CourseEntry class, and then the program will call methods of the CourseEntry class to display the data in the JTable.

### 4.3.10 View course information function

The program of this system uses the CourseJPanel class to instantiate a course information page. On this page, the program calls the getCourse method of the SqliteDatabase class to get the information of the current course in the Course table, which will be an instance of the CourseEntry class, and calls methods of the CourseEntry class to display the course information.

### 4.3.11 Edit course information function

In course information page, based on the role, instructors have permission to edit the course information. The program will call the updateCourse method of the SqliteDatabase class to update the information of this course in the Course table.

### 4.3.12 View list of students

In course information page, instructors have permission to view all students under this course. The program will use getUsers method of the SqliteDatabase class to get all the students of the current course, which will be a list of instances of the UserEntry class, and then the program will call methods of the UserEntry class to display the data in the JTable.

### 4.3.13 Approve the course registration request

In the home page, the instructor can process students' course registration requests. When they click the approve button for a course, the program will move this data from the Requested table to the Enrolled table.

### 4.3.14 Decline the course registration request

When the instructor click the decline button for a course, the program will delete this data from the Requested table.

# 5. Logic Specification

In this section, we will present various flowcharts of the course registration system to visualize the UI construction process, page router mechanism and some important functions of the system.

## 5.1 UI Building Flow

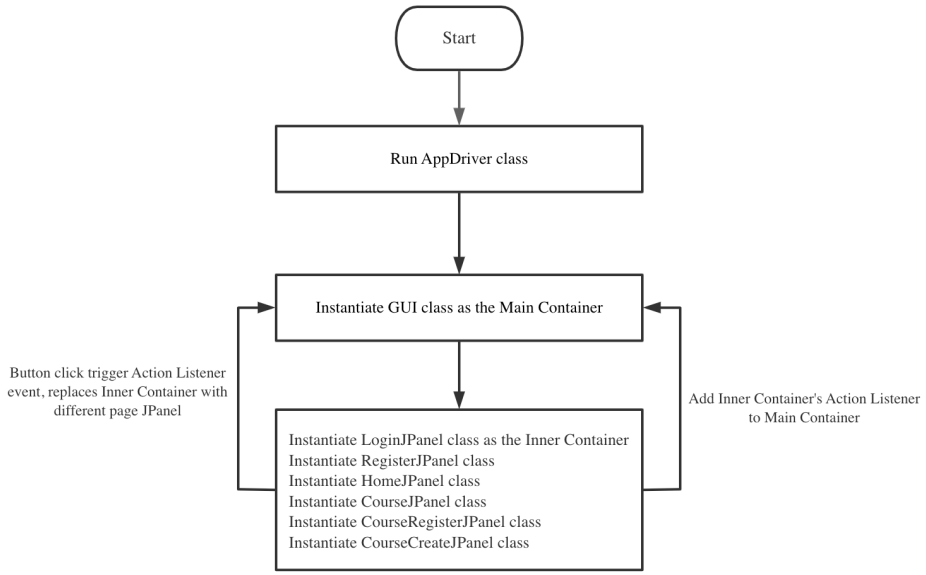


Figure 13: The UI building flow

## 5.2 Page Router Flow

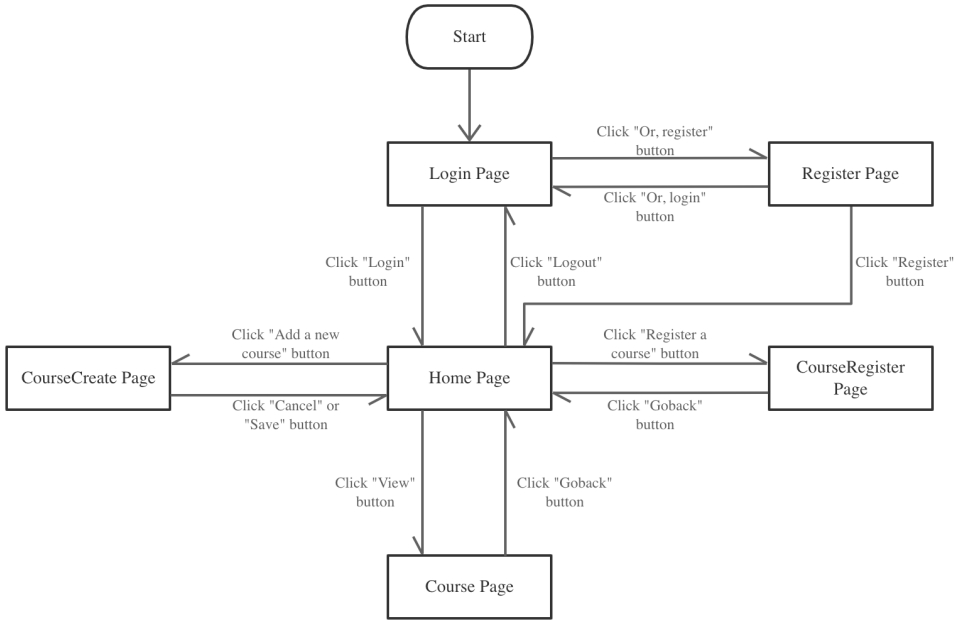


Figure 14: The page router flow

## 5.3 Key Function Flows

### 5.3.1 Create User Flow

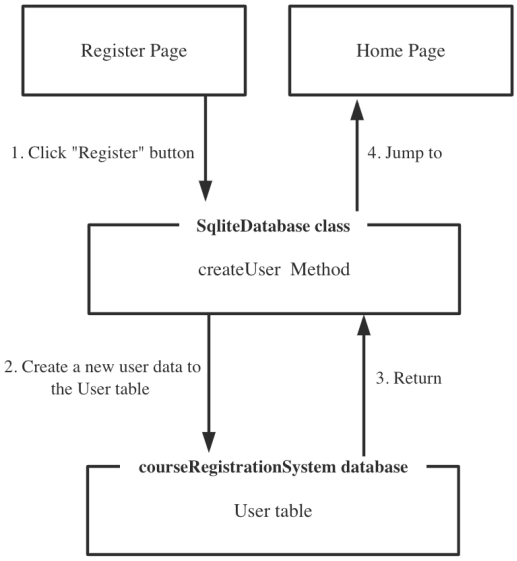


Figure 15: The create user flow

### 5.3.2 Login Flow

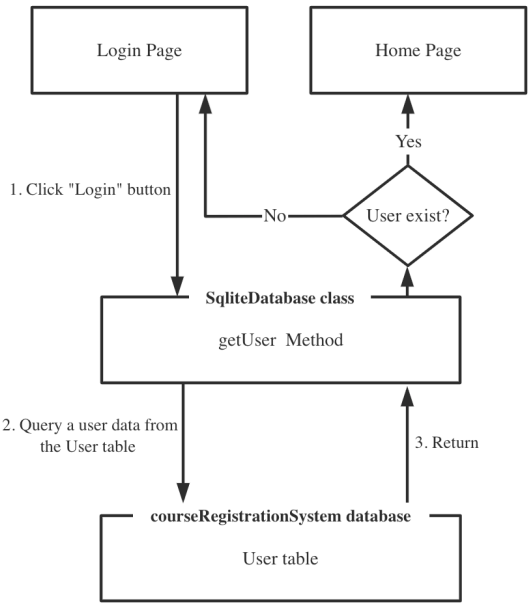


Figure 16: The login flow

## 5.3.3 Create Course Flow

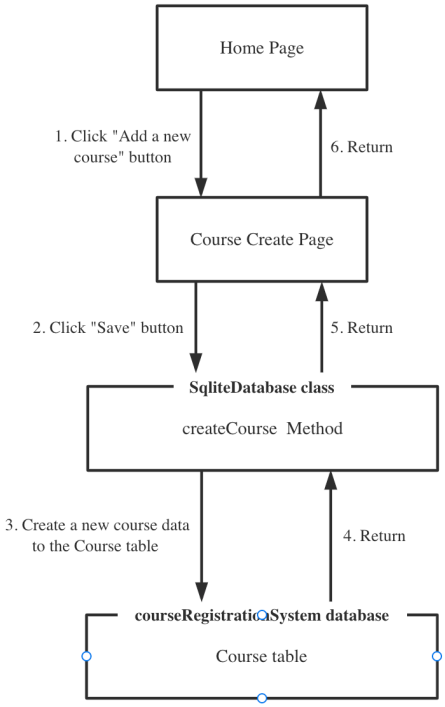


Figure 17: The create course flow

## 5.3.4 Register Course Flow

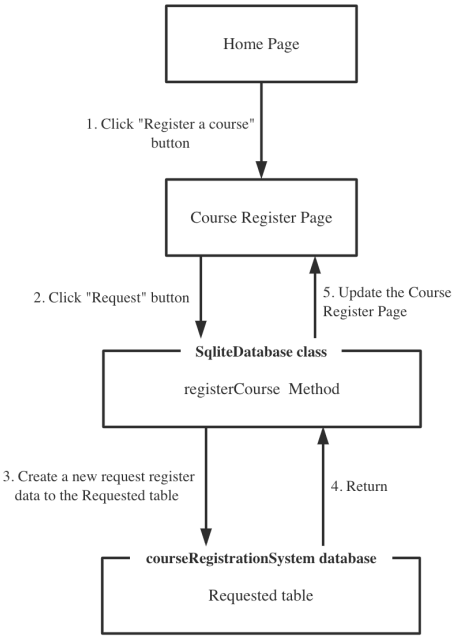


Figure 18: The register course flow

## 5.3.5 Drop Course Flow

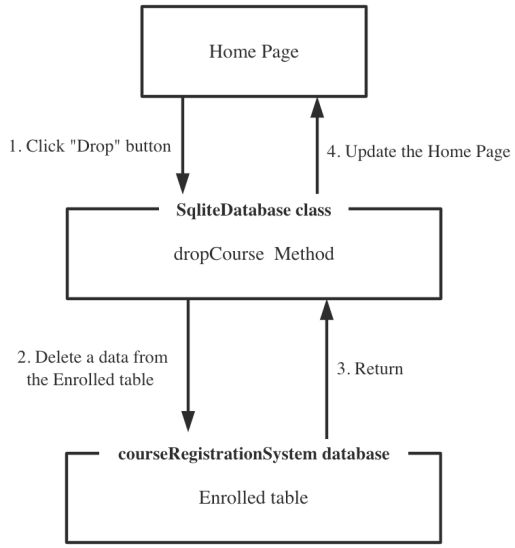


Figure 19: The drop course flow

## 5.3.6 Approve Course Flow

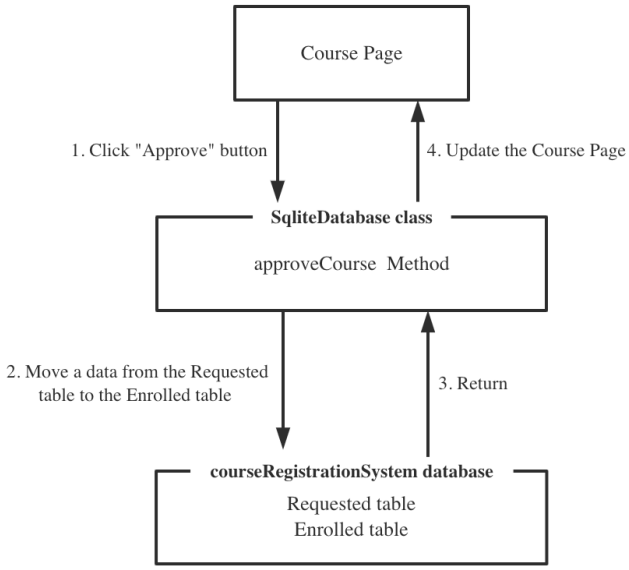


Figure 20: The approve course flow

## 5.3.7 Decline Course Flow

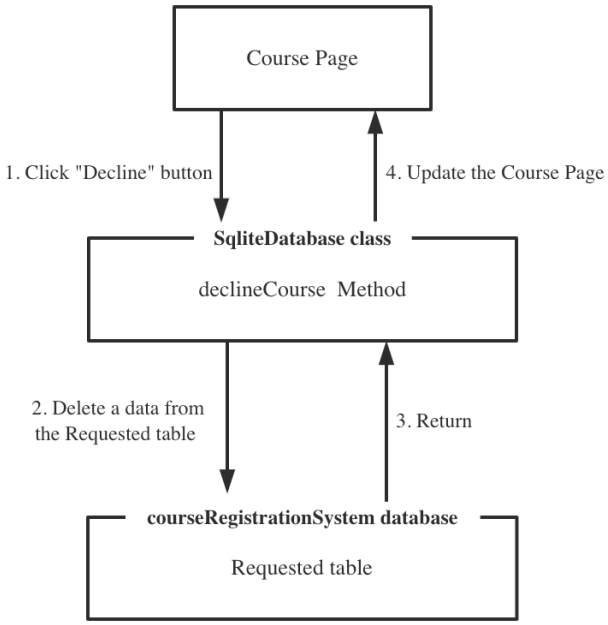


Figure 21: The decline course flow