The Hashemite University, Zarqa, Jordan Faculty of Prince Al-Hussein Bin Abdallah II for Information Technology Software Engineering Department

Exam Scheduling System

A project submitted. In partial fulfillment of the requirements for the B.Sc. Degree in Software Engineering

Submitted By:

Salasabeel Abdullah Adnan Hassouneh	2138744
Rama Salah Ahmad Al-Sheikh	2138897
Ameera Nafed Abd Al-Fattah Mohammad	2143217
Sara Yaseen Ibrahim Johar	2137171

Supervised by:

Dr. Bashar Abdul Kareem Al-Shboul

Jan/2025

CERTIFICATE

It is hereby certified that the project titled < **Exam Scheduling System** >, submitted by undersigned, in partial fulfillment of the award of the degree of “bachelor’s in software engineering” embodies original work done by them under my supervision.
All the analysis, design and system development have been accomplished by the undersigned. Moreover, this project has not been submitted to any other college or university.

Salasabeel Abdullah Adnan Hassouneh 2138744
Rama Salah Ahmad Al-Sheikh 2138897
Ameera Nafed Abd Al-Fattah Mohammad 2143217
Sara Yaseen Ibrahim Johar 2137171

ABSTRACT

This project is a special website designed for Hashemite University/Prince Al Hussein bin Abdullah II Faculty for Information Technology; it aims to reduce conflicts in exam times as much as possible by making the exam reservation process more organized.

Exam scheduling system will provide an easy-to-use interface for all users, the website supplied by a notification messages sent to coordinator contains the important and specific dates for making exams , then the website will dismiss any date for making exams outside the specific period according to the exam type, it will give the ability for coordinator to reserve an exam using the website with consideration gives to available classrooms, time slots and invigilators, if the exam has been reserved when another exam is already scheduled, a notification will be sent to the coordinator with a message show the conflicting. Also, the teacher can show the exam schedule and request edit for the reservation and send this request to coordinator to respond to it. The invigilator can show the exam schedule and get notifications to monitor the exam, he can enter excuses if he is not free at this time. And students can show the schedule and get notifications about their courses and exams.

The project aims to simplify the exam reservation process, enhance efficiency, and improve academic experience for students and faculty members.

ACKNOWLEDGEMENT

Thanks, and appreciation. We would like to thank many people for their time, guidance, and support throughout this process.

We thank God Almighty for granting us the blessing and helping us to reach this level.

It gives us great pleasure to express our sincere thanks and appreciation to our supervisor **Dr. Bashar Abdul Kareem Al-Shboul** for his support, guidance, assistance, and suggestions. We are also grateful to all professors, educators, and colleagues at the College of Information Technology, for their valuable support and contributions, big thanks to our family and friends for their time and support.

Table of contents

Contents

List of Tables

List of Figure

Chapter 1: Introduction

1.1 Overview

There is no doubt that during the previous and recent period the number of students has increased, as a result it is mandatory to do multi sessions exams or schedule more than one exam on the same day and maybe at the same time. This system aims to help with scheduling exams in college without conflicts in time, also, to manage the exam reservation process.

1.2 Project motivation

Q1: What are the reasons behind your choice to develop this project?

In recent years it has been noticed that many students suffer from conflicts in their exam times, because the instructors do not know the time or date for other courses' exams.

Students show their dissatisfaction with this problem because it is affecting the quality of their studying process, and sometimes the student is forced to do make up exam.

Q2. Why your project is important?

This system will reduce the conflicts between exams and make reserving exams more organized, and the student will feel satisfied when knowing that no more exams will be at the same time.

Q3. What is the new idea that has been proposed by this project?

At Hashemite University, there is a similar system only used for final exams by admission and registration unit, but this system will be used in the IT college directly.

1.3 Problem Statement

The students may have two or more exams on the same day and may be at the same time, so students will be forced to take one of them as a make-up exam then students will not be satisfied in this trial, students usually hate make up exams.

1.4 Project Aim and Objectives

The main goal for this project is to organize the reserving and scheduling exam process and make it more satisfying for students, coordinators and teachers.

The goal can be achieved by enabling reserve exams when another exam is already reserved at the same time, especially when the two courses are for the same year, also, enable the coordinator to request an exam when the selected date for it is out of the specified period, according to the exam type first, second, midterm.

1.5 Project Scope

Coordinator: He can register on the site and access it through the registration screen, and he requests reservation, manage it, and view the exam list.

Administrator: He can register on the site and access it through the registration screen, also he can add user or course or classroom.

Student: He can register on the site and access it through the registration screen and view the reservation exam list.

Teacher: He can register on the site and access it through the registration screen, view the reservation exam list and request change.

Invigilators: He can register on the site and access it through the registration screen, he can view his exams to monitor, also he can make excuses for monitoring any exam because of his own time or conditions.

1.6 Project Software and Hardware Requirements

Hardware requirements:

1. Device connected to internet.

2. Core i3 or above.

3. 4GB RAM or above.

Software search engines:

can work on Windows, Mac and Linux.

1.7 Project Limitation

1. Limited time for project construction.
2. Require large database to include all students.
3. Limited number for classroom.

Not included in the project:

1. The application does not available without the internet.

1.8 Project Expected Output

1. Optimized exam schedule.
2. Avoid time conflicts between exams.
3. Facilitate Invigilators assignment.
4. Facilitate Room assignment.
5. Send Notifications to both lecturer and student.

1.9 project schedule

Table 1: Project Schedule

Activity	Start date	Finish date
Introduction	28-3-2024	31-3-2024
Literature Review	1-4-2024	4-4-2024
Requirement Engineering and Analysis	5-4-2024	14-4-2024
Architecture and Design	15-4-2024	5-5-2024
Implementation Plan	24-6-2024	24-11-2024
Testing plan	29-11-2024	14-12-2024
Conclusion and Result	19-12-2024	24-12-2024

1.10 Project, product, and schedule risks

1. A Problem in linking the right jobs to the right person
2. A link between the administrator, the teacher, and the student (such as: setting appropriate times, not having more than one exam conflict at the same time), and this requires time and great effort to avoid the occurrence of errors.
3. Accuracy and focus in obtaining data, saving it, and adding it to databases
4. May need approval from the college dean to start implementing the project, as this requires a lot of time.
5. Giving us permission to access the university's databases to obtain rooms information and the number of students and teachers.

1.11 Report Organization

1. Chapter 2 introduces a discussion on reviewing some of the available literature related to the project.

2. Chapter 3 lists the requirements analysis that involves requirements elicitation, non- functional user requirements, stockholders, and the use case diagram and workflow for each use case.

3. Chapter 4 presents the System architecture, design, all components, and the UML diagrams generally (software design) all it is described in this chapter.

4. Chapter 5 will discuss the implementation plan and the programming language that we used to develop this project.

5. Chapter 6 talks about testing plans and software testing.

6. Chapter 7 presents future work and concludes the report.

Chapter 2: Literature Review

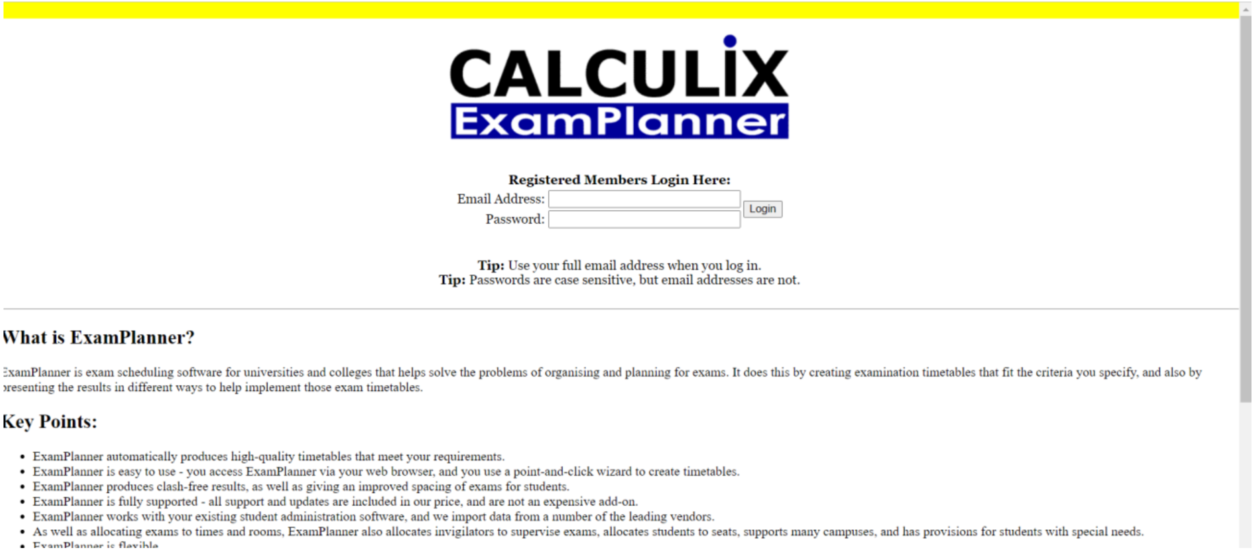
2.1 introduction

This website helps to optimize, organize and facilitate exam scheduling and reservation process. This website will benefit both coordinators and students, coordinators will no longer reserve exams manually which result in conflicts and problems for students, and students will be more comfortable with their exams time.

There are many websites that are similar in the idea to this website. From our point of view, we will explain the difference between them, the features that distinguish our project from others, the problems faced by the current system, and the proposed solutions.

2.2 Existing systems

- **Exam Planner**
  Exam Planner is exam scheduling software for universities and colleges that helps solve the problems of organizing and planning for exams. It does this by creating examination timetables that fit the criteria you specify, and by presenting the results in different ways to help implement those exam timetables [1].



Registered Members Login Here:
Email Address:
Password:

Tip: Use your full email address when you log in.
Tip: Passwords are case sensitive, but email addresses are not.

What is ExamPlanner?

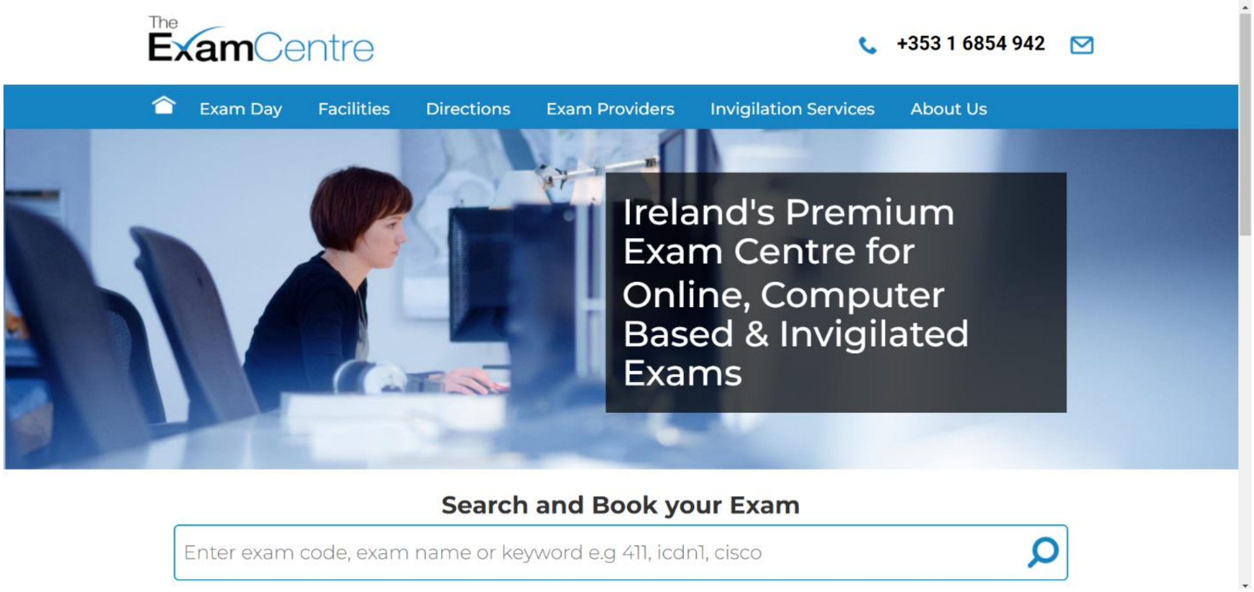
ExamPlanner is exam scheduling software for universities and colleges that helps solve the problems of organising and planning for exams. It does this by creating examination timetables that fit the criteria you specify, and also by presenting the results in different ways to help implement those exam timetables.

Key Points:

- ExamPlanner automatically produces high-quality timetables that meet your requirements.
- ExamPlanner is easy to use - you access ExamPlanner via your web browser, and you use a point-and-click wizard to create timetables.
- ExamPlanner produces clash-free results, as well as giving an improved spacing of exams for students.
- ExamPlanner is fully supported - all support and updates are included in our price, and are not an expensive add-on.
- ExamPlanner works with your existing student administration software, and we import data from a number of the leading vendors.
- As well as allocating exams to times and rooms, ExamPlanner also allocates invigilators to supervise exams, allocates students to seats, supports many campuses, and has provisions for students with special needs.
- ExamPlanner is flexible

Figure[1] :Exam Planner website

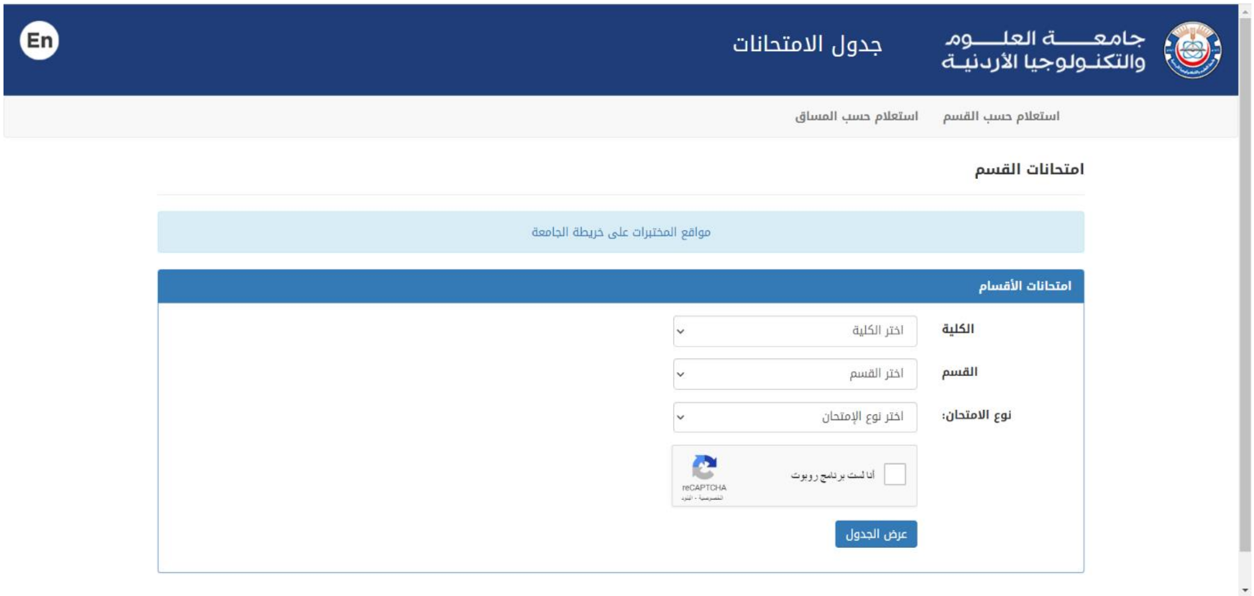
- **The Exam Centre**
  The Exam Centre was established in 2009 and has become Ireland’s largest dedicated exam center designed specifically for those wishing to take computer based, online or paper-based exams, you can search and book your exam based on available timeslot [2].



Ireland's Premium Exam Centre for Online, Computer Based & Invigilated Exams

Figure[2]:The Exam Centre

- **Jordan University of Science and Technology** Exam System [3].



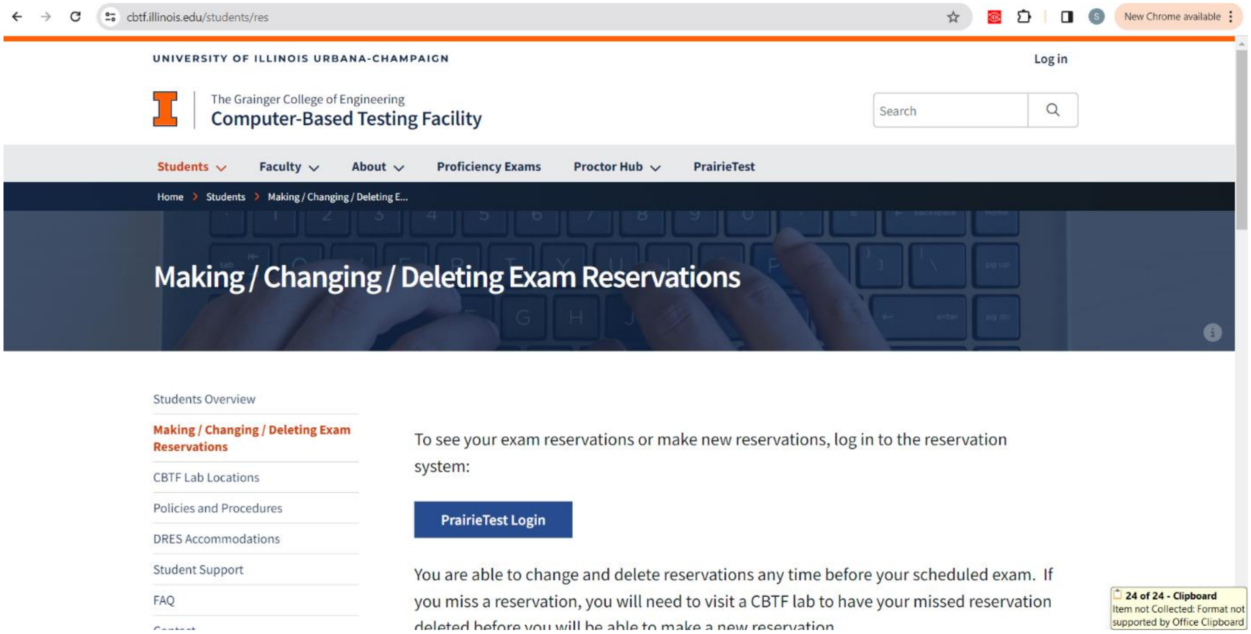
امتحانات القسم

امتحانات الأقسام

الكلية	اختر الكلية
القسم	اختر القسم
نوع الامتحان:	اختر نوع الإمتحان

Figure [3]:Jordan University of Science and Technology: view schedules

- **The university of Illinois Urbana-Champaign Exam reservation System**

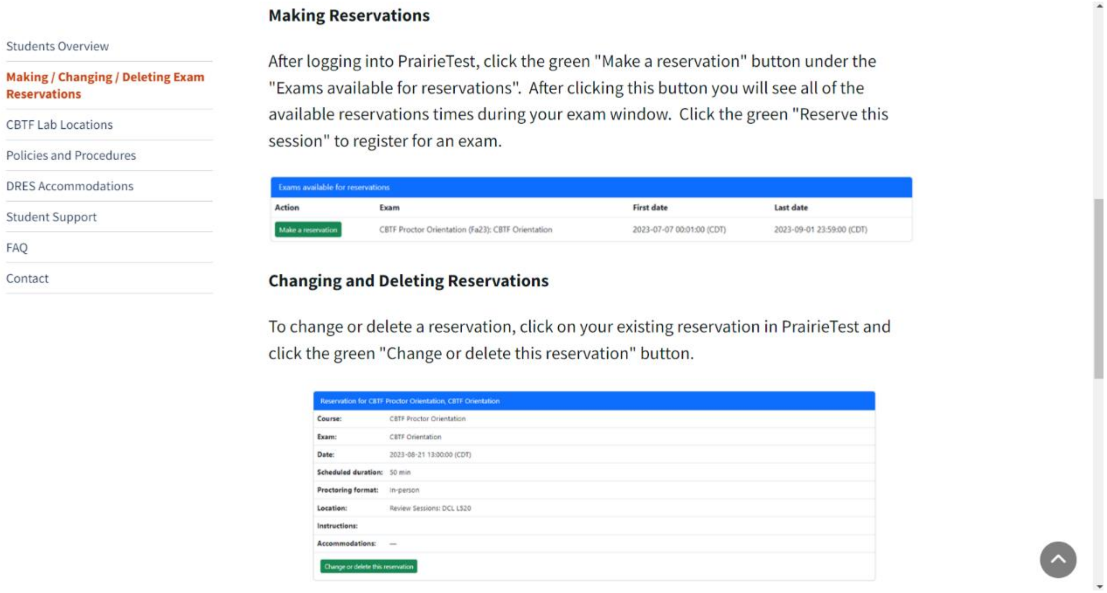

cbtf.illinois.edu/students/res
New Chrome available
UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN
Log in
The Grainger College of Engineering
Computer-Based Testing Facility
Search
Students
Faculty
About
Proficiency Exams
Proctor Hub
PrairieTest
Home > Students > Making / Changing / Deleting E...
Students Overview
Making / Changing / Deleting Exam Reservations
CBTF Lab Locations
Policies and Procedures
DRES Accommodations
Student Support
FAQ
PrairieTest Login
24 of 24 - Clipboard
Item not Collected: Format not supported by Office Clipboard

Making / Changing / Deleting Exam Reservations

To see your exam reservations or make new reservations, log in to the reservation system:

You are able to change and delete reservations any time before your scheduled exam. If you miss a reservation, you will need to visit a CBTF lab to have your missed reservation deleted before you will be able to make a new reservation.

Figure[4]:The university of Illinois Urbana-Champaign Exam reservation System(1)



Making Reservations

After logging into PrairieTest, click the green "Make a reservation" button under the "Exams available for reservations". After clicking this button you will see all of the available reservations times during your exam window. Click the green "Reserve this session" to register for an exam.

Exams available for reservations

Action	Exam	First date	Last date
Make a reservation	CBTF Proctor Orientation (Fa23): CBTF Orientation	2023-07-07 00:01:00 (CDT)	2023-09-01 23:59:00 (CDT)

Changing and Deleting Reservations

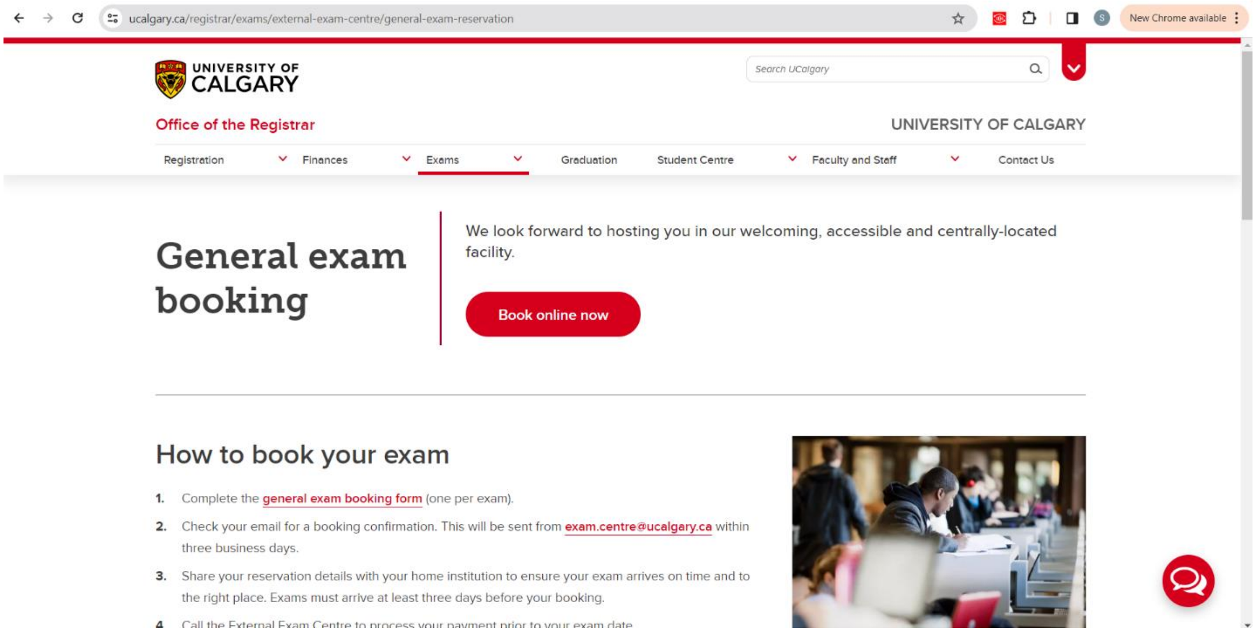
To change or delete a reservation, click on your existing reservation in PrairieTest and click the green "Change or delete this reservation" button.

Reservation for CBTF Proctor Orientation, CBTF Orientation

Course:	CBTF Proctor Orientation
Exam:	CBTF Orientation
Date:	2023-08-21 13:00:00 (CDT)
Scheduled duration:	50 min
Proctoring format:	in-person
Location:	Review Sessions: DCL L520
Instructions:	
Accommodations:	—

Figure[5]:The university of Illinois Urbana-Champaign Exam reservation System(2)

- **University of Calgary general Exam Booking**



General exam booking

We look forward to hosting you in our welcoming, accessible and centrally-located facility.

How to book your exam

1. Complete the general exam booking form (one per exam).
2. Check your email for a booking confirmation. This will be sent from exam.centre@ucalgary.ca within three business days.
3. Share your reservation details with your home institution to ensure your exam arrives on time and to the right place. Exams must arrive at least three days before your booking.
4. Call the External Exam Centre to process your payment prior to your exam date

Figure[6]:University of Calgary general Exam Booking

2.3 Overall Problems of Existing Systems

1. These websites don’t send notifications when any exam is reserved.
2. These websites don’t send notifications when the exams period starts.
3.These websites allow locate exams in different buildings.
4. Invigilators can’t reply to requests to accept it or deny it.

2.4 Overall Solution Approach

1. Our website will send notifications to students when any exam is reserved.
2. Our website will send notifications to the coordinator when the exams period starts to start reservations and scheduling.
3. Our website allows to locate exams only in IT college.
4. Invigilators can reply to a request to accept it or deny it.

Chapter 3: Requirement Engineering and Analysis

This chapter will present the list of individuals, or Business owners, who may affect the project or be affected by it and specify the type of stakeholder and show the use case diagram that shows the relationship between the user and the different use cases and shows functional and nonfunctional requirements.

3.1 Stakeholders

Table 2 : Stakeholders

Stakeholder	**Type**
Student	primary
Administrator	primary
Coordinator	primary
Teacher	primary
Invigilators	primary

1) **Student**:
   the person who studies at the information technology college.
2) **Administrator**:
   The person who manages the project and the resources.
3) **Coordinator**:
   the person who is responsible for identifying the characteristics of the exam.
4) **Teacher**:
   the person who teaches the subject.
5) **Invigilators**:
   people who monitor exams in the college.

3.2 Use case Diagram

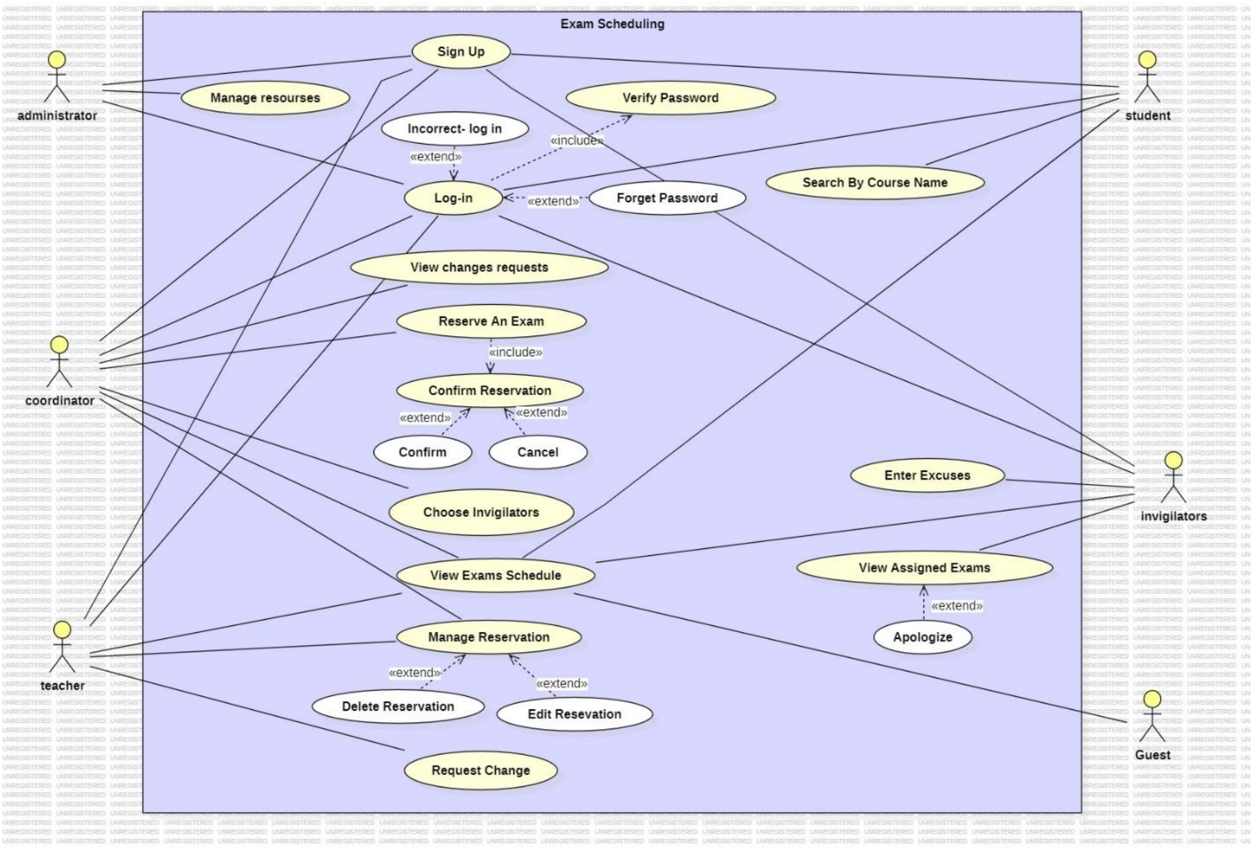


Figure [7] : Use Case Diagram

3.2.1 Use Case Section

Table 3: Basic Flow Sign Up

Use case	**Sign Up**
Description	Users can register an account in the site.
Actors	Administrator, Coordinator, Student, Invigilator, Teacher.
Precondition	Users must be connected to the internet and open the website on his device.
Basic Flow	1. Clicks the “Sign Up” on the page. 2. The system displays a signup page that allows users to fill in their ID number and password and email. 3. If the user is a guest, he can click on (Enter as a guest). 3. clicks “Sign Up” button. 4. The system displays the main page.
Postcondition	The user has an account and the user's details are stored in the database.

Table 4: Basic Flow & Alternative Flow Login

Use Case	**Log in**
Description	User tries to log into the system
Actors	Administrator, Coordinator, Student, Invigilator, Teacher.
Precondition	User has already had an account in the database of the web site.
Basic Flow	1. Click the “Login” on the page. 2. The system displays login pages to allow the user to fill in data (ID, password). 3. The user clicks “log in” button. 4. include (verify Password). 5. The system displays the home page.
Postcondition	The user login successful
Alternative flow	**[Incorrect log in]** 1. **After step 3** of the main flow. 2. The system does not find any matching details in the database. 3. The system redisplays the login page with an invalid login message. 4. The user can re-enter the correct details. **[Forget Password]** 1.**After step 1**, User selects the "Forgot Password" option on the login screen if he forgets his Password. 2. System prompts the user to enter their email address associated with the account. 3. User enters their email address and submits the form. 4. System validates the email address and sends a password reset link to the user's email address. 5. User checks their email, clicks on the password reset link, and is directed to a page where they can enter a new password. 6. User enters a new password and confirms it. 7. System verifies the new password and updates the user's account with the new password. 8. User receives confirmation that their password has been successfully reset and can now log in with the new password.

Table 5: Basic Flow & Alternative flow Reserve an Exam

Use Case	**Reserve an Exam**
Description	Allow users to create an exam schedule.
Actors	Coordinator
Precondition	Users must relate to the internet and login and the exam period start.
Basic Flow	1. User selects the option “Reserve an Exam” to create a new exam reservation. 2. User selects the exam course. 3. Users select the exam type (First, Second, Midterm). 4. system views the available dates, times, and available slots. 5. User selects a date, time, and available slot for the exam. 6. System view available classroom at the chosen time then user select one (he can reserve more than one room). He also can choose if he wants to reserve a classroom or lab or a college theater. 7. User confirms the reservation. 8. System create the reservation and updates the exam schedule.
Postcondition	The exam reservation is successfully created and added to the schedule.
Alternative flow	**[Resolve conflicts]** **After step3**, If the selected slot is not available (already reserved or conflicting with other courses that are taken on the same level of his course), the user is prompted to choose another slot.

Table 6: Basic Flow Resource Management

Use Case	**Resource Management**
Description	Allow the admin to manage the system resources.
Actors	Administrator
Precondition	Administrator log into the system and authenticated and authorized to access the system.
Basic Flow	1. Decide desire resource to manage
Postcondition	1.Users, courses, or classrooms are successfully added to the system. 2. Exam periods are reserved with specified durations.
Alternative flow	**After step 1**, the Administrator can Add users by entering user ID and choose faculty role, or Add courses by entering course names and ID, or Add classrooms by entering classroom ID. Reserve exam periods and specify their durations.

Table 7 : Basic Flow Choose Invigilators

Use Case	**Choose Invigilators**
Description	Allow User to choose people for monitoring exam.
Actors	coordinator
precondition	The coordinator must be connected to internet and logged in and the exam must be scheduled.
Basic flow	1. After the desired exam had been scheduled and reserved. 2. The coordinator selects the invigilator for the desired reservation.
postcondition	A notification will be sent to the invigilator with time of monitoring an exam.

Table 8: Basic Flow View Exam Schedule

Use case	**View exams schedule**
Description	Allow users to display the exam schedule
actors	Coordinator, student, invigilator, Teacher, guest.
Precondition	The user must be connected to internet.
Basic flow	1. The user opens the site. 2. The user click the “view exam schedule” button.
Post condition	The exam schedule is viewed.

Table 9 :Basic Flow & Alternative flow Manage Reservation

Use case	**Manage reservation**
description	Allow the user to edit and manage or delete the reservation.
Actors	Coordinator, teacher.
Precondition	The user must be connected to internet and logged in.
Basic flow	1. The user click manages reservation. 2. The system displays the exam reservation list. 3.The user selects the desired exam. 4.The user can edit or delete the selected exam.
Postcondition	The reservation may be edited or deleted.
Alternative flow	[delete reservation] **At step 4** the user can delete the reservation and set another one. [edit reservation] **At step 4** the user can modify the reservation by editing time, date, classrooms, invigilators or anything else then save changes.

Table 10: Basic Flow Request Change

Use case	**Request Change**
Description	allows teachers to request modifications to exam reservation.
Actors	Teacher
Preconditions	The coordinator must have scheduled the exams
Basic flow	1. The teacher logs into the system. 2. Selects the exam they wish to reschedule. 3. Specifies the new date and time for the exam. 4. send request to coordinator. 5. Obtain the coordinator’s approval to make changes to the exam schedule.
postconditions	The request sends successfully to the coordinator.

Table 11: Basic Flow Enter Excuse

Use case	**Enter Excuse**
Description	The excuses from invigilators regarding their inability to invigilate exams.
Actors	Invigilators
Preconditions	The invigilator had previously logged into the system.
Basic flow	1. Navigates to the Enter Excuse section. 2. Specifies the date of the excuse and its reason. 3. send the excuse.
postconditions	The invigilator's excuse is successfully recorded in the system.

Table 12: Basic Flow & Alternative flow View Assigned Exam

Use case	**View Assigned Exame**
Description	The invigilators view their assigned exam schedule to monitor.
Actors	Invigilators
Preconditions	The invigilators have been assigned to monitor an exam.
Basic flow	1. The invigilators log into the system. 2. Navigate to the Assigned Exams screen. 3. View exams assigned to him.
Alternative flow	**[Apologize]** **After step 3, if he can’t monitor the exam:** 1. users can apologize for specific exams. 2. Select the exam, they can’t monitor it. 3. enter the reason for the excuse. 4.send excuses to the coordinator.
Postconditions	The invigilator's excuse is successfully sent.

Table 13 Basic Flow View Change Requests

Use case	**View change Requests**
Description	Allow Coordinator to Accept or deny a request from teacher to change on specific reservation
Actors	Coordinator
Preconditions	-Login to the system. -request reach from teacher.
Basic flow	1. Open the list of requests for change. 2. select request 3.Accept Request or deny it.
postconditions	The confirmation sends to the teacher to gain access and start editing on reservation. or receive rejection message for his request

3.3 Non-Functional Requirements

- **performance**

  - The website must be responsive.

  - The website must support both Arabic and English languages.

  - No need for high quality devices to open the website.

- **Maintainable**

  - The website must be maintainable and developable.

  - The website must be flexible to add new features.

- **Availability**

  - The website must be available at any time if it is connected to the Internet.

- **Usability**

  - It gives users the ability to enter and exit the site easily.

  - The website should have an easy-to-use interface.

  - Instant translation of texts.

- **Security and protection**

  - Enter and exit the site securely.

  - Use safety requirements related to accessing and processing data.

- **Scalability**

  -The website must work on all operating systems.

Chapter 4: Design

4.1 Overview

This chapter will explain the components of the website and how they relate to each other through some diagrams and a full description of the website interfaces.

4.2 Software Architecture

4.2.1 Logical view (Class diagram)

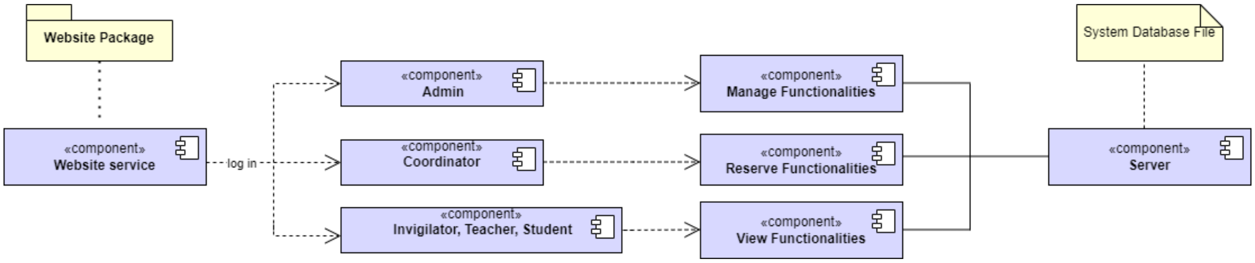


Figure [8] :Logical view diagram

4.2.2 Process View

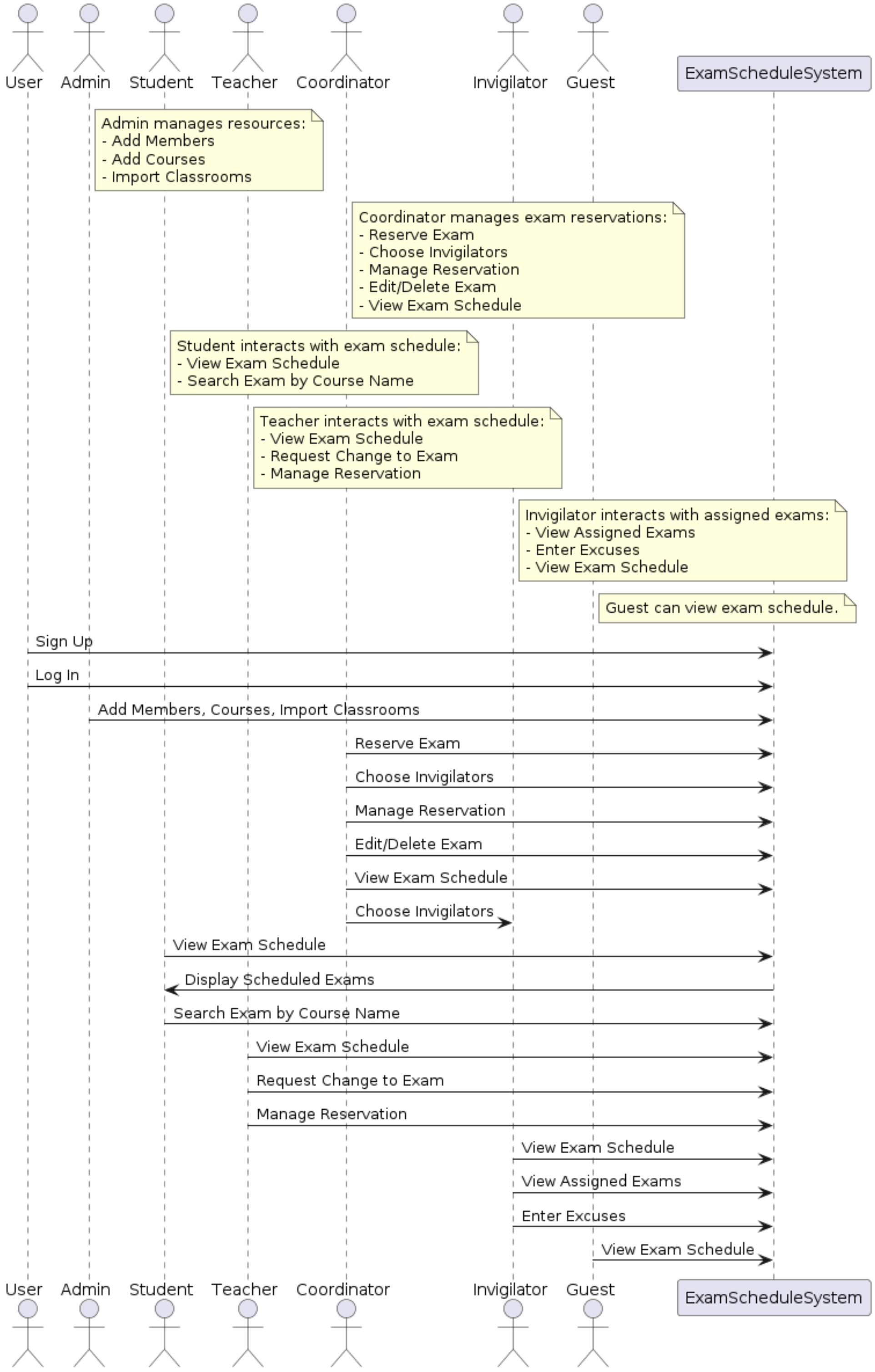


Figure [9] :Process View diagram

4.2.3 Physical View

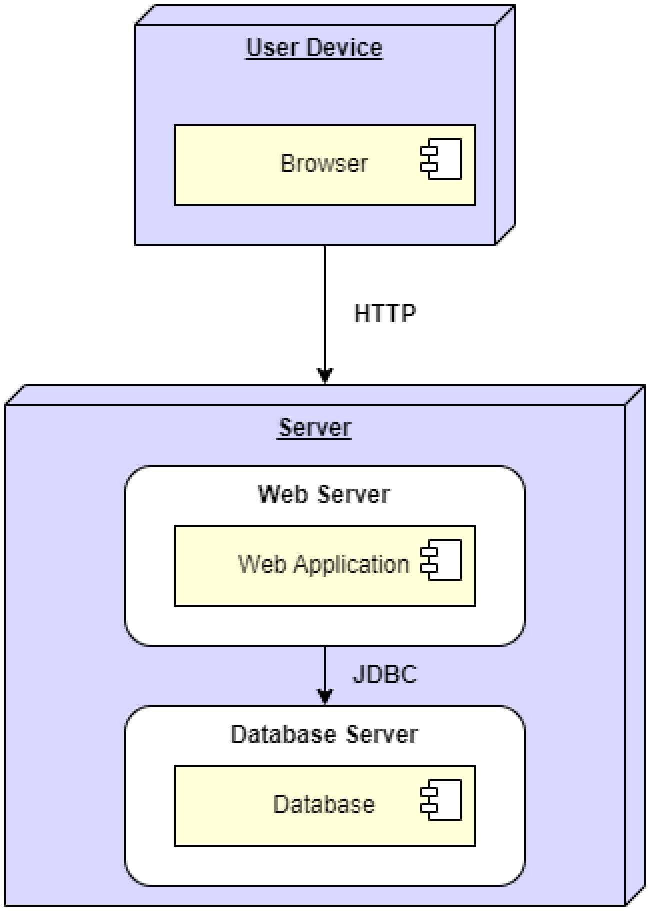


Figure [10] : Physical View diagram

4.3 Software design

4.3.1 UML sequence/communication diagram

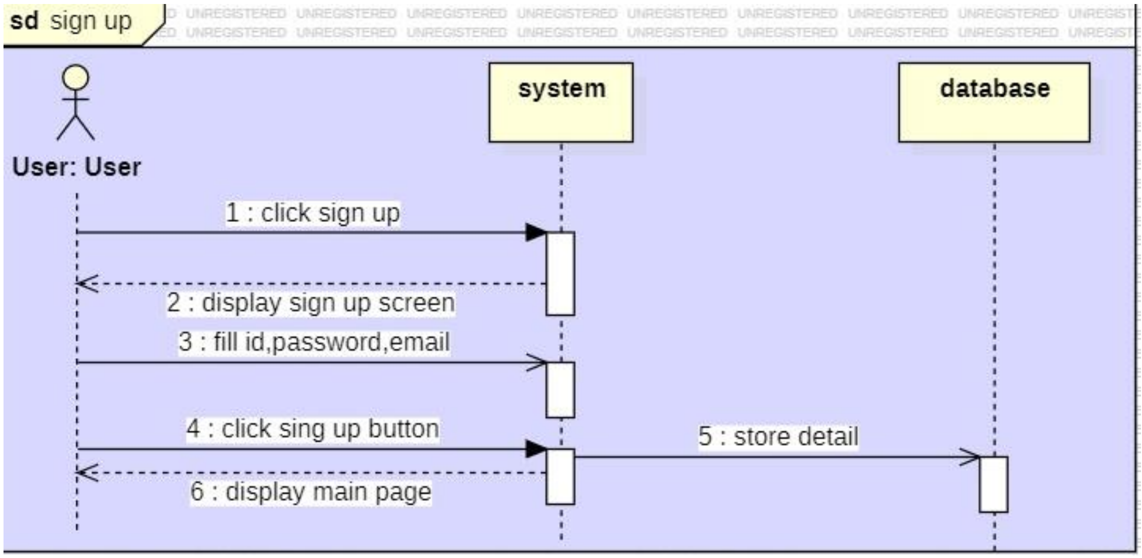

sd sign up
UNREGISTERED UNREGISTERED UNREGISTERED UNREGISTERED UNREGISTERED UNREGISTERED UNREGISTERED UNREGISTERED
User: User
system
database
1 : click sign up
2 : display sign up screen
3 : fill id,password,email
4 : click sing up button
5 : store detail
6 : display main page

Figure [11]:Sign Up Sequence diagram

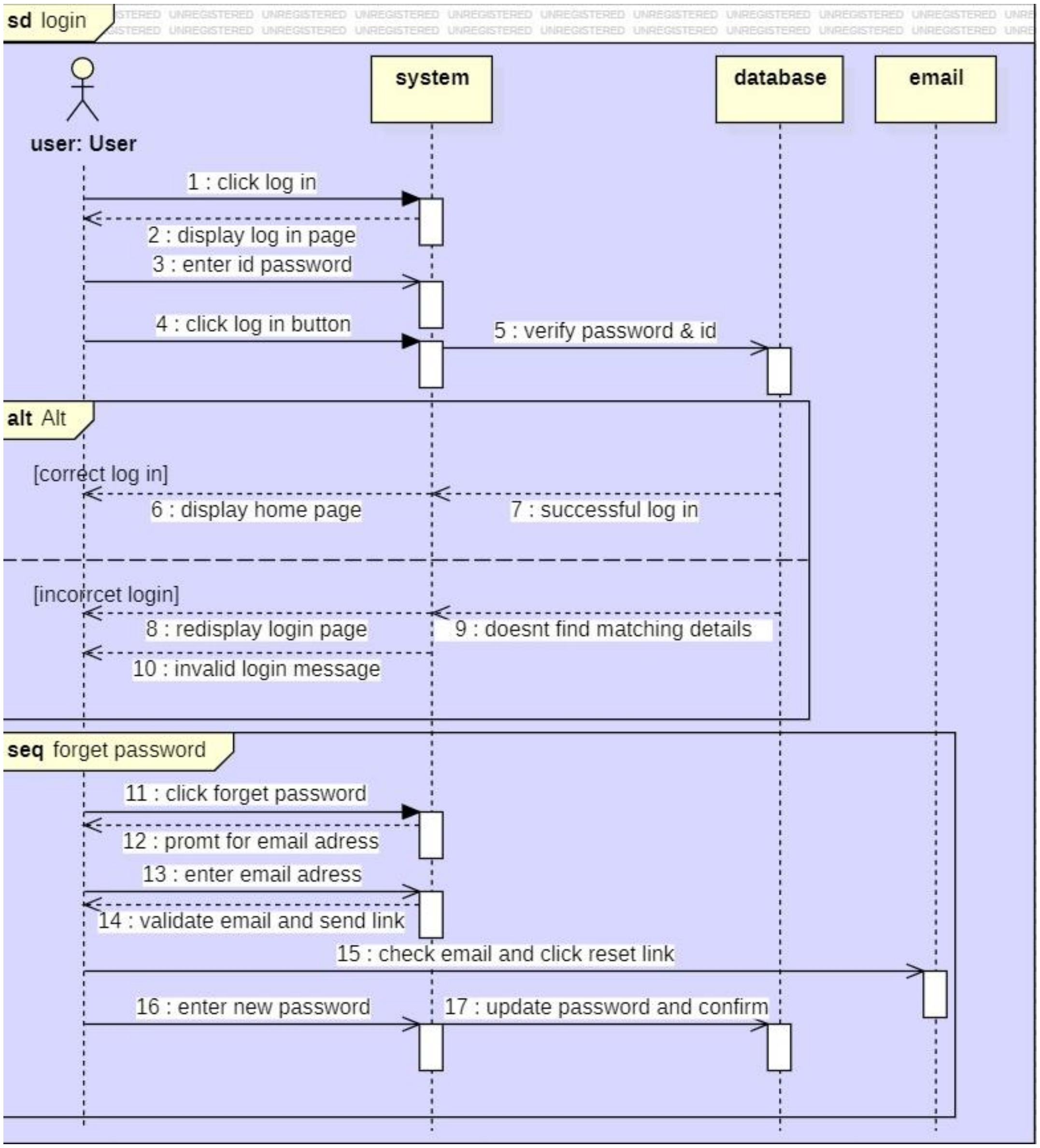


Figure [12]:Log In Sequence diagram

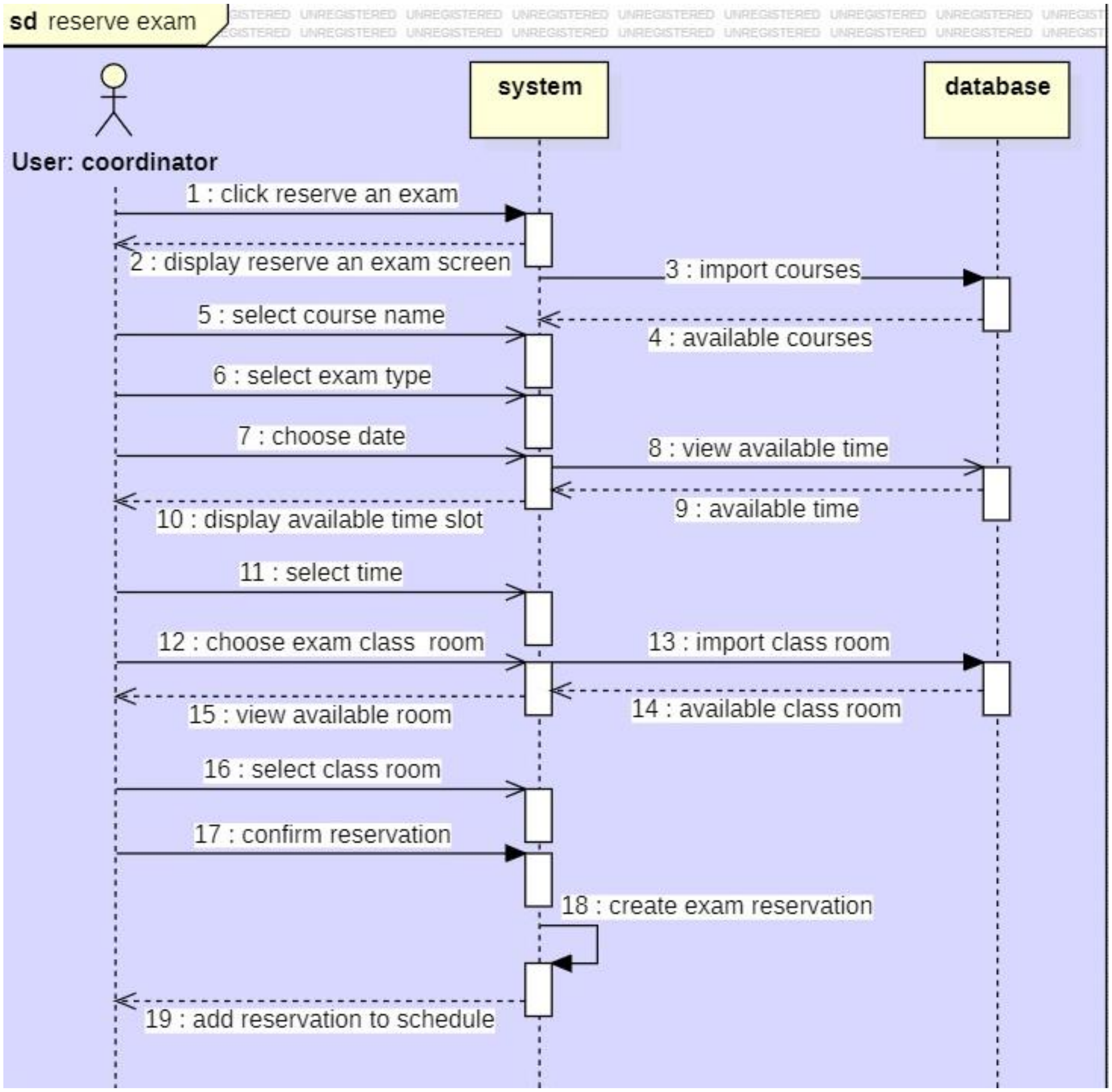


Figure [13] :Reserve Exam Sequence diagram

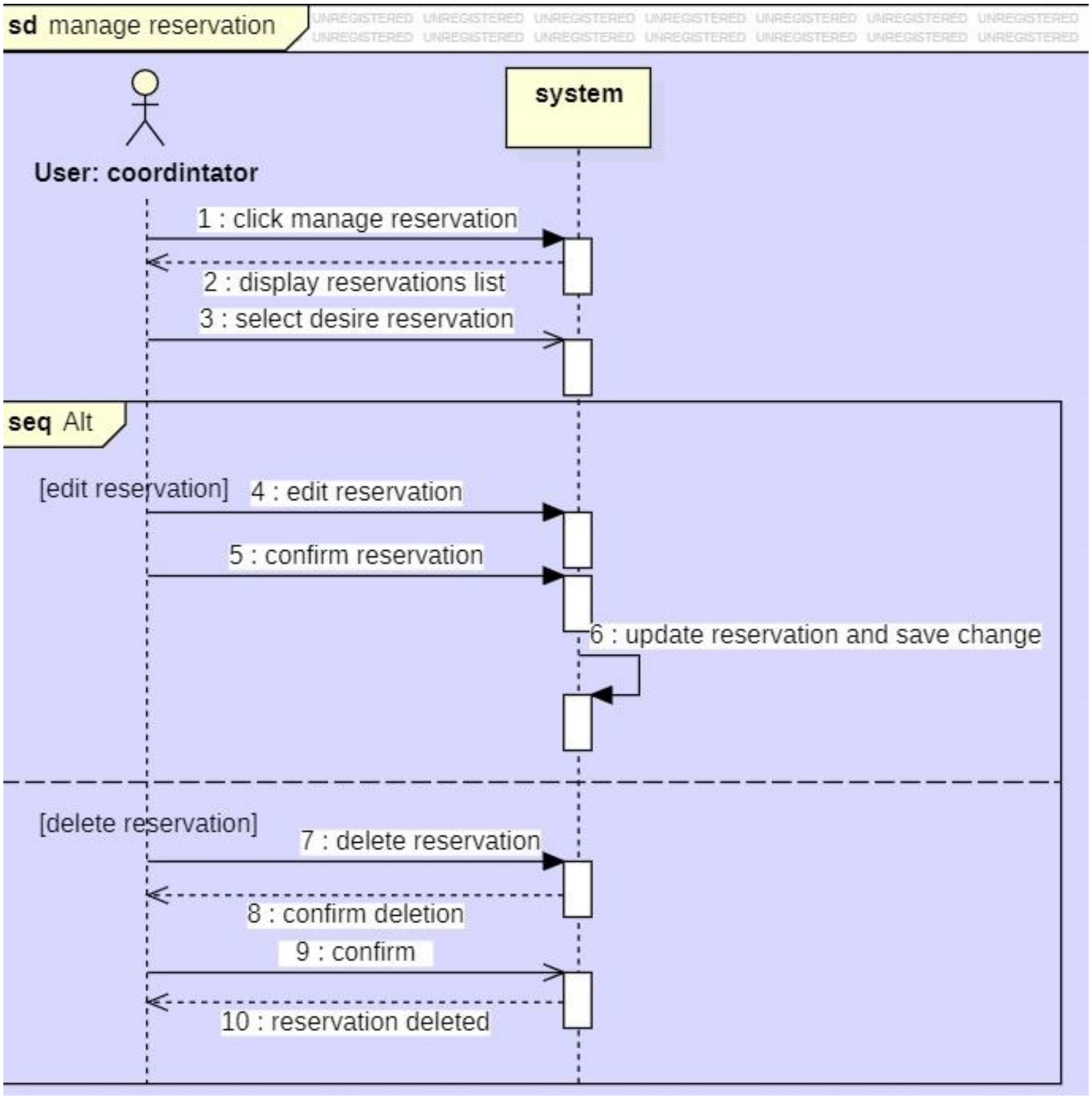

sd manage reservation
UNREGISTERED UNREGISTERED UNREGISTERED UNREGISTERED UNREGISTERED UNREGISTERED UNREGISTERED
UNREGISTERED UNREGISTERED UNREGISTERED UNREGISTERED UNREGISTERED UNREGISTERED UNREGISTERED
User: coordintator
system
1 : click manage reservation
2 : display reservations list
3 : select desire reservation
seq Alt
[edit reservation]
4 : edit reservation
5 : confirm reservation
6 : update reservation and save change
[delete reservation]
7 : delete reservation
8 : confirm deletion
9 : confirm
10 : reservation deleted

Figure [14]: Manage Reservation Sequence diagram

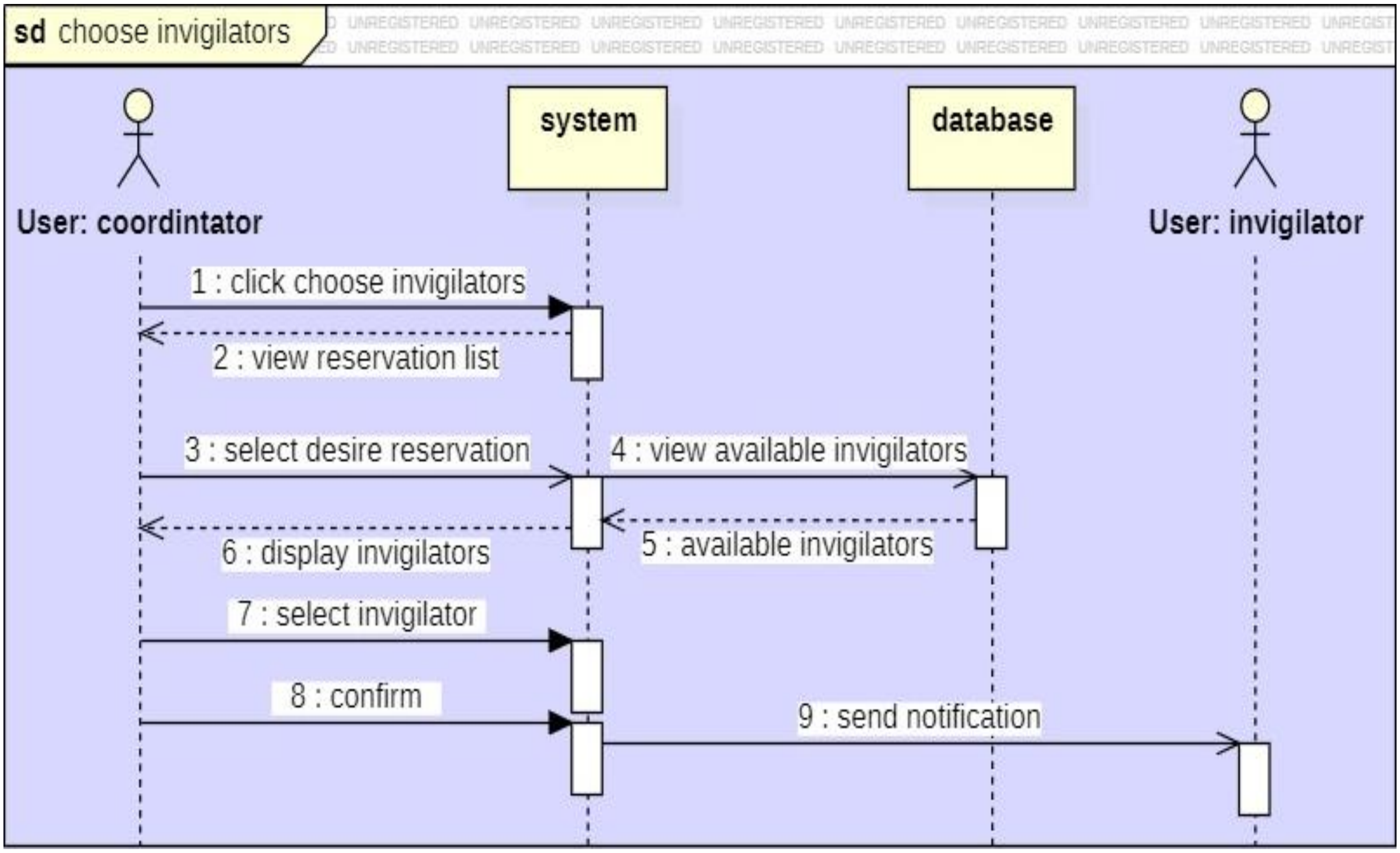


Figure [15]: Choose invigilators Sequence diagram

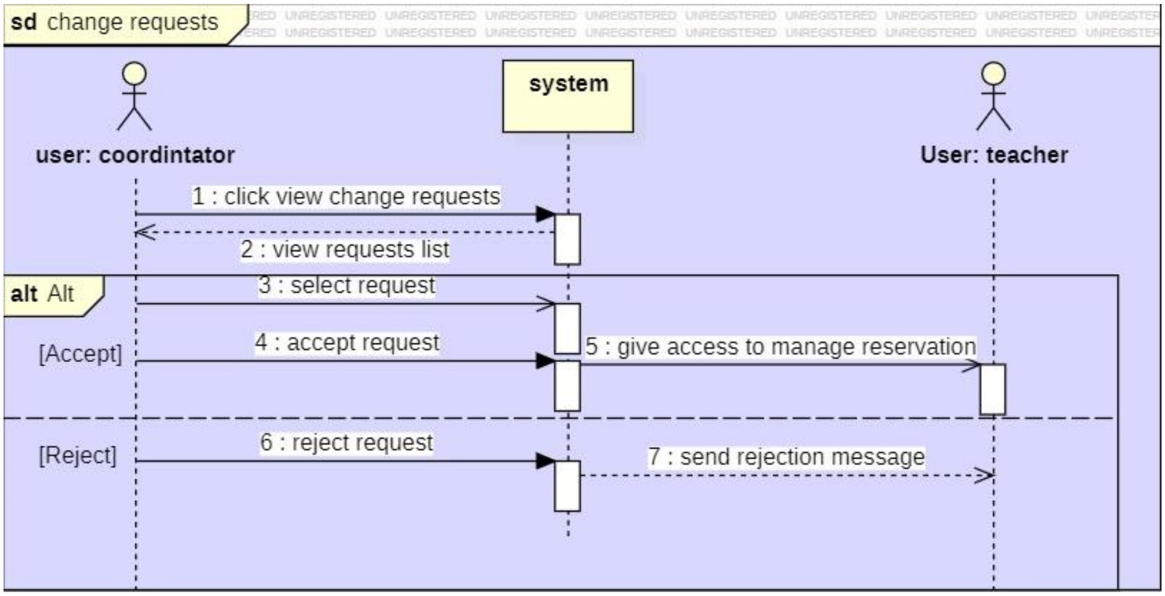


Figure [16]:View Change Requests Sequence diagram

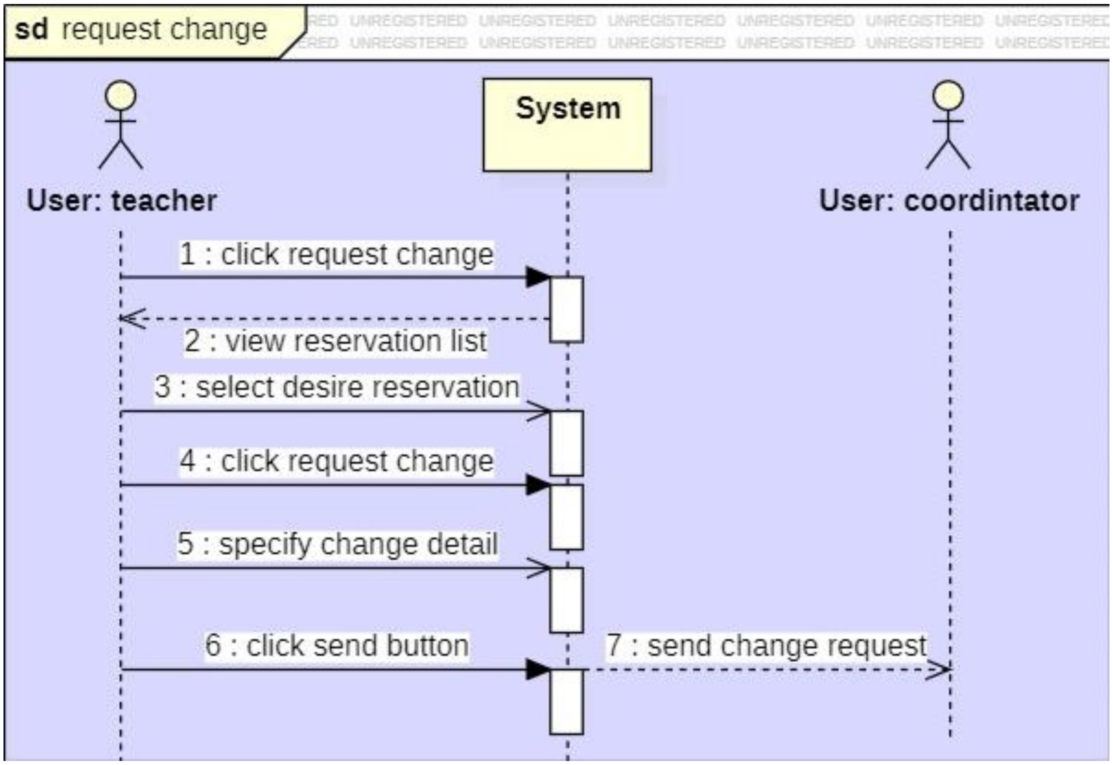

sd request change
UNREGISTERED UNREGISTERED UNREGISTERED UNREGISTERED UNREGISTERED UNREGISTERED
System
User: teacher
User: coordintator
1 : click request change
2 : view reservation list
3 : select desire reservation
4 : click request change
5 : specify change detail
6 : click send button
7 : send change request

Figure [17]: Request Change Sequence diagram

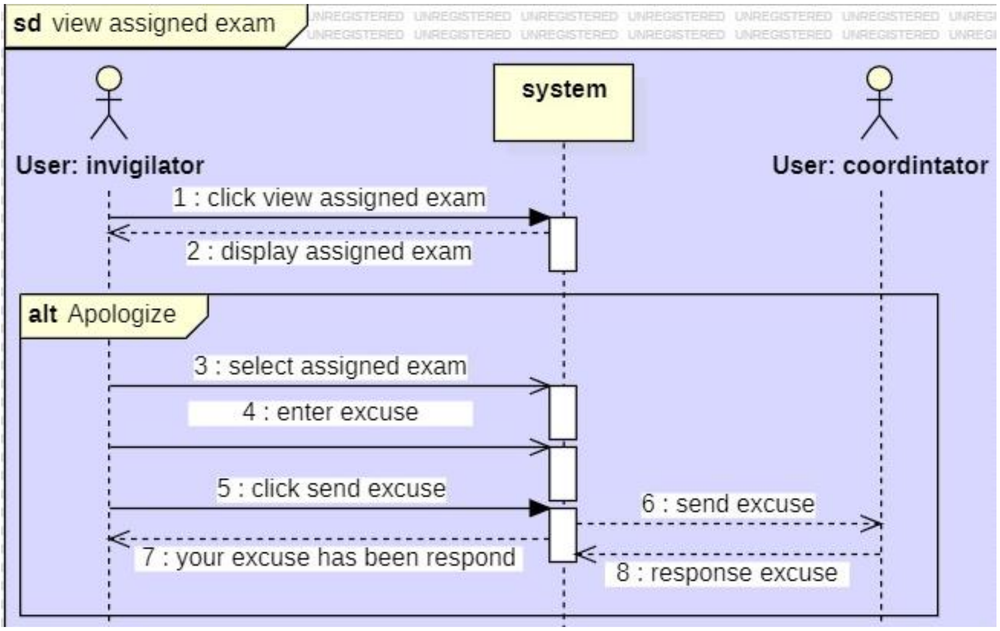


Figure [18]: View Assigned Exam Sequence diagram

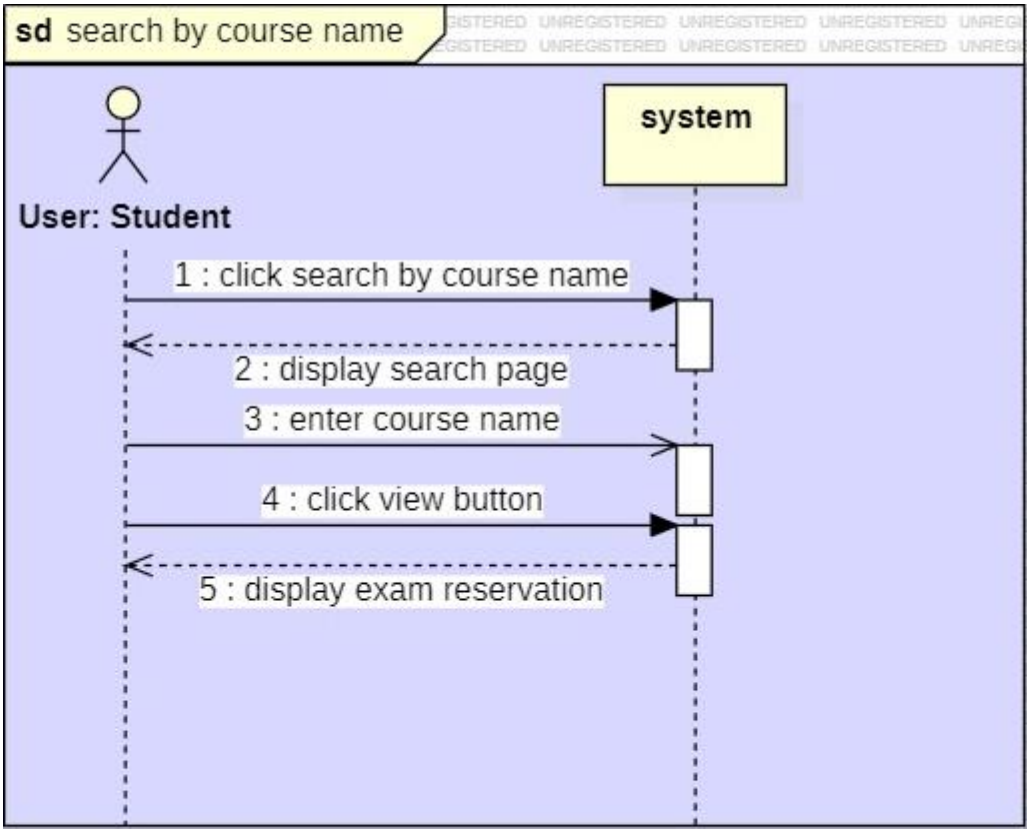

sd search by course name
UNREGISTERED UNREGISTERED UNREGISTERED UNREGISTERED UNREGISTERED
User: Student
system
1 : click search by course name
2 : display search page
3 : enter course name
4 : click view button
5 : display exam reservation

Figure [19]: Search By Course Name Sequence diagram

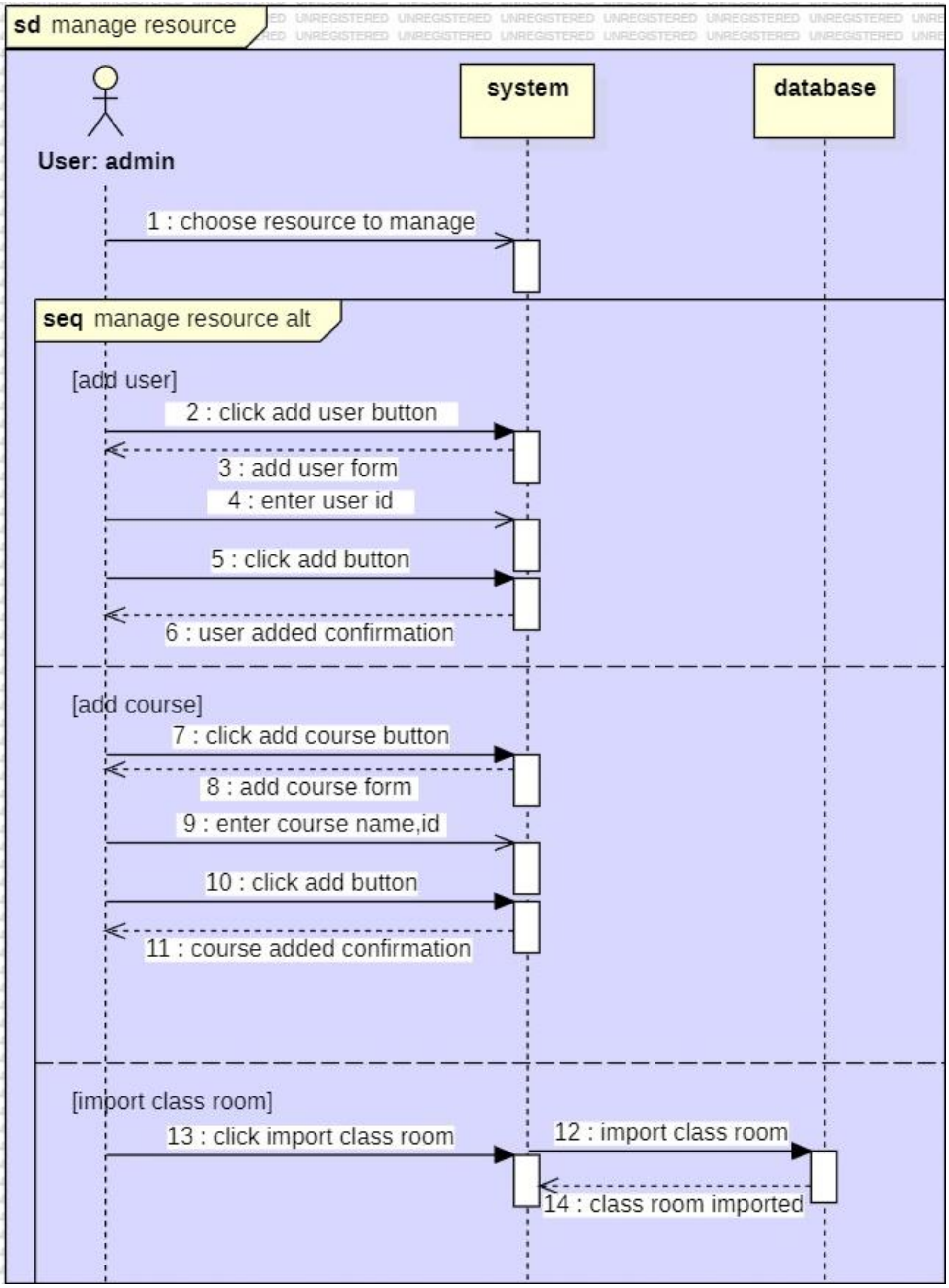


Figure [20]: Manage Resource Sequence diagram

4.3.2 Class Diagram

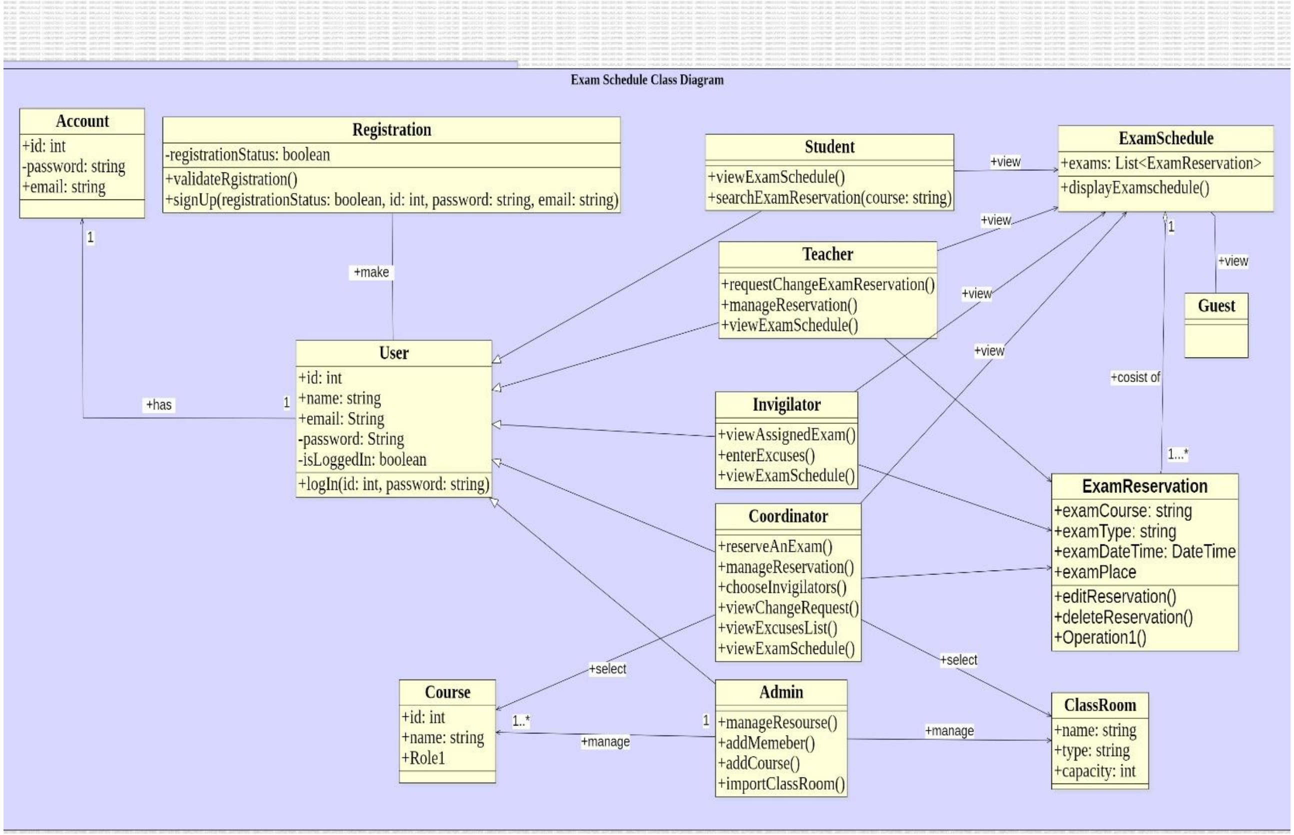


Figure [21]: Class Diagram

4.3.3 ER Diagram

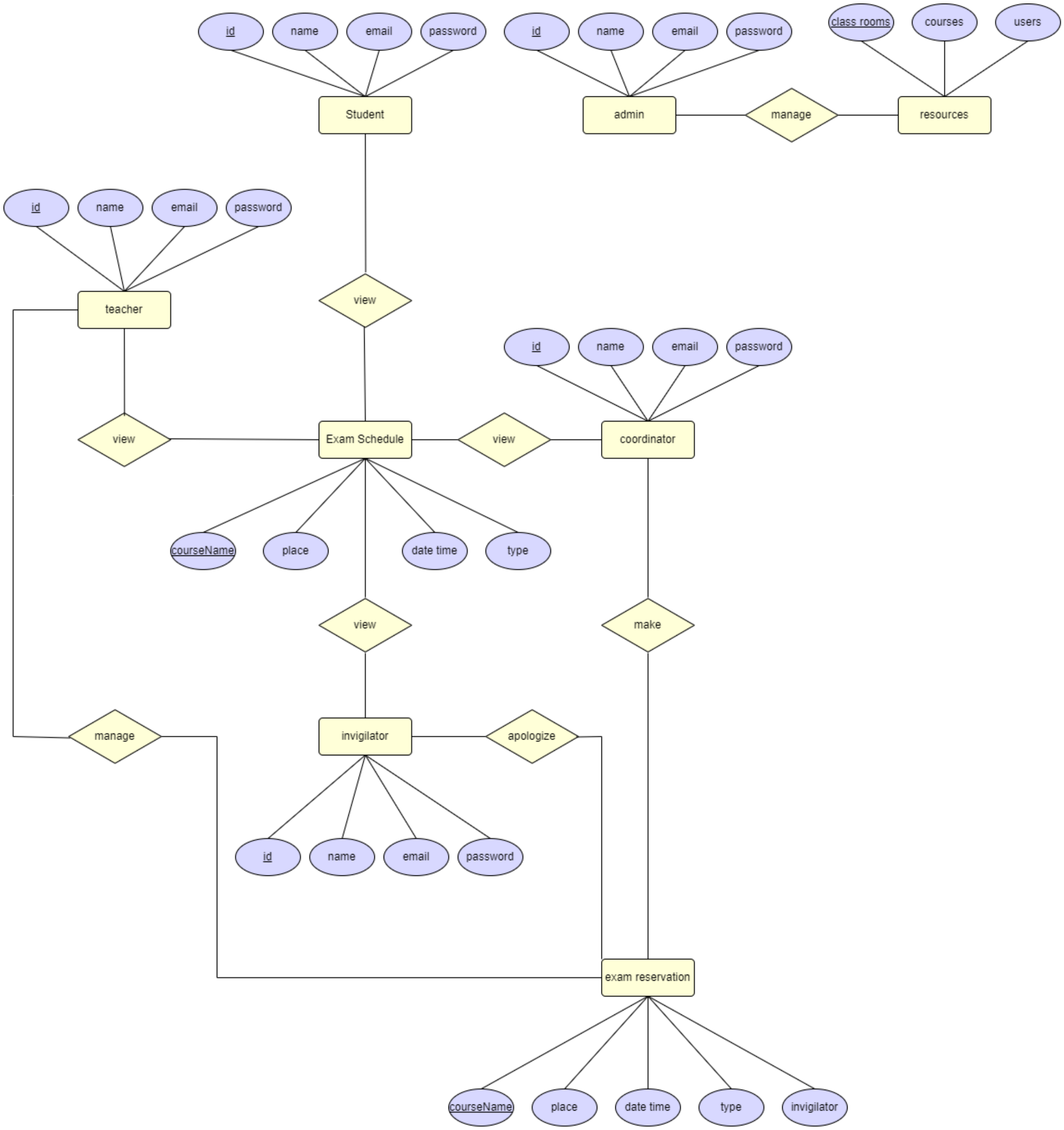


Figure [22]: ER Diagram

Chapter 5: Implementation Plan

In this chapter, the project implementation plan is described, describing the main tasks and components needed for implementation from hardware, software and resources, defining the programming languages and techniques required to support and facilitate the implementation process.

5.1 Description of Implementation

This website is divided into students, coordinator, teacher, admin, invigilator, and guest, each one has a certain role.

1- **Create an account and enter the website:**

First, an account is created on the website for the student, coordinator, teacher, admin, invigilator, to use the website, the user chooses to log in as: student, faculty, or guest. then using the ID number, and password to try entering the website, in all cases the information will be checked for correctness and stored in a database. If the ID is already there and has a compatible password, the user can enter the website and use it according to its role. Otherwise, it will return an error message with its content. As for the guest, he can show the exam schedule without logging in process.

2- **Using the application:**

the user can use the available functionality according to the selected role , the student can view the exam schedule, the administrator can manage the resources , the teacher can view the exam schedule , manage reservation and request change, the coordinator can view the exam schedule, reserve an exam ,view request changes , view excuses list ,choose invigilators and manage reservation , the invigilators can view exam schedule , enter excuses and view assigned exams and apologize.

3- **Logout**:

The user can log out of the website when they are finished using it

5.2 Programming language and technology

ASP.NET Core is an open-source modular web-application framework. Its predecessor .NET Core applications can run on Windows, macOS, and Linux operating systems. This makes it a versatile choice for developers targeting multiple platforms. It is a redesign of ASP.NET .

1.Net core MVC

.NET Core MVC is a model-view-controller (MVC) framework for building web applications. MVC is a design pattern that separates the application's concerns into three distinct parts:

Model: Represents the data of the application.

View: Presents the data to the user.

Controller: Handles user interactions and updates the model and view.

2. C# Language

C# is a programming language developed by Microsoft. It is a general-purpose language that can be used for a variety of tasks, including web development. In this project, C# will be used to write the code for the backend, which is the server-side part of the application.

3. SQL Server

It is a relational database management system (RDBMS) developed by Microsoft. It's one of the most popular and widely used database systems globally. Businesses of all sizes use SQL Server to store, manage, and analyze their data.

The project will be using HTML, CSS, JavaScript, and Bootstrap for the front end, which is the client-side part of the application.

1. **HTML:**

   Hyper Text Markup Language (HTML) is the standard markup language for creating web pages. It defines the structure and content of a web page.

2. **CSS:**

   Cascading Style Sheets (CSS) is a language used to style web pages. It controls the appearance of HTML elements, such as font size, color, and layout.

3. **JavaScript:**

   JavaScript is a scripting language that adds interactivity to web pages. It allows you to create dynamic and responsive elements that react to user input.

4. **Bootstrap:**

   is a popular CSS framework that provides pre-built components for building web pages. It can help you create responsive and consistent designs quickly and easily.

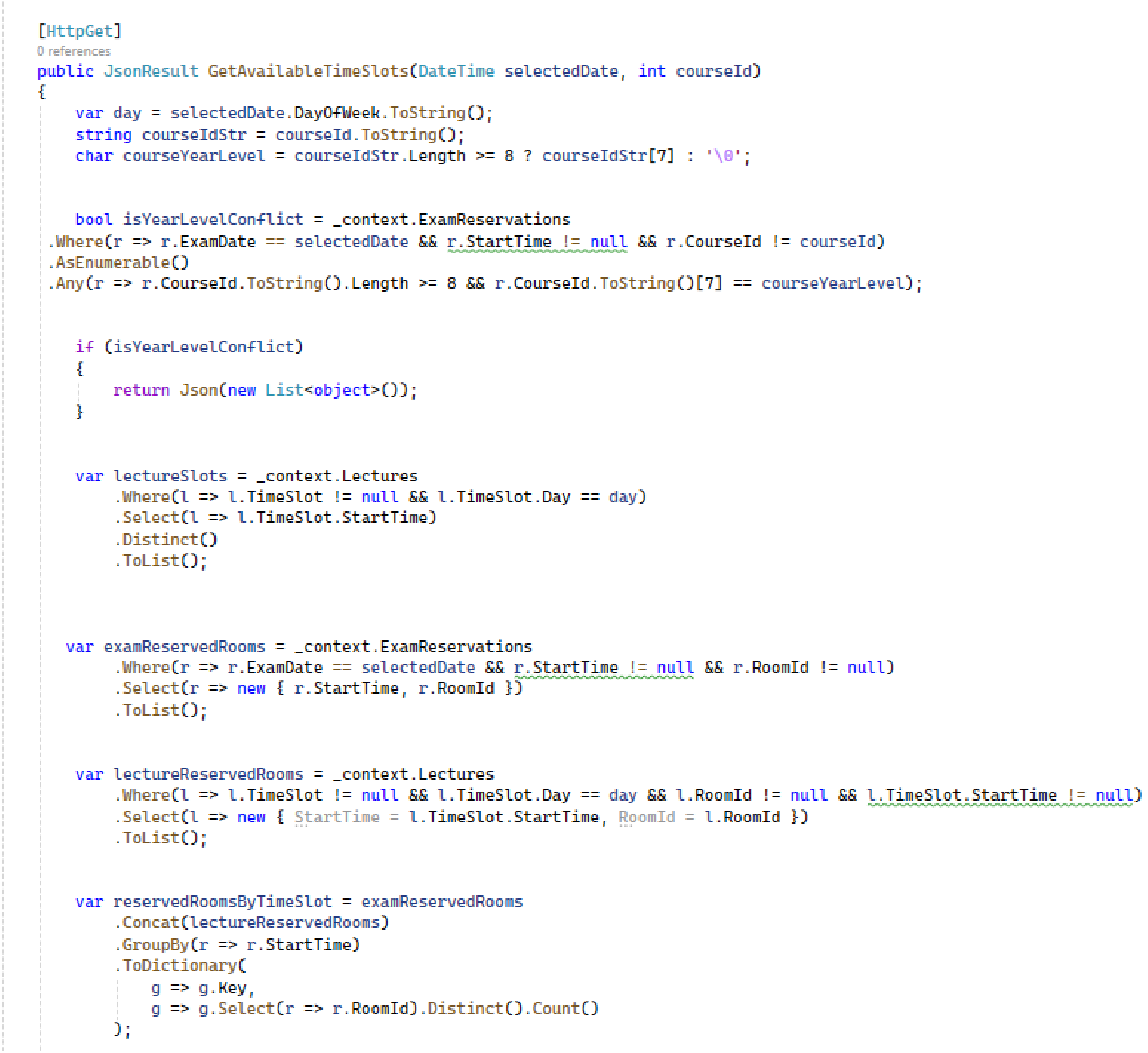
5. **AJAX**

   Asynchronous JavaScript and XML (AJAX) AJAX is not a programming language, it is a combination of web application development technologies that make web applications more responsive to user interaction**.**

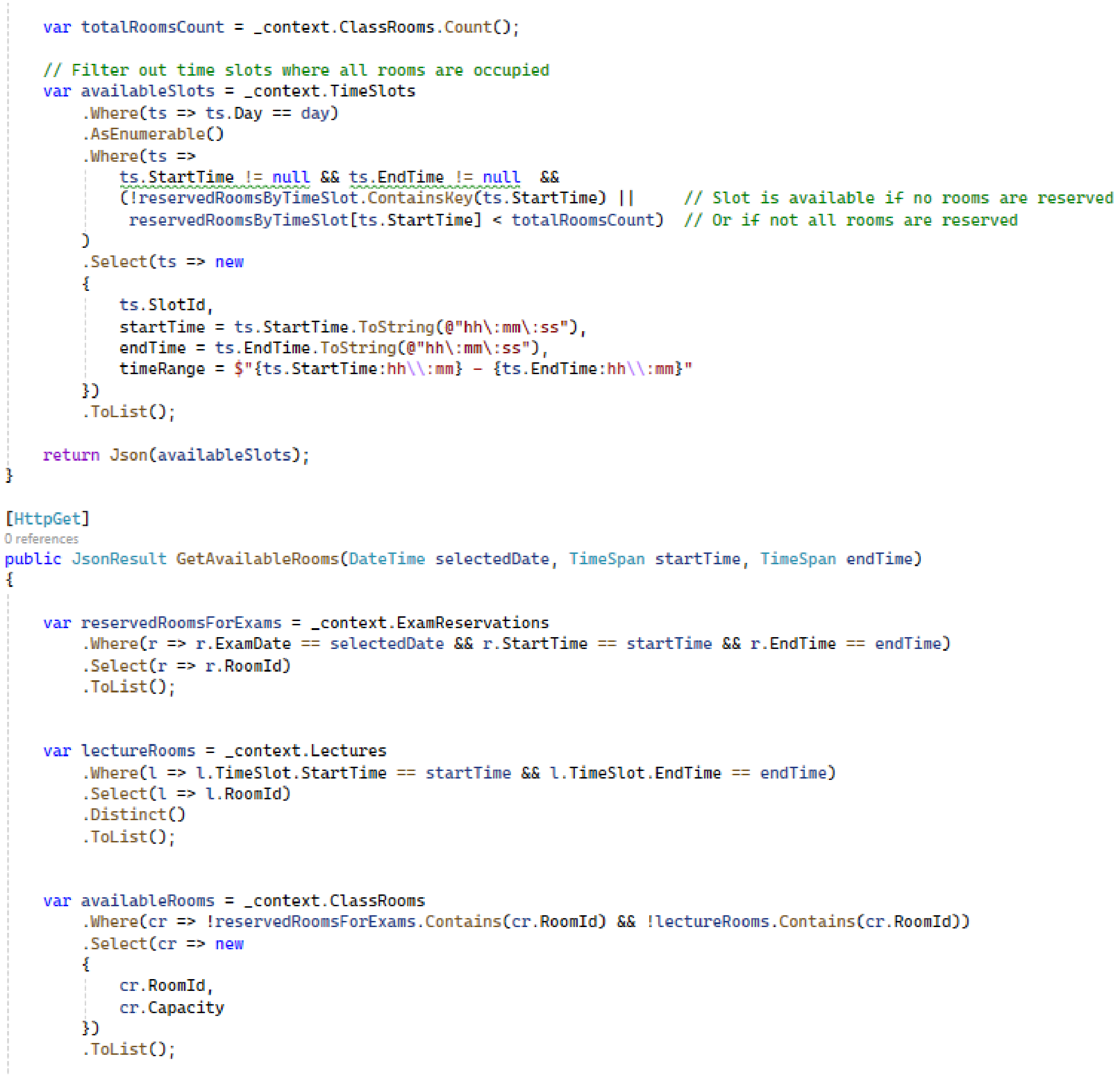
6. **JSON**

   JavaScript Object Notation (JSON) is a standard text-based format for representing structured data based on JavaScript object syntax. It is commonly used for transmitting data in web applications

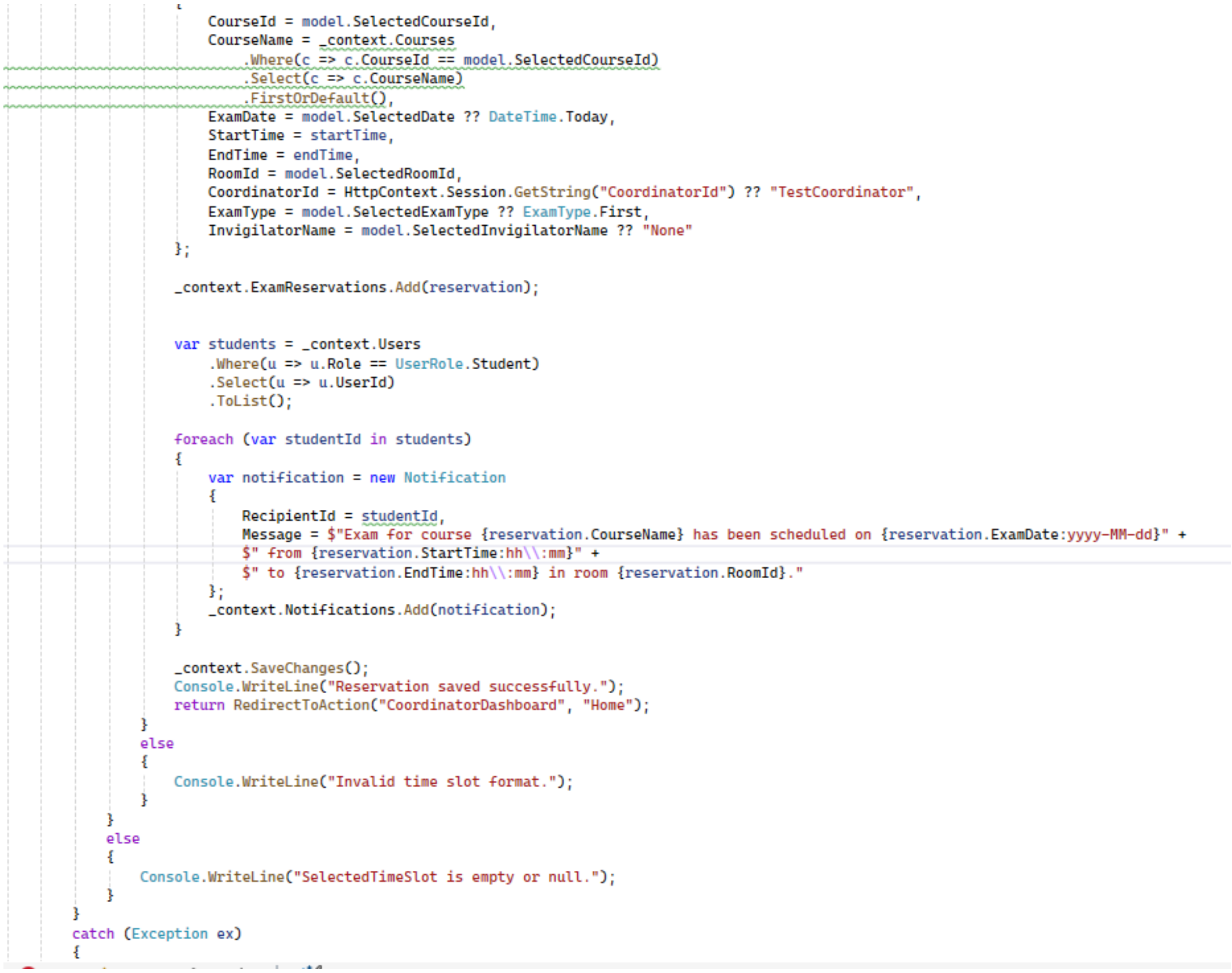
5.3 Implementation



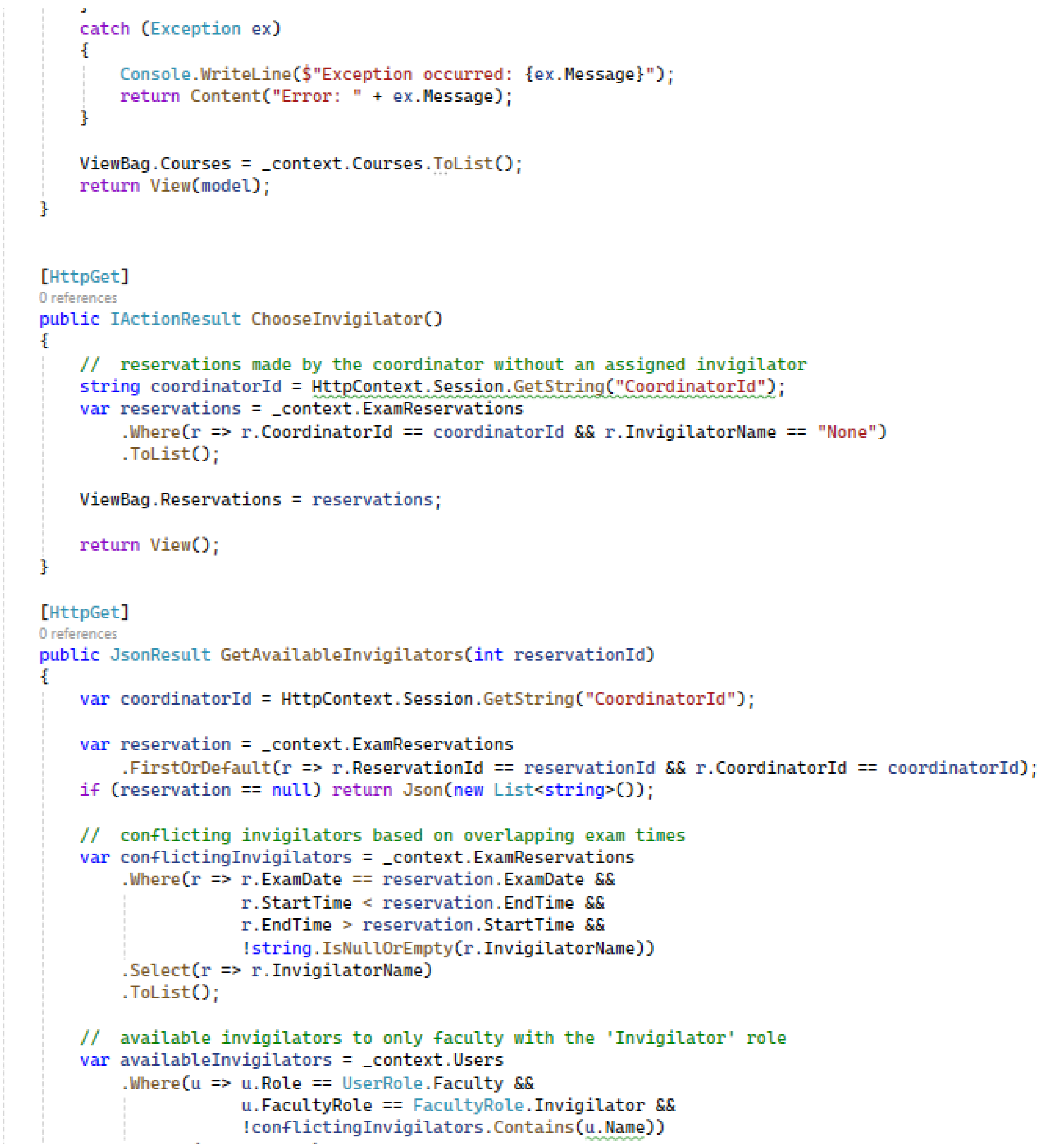
```
[HttpGet]
0 references
public JsonResult GetAvailableTimeSlots(DateTime selectedDate, int courseId)
{
    var day = selectedDate.DayOfWeek.ToString();
    string courseIdStr = courseId.ToString();
    char courseYearLevel = courseIdStr.Length >= 8 ? courseIdStr[7] : '\0';


    bool isYearLevelConflict = _context.ExamReservations
 .Where(r => r.ExamDate == selectedDate && r.StartTime != null && r.CourseId != courseId)
 .AsEnumerable()
 .Any(r => r.CourseId.ToString().Length >= 8 && r.CourseId.ToString()[7] == courseYearLevel);


    if (isYearLevelConflict)
    {
        return Json(new List<object>());
    }


    var lectureSlots = _context.Lectures
        .Where(l => l.TimeSlot != null && l.TimeSlot.Day == day)
        .Select(l => l.TimeSlot.StartTime)
        .Distinct()
        .ToList();



   var examReservedRooms = _context.ExamReservations
        .Where(r => r.ExamDate == selectedDate && r.StartTime != null && r.RoomId != null)
        .Select(r => new { r.StartTime, r.RoomId })
        .ToList();


    var lectureReservedRooms = _context.Lectures
        .Where(l => l.TimeSlot != null && l.TimeSlot.Day == day && l.RoomId != null && l.TimeSlot.StartTime != null)
        .Select(l => new { StartTime = l.TimeSlot.StartTime, RoomId = l.RoomId })
        .ToList();


    var reservedRoomsByTimeSlot = examReservedRooms
        .Concat(lectureReservedRooms)
        .GroupBy(r => r.StartTime)
        .ToDictionary(
            g => g.Key,
            g => g.Select(r => r.RoomId).Distinct().Count()
        );
```



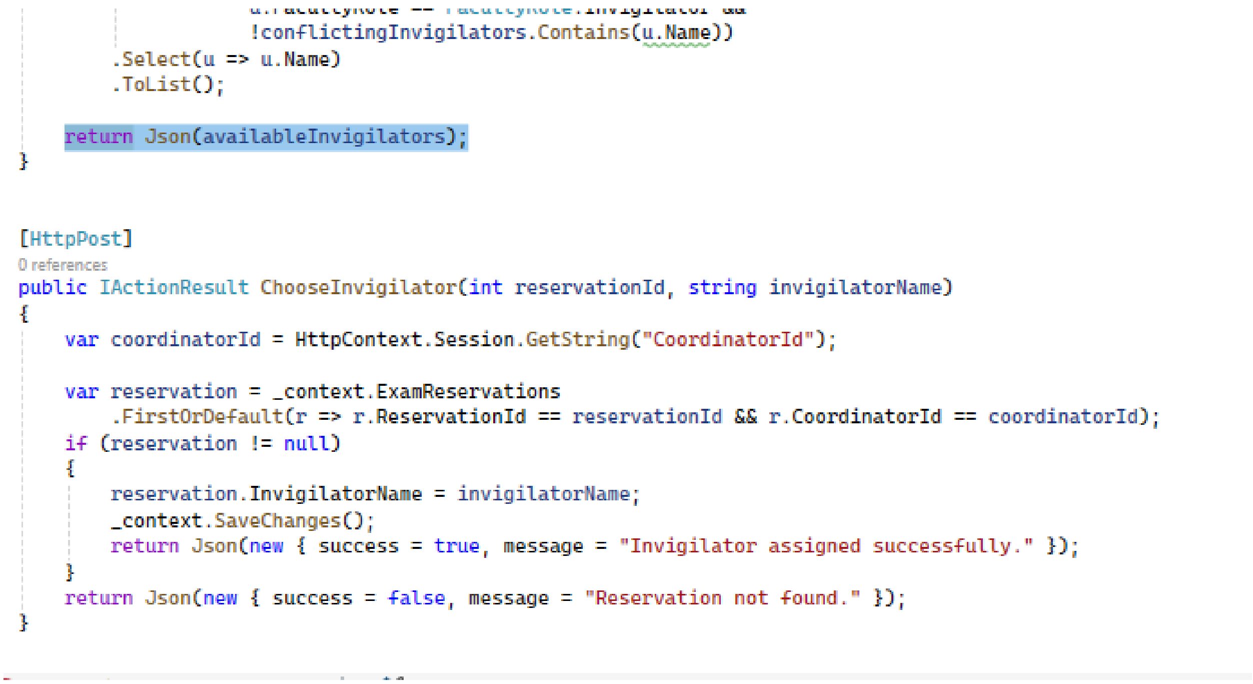
```
    var totalRoomsCount = _context.ClassRooms.Count();

    // Filter out time slots where all rooms are occupied
    var availableSlots = _context.TimeSlots
        .Where(ts => ts.Day == day)
        .AsEnumerable()
        .Where(ts =>
            ts.StartTime != null && ts.EndTime != null  &&
            (!reservedRoomsByTimeSlot.ContainsKey(ts.StartTime) ||     // Slot is available if no rooms are reserved
             reservedRoomsByTimeSlot[ts.StartTime] < totalRoomsCount)  // Or if not all rooms are reserved
        )
        .Select(ts => new
        {
            ts.SlotId,
            startTime = ts.StartTime.ToString(@"hh\:mm\:ss"),
            endTime = ts.EndTime.ToString(@"hh\:mm\:ss"),
            timeRange = $"{ts.StartTime:hh\\:mm} - {ts.EndTime:hh\\:mm}"
        })
        .ToList();

    return Json(availableSlots);
}

[HttpGet]
0 references
public JsonResult GetAvailableRooms(DateTime selectedDate, TimeSpan startTime, TimeSpan endTime)
{

    var reservedRoomsForExams = _context.ExamReservations
        .Where(r => r.ExamDate == selectedDate && r.StartTime == startTime && r.EndTime == endTime)
        .Select(r => r.RoomId)
        .ToList();


    var lectureRooms = _context.Lectures
        .Where(l => l.TimeSlot.StartTime == startTime && l.TimeSlot.EndTime == endTime)
        .Select(l => l.RoomId)
        .Distinct()
        .ToList();


    var availableRooms = _context.ClassRooms
        .Where(cr => !reservedRoomsForExams.Contains(cr.RoomId) && !lectureRooms.Contains(cr.RoomId))
        .Select(cr => new
        {
            cr.RoomId,
            cr.Capacity
        })
        .ToList();
```



```
            {
                CourseId = model.SelectedCourseId,
                CourseName = _context.Courses
                    .Where(c => c.CourseId == model.SelectedCourseId)
                    .Select(c => c.CourseName)
                    .FirstOrDefault(),
                ExamDate = model.SelectedDate ?? DateTime.Today,
                StartTime = startTime,
                EndTime = endTime,
                RoomId = model.SelectedRoomId,
                CoordinatorId = HttpContext.Session.GetString("CoordinatorId") ?? "TestCoordinator",
                ExamType = model.SelectedExamType ?? ExamType.First,
                InvigilatorName = model.SelectedInvigilatorName ?? "None"
            };

            _context.ExamReservations.Add(reservation);


            var students = _context.Users
                .Where(u => u.Role == UserRole.Student)
                .Select(u => u.UserId)
                .ToList();

            foreach (var studentId in students)
            {
                var notification = new Notification
                {
                    RecipientId = studentId,
                    Message = $"Exam for course {reservation.CourseName} has been scheduled on {reservation.ExamDate:yyyy-MM-dd}" +
                    $" from {reservation.StartTime:hh\\:mm}" +
                    $" to {reservation.EndTime:hh\\:mm} in room {reservation.RoomId}."
                };
                _context.Notifications.Add(notification);
            }

            _context.SaveChanges();
            Console.WriteLine("Reservation saved successfully.");
            return RedirectToAction("CoordinatorDashboard", "Home");
        }
        else
        {
            Console.WriteLine("Invalid time slot format.");
        }
    }
    else
    {
        Console.WriteLine("SelectedTimeSlot is empty or null.");
    }
}
catch (Exception ex)
{
```



```
    catch (Exception ex)
    {
        Console.WriteLine($"Exception occurred: {ex.Message}");
        return Content("Error: " + ex.Message);
    }

    ViewBag.Courses = _context.Courses.ToList();
    return View(model);
}


[HttpGet]
0 references
public IActionResult ChooseInvigilator()
{
    //  reservations made by the coordinator without an assigned invigilator
    string coordinatorId = HttpContext.Session.GetString("CoordinatorId");
    var reservations = _context.ExamReservations
        .Where(r => r.CoordinatorId == coordinatorId && r.InvigilatorName == "None")
        .ToList();

    ViewBag.Reservations = reservations;

    return View();
}

[HttpGet]
0 references
public JsonResult GetAvailableInvigilators(int reservationId)
{
    var coordinatorId = HttpContext.Session.GetString("CoordinatorId");

    var reservation = _context.ExamReservations
        .FirstOrDefault(r => r.ReservationId == reservationId && r.CoordinatorId == coordinatorId);
    if (reservation == null) return Json(new List<string>());

    //  conflicting invigilators based on overlapping exam times
    var conflictingInvigilators = _context.ExamReservations
        .Where(r => r.ExamDate == reservation.ExamDate &&
                    r.StartTime < reservation.EndTime &&
                    r.EndTime > reservation.StartTime &&
                    !string.IsNullOrEmpty(r.InvigilatorName))
        .Select(r => r.InvigilatorName)
        .ToList();

    //  available invigilators to only faculty with the 'Invigilator' role
    var availableInvigilators = _context.Users
        .Where(u => u.Role == UserRole.Faculty &&
                    u.FacultyRole == FacultyRole.Invigilator &&
                    !conflictingInvigilators.Contains(u.Name))
```



```
                    u.FacultyRole == FacultyRole.Invigilator &&
                    !conflictingInvigilators.Contains(u.Name))
            .Select(u => u.Name)
            .ToList();

        return Json(availableInvigilators);
    }


    [HttpPost]
    0 references
    public IActionResult ChooseInvigilator(int reservationId, string invigilatorName)
    {
        var coordinatorId = HttpContext.Session.GetString("CoordinatorId");

        var reservation = _context.ExamReservations
            .FirstOrDefault(r => r.ReservationId == reservationId && r.CoordinatorId == coordinatorId);
        if (reservation != null)
        {
            reservation.InvigilatorName = invigilatorName;
            _context.SaveChanges();
            return Json(new { success = true, message = "Invigilator assigned successfully." });
        }
        return Json(new { success = false, message = "Reservation not found." });
    }
```

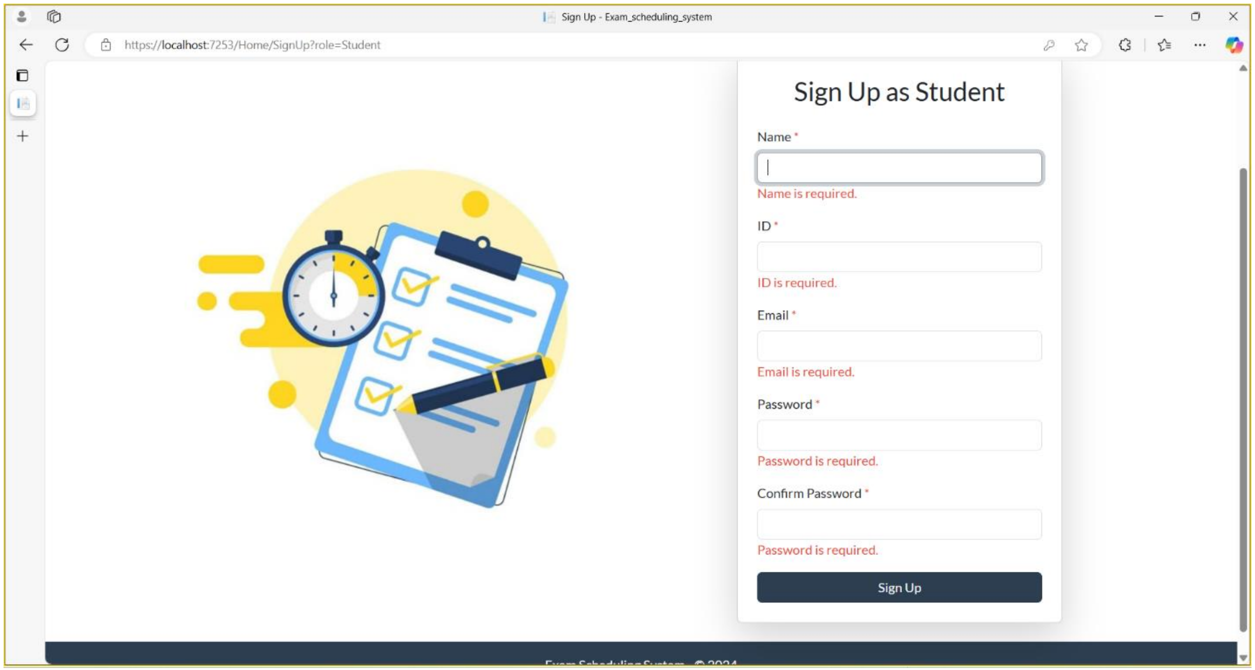
CHAPTER 6: Testing Plan

6.1 BlackBox Testing

Table 14 : Black box testing

Test case ID	Test case description	Test data	Expected result	Actual result	status
6.1.1	Fill wrong email format	********@gmail.com	Error Message:” invalid email format must be like *******@std.hu.edu.jo”.	As expected	pass
6.1.2	Sign up with empty data	Empty elements	Error message: “this field required.”.	As expected	pass
6.1.3	Sign up with an invalid ID number	ID: 21443800	Error message:” student ID must be exactly 7 characters”.	As expected	pass
6.1.4	Sign up with passwords less than 8 characters.	12345	Error Message:” password must be at least 8 characters long”.		
6.1.5	Not compatible password	2020443 2020445	Error message:” passwords don’t match”.	As expected	pass
6.1.6	Sign up with existing email	2138744@std.hu.edu.jo	Error message:” a user with this email already exist”.	As expected	pass
6.1.7	Log in with invalid ID for role	12120	Error message:” incorrect role for the provided ID”.	As expected	pass
6.1.8	Log in with invalid ID or password	55555	Error message:” invalid ID or password”.	As expected	pass
6.1.9	Sign up with existing ID	2138744	Error message:” a user with this ID already exist”.	As expected	pass
6.1.10	Enter letters in ID field	2144f87	Error message:” invalid ID or password”.	As expected	pass
6.1.11	Enter email in ID field	2144380@std.hu.edu.jo	Error message:” invalid ID or password”.	As expected	pass
6.1.12	Log in with	Empty ID	Error message:” this field	As	pass

	empty ID or password		is required”.	expected	
6.1.13	Wrong course name	C++	Error massage: ” No exams found for the specified course name”.	As expected	pass
6.1.14	Add course name already exist	Software Testing	Error message: “this course name already exists”	As expected	pass
6.1.15	Adding course ID already exist	2010031221	Error message: “this course ID already exist”	As expected	pass
6.1.16	Invalid course ID	CCCCCC 20103355R	Error message:” the input value is not valid for course ID”.	As expected	pass
6.1.17	Invalid course name	144506	Error message:” course name must contain only letter”.	As expected	pass
6.1.18	Enter empty field in course name	Empty element	Error message:” The course name field is required”.	As expected	pass
6.1.19	Empty field at course department	Empty field	Error message:” The course department field is required”.	As expected	pass
6.1.20	Empty data in room ID	Empty data	Error message: “the room id field is required”.	As expected	pass
6.1.21	Empty data for capacity	Empty data	Error message” the capacity field is required”.	As expected	pass
6.1.22	Add classroom already exist	Room 303	Error message:” the classroom already exists”.	As expected	pass
6.1.23	Add member with invalid id	111234	Error message:”Faculty ID must be exactly 5 characters.”	As expected	pass


Sign Up - Exam_scheduling_system
https://localhost:7253/Home/SignUp?role=Student
Sign Up
Exam Scheduling System - © 2024

Sign Up as Student

Name *

Name is required.

ID *

ID is required.

Email *

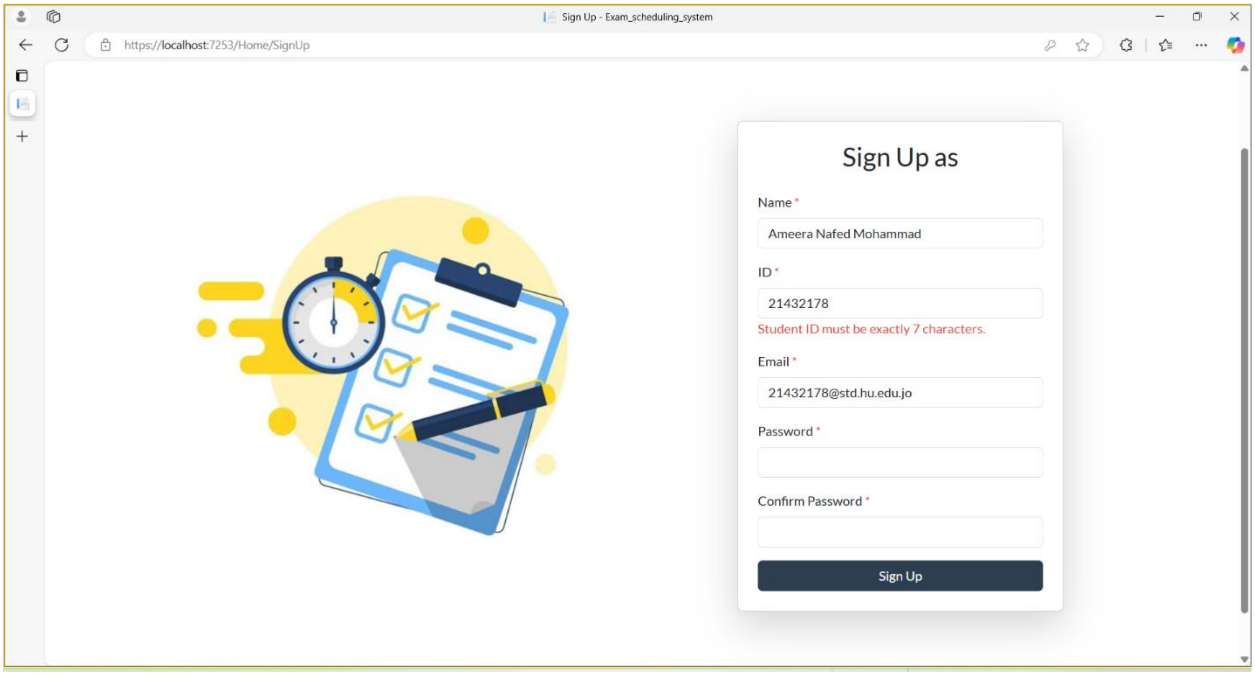
Email is required.

Password *

Password is required.

Confirm Password *

Password is required.


Sign Up - Exam_scheduling_system
https://localhost:7253/Home/SignUp
Sign Up

Sign Up as

Name *

Ameera Nafed Mohammad

ID *

21432178

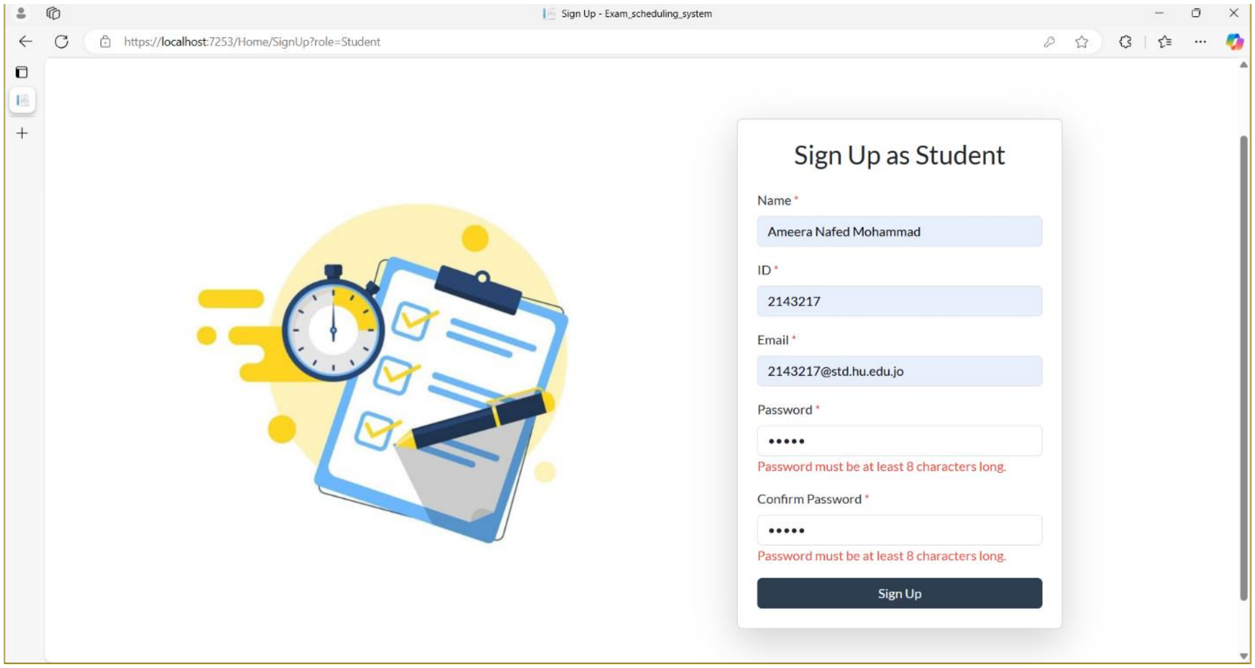
Student ID must be exactly 7 characters.

