Palestine Technical University - Kadoorie

Faculty of Engineering and Technology

Department of Computer Systems Engineering

Management and organization of operations in the Department of Computer Systems Engineering

By:

Fatima Omar Abuasbeh	201911043
Malak Nedal Tabeasheh	201911453
Maymoona Ayoub Bani Oudeh	201910525

Supervisor:

Dr. Mohammad Khalil

This document was submitted in Fulfillment of the requirements for the "Introduction to Graduation Project" course

Tulkarm, Palestine
June 2023

ACKNOWLEDGEMENT

We dedicate our work in this to:

Our homeland is Palestine, which inspired us all these years.

Al-Aqsa Mosque and Gaza steadfastness.

Our martyrs and prisoners are the symbols of sacrifices.

Our great teacher Dr. Mohammad Khalil, who counseled us in this project, and all teachers who illuminated our way.

Our university, Palestine Technical University - Kadoorie, where spent our study and diligence and met great people.

Our beloved parents and family, who believed in us.

And everyone who kept sweet words for us.

ABSTRACT

The proposed project, the CSED (Computer Systems Engineering Department Portal), is an innovative solution designed to address the existing gaps in the academic portals and Learning Management Systems (LMS) currently in use at Palestine Technical University - Kadoorie. Exclusively developed for the Computer Systems Engineering Department, CSED incorporates three key features aimed at enhancing departmental engagement and streamlining internal communication.

The first feature allows the head of the department to disseminate crucial announcements and updates efficiently. This tackles the challenge within current academic portals where important information may get lost in the sea of university-wide updates.

Secondly, CSED includes a streamlined appointment booking system for students to meet with academics. This addresses the communication obstacles often encountered in standard LMS, with academics managing their availability and students easily scheduling appointments, thus refining the academic consultation process.

The third feature is a unique photo archive that houses a visual record of the department's activities and accomplishments. This element fosters a sense of community and cohesion often lacking in impersonal LMS platforms.

Exclusively accessible to department members and managed by the department head, the CSED platform fills the void for a secure, department-specific portal. By melding streamlined communication, simplified academic interaction, and community building, the CSED platform is set to surpass the functionalities of traditional academic portals and LMS. It promises to provide an enhanced academic environment for the Computer Systems Engineering Department at Palestine Technical University - Kadoorie.

Keywords: CSED, LMS, Announcements, Appointment, Photo archive

Table of Contents

CHAPTER 1

INTRODUCTION

In this chapter, we will discuss the outlines of our project.

1.1 Overview

The Computer Systems Engineering (CSE) department is typically a part of the engineering faculty in a university. It focuses on the study and application of principles and techniques related to computer systems and their integration into various domains. Here is some information about the Computer Systems Engineering department in our university:

Curriculum: The department offers a comprehensive curriculum that covers a range of topics, including computer hardware, software engineering, computer networks, operating systems, embedded systems, digital systems design, computer architecture, and system integration. The curriculum is designed to provide students with a solid foundation in both hardware and software aspects of computer systems.

Specializations: Some universities may offer specializations or elective tracks within the Computer Systems Engineering department. These specializations allow students to focus on specific areas of interest, such as embedded systems, robotics, computer architecture, cyber security, or network engineering.

Laboratory Work: Computer Systems Engineering programs often emphasize practical skills, and students are required to complete laboratory work to reinforce theoretical concepts. They may work with hardware components, design and develop software systems, simulate computer networks, or build and program embedded systems.

Projects and Research: Many universities encourage students to engage in projects and research activities. These could involve designing and implementing a computer system, optimizing system performance, developing software applications, or exploring emerging technologies. Such projects offer students hands-on experience and the opportunity to apply their knowledge in real-world scenarios.

1.2 Problem Statement

The Computer Systems Engineering Department at Palestine Technical University - Kadoorie currently faces several challenges related to effective and efficient communication and engagement. While the university has broad-based academic portals and a Learning Management System (LMS), these platforms do not offer a focused, department-specific communication channel.

Critical departmental announcements often get overshadowed amidst a myriad of university-wide updates, resulting in these important updates not reaching their intended recipients. Using social media platforms such as Facebook for disseminating such information is not a desirable or professional solution, given the need for maintaining academic decorum and ensuring information privacy.

Moreover, the current LMS lacks a feature to facilitate the scheduling of appointments between students and their professors. This void leads to a reliance on less organized, time-consuming, and often ineffective communication methods for appointment scheduling.

In addition, there is a notable absence of a centralized archive for the department's visual records – a single repository that maintains photos of departmental activities, events, and milestones. The lack of such a facility results in a scattered and disorganized collection of photos, hindering access and appreciation of these important visual records.

These challenges highlight a clear need for a secure, dedicated platform that can streamline department-specific communication, provide a system for efficiently scheduling appointments, and house a consolidated photo archive. The creation of such a platform is critical for improving internal communication, fostering a stronger sense of community, and enhancing the overall academic experience within the Computer Systems Engineering Department at Palestine Technical University - Kadoorie.

1.3 Objectives

- **Department-Specific Communication:** To create a centralized communication channel dedicated solely to the Computer Systems Engineering Department. This will ensure that all important announcements reach the relevant members efficiently and effectively.

- **Streamlined Appointment Booking:** To develop a feature that allows students to book appointments with their academics in a streamlined and organized manner. This will enhance the efficiency of academic consultation and foster a more systematic approach to student-faculty interaction.

- **Photo Archive:** To build a consolidated and organized photo archive for the department. This archive will serve as a repository for photos from departmental events, activities, and milestones, fostering a sense of community and shared history among department members.

- **Secure Access:** To ensure that the portal is securely accessible only to members of the Computer Systems Engineering Department, by maintaining the privacy and relevance of information shared and interactions facilitated on the platform.

- **User-Friendly Interface:** To design a platform with a user-friendly interface that simplifies navigation and usage for all members, regardless of their technical proficiency.

- **Admin Oversight:** To enable administrative oversight for the head of the department, ensuring smooth functioning, maintenance, and moderation of content on the platform.

- **Scalability:** To ensure the platform is scalable, able to accommodate growth in department size and evolving departmental needs over time.

By achieving these objectives, the CSED project aims to facilitate communication, embed the major's achievements, and enhance academic interaction within the Computer Systems Engineering Department at Palestine Technical University - Kadoorie.

1.4 Scope of Work

- **System Analysis and Design:** The project begins with a comprehensive system analysis to identify the department's specific needs and design a system that aligns with these requirements. This will involve collaborative discussions with the department's key stakeholders, including the head of the department, academics, and students.

- **Platform Development:** The next stage is the development of the CSED platform, ensuring it meets the requirements identified in the system analysis stage. This involves creating a dedicated communication channel, a streamlined appointment booking system, a consolidated photo archive, and a secure access system. The platform will be developed with a user-friendly interface to ensure ease of use by all department members.

- **Admin Control Setup:** An administrative control panel will be set up for the head of the department. This panel will allow the department head to oversee the platform's operations, moderate content, and make necessary updates as required.

- **Testing and Quality Assurance:** Rigorous testing will be conducted to ensure that the platform functions as expected without glitches. This includes functional testing, usability testing, performance testing, and security testing. Necessary adjustments will be made based on the test results to ensure the platform meets the desired quality standards.

- **Training and Documentation:** Training will be provided to the department head, academics, and students to familiarize them with the platform's features and usage. Comprehensive documentation will be developed, outlining the platform's functionalities, usage guidelines, troubleshooting, and maintenance procedures.

- **Deployment and Maintenance:** The platform will be deployed for the department's use upon successful testing and training completion. Regular maintenance will be provided to ensure the platform continues to function smoothly and securely, and to implement any necessary updates or enhancements in response to user feedback or evolving departmental needs.

1.5 Procedure

Simple step-by-step procedure to realize the CSED project:

- **Requirements Gathering:** First, talking to the key stakeholders, which include the head of the department, academics, and students, to understand their needs and preferences for the new platform.

- **Make a questionnaire:** The questionnaire we developed for this project serves as a critical tool to understand the computer system engineering students in our university better. It is designed to delve into their current challenges and frustrations regarding appointment scheduling and resource sharing within their university context. [1-1]

- **Designing the Platform:** Based on the information gathered, create a design outline of the CSED platform. This design should include a communication channel, an appointment booking system, a photo archive, and a secure access system.

- **Developing the Platform:** Start building the platform according to the design. This includes writing the code, setting up databases, and integrating the necessary software components.

- **Setting up Admin Controls:** Develop an administrative control panel for the head of the department, allowing them to oversee, moderate, and update content on the platform.

- **Testing the Platform:** Conduct rigorous testing to identify and fix any issues. Check if everything works as intended and make necessary adjustments.

- **Training Users:** Organize training sessions for the department head, academics, and students to help them understand how to use the platform effectively.

- **Deploying the Platform:** Once everything is ready and everyone is trained, launch the platform for the department's use.

- **Maintaining the Platform:** Regularly check the platform to ensure it's running smoothly. Provide updates or enhancements as necessary.

CHAPTER 2

RELATED WORKS

In this chapter, we will discuss the literature review (related works)

2.1 Moodle (LMS)

The PTUK LMS is built on Moodle, a popular open-source LMS widely used by educational institutions around the world. Moodle provides a platform for online learning, and it comes with many features like course management, grading systems, and support for various learning resources such as files, forums, quizzes, and assignments. It allows for collaborative learning, with students able to interact with each other and their instructors. However, the PTUK LMS, like many Moodle implementations, may not cater to some specific needs of individual departments or offer certain functionalities. [2-1]

The CSED project is designed to cater to the specific needs of the Computer Systems Engineering Department, filling the gaps left by the PTUK LMS.

By filling these gaps, the CSED project can complement the PTUK LMS to provide a comprehensive academic platform for the Computer Systems Engineering Department. It can be tailored to align more closely with the workflows and specific needs of the department, potentially providing a more user-friendly experience for both students and academics.

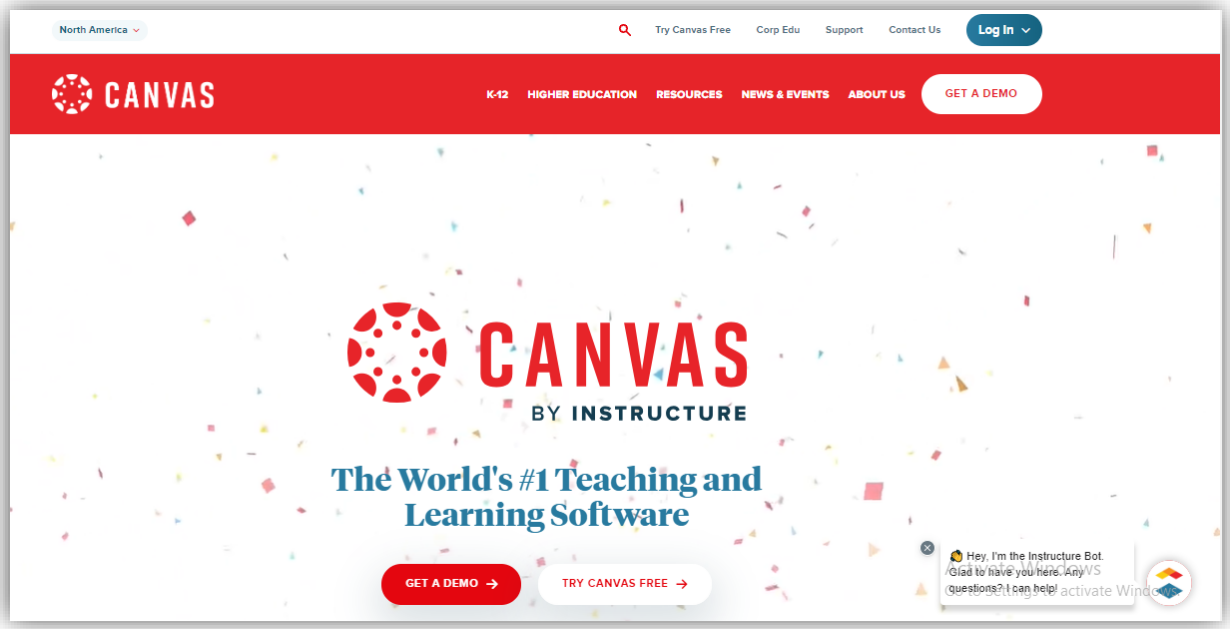
In conclusion, while the PTUK LMS serves as a general educational platform, CSED is designed to cater to specific needs of the Computer Systems Engineering Department, thus enhancing the overall academic experience at the department level. Regular feedback and iterative improvements can ensure that the platform continues to evolve and better serve its users.

2.2 Edugate

The academic platform of Palestine Technical University - Kadoorie (PTUK). This platform is a student information system that provides various functionalities including course registration, access to grades, and other administrative tasks related to students' academic journey. Edugate, as an academic portal, serves as the primary interface between students and the administrative aspects of their education at PTUK. The system likely includes features such as student registration, course enrollment, grade access, and tracking of academic progress. While such a system is crucial for the functioning of the university, its broad administrative scope might overlook the specific needs of individual departments or provide a platform for direct academic interaction. [2-2]

The CSED project, on the other hand, aims to provide a platform that is tailored specifically to the needs of the Computer Systems Engineering Department. While Edugate serves the broad administrative needs of students, CSED fills the gaps by providing a platform tailored specifically to the needs of the Computer Systems Engineering Department, enhancing the overall academic experience for both students and academics within the department. Through regular feedback and iterative improvements, the platform can continue to evolve to better serve its users.

2.3 Canvas



Canvas provides a flexible, customizable, and intuitive environment for learning and teaching. It offers a wide range of features, including:

Course management: professors can upload course materials, assignments, and conduct assessments.

Collaboration tools: students can interact with each other and with teachers through discussion boards, group projects, etc.

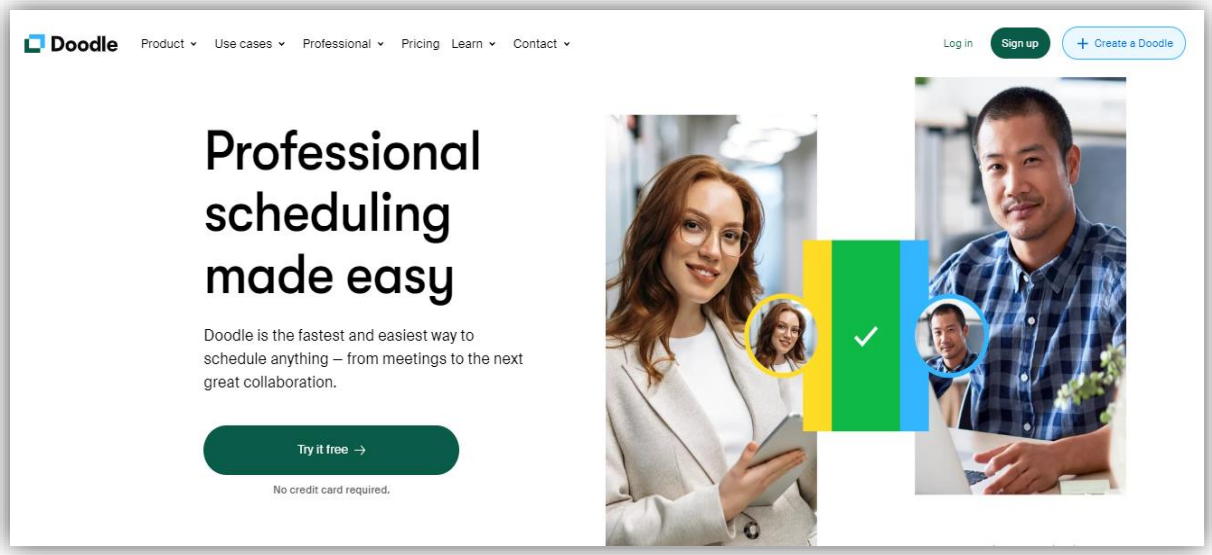
Integrated calendar: allows students and teachers to track deadlines and schedule appointments.

Communication: teachers can post announcements and updates.

The CSED platform is more specialized aimed specifically at the Computer Systems Engineering Department at PTUK. It focuses on department-specific needs that are not addressed by generalized LMS platforms like Canvas.

So, while Canvas can serve as an excellent reference point due to its extensive feature set and usability, remember that the CSED project is unique in its focus on the specific needs of the Computer Systems Engineering Department at PTUK. [2-3]

2.4 Doodle



Doodle is primarily focused on scheduling. It's widely used in a professional context to help teams find suitable times for meetings by eliminating the back-and-forth typically associated with scheduling. Organizers can propose several meeting times, and participants can vote on the times that work best for them. The system then suggests the optimal meeting time based on participants' responses. While Doodle is a powerful tool for this purpose, it's not specifically designed to cater to the needs of academic departments, and it lacks several features that would be useful in such a context. [2-4]

On the other hand, the CSED project aims to provide a comprehensive academic platform for the Computer Systems Engineering Department at PTUK. While it includes a system for scheduling appointments with academics similar to Doodle's functionality, it also offers several additional features tailored specifically to the needs of the department, while Doodle is a powerful scheduling tool, the CSED platform offers a more holistic solution for the needs of an academic department. CSED combines scheduling functionality with other features like department-specific announcements, a photo archive, and user account management to provide a comprehensive academic platform for the Computer Systems Engineering Department at PTUK.

2.5 Comparison

In short, we have summarized the comparisons between our site and similar sites in this table:

Feature\Platform	Canvas	Doodle	PTUK (LMS)	PTUK Edugate	Our Project (CSED)
*User Authentication	✓	✓	✓	✓	✓
Calendar Integration	✓	✓	✓	X	✓
Appointment Booking	X	✓	X	X	✓
Personalized User Profiles	✓	X	✓	✓	✓
Custom Announcement System	✓	X	✓	✓	✓
*Course Management	✓	X	✓	✓	X
Photo Archive	X	X	X	X	✓
Document Sharing	✓	X	✓	✓	✓
*Role-based Access Control	✓	✓	✓	✓	✓
Mobile App	✓	✓	✓	X	X
Personalized Notifications	✓	X	✓	✓	✓
Arabic language support	X	X	✓	✓	✓

***User authentication**: the process of verifying the identity of an individual accessing a system or application.

***Course management**: the administration and organization of educational courses, including tasks such as creating, scheduling, and managing course content, assignments, and student enrollments.

***Role-based access control (RBAC)**: is a method of managing and controlling access to resources based on the roles and responsibilities of users, assigning permissions and restrictions accordingly.

CHAPTER 3

SYSTEM REQUIREMENTS

In this chapter, we will discuss the functional and nonfunctional requirements

3.1 Functional Requirements

1-User Registration and Authentication:

- Students, doctors, and admins can create accounts and log in securely.
- Admins have the ability to manage user accounts.

2-Schedule Management:

- Doctors can update their availability and manage their schedule.
- Students can view doctor schedules and book appointments.

3-Appointment Handling:

- Students can request appointments with specific details.
- Doctors receive appointment requests and can accept or reject them.

4-Announcement Posting:

- Admins can post important announcements to the system.
- Doctors can propose ads for admin approval.

5-Photo Archive Management:

- Admins can manage the photo archive.
- Doctors can propose photos or documents to be added to the archive, subject to admin approval.
- Students can view and utilize the archive.

3.2 Non-Functional Requirements

1-Usability:

- The system should be user-friendly with a clear, intuitive interface.
- Information should be well-organized and easily accessible.
- The system should provide helpful error messages

2-Security:

- User data should be protected and stored securely.
- All transactions should be encrypted for added security.
- System access levels should be implemented to limit functions based on user roles (admin, doctor, and student).

3-Performance:

- The system should be responsive and capable of handling multiple simultaneous requests.
- Load times should be minimized to provide a smooth user experience.

4-Scalability:

- The system should be designed to accommodate a growing number of users and increased data volume.

5-Reliability:

- The system should have high availability; ensuring users can access it at any time.
- The system should handle errors gracefully and recover quickly from any crashes or failures.

6-Compatibility:

- The system should be compatible with different operating systems and browsers.

7-Maintainability:

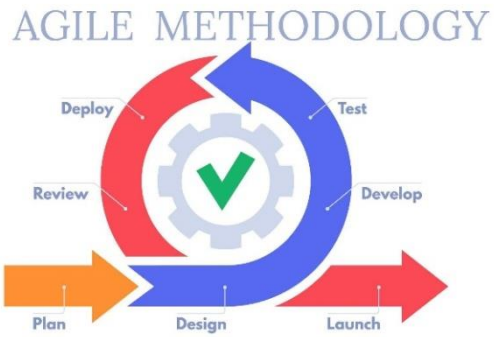
- The system should be designed in a modular manner to allow for easy updates and maintenance.

CHAPTER 4

METHODOLOGY AND TECHNOLOGY

4.1 Methodology

For our project to be done, we plan to follow the Agile Method. Agile is one of the most popular approaches to project management due to its flexibility, adaptability to change, and high level of customer input. The Agile methodology is a way to manage a project by breaking it up into sequences of repeated cycles (iterations). It involves constant collaboration with stakeholders and continuous improvement at every stage. Once the work begins, teams' cycle through a process of planning, executing, and evaluating. Continuous collaboration is vital, both with team members and project stakeholders. Methodology in Agile project management it's a process for managing a project that involves constant collaboration and working in iterations. Agile project management works off the basis that a project can be continuously improved upon throughout its life cycle, with changes being made quickly and responsively.[4-1]



It depends on presenting real primary products during successive periods of time to the customer (customer), which the customer (customer) can try and evaluate and give feedback or existing errors, and therefore it is easy to overcome these errors or observations being in an early stage of work, and you can progress to the next step steadily Bigger and complete the next version by adding some features and features until the project is finished.

Agile isn't defined by a set of ceremonies or specific development techniques. Rather, agile is a group of methodologies that demonstrate a commitment to tight feedback cycles and continuous improvement. And since the CSED system is part of an ever-evolving scope, we have chosen the agile methodology approach as it is the most appropriate for such systems.

4.2 Implementation

In this section we will discuss which we plan to work, all used technologies, programming languages, frameworks, and database we plan to use in building our website.

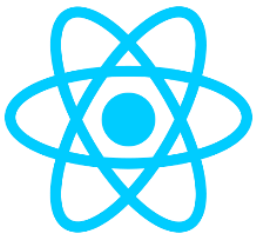
4.2.1 Overview of Implementation Stack

- **Front-End Technologies**
  - React.js
  - React Material UI
- **Back-End Technology**
  - ASP.NET framework
  - MVC
- **Database**
  - SQL Server database
  - ORM
- **Integrated Development Environment**
  - Visual Studio Code
  - Visual Studio

4.2.2 Front-End Technologies

Front-end web development, also known as client-side development is the practice of producing React for a website or Web Application so that a user can see and interact with them directly. The challenge associated with front end development is that the tools and techniques used to create the front end of a website change constantly and so the developer needs to constantly be aware of how the field is developing. [4.2]

4.2.2.1 React



React is an open-source JavaScript framework, that isn't actually a framework. But it is currently the most commonly used front-end development technology in the world. React, originally developed by and still maintained by Facebook and supported by an active open-source community, is in fact a JS 'library'.

ReactJS is responsible for building a hierarchy of, or rendering UI components and provides support for both frontend and server-side.

Like all JavaScript frameworks and libraries, React primary use case is in the creation of dynamic user interfaces, which update on the client (user)-side. The page doesn't have to be refreshed in the browser but automatically updates. Exactly like a Facebook feed does. This is why the social media giant invested in developing its own JavaScript library, specifically tailored to the needs of its own apps. [4-5]

Why Should You Use React? [4-6]

- **React is Flexible**

  React is remarkably flexible. Once you have learned it, you can use it on a vast variety of platforms to build quality user interfaces. React is a library, NOT a framework. Its library approach has allowed React to evolve into such a remarkable tool.

  "Learn React Once and Write Everywhere".

- **React Has Great Performance**

  The React team realized that JavaScript is fast, but updating the DOM makes it slow. React minimizes DOM changes. And it has figured out the most efficient and intelligent way to update DOM. Before React, most frameworks and libraries would update the DOM unintelligently to reflect a new state. This resulted in changes to a significant portion of the page.

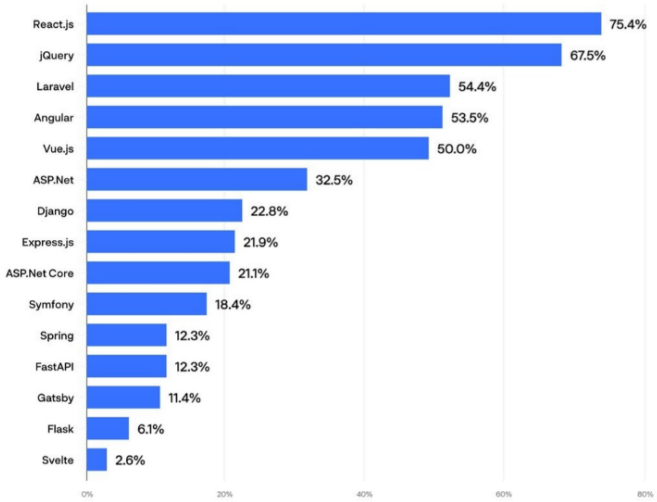
- **React is Easy to Test**

  React design is very user friendly for testing.

  - Traditional UI browser testing is a hassle to setup. On the other hand, you require very little or no configuration for testing in React.
  - Traditional UI browser requires browsers for testing, but you can test React components quickly and easily using the node command-line.
  - Traditional UI browser testing is slow. But command-line testing is fast, and you can run a considerable amount of test suites at a time.
  - Traditional UI browser testing is often time consuming and challenging to maintain. React test can be written quickly using tools like Jest & Enzyme.

- **React Has Broader Community Support**

  Since 2015, React popularity has grown steadily. It has a massive active community and its GitHub Repository has over 164k Stars. It is one of the Top 5 Repositories on GitHub.



4.2.2.2 React

Material UI

Material UI is an open-source React component library that implements Google's Material Design.

It includes a comprehensive collection of prebuilt components that are ready for use in production right out of the box.

Material UI is beautiful by design and features a suite of customization options that make it easy to implement your own custom design system on top of our components.

Advantages of Material UI

- Ship faster: Over 2,500 open-source contributors have poured countless hours into these components. Focus on your core business logic instead of reinventing the wheel.
- Beautiful by default: meticulous about our implementation of Material Design, ensuring that every Material UI component meets the highest standards of form and function, but diverge from the official spec where necessary to provide multiple great options.
- Customizability: the library includes an extensive set of intuitive customizability features.
- Cross-team collaboration: Material UI's intuitive developer experience reduces the barrier to entry for back-end developers and less technical designers, empowering teams to collaborate more effectively. The design kits streamline your workflow and boost consistency between designers and developers.
- Trusted by thousands of organizations: Material UI has the largest UI community in the React ecosystem. It's almost as old as React itself (its history stretches back to 2014) and we're in this for the long haul. You can count on community's support for years to come (e.g., Stack Overflow). [4-7]

4.2.3 Back-End Technologies

Back-end development means working on server-side software, which focuses on everything you can’t see on a website. Back-end developers ensure the website performs correctly, focusing on databases, back-end logic, and application programming interface (APIs), architecture, and servers. They use code that helps browsers communicate with databases, store, understand, and delete data. [4-8]

4.2.3.1 ASP.NET Framework

ASP.NET is an open-source, server-side web-application framework designed for web development to produce dynamic web pages. It was developed by Microsoft to allow programmers to build dynamic web sites, applications and services. The name stands for Active Server Pages Network Enabled Technologies.

It was first released in January 2002 with version 1.0 of the .NET Framework and is the successor to Microsoft's Active Server Pages (ASP) technology. ASP.NET is built on the Common Language Runtime (CLR), allowing programmers to write ASP.NET code using any supported .NET language. The ASP.NET SOAP extension framework allows ASP.NET components to process SOAP messages.
[4-9]

Why we will use ASP.NET Framework?[4-10]

1. ASP.NET offers amazing flexibility when it comes to language preferences. Developers can choose the language that they best prefer and work on it to develop their application. Developers can also use multiple languages together.
2. ASP follows an efficient coding process and provides a solution for the ages. The platform helps simplify the usage of code in a complex application. Developers can decrease the amount of code using .NET tools and strategies to get the same results.
3. ASP.NET can simplify the deployment process and make it fairly easy for all involved. Developers do not have to register different components, as all info pertaining to the configuration is already available.
4. The pre-built authentication provided on ASP.NET makes sure that all of your applications are safe and secure.
5. All processes are closely managed by the ASP.NET runtime. A new process is created if one dies.

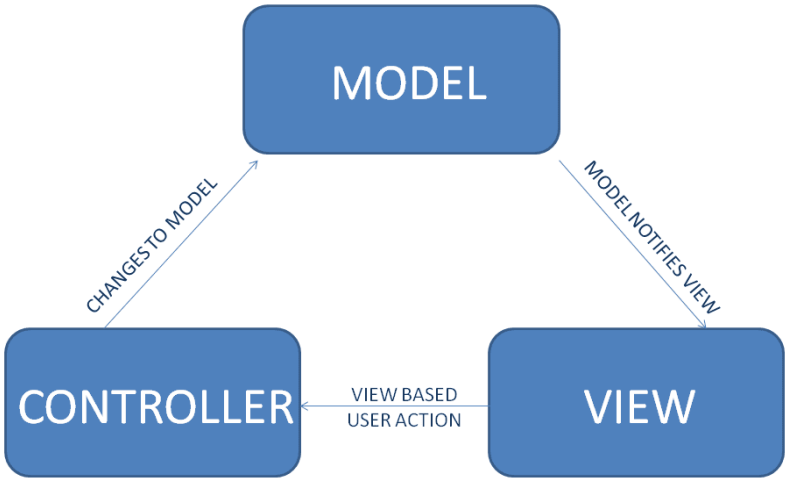
ASP.NET is surely one of the best frameworks and programming solutions in town. The reasons mentioned here can help pull many developers in favor of the framework.

4.2.3.2 MVC

The Model-View-Controller (MVC) architectural pattern separates an application into three main components: the model, the view, and the controller. The ASP.NET MVC framework provides an alternative to the ASP.NET Web Forms pattern for creating MVC-based Web applications. The ASP.NET MVC framework is a lightweight, highly testable presentation framework that (as with Web Forms-based applications) is integrated with existing ASP.NET features, such as master pages and membership-based authentication.

MVC is a standard design pattern that many developers are familiar with. Some types of Web applications will benefit from the MVC framework. Others will continue to use the traditional ASP.NET application pattern that is based on Web Forms and post backs. Other types of Web applications will combine the two approaches; neither approach excludes the other.

The MVC framework includes the following components:



- **Models**

  Model objects are the parts of the application that implement the logic for the application s data domain. Often, model objects retrieve and store model state in a database. For example, a Product object might retrieve information from a database, operate on it, and then write updated information back to a Products table in SQL Server.

  In small applications, the model is often a conceptual separation instead of a physical one. For example, if the application only reads a data set and sends it to the view, the application does not have a physical model layer and associated classes. In that case, the data set takes on the role of a model object.

- **Views**

  Views are the components that display the application s user interface (UI). Typically, this UI is created from the model data. An example would be an edit view of a Products table that displays text boxes, drop-down lists, and check boxes based on the current state of a Products object.

- **Controllers**

  Controllers are the components that handle user interaction, work with the model, and ultimately select a view to render that displays UI. In an MVC application, the view only displays information; the controller handles and responds to user input and interaction. For example, the controller handles query-string values, and passes these values to the model, which in turn queries the database by using the values.

The MVC pattern helps you create applications that separate the different aspects of the application (input logic, business logic, and UI logic), while providing a loose coupling between these elements. The pattern specifies where each kind of logic should be located in the application. The UI logic belongs in the view. Input logic belongs in the controller. Business logic belongs in the model. This separation helps you manage complexity when you build an application, because it enables you to focus on one aspect of the implementation at a time. For example, you can focus on the view without depending on the business logic. [4-11]

4.2.4 Database

4.2.4.1 SQL Server

SQL Server is a relational database management system (RDBMS) developed by Microsoft. It is primarily designed and developed to compete with MySQL and Oracle database. SQL Server supports ANSI SQL, which is the standard SQL (Structured Query Language) language. However, SQL Server comes with its own implementation of the SQL language, T-SQL (Transact-SQL).

T-SQL is a Microsoft propriety Language known as Transact-SQL. It provides further capabilities of declaring variable, exception handling, stored procedure, etc. [4-12]

Why we will use relational database?

Our website data is simple and well organized and there should not be empty values in its file, so the relational database option was better.

4.2.4.2 ORM

What is an ORM Tool?

An ORM tool is software designed to help OOP developers interact with relational databases. So instead of creating your own ORM software from scratch, you can make use of these tools.

Here's an example of SQL code that retrieves information about a particular user

```
"SELECT id, name, email, country, phone_number FROM users WHERE id = 20"
```

From a database:

The code above returns information about a username, email, country, and phone number from a table called users. Using the WHERE clause, we specified that the information should be from a user with an id of 20.

On the other hand, an ORM tool can do the same query as above with simpler methods. That is:

```
users.GetById(20)
```

So, the code above does the same as the SQL query. Note that every ORM tool is built differently so the methods are never the same, but the general purpose is similar.
ORM tools can generate methods like the one in the last example. [4-13]

4.2.5 Integrated Development Environment

An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of at least a source code editor, build automation tools and a debugger. [4-14]

4.2.5.1 Visual Studio Code

Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging.
First and foremost, it is an editor that gets out of your way. The delightfully frictionless edit-build-debug cycle means less time fiddling with your environment, and more time executing on your ideas. [4-15]
And we will use this environment for front end.

4.2.5.2 Visual Studio

Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs including websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silver light. It can produce both native code and managed code. [4-16]
And we will use this environment for back end.

CHAPTER 5

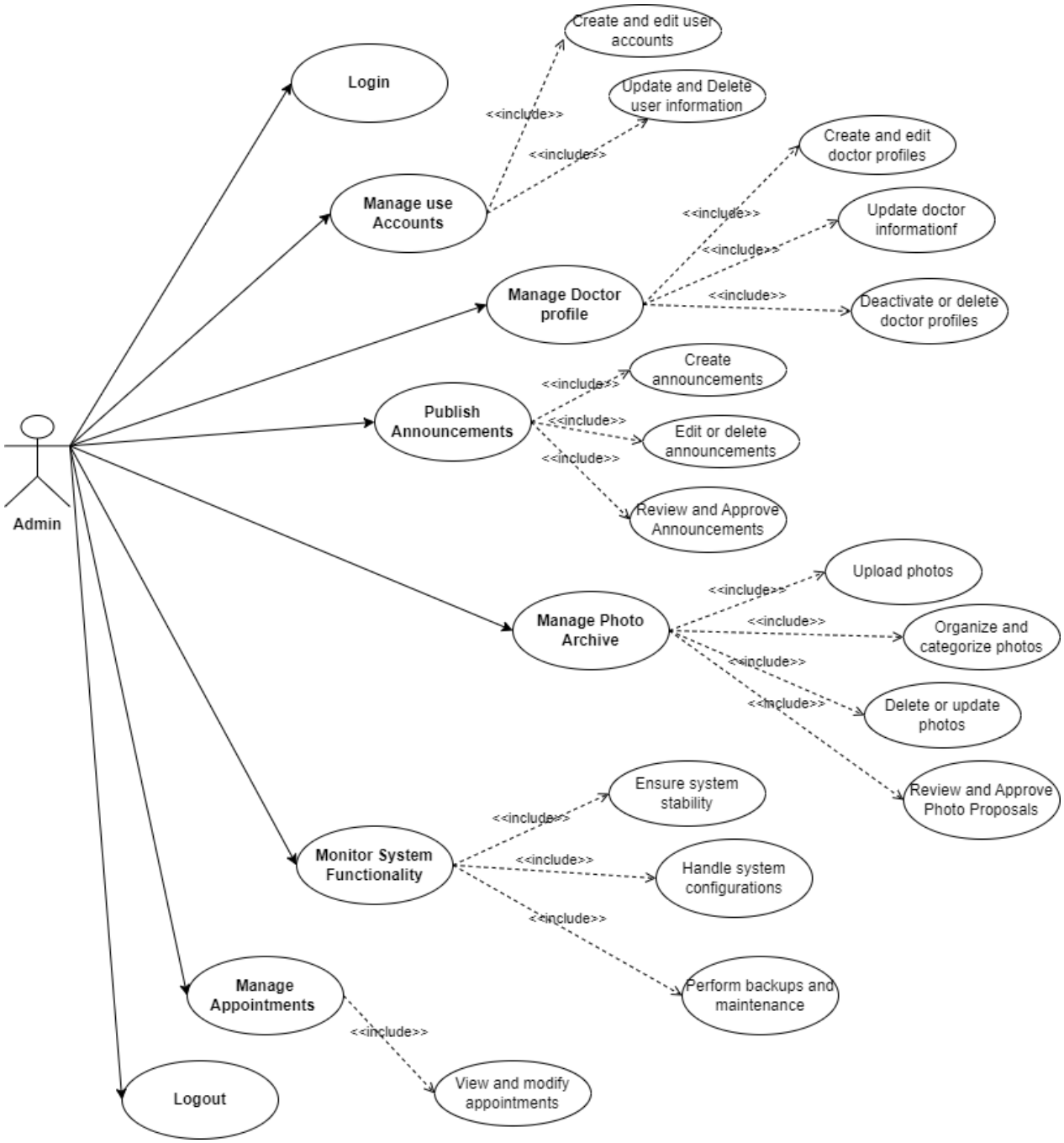
SOFTWARE DESIGN

5.1 Actors

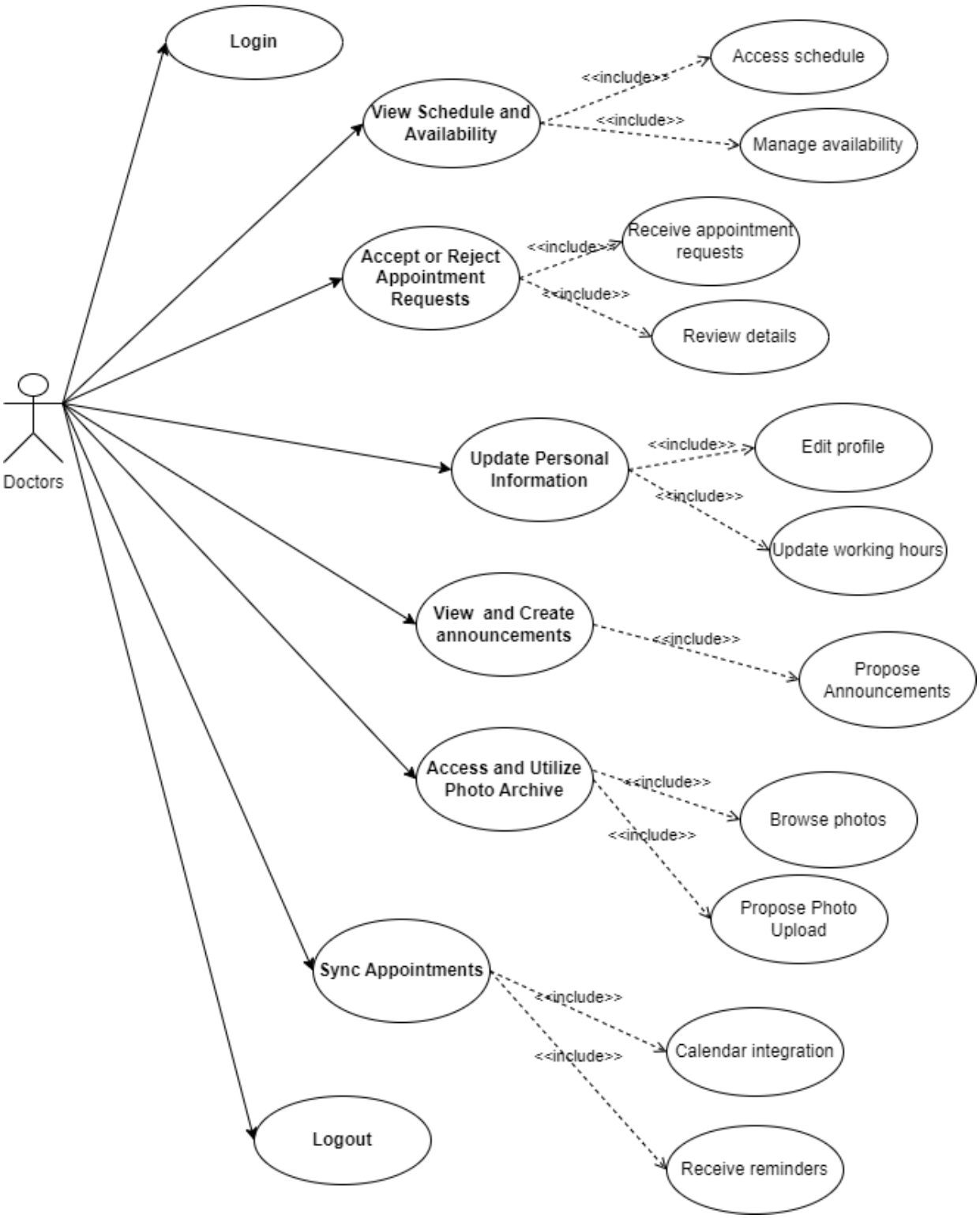
Actor	Description
Student	The primary users of the system who need to schedule appointments with doctors, view the academic photo archive, and receive important announcements. They can also utilize the resources from the photo archive for their academic activities, research, or presentations.
Doctor	Doctors are another primary user of the system. They need to manage their schedule, accept or reject appointment requests from students, and utilize the resources from the photo archive for their academic activities, research, or presentations. They also have the ability to post advertisements and upload photos to the archive, pending approval from the administrator.
Admin	The Admin manages the overall functioning of the system. They are responsible for managing the photo archive, approving doctor advertisement submissions and photo uploads, and posting important system-wide announcements. They have the highest level of control in the system.
System	The system is the platform itself that facilitates the interaction between students, doctors, and the admin. It is responsible for executing the requests of the actors, such as scheduling appointments, managing the photo archive, posting announcements, and so on.

5.2 Use Case Diagrams

5.2.1 Admin use case diagram



5.2.2 Doctor use case diagram



5.2.3 Student use case diagram

5.3 Use Case Descriptions

5.3.1 Admin

Use Case Name:	Manage User Accounts
Description:	The Admin manages the user accounts, which involves creating new accounts, updating existing ones, and assigning roles
Scenario:	An existing student or doctor needs to be added to the system, or a current user's details need to be updated.
Event Launch:	This event is triggered when a new user is to be added to the system or when an existing user's information needs updating.
Brief Description:	The Admin adds a new user to the system by entering the relevant details and assigning a role. For existing users, the Admin can update any necessary information.
Actors:	Admin
Relevant Use Case:	Admin login
Stakeholders:	Admin, Students, Doctors
Pre-Conditions:	The Admin is logged in and has access to the management features of the system.
Post-Conditions:	A new user is added to the system, or the details of an existing user are updated
Flow of Activates:	➢ Admin logs into the system. ➢ Admin accesses the user management module. ➢ Admin chooses to add a new user or update an existing one. ➢ Admin enters or updates the required details and saves them to the system.
Exception Conditions:	If the required information for a new user is not provided or if the system encounters an error when saving the updated details.

Use Case Name:	Management Doctor Profiles
Description:	The Admin manages doctor profiles, including creating, updating, and deactivating or deleting profiles.
Scenario:	A new doctor needs to be added to the system, or an existing doctor's profile needs to be updated or deactivated.
Event Launch:	This event is triggered when a new doctor is to be added to the system or when an existing doctor's profile needs updating or deactivating.
Brief Description:	The Admin adds a new doctor profile to the system or updates an existing one. In some cases, the Admin might need to deactivate a profile.
Actors:	Admin
Relevant Use Case:	Admin login
Stakeholders:	Admin, Doctors
Pre-Conditions:	The Admin is logged in and has access to the management features of the system.
Post-Conditions:	A new doctor profile is added to the system, or the details of an existing profile are updated or deactivated.
Flow of Activates:	➢ Admin logs into the system. ➢ Admin accesses the doctor management module. ➢ Admin chooses to add a new doctor, update an existing one, or deactivate a profile. ➢ Admin enters or updates the required details and saves them to the system.
Exception Conditions:	If the required information for a new doctor profile is not provided or if the system encounters an error when saving the updated details.

Use Case Name:	Review and Approve Announcements
Description:	The Admin reviews and approves or rejects the announcement proposals made by the Doctors.
Scenario:	There is an announcement proposal pending from a Doctor.
Event Launch:	This event is triggered when a Doctor proposes a new announcement.
Brief Description:	The Admin reviews the proposed announcements and either approves or rejects them.
Actors:	Admin
Relevant Use Case:	Admin Login
Stakeholders:	Admin, Students, Doctors
Pre-Conditions:	The Admin is logged in and there are announcement proposals pending.
Post-Conditions:	The proposed announcement has either been approved and published on the system, or rejected.
Flow of Activates:	➢ Admin logs into the system. ➢ Admin accesses the review announcements section. ➢ Admin reviews each proposed announcement and decides to approve or reject it.
Exception Conditions:	If there are no announcement proposals pending or if the system encountered an error when trying to approve or reject the announcement.

Use Case Name:	Monitor System Functionality
Description:	The Admin monitors the system's performance, adjusts system settings, and performs necessary maintenance tasks.
Scenario:	The Admin needs to monitor the system's performance, adjust settings, or perform maintenance tasks.
Event Launch:	This event is triggered as part of the Admins routine tasks or when an issue with the system's performance is detected.
Brief Description:	The Admin monitors the system's performance, makes necessary adjustments to system settings, and performs maintenance tasks as needed.
Actors:	Admin
Relevant Use Case:	Admin login
Stakeholders:	Admin, Students, Doctors
Pre-Conditions:	The Admin is logged in and has access to the system's performance monitoring and settings adjustment features.
Post-Conditions:	The system's performance is optimized, settings are adjusted, and maintenance tasks are performed as needed.
Flow of Activates:	➢ Admin logs into the system. ➢ Admin accesses the system's performance monitoring module. ➢ Admin monitors the system's performance, adjusts system settings, and performs maintenance tasks as needed.
Exception Conditions:	If the system encounters an error when monitoring performance, adjusting settings, or performing maintenance tasks.

Use Case Name:	Review and Approve Photo Proposals
Description:	The Admin reviews and approves or rejects the photo proposals made by the Doctors.
Scenario:	There is a photo proposal pending from a Doctor.
Event Launch:	This event is triggered when a Doctor proposes a new photo or document.
Brief Description:	The Admin reviews the proposed photos and documents and either approves or rejects them.
Actors:	Admin
Relevant Use Case:	Admin login
Stakeholders:	Admin, Doctors, Students
Pre-Conditions:	The Admin is logged in and there are photo proposals pending.
Post-Conditions:	The proposed photo or document has either been approved and added to the archive, or rejected.
Flow of Activates:	➢ Admin logs into the system. ➢ Admin accesses the review photo proposals section. ➢ Admin reviews each proposed photo or document and decides to approve or reject it
Exception Conditions:	If there are no photo proposals pending or if the system encountered an error when trying to approve or reject the proposal.

5.3.2 Student

Use Case Name:	Book Appointments
Description:	The student requests appointments with their preferred doctor by selecting an available time slot and providing relevant details.
Scenario:	The student wishes to book an appointment with a doctor.
Event Launch:	This event is initiated when a student logs in and decides to book an appointment.
Brief Description:	The student logs in, searches for a doctor, checks their availability, and books an appointment.
Actors:	Student
Relevant Use Case:	Student login, Browse and search doctors, View Doctor Availability and Schedule
Stakeholders:	Student, Doctor, Admin
Pre-Conditions:	The student is logged in and has access to the doctor's profiles and their schedules.
Post-Conditions:	An appointment is booked, and the doctor receives a notification about the appointment.
Flow of Activates:	➢ Student logs into the system. ➢ Student browses and searches for doctors. ➢ Student views the doctor's availability and schedule. ➢ Student selects an available time slot and provides necessary details for the appointment. ➢ Student submits the appointment request.
Exception Conditions:	If the system encounters an error when booking the appointment or if the selected time slot is not available.

Use Case Name:	Access and Utilize Photo Archive
Description:	The student can access the photo archive and browse through the collection of academic photos or important documents.
Scenario:	The student wishes to access the photo archive for academic activities, research, or presentations.
Event Launch:	This event is initiated when a student logs in and decides to access the photo archive.
Brief Description:	The student logs in, accesses the photo archive, and utilizes the photos and documents as needed.
Actors:	Student
Relevant Use Case:	Student login
Stakeholders:	Student, Admin
Pre-Conditions:	The student is logged in and has access to the photo archive.
Post-Conditions:	The student successfully retrieves and utilizes the required photos or documents from the archive.
Flow of Activates:	➢ Student logs into the system. ➢ Student accesses the photo archive. ➢ Student browses through the photos or documents and selects the required items.
Exception Conditions:	If the system encounters an error when accessing the photo archive or retrieving specific items.

5.3.3 Doctor

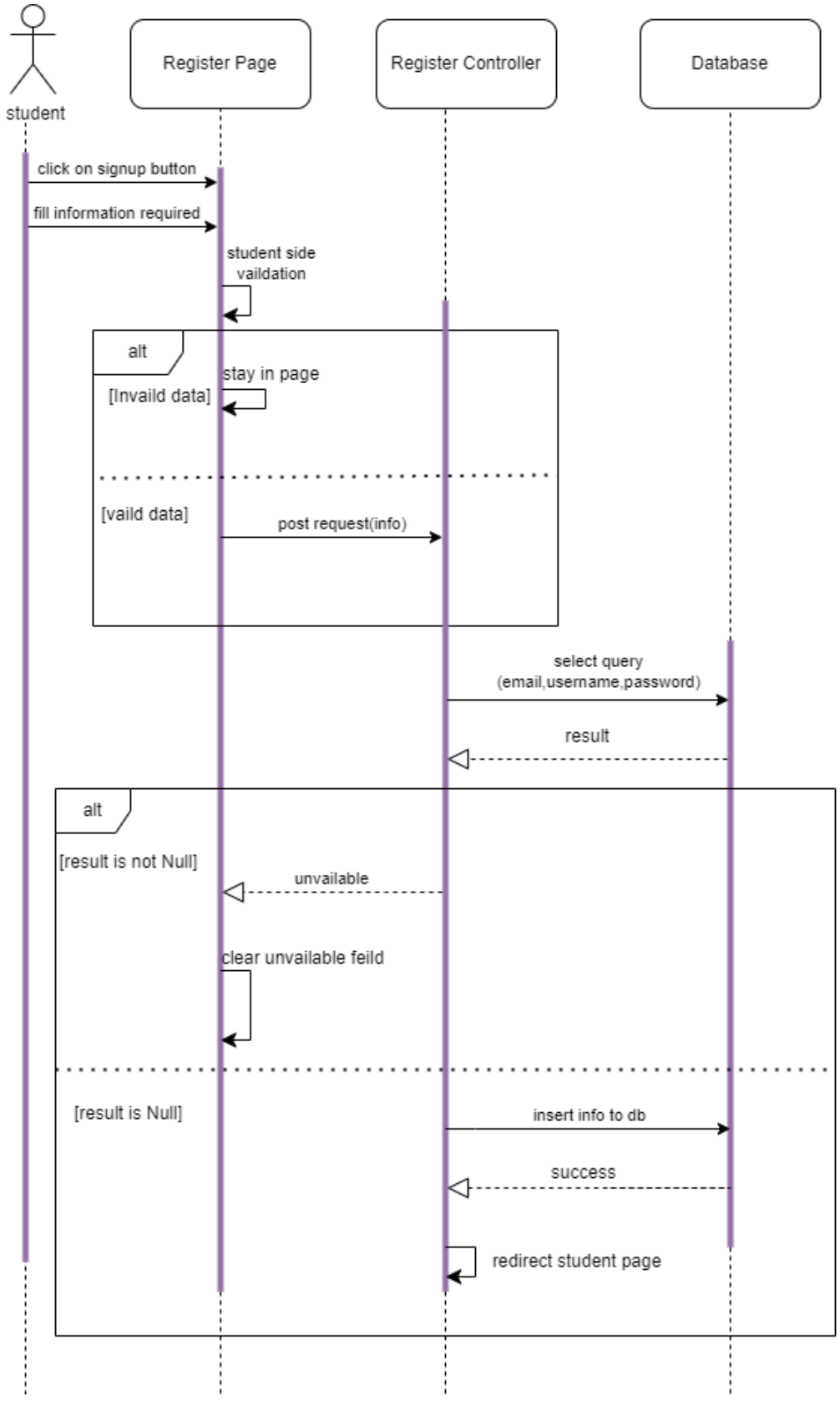
Use Case Name:	Accept or Reject Appointment Requests
Scenario:	A doctor wants to review and respond to appointment requests from students.
Event Launch:	The doctor receives a new appointment request notification
Brief Description:	The doctor reviews the details of the appointment request and decides to accept or reject it.
Actors:	Doctor, Student
Relevant Use Case:	Book Appointments (Student)
Stakeholders:	Doctor, Student, Admin
Pre-Conditions:	The doctor is logged into the system, and there is at least one appointment request to review.
Post-Conditions:	The doctor accepts or rejects the appointment request, and the student is notified of the decision.
Flow of Activates:	➢ Doctor logs into the system ➢ Doctor navigates to appointment requests ➢ System shows the details of each appointment request ➢ Doctor reviews the details of each appointment request ➢ Doctor decides to accept or reject each appointment request ➢ System notifies the student of the decision
Exception Conditions:	None.

Use Case Name:	Propose Photo Upload
Description:	The Doctor proposes a photo to be added to the archive, which is then subject to approval by the Admin.
Scenario:	The Doctor has a photo or document that they believe would be beneficial to the archive.
Event Launch:	This event is triggered when a Doctor proposes a new photo or document upload to the system.
Brief Description:	The Doctor selects a new photo or document and proposes it to the Admin for review and potential addition to the archive.
Actors:	Doctor, Admin
Relevant Use Case:	Doctor Login,Review and Approve Photo Proposals
Stakeholders:	Doctor, Student, Admin
Pre-Conditions:	The Doctor is logged in and has access to the system's photo proposal features.
Post-Conditions:	A new photo or document is proposed for the system archive and is awaiting review from the Admin.
Flow of Activates:	➢ Doctor logs into the system. ➢ Doctor accesses the photo proposal section. ➢ Doctor selects a new photo or document and submits it for review.
Exception Conditions:	If the Doctor didn't provide a valid photo or document for the proposal, or if the system encountered an error when submitting the proposal.