Email *

21432178@std.hu.edu.jo

Password *

Confirm Password *


Sign Up - Exam_scheduling_system
https://localhost:7253/Home/SignUp?role=Student

Sign Up as Student

Name *

Ameera Nafed Mohammad

ID *

2143217

Email *

2143217@std.hu.edu.jo

Password *

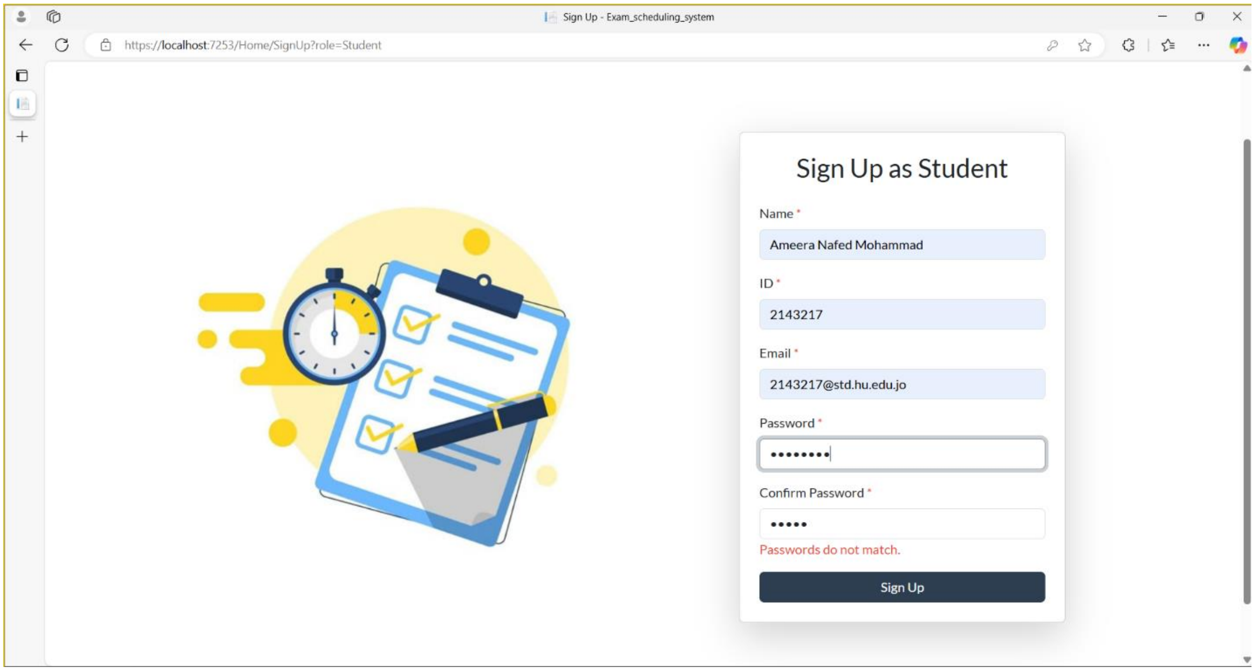
•••••

Password must be at least 8 characters long.

Confirm Password *

•••••

Password must be at least 8 characters long.


Sign Up - Exam_scheduling_system
https://localhost:7253/Home/SignUp?role=Student

Sign Up as Student

Name *

Ameera Nafed Mohammad

ID *

2143217

Email *

2143217@std.hu.edu.jo

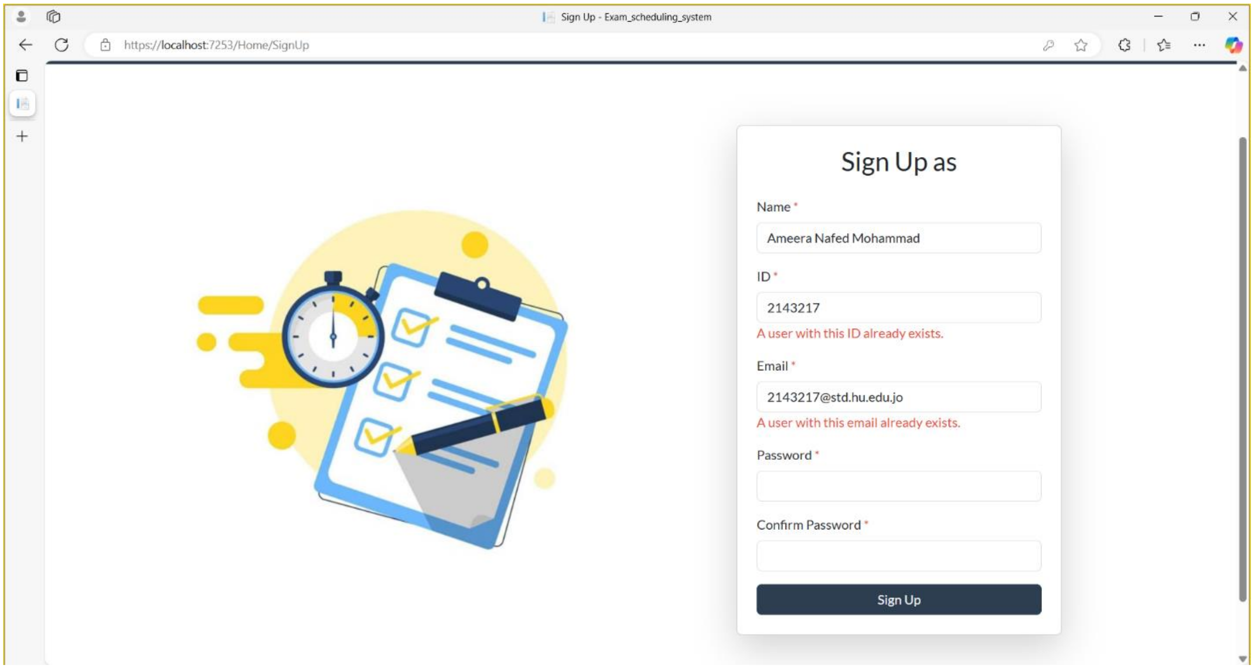
Password *

••••••••

Confirm Password *

•••••

Passwords do not match.


Sign Up - Exam_scheduling_system
https://localhost:7253/Home/SignUp

Sign Up as

Name *

Ameera Nafed Mohammad

ID *

2143217

A user with this ID already exists.

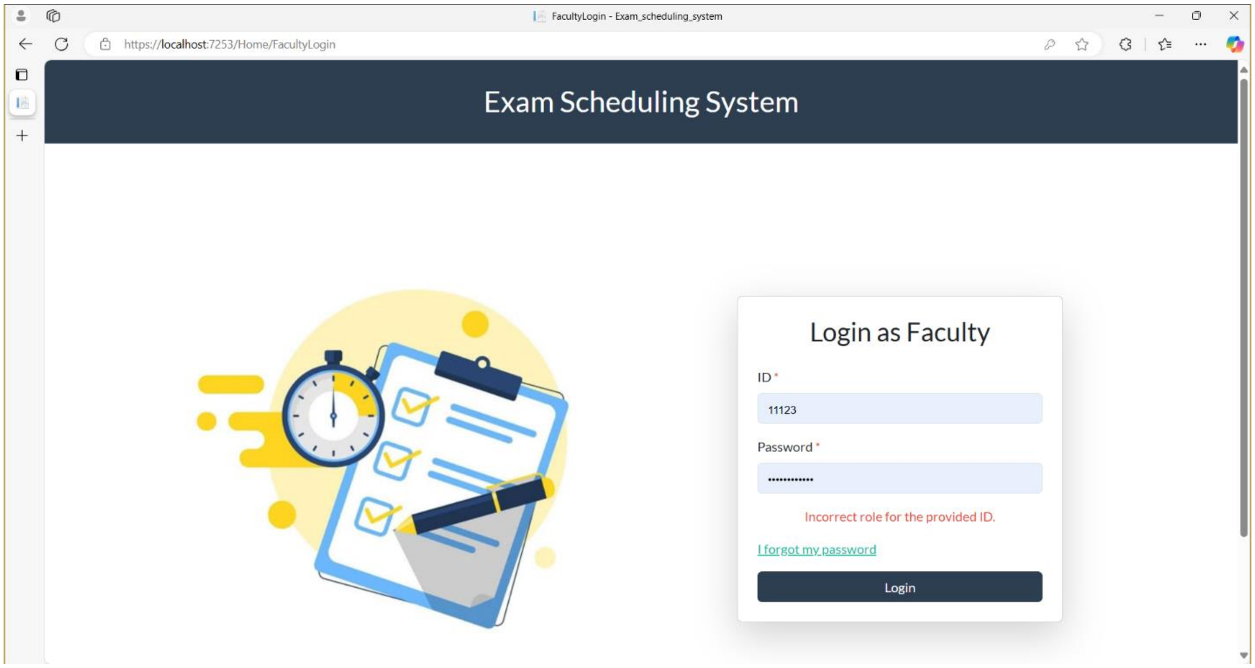
Email *

2143217@std.hu.edu.jo

A user with this email already exists.

Password *

Confirm Password *


FacultyLogin - Exam_scheduling_system
https://localhost:7253/Home/FacultyLogin

Exam Scheduling System

Login as Faculty

ID *

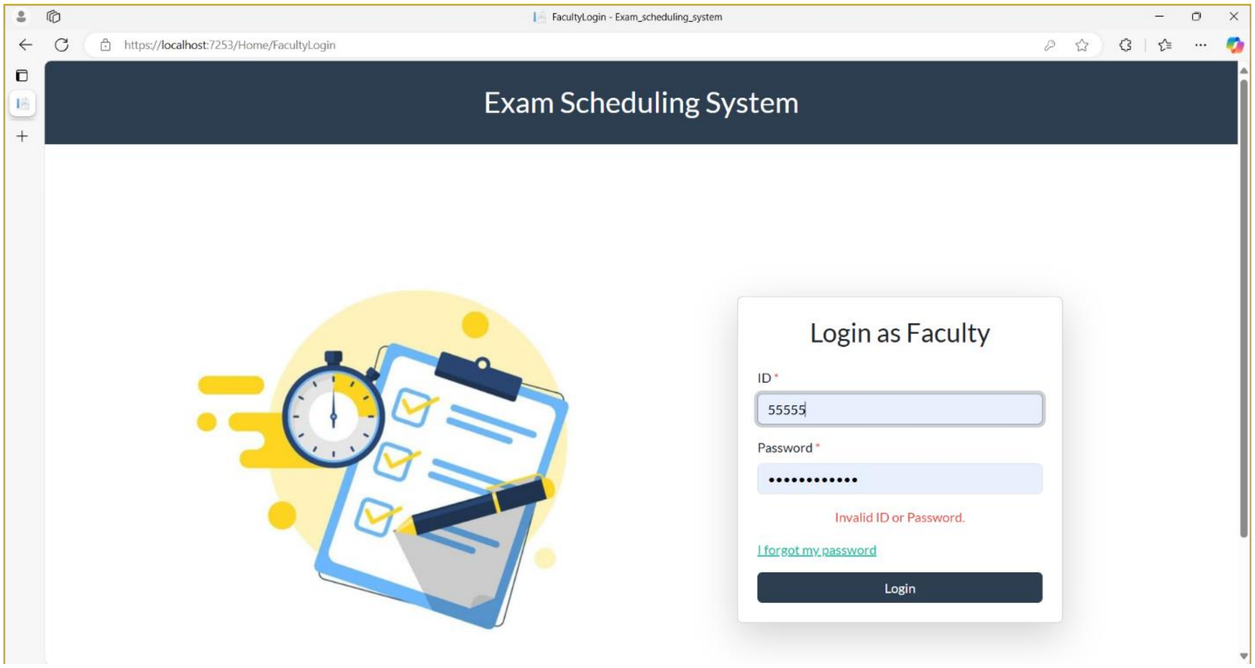
11123

Password *

•••••••••••

Incorrect role for the provided ID.

I forgot my password
Login



FacultyLogin - Exam_scheduling_system

https://localhost:7253/Home/FacultyLogin

Exam Scheduling System

Login as Faculty

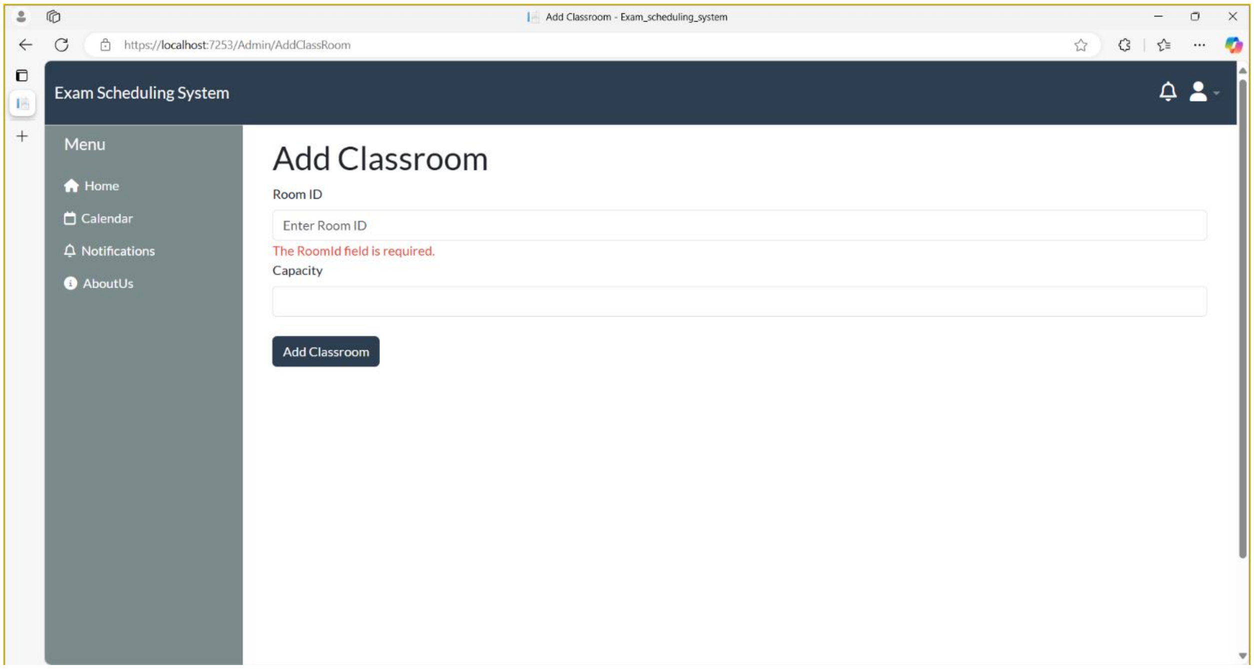
ID *

55555

Password *

••••••••••••

Invalid ID or Password.



Add Classroom - Exam_scheduling_system

https://localhost:7253/Admin/AddClassRoom

Exam Scheduling System

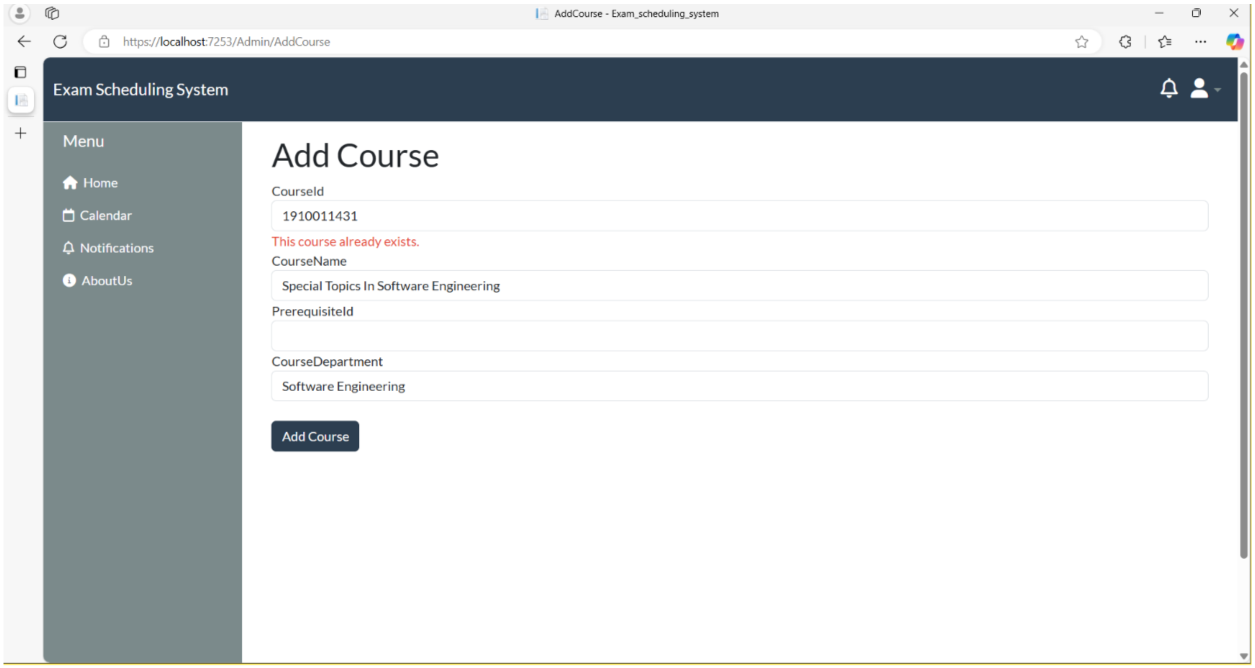
Add Classroom

Room ID

Enter Room ID

The RoomId field is required.

Capacity


AddCourse - Exam_scheduling_system
https://localhost:7253/Admin/AddCourse
Exam Scheduling System
Menu
Home
Calendar
Notifications
AboutUs
Add Course

Add Course

CourseId

1910011431

This course already exists.

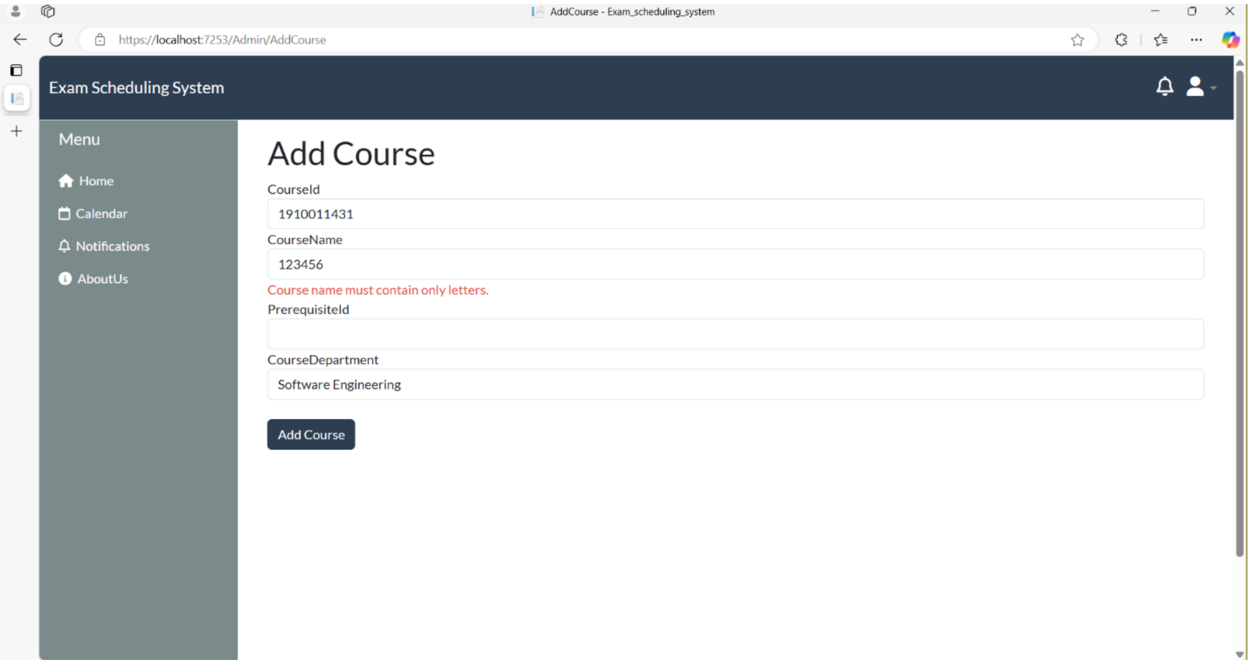
CourseName

Special Topics In Software Engineering

PrerequisiteId

CourseDepartment

Software Engineering


AddCourse - Exam_scheduling_system
https://localhost:7253/Admin/AddCourse
Exam Scheduling System
Menu
Home
Calendar
Notifications
AboutUs
Add Course

Add Course

CourseId

1910011431

CourseName

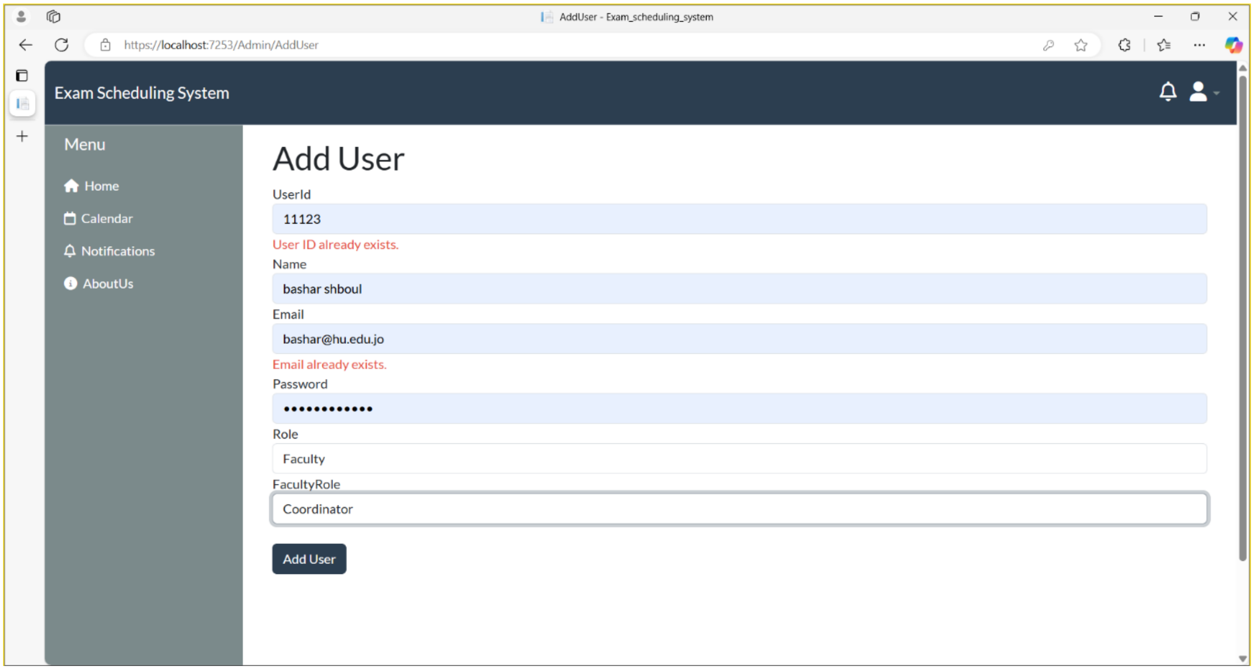
123456

Course name must contain only letters.