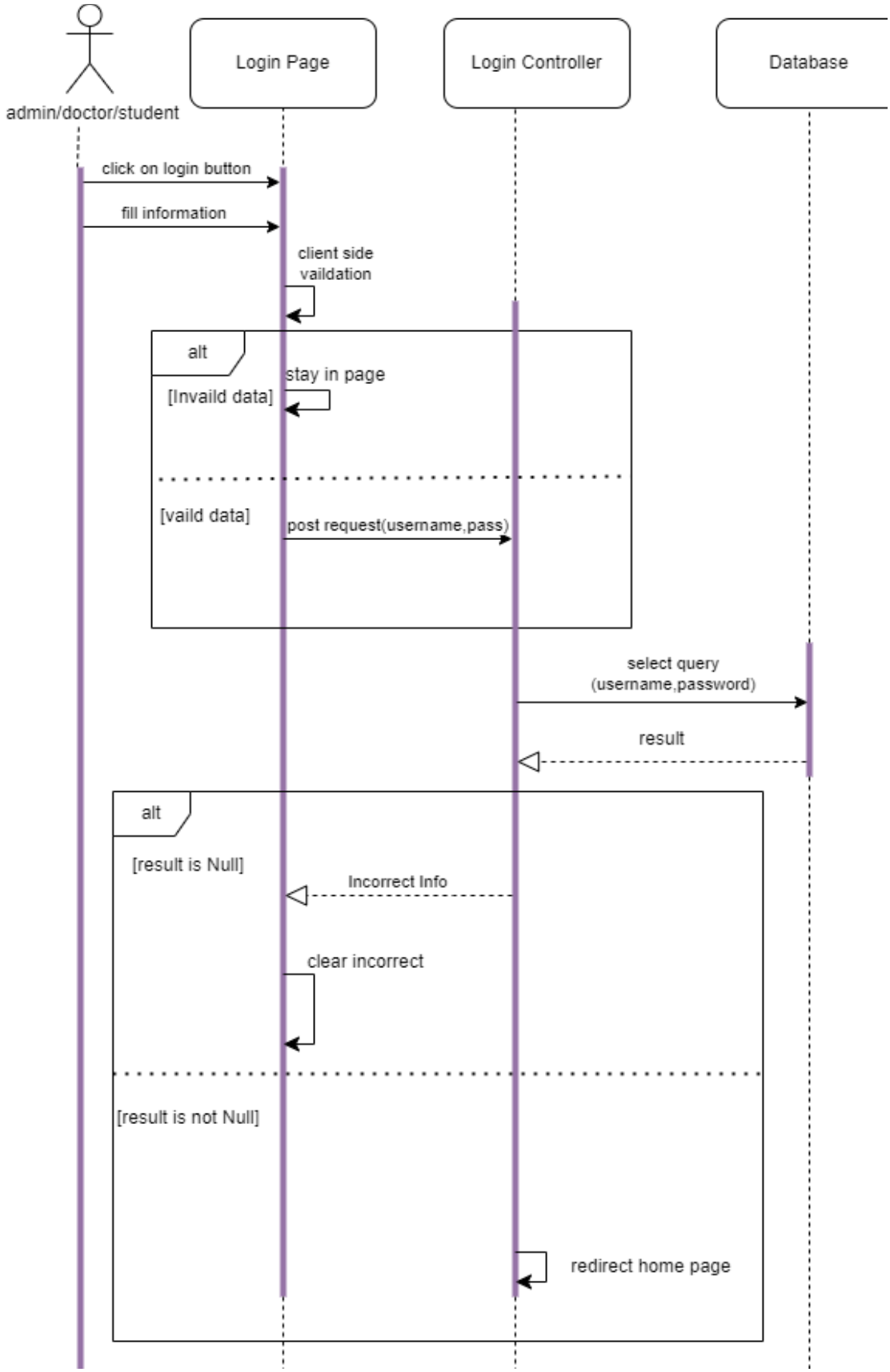
Use Case Name:	Propose Announcements
Description:	The Doctor proposes system announcements, which are then subject to approval by the Admin.
Scenario:	There is important information or an announcement that a Doctor believes should be shared with all system users.
Event Launch:	This event is triggered when a Doctor proposes a new announcement on the system.
Brief Description:	The Doctor creates a new announcement and proposes it to the Admin for review and potential publishing on the system.
Actors:	Doctor, Admin
Relevant Use Case:	Doctor Login, Review and Approve Announcements
Stakeholders:	Doctor, Admin, Students
Pre-Conditions:	The Doctor is logged in and has access to the system's announcement proposal features.
Post-Conditions:	A new announcement is proposed in the system and is awaiting review from the Admin.
Flow of Activates:	➢ Doctor logs into the system. ➢ Doctor accesses the announcement proposal section. ➢ Doctor creates a new announcement and submits it for review.
Exception Conditions:	If the Doctor didn't provide the required information for a new announcement or if the system encountered an error when submitting the announcement.

5.4 Sequence Diagrams

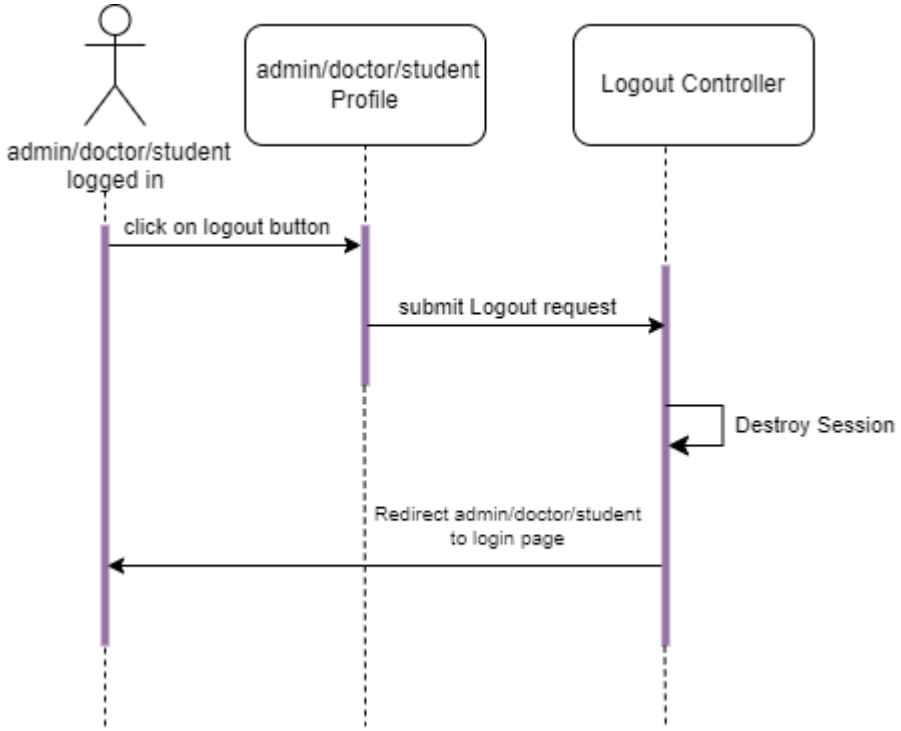
5.4.1 Student Registration sequence diagram



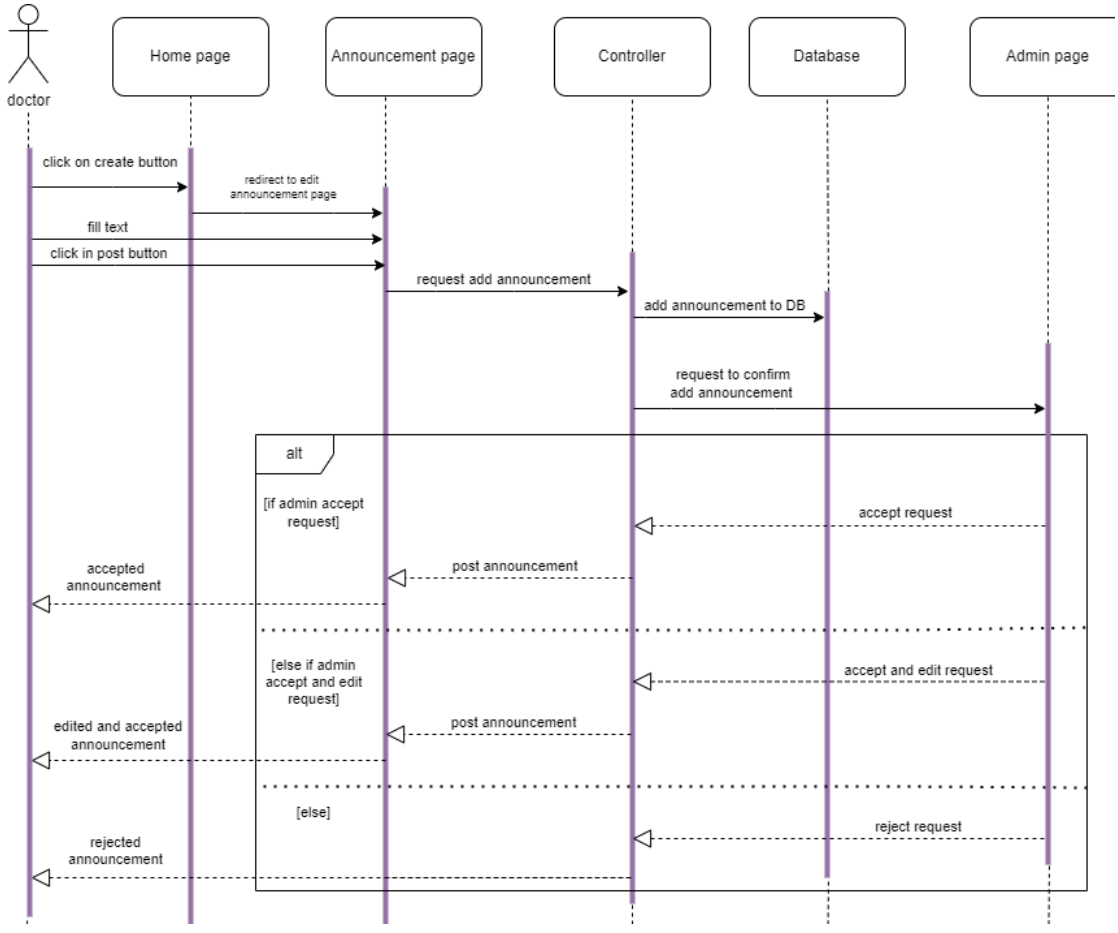
5.4.2 Login sequence diagram



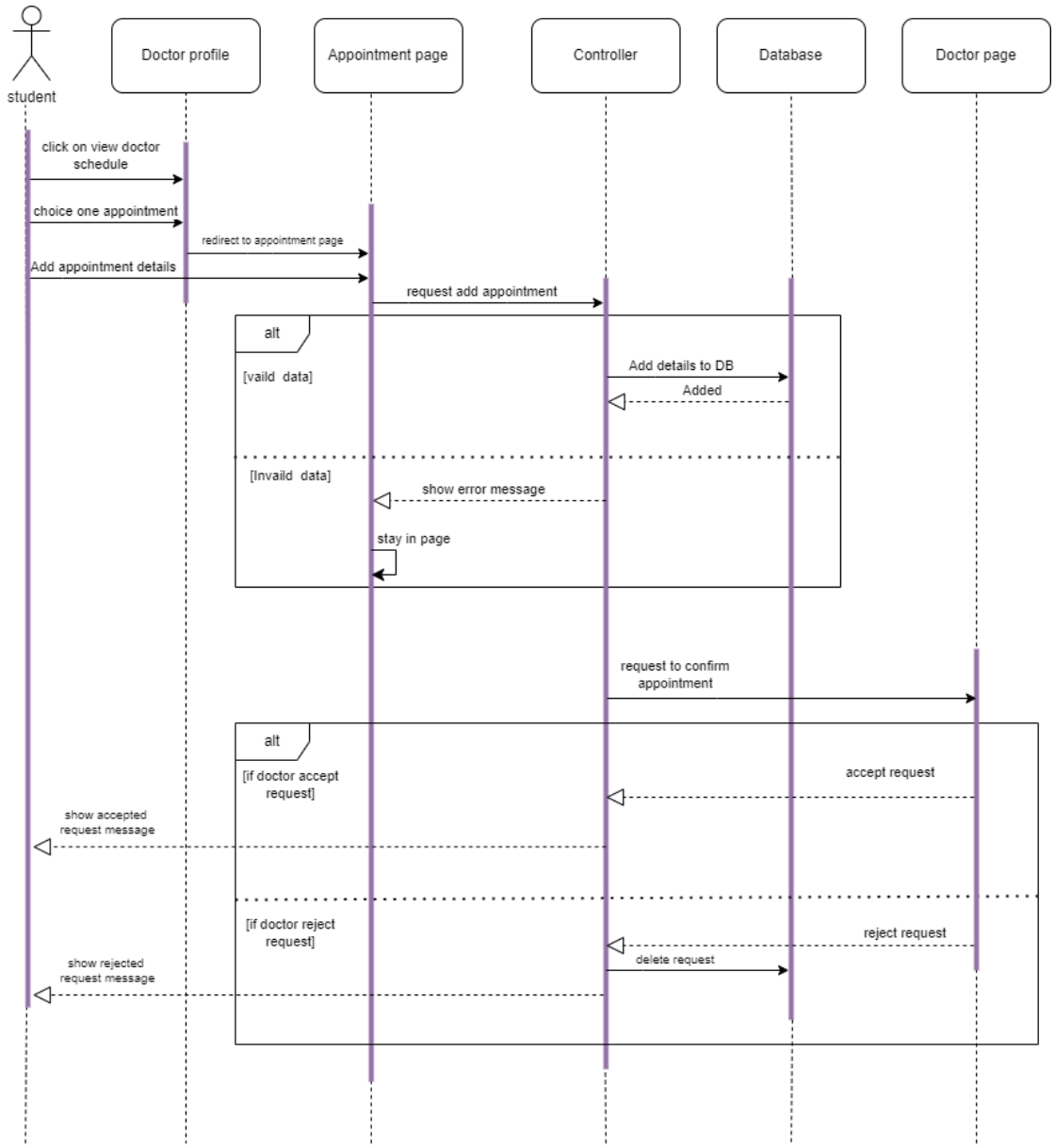
5.4.3 Logout sequence diagram



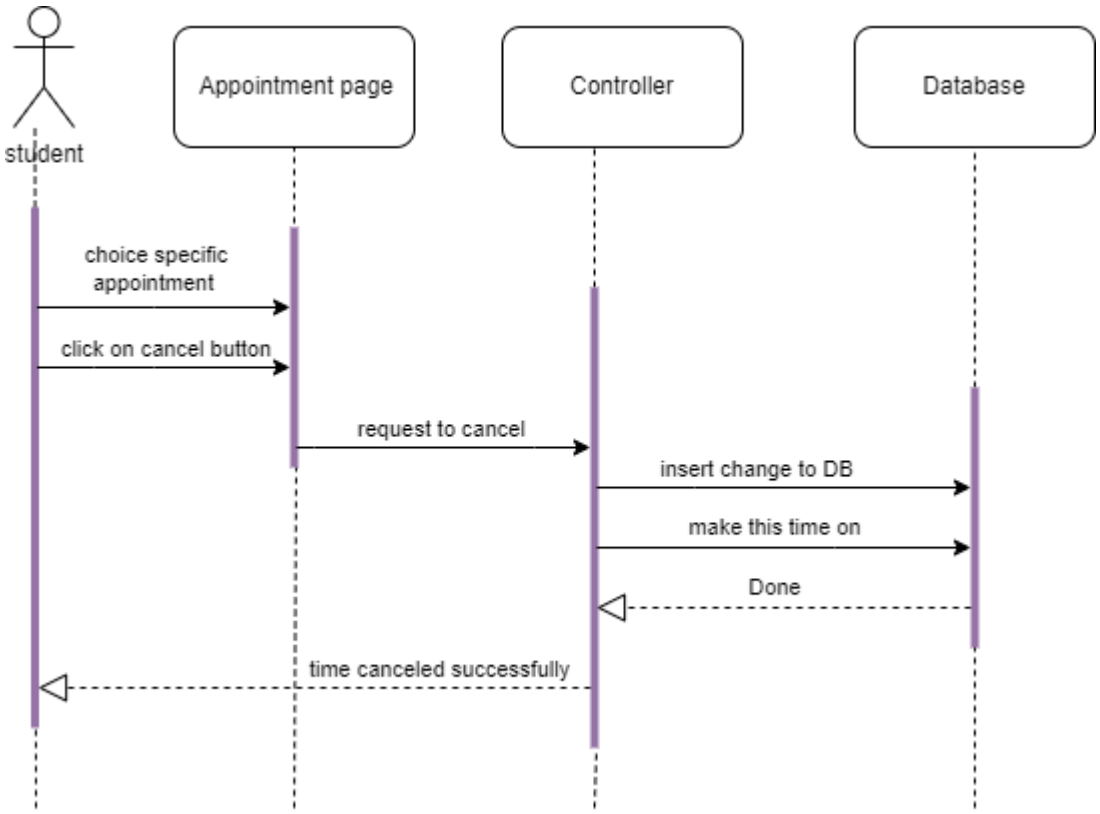
5.4.4 Announcement sequence diagram



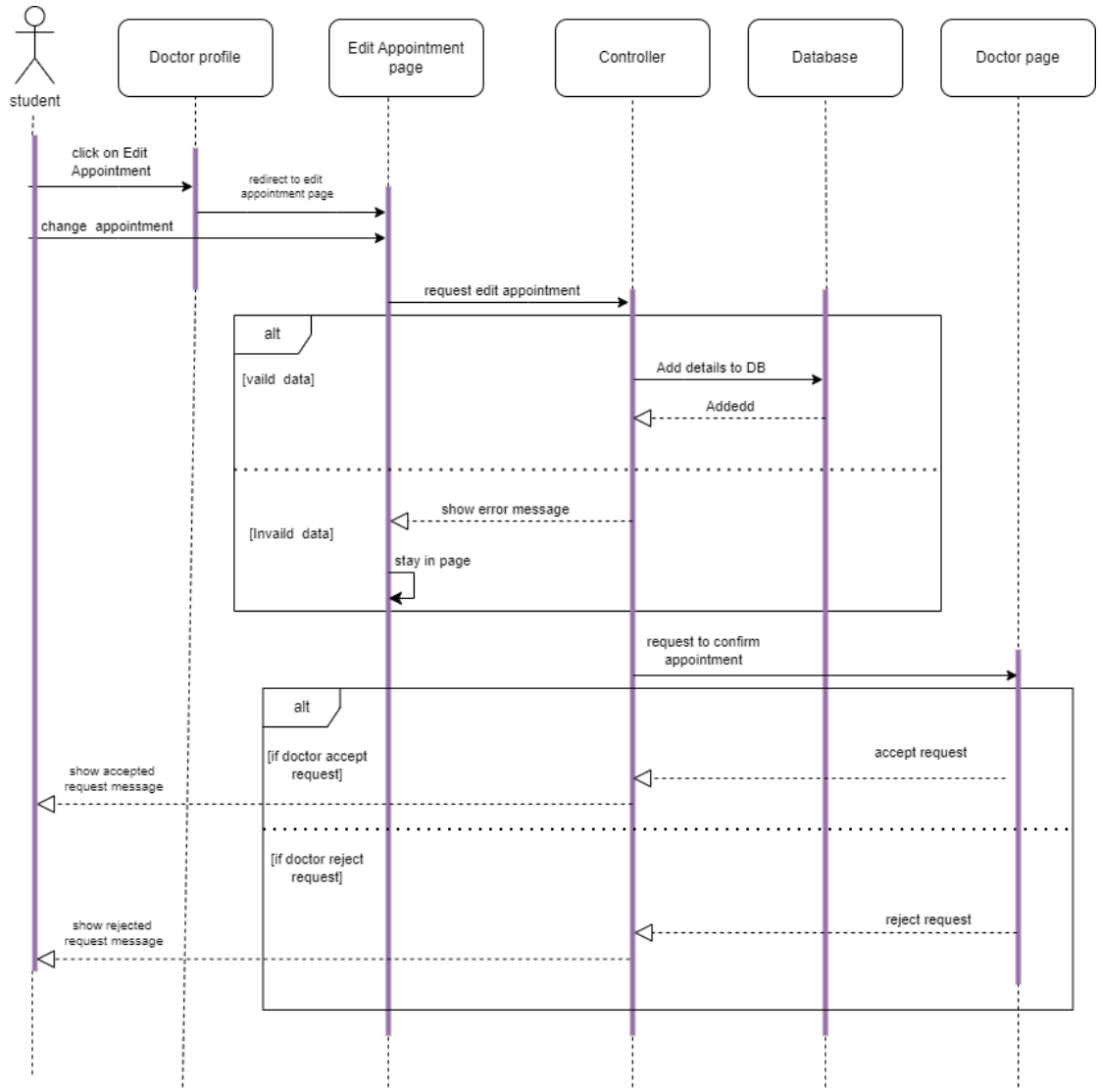
5.4.5 Appointments sequence diagram



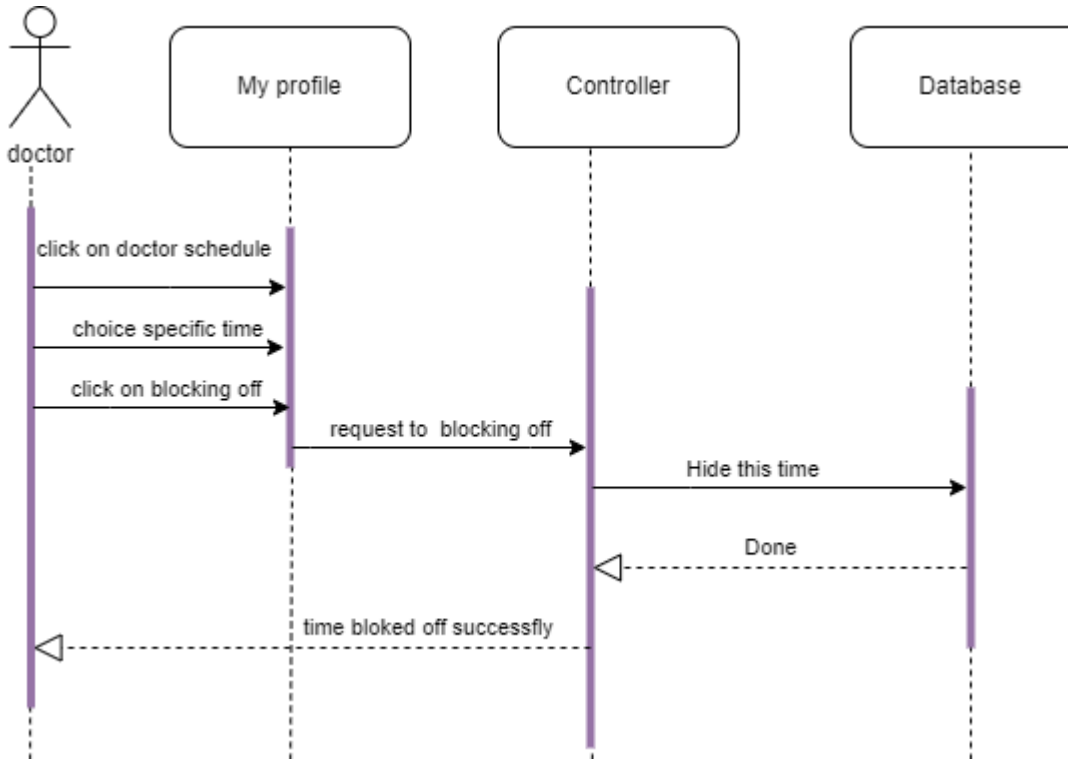
5.4.6 Cancel appointments sequence diagram



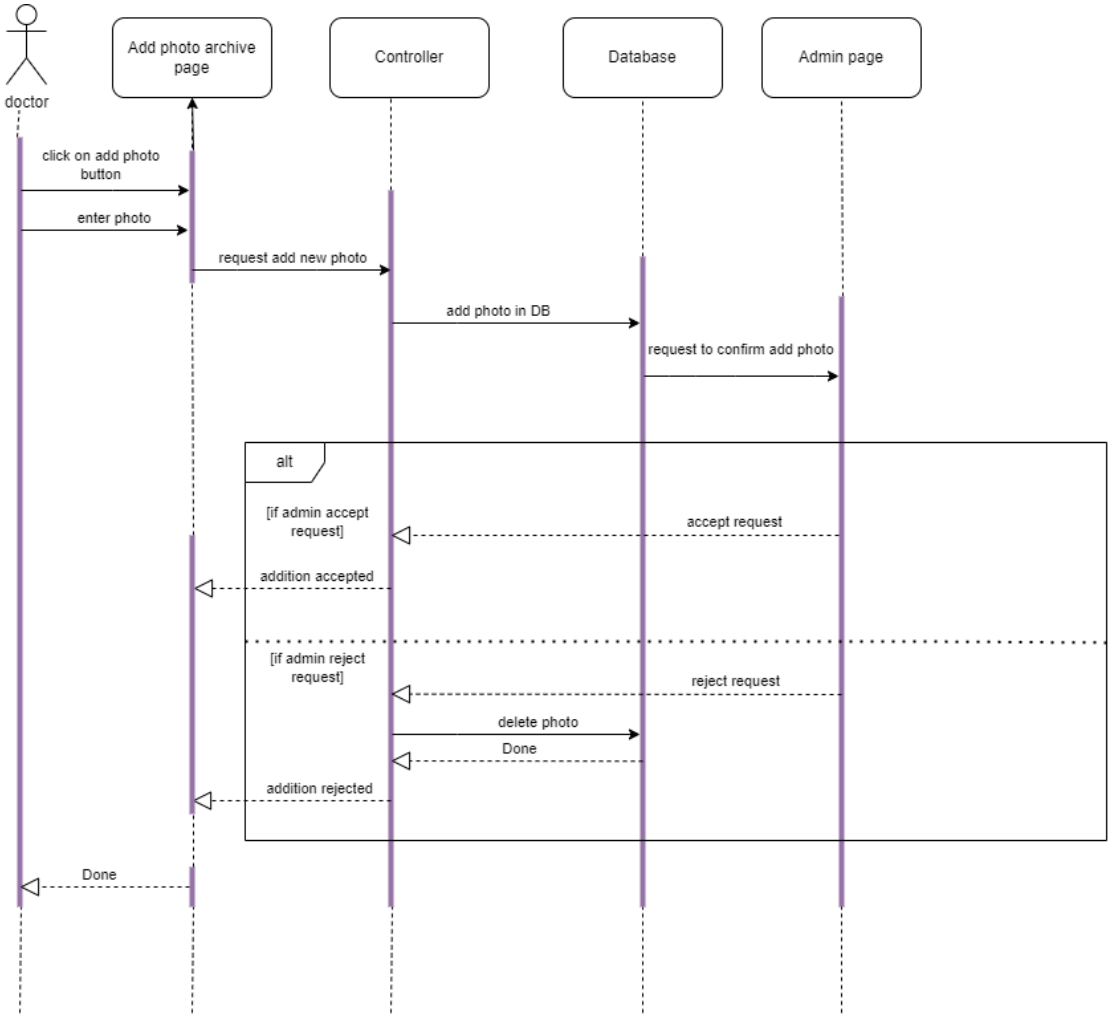
5.4.7 Reschedule appointments sequence diagram



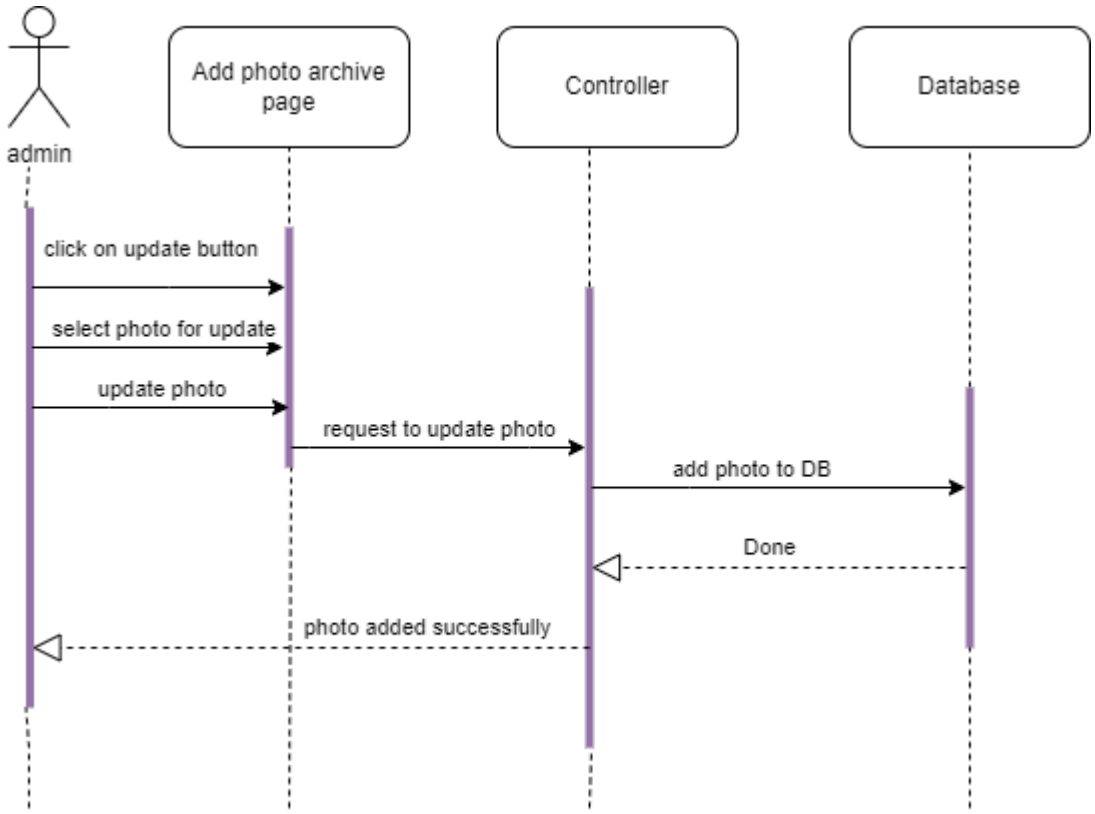
5.4.8 Blocking off specific time sequence diagram