PrerequisiteId

CourseDepartment

Software Engineering


AddUser - Exam_scheduling_system
https://localhost:7253/Admin/AddUser
Exam Scheduling System
Menu
Home
Calendar
Notifications
AboutUs

Add User

UserId: 11123

User ID already exists.

Name: bashar shboul

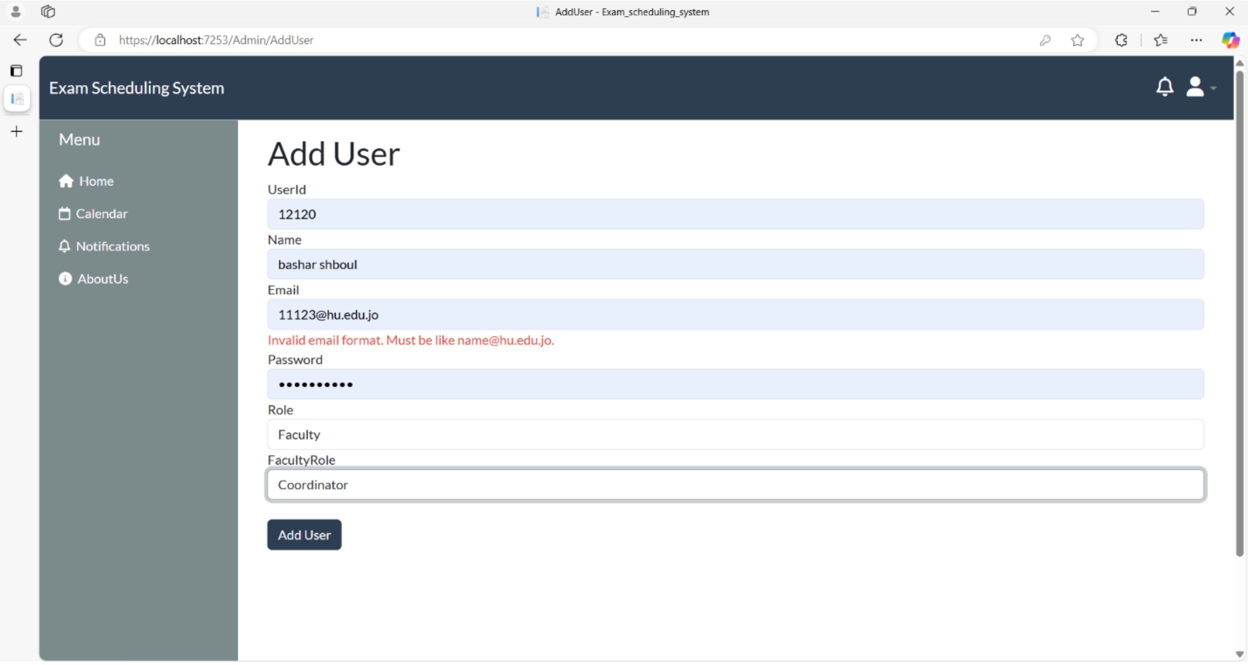
Email: bashar@hu.edu.jo

Email already exists.

Password: ••••••••••••

Role: Faculty

FacultyRole: Coordinator


AddUser - Exam_scheduling_system
https://localhost:7253/Admin/AddUser
Exam Scheduling System
Menu
Home
Calendar
Notifications
AboutUs

Add User

UserId: 12120

Name: bashar shboul

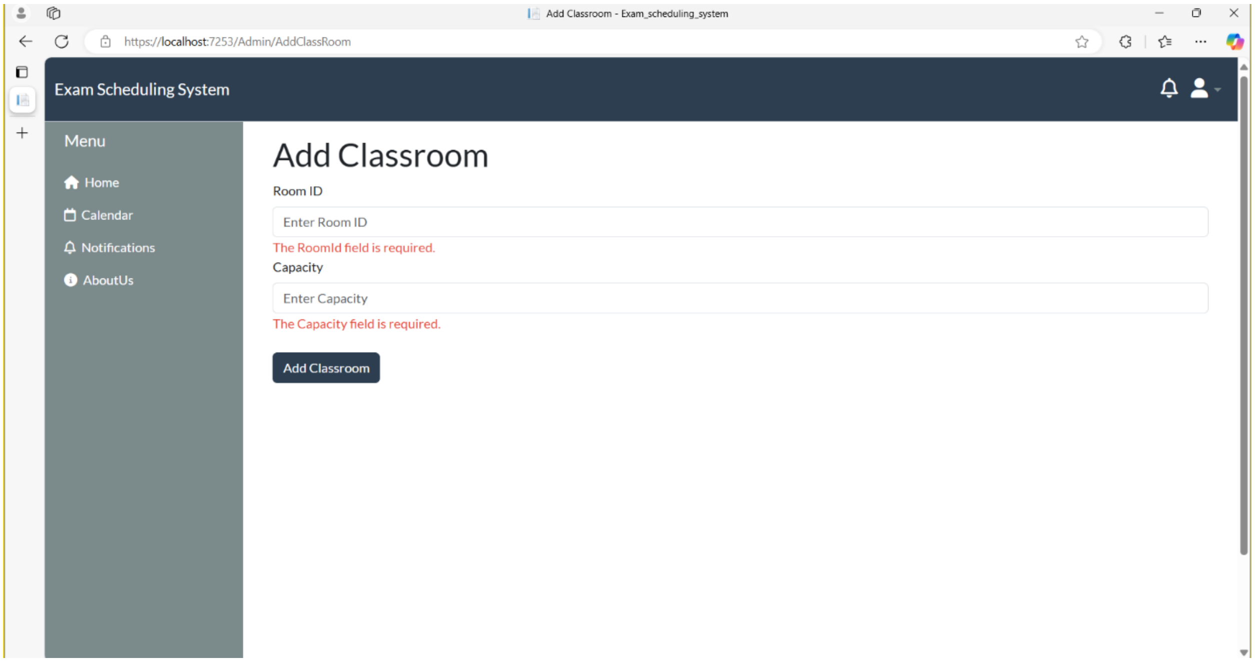
Email: 11123@hu.edu.jo

Invalid email format. Must be like name@hu.edu.jo.

Password: ••••••••••

Role: Faculty

FacultyRole: Coordinator



Add Classroom - Exam_scheduling_system
https://localhost:7253/Admin/AddClassRoom
Exam Scheduling System
Menu
Home
Calendar
Notifications
AboutUs

Add Classroom

Room ID

Enter Room ID

The RoomId field is required.

Capacity

Enter Capacity

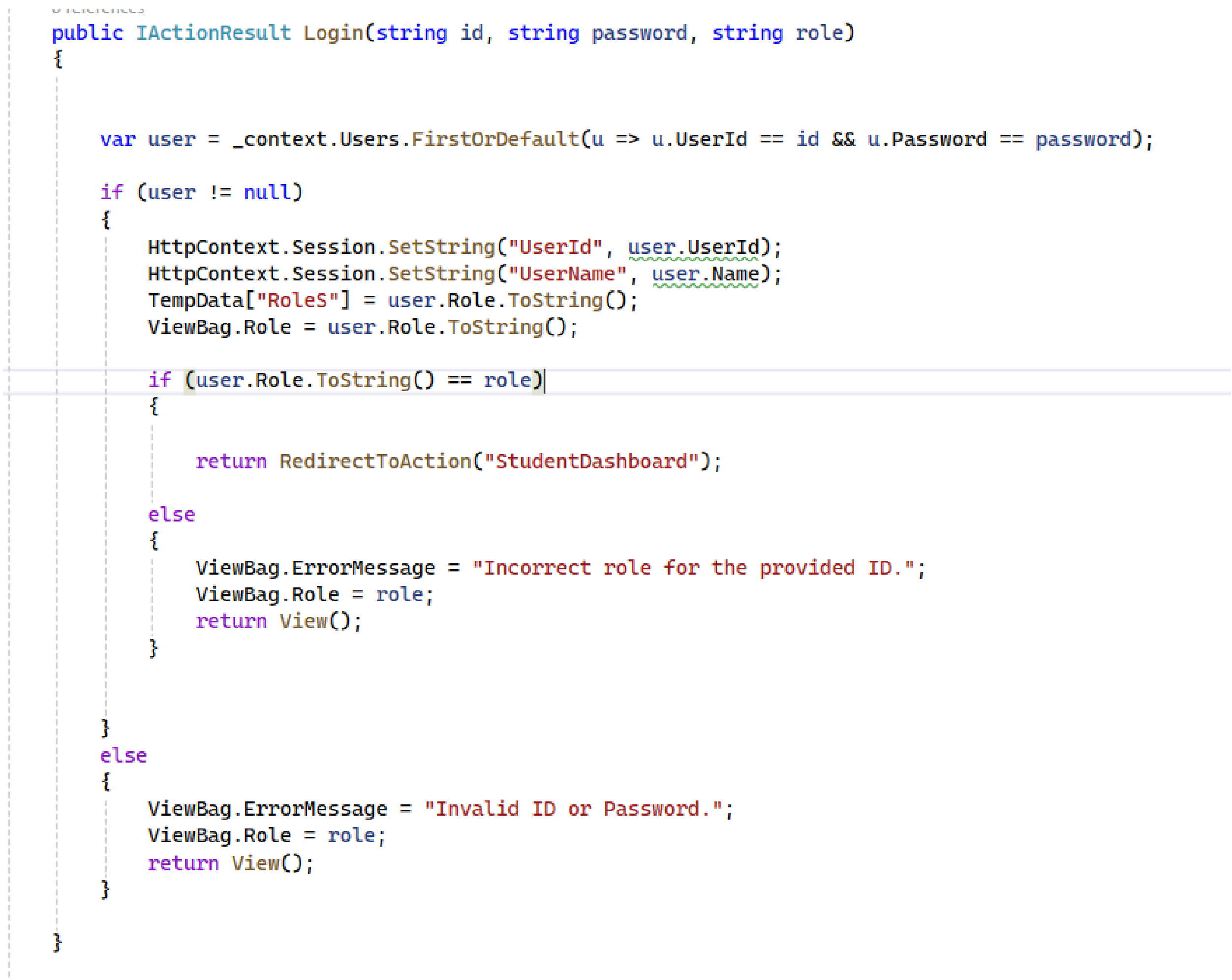
The Capacity field is required.

Add Classroom

6.2 white Box

6.2.1 : Log in with invalid ID or password

6.2.2 : Log in with invalid ID for role



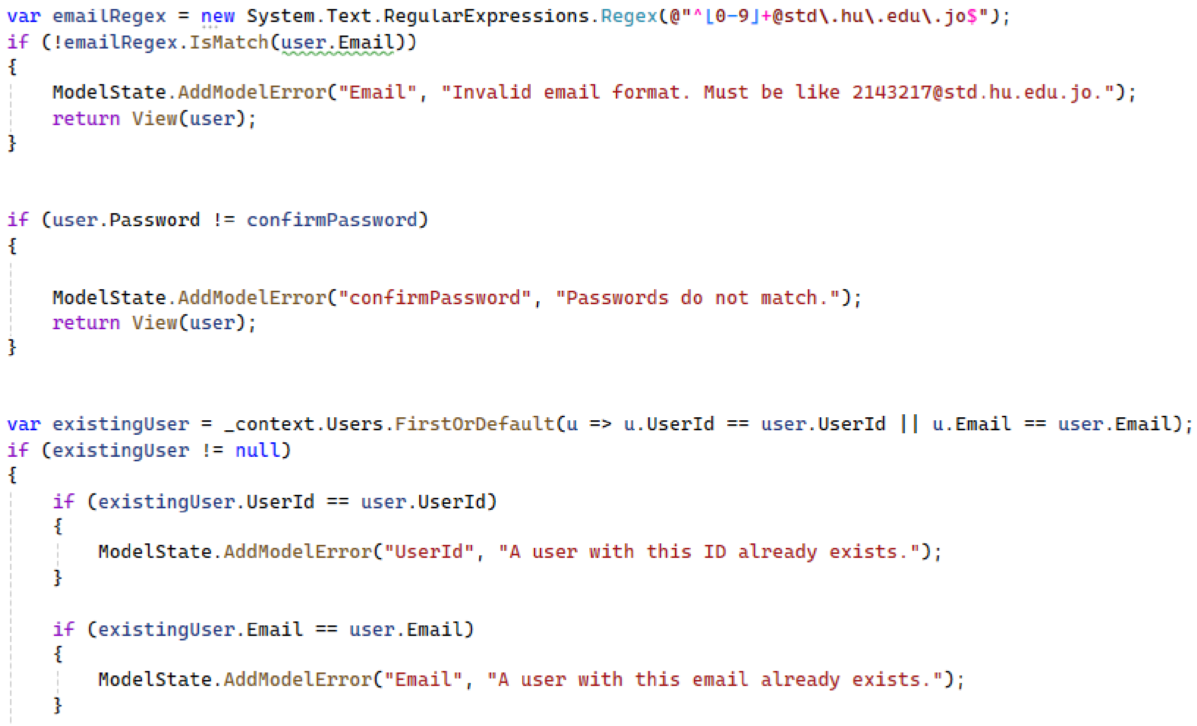
```
0 references
public IActionResult Login(string id, string password, string role)
{


    var user = _context.Users.FirstOrDefault(u => u.UserId == id && u.Password == password);

    if (user != null)
    {
        HttpContext.Session.SetString("UserId", user.UserId);
        HttpContext.Session.SetString("UserName", user.Name);
        TempData["RoleS"] = user.Role.ToString();
        ViewBag.Role = user.Role.ToString();

        if (user.Role.ToString() == role)
        {

            return RedirectToAction("StudentDashboard");

        else
        {
            ViewBag.ErrorMessage = "Incorrect role for the provided ID.";
            ViewBag.Role = role;
            return View();
        }


    }
    else
    {
        ViewBag.ErrorMessage = "Invalid ID or Password.";
        ViewBag.Role = role;
        return View();
    }

}
```

6.2.3: Fill wrong email format

6.2.4: Not compatible password

6.2.5: Sign up with existing email

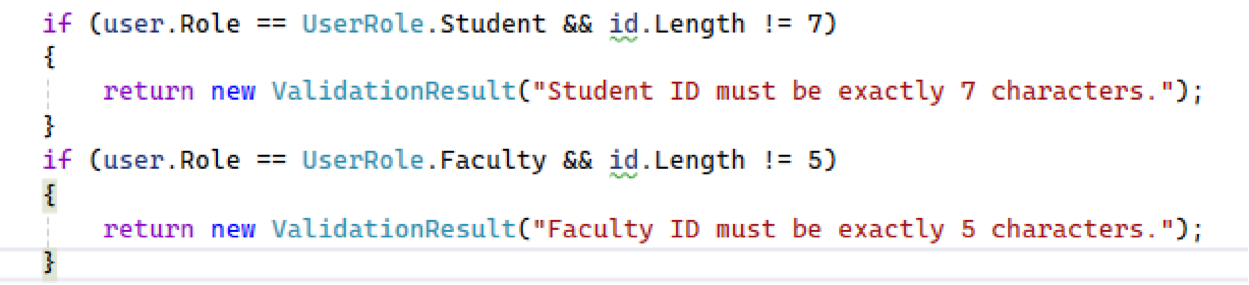
6.2.6: Sign up with existing ID



```
var emailRegex = new System.Text.RegularExpressions.Regex(@"^[0-9]+@std\.hu\.edu\.jo$");
if (!emailRegex.IsMatch(user.Email))
{
    ModelState.AddModelError("Email", "Invalid email format. Must be like 2143217@std.hu.edu.jo.");
    return View(user);
}


if (user.Password != confirmPassword)
{

    ModelState.AddModelError("confirmPassword", "Passwords do not match.");
    return View(user);
}


var existingUser = _context.Users.FirstOrDefault(u => u.UserId == user.UserId || u.Email == user.Email);
if (existingUser != null)
{
    if (existingUser.UserId == user.UserId)
    {
        ModelState.AddModelError("UserId", "A user with this ID already exists.");
    }

    if (existingUser.Email == user.Email)
    {
        ModelState.AddModelError("Email", "A user with this email already exists.");
    }
```

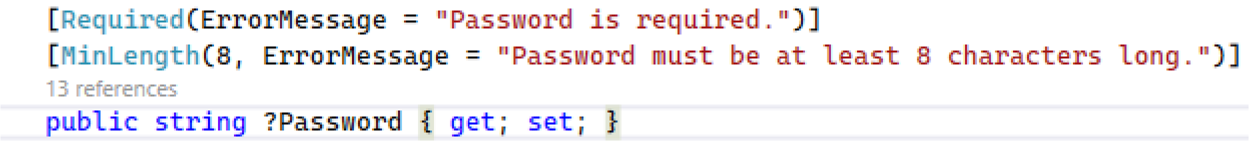
6.2.7: Sign up with ID not exactly 7 characters for student

6.2.8: Add Faculty Member with ID not exactly 5 characters



```
if (user.Role == UserRole.Student && id.Length != 7)
{
    return new ValidationResult("Student ID must be exactly 7 characters.");
}
if (user.Role == UserRole.Faculty && id.Length != 5)
{
    return new ValidationResult("Faculty ID must be exactly 5 characters.");
}
```

6.2.9: Sign Up with password less than 8 characters



```
[Required(ErrorMessage = "Password is required.")]
[MinLength(8, ErrorMessage = "Password must be at least 8 characters long.")]
13 references
public string ?Password { get; set; }
```

Chapter 7: Conclusion And Results

Finally, the website aims to reduce the conflicts with exams in Information Technology College as much as possible and try to make the exam reservation process well organized.

7.1 Summary of accomplished project.

This document describes the system as a whole supported with diagrams (use case diagram, class diagram , sequence diagram , and ER diagram) , these diagrams done after the idea has been analyzed , then the prototype go through three stages to be ready , first the wireframe, design the website structure using papers ,then mockup design the website using Figma with adding content , Finally connect the screens and buttons together and make it interactive. Then we convert the prototype to an executable website that will be ready to use in the college and solve the problem.

7.2 future work

There are many things that aim to achieve in the future updates on the website, such as:

- Make the exam scheduling process automated by one click.
- Publishing the website to be used in all colleges in the university.
- Make the coordinator to be able to determine the number of invigilators.
- Add the property filtration for the exam table.

References:

[1] http://www.examplanner.com/
[2] https://www.theexamcentre.ie/
[3] https://services.just.edu.jo/ExamsSchedule/
[4] https://cbtf.illinois.edu/students/res
[5] https://www.ucalgary.ca/registrar/exams/external-exam-centre/general-exam-reservation