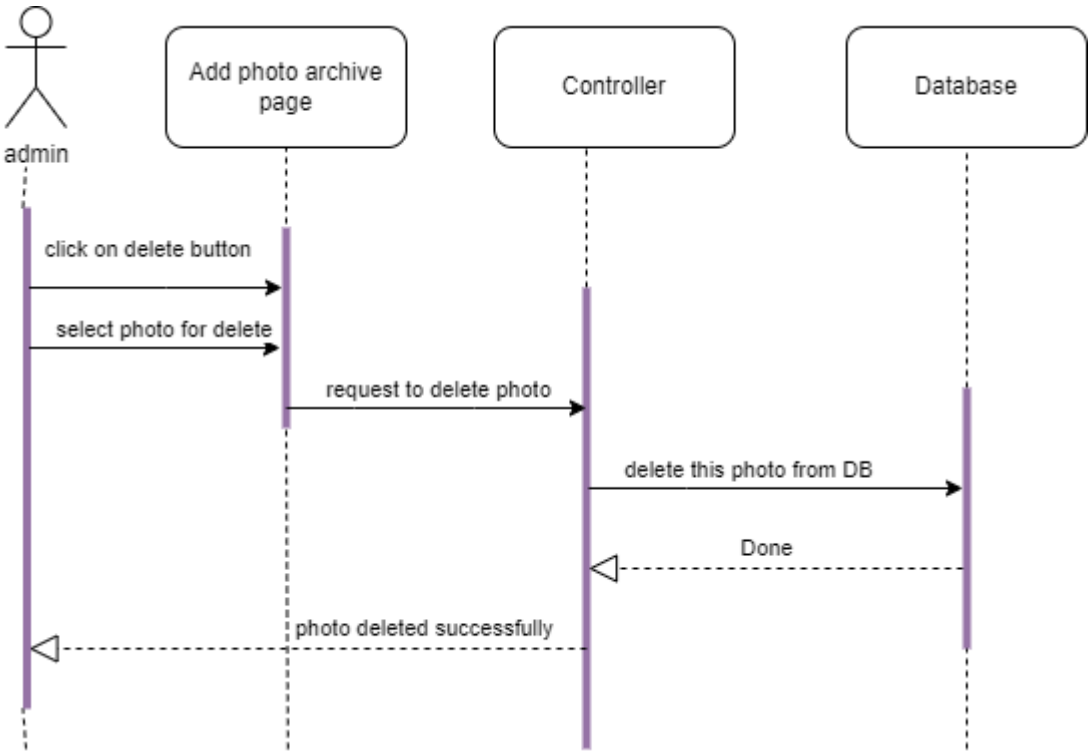
5.4.9 Add archive photo sequence diagram



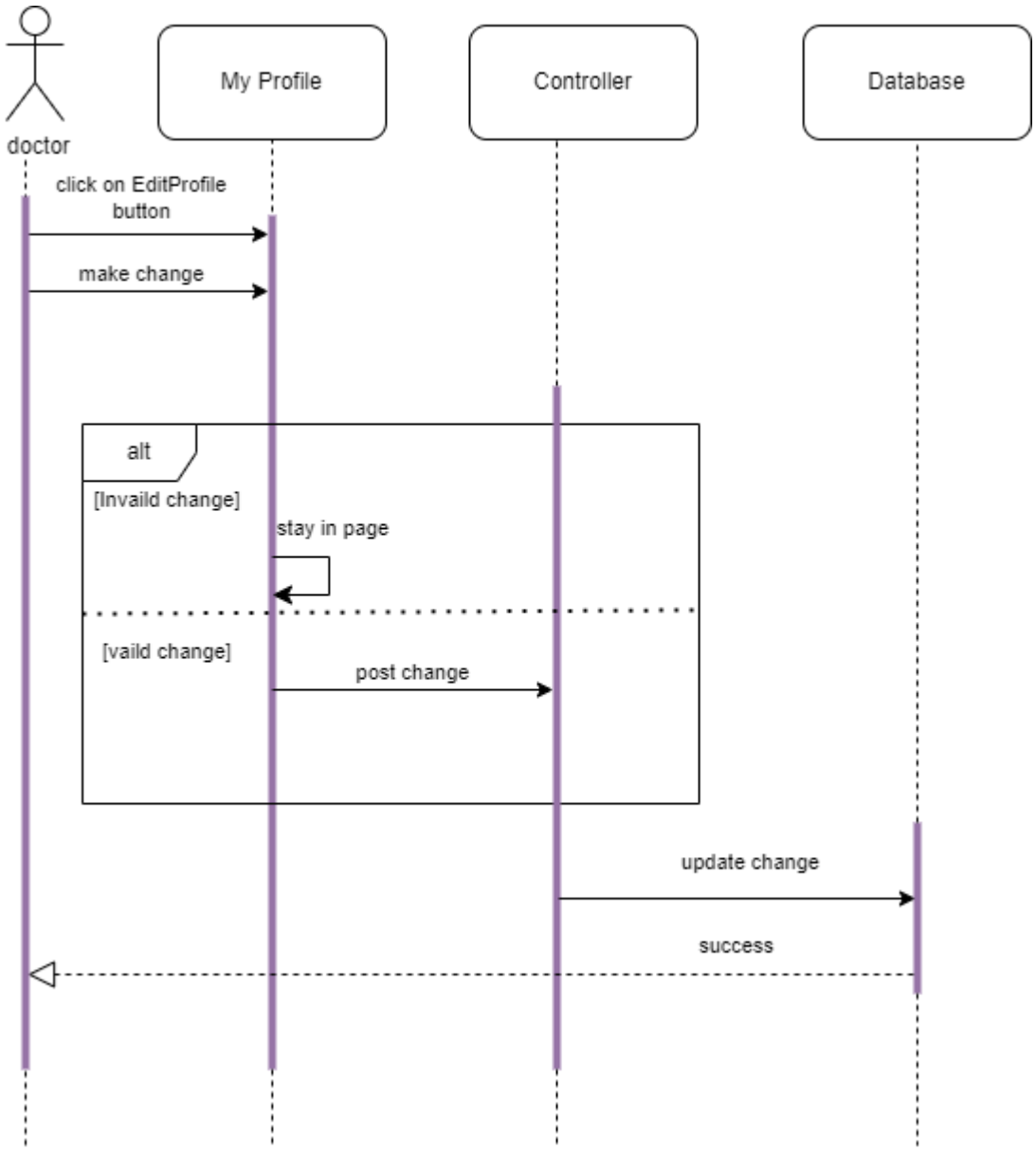
5.4.10 Update photo archive sequence diagram



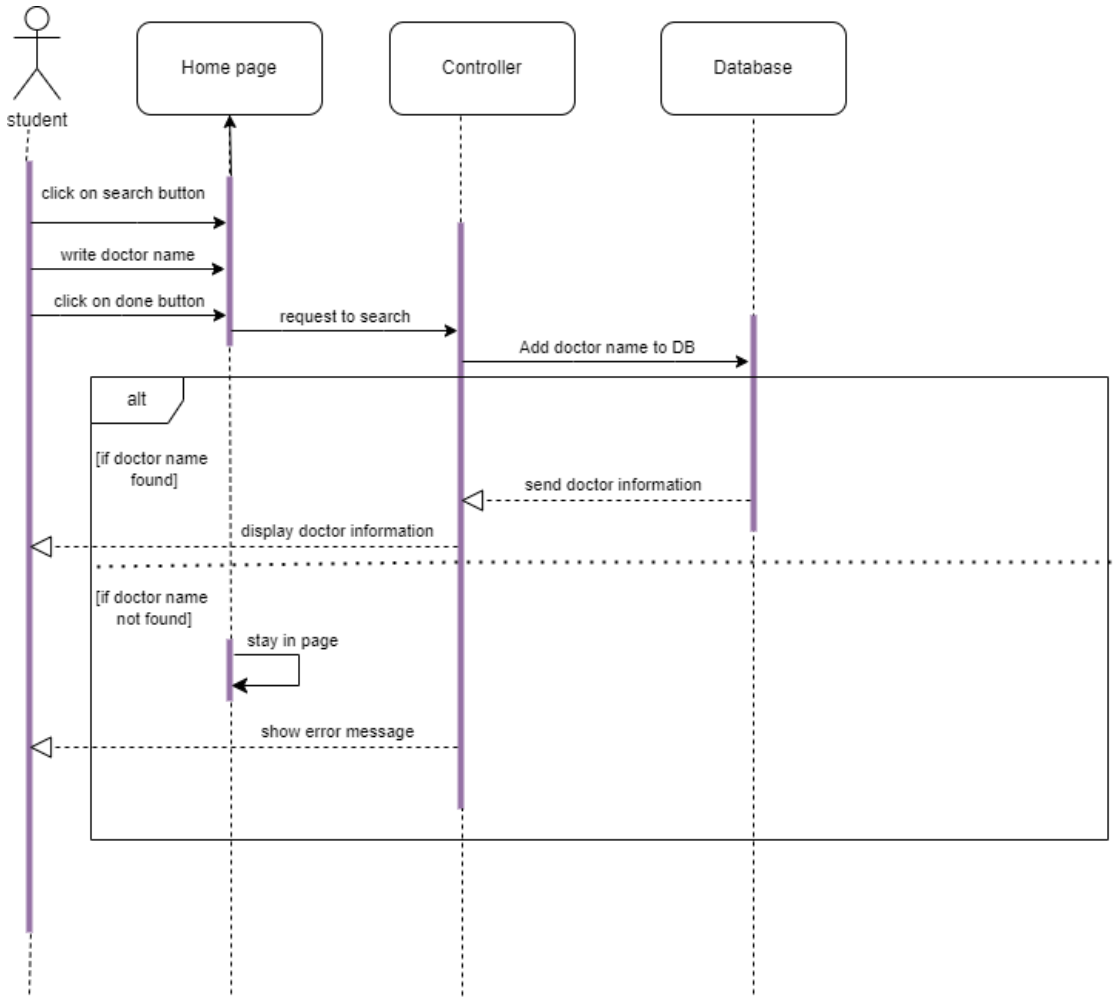
5.4.11 Delete Photo archive sequence diagram



5.4.12 Update personal doctor information sequence diagram

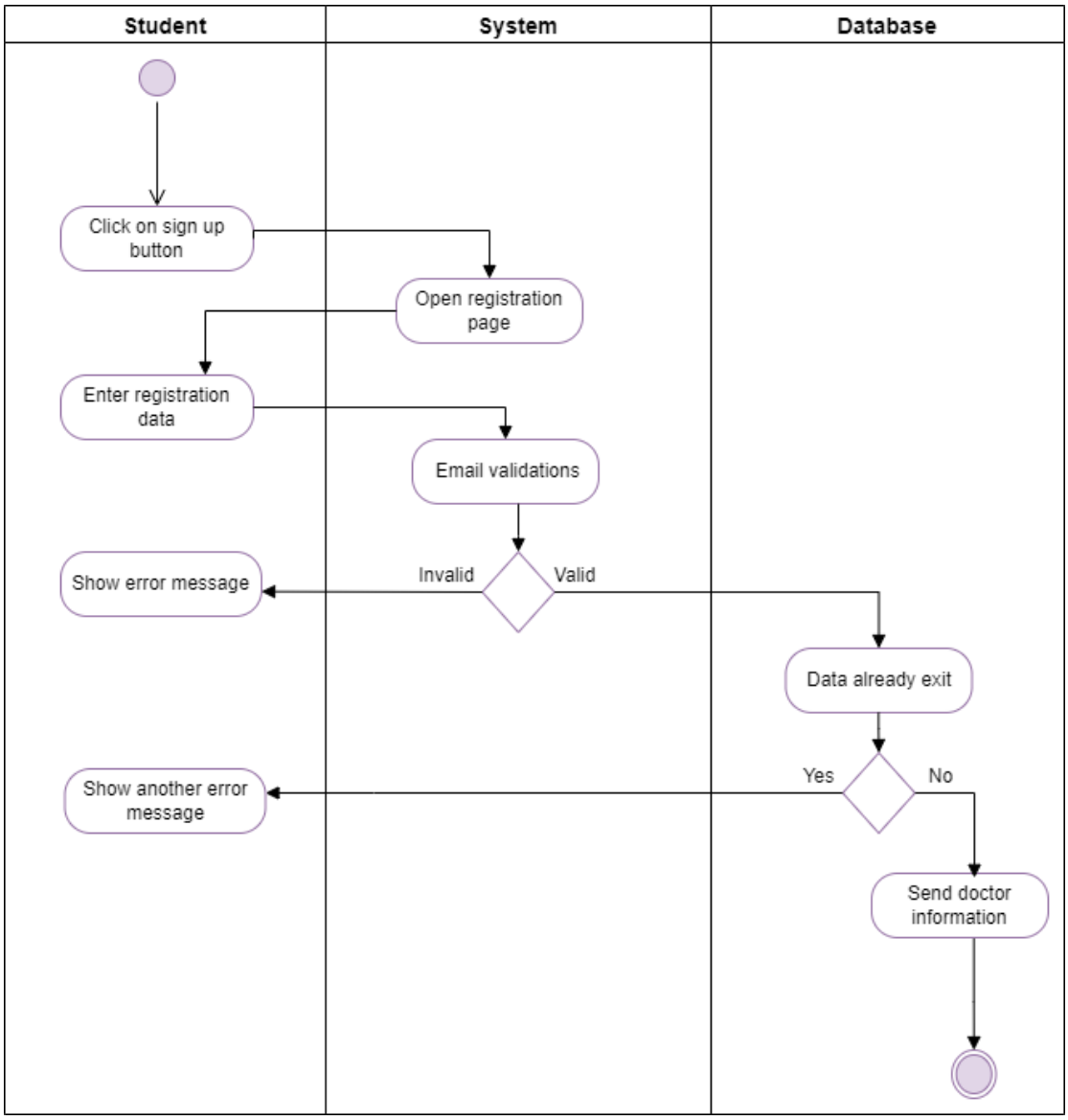


5.4.13 Browse and search doctors sequence diagram

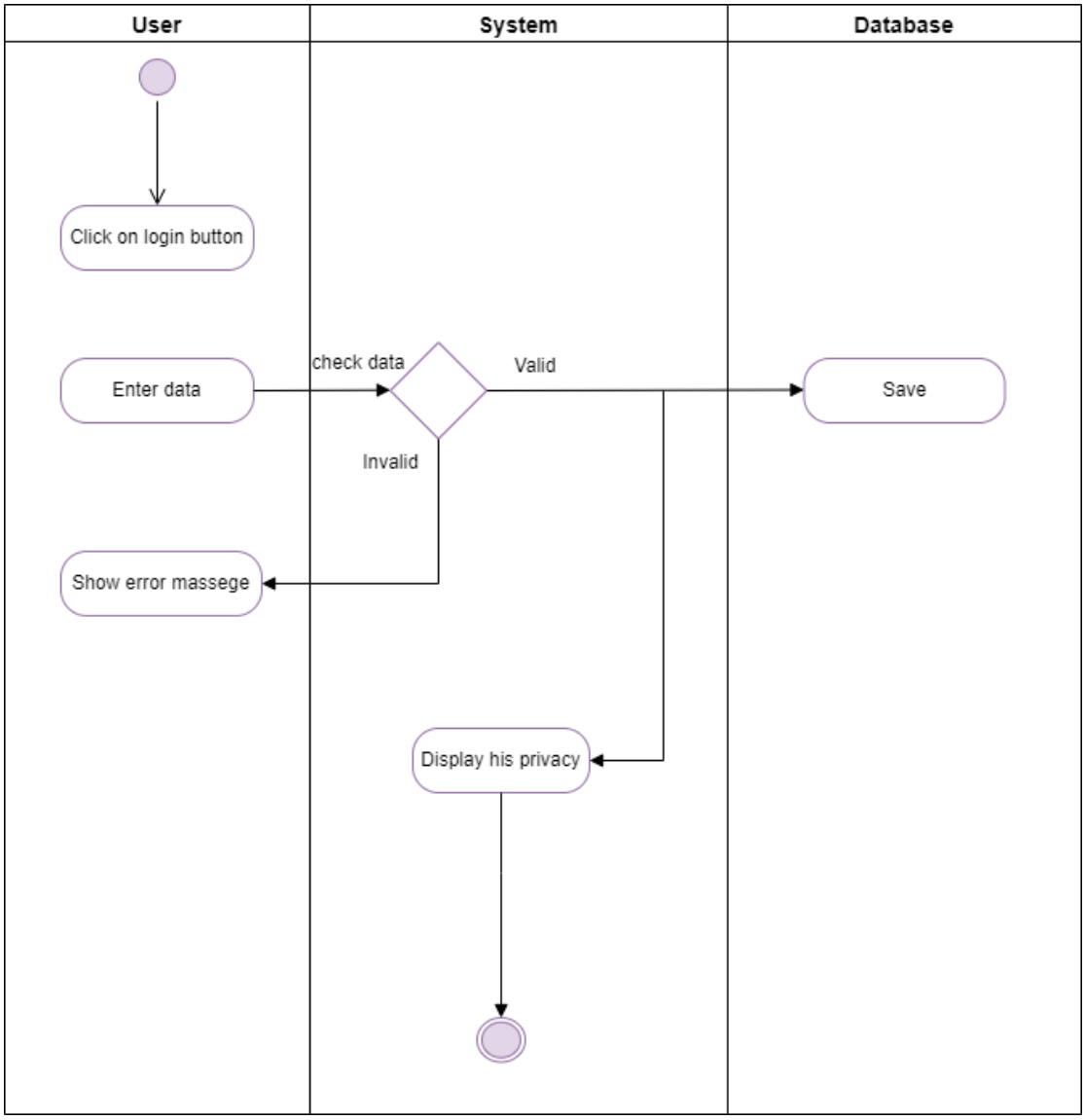


5.5 Activity Diagrams

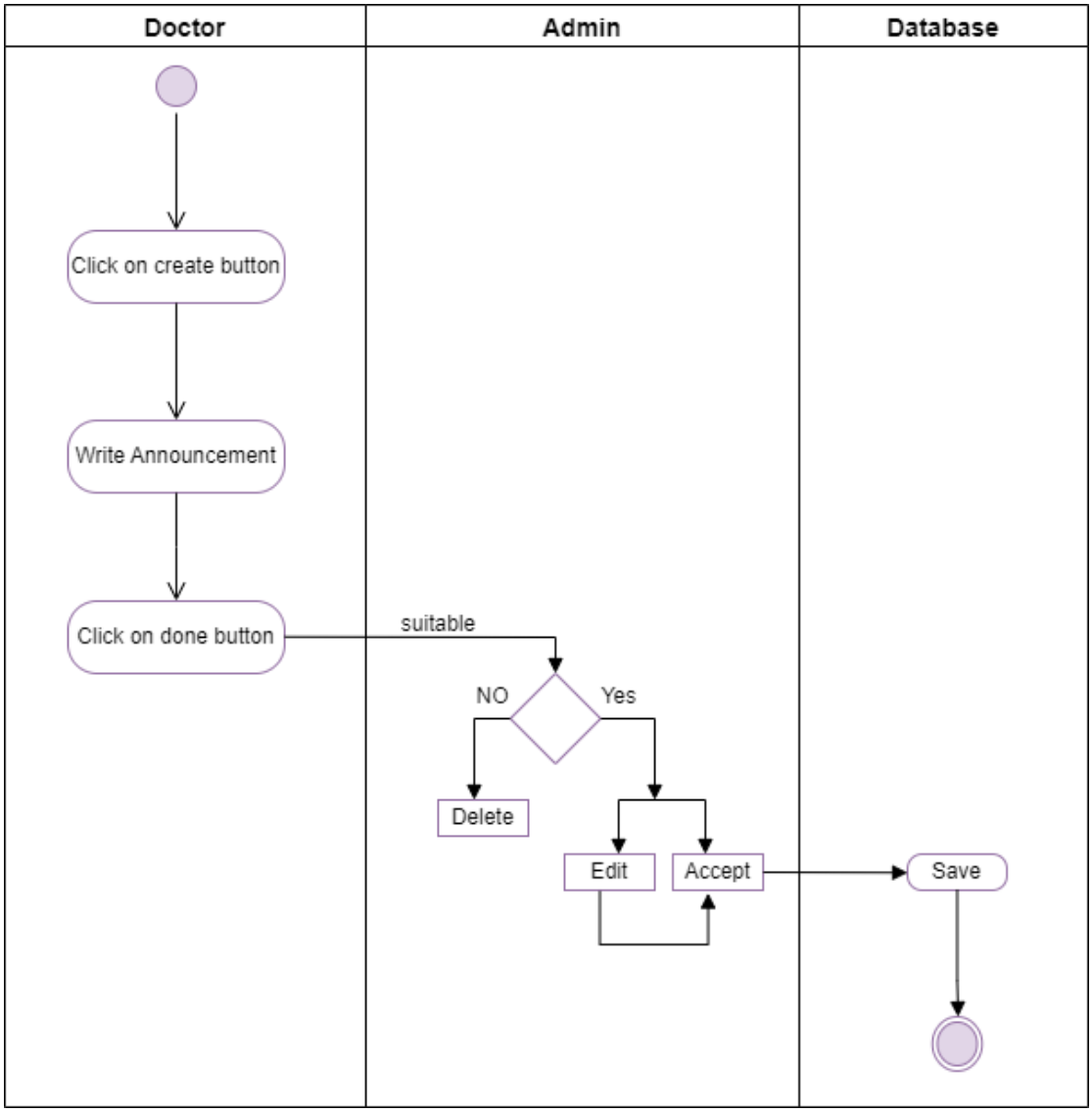
5.5.1 Student Registration activity diagram



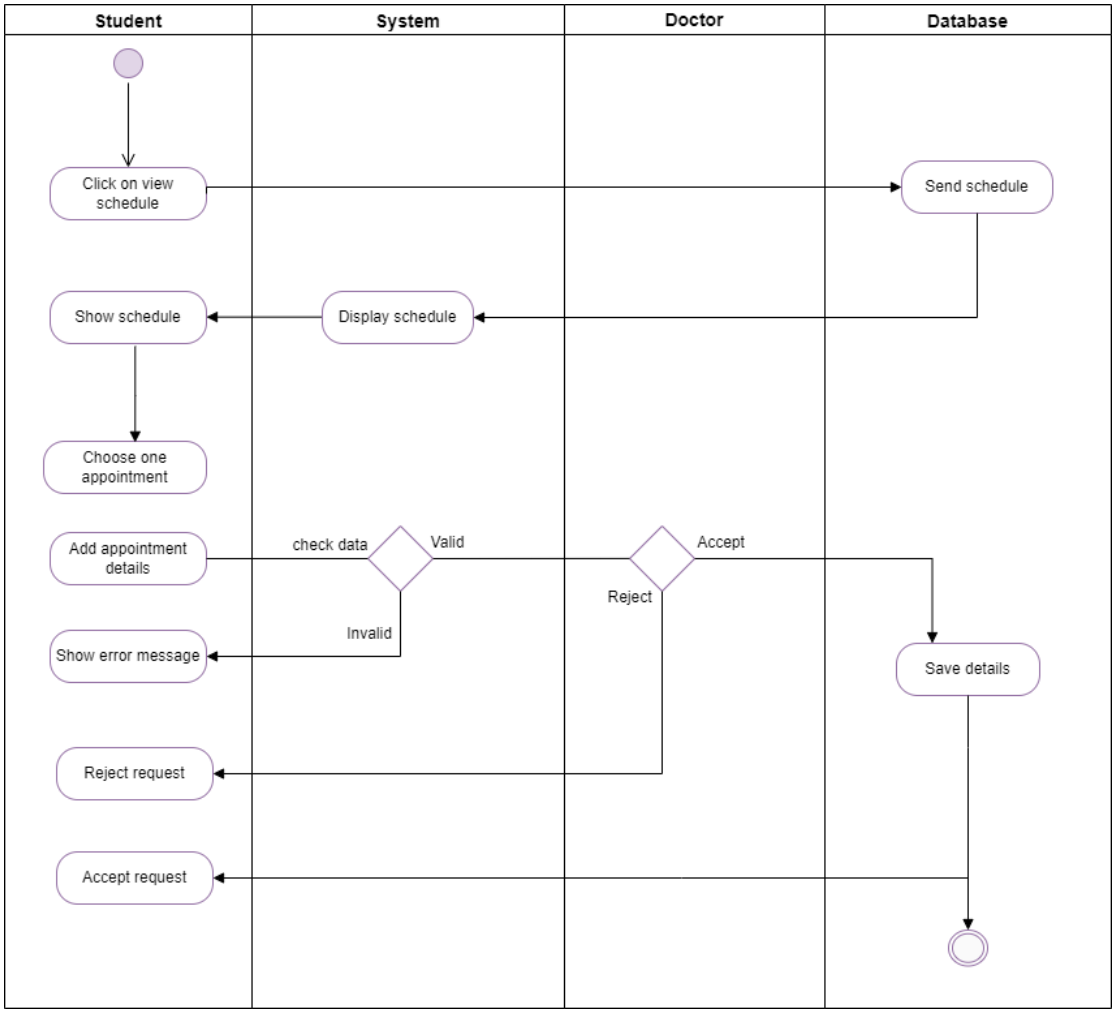
5.5.2 Login activity diagram



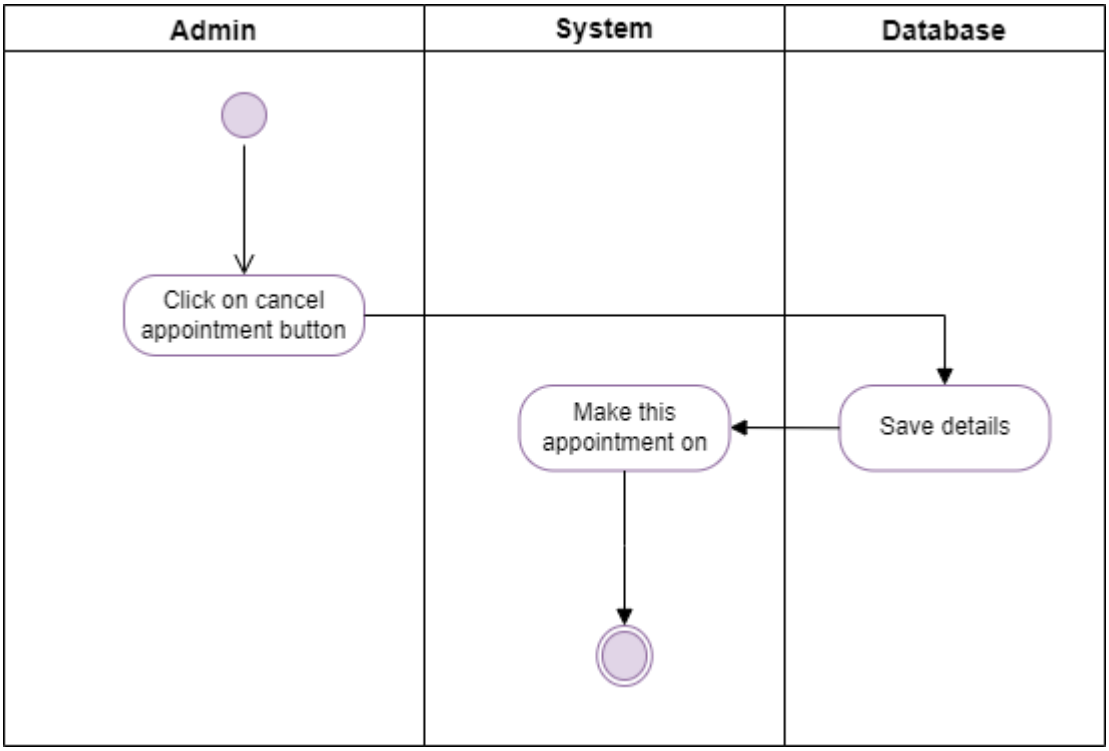
5.5.3 Announcement activity diagram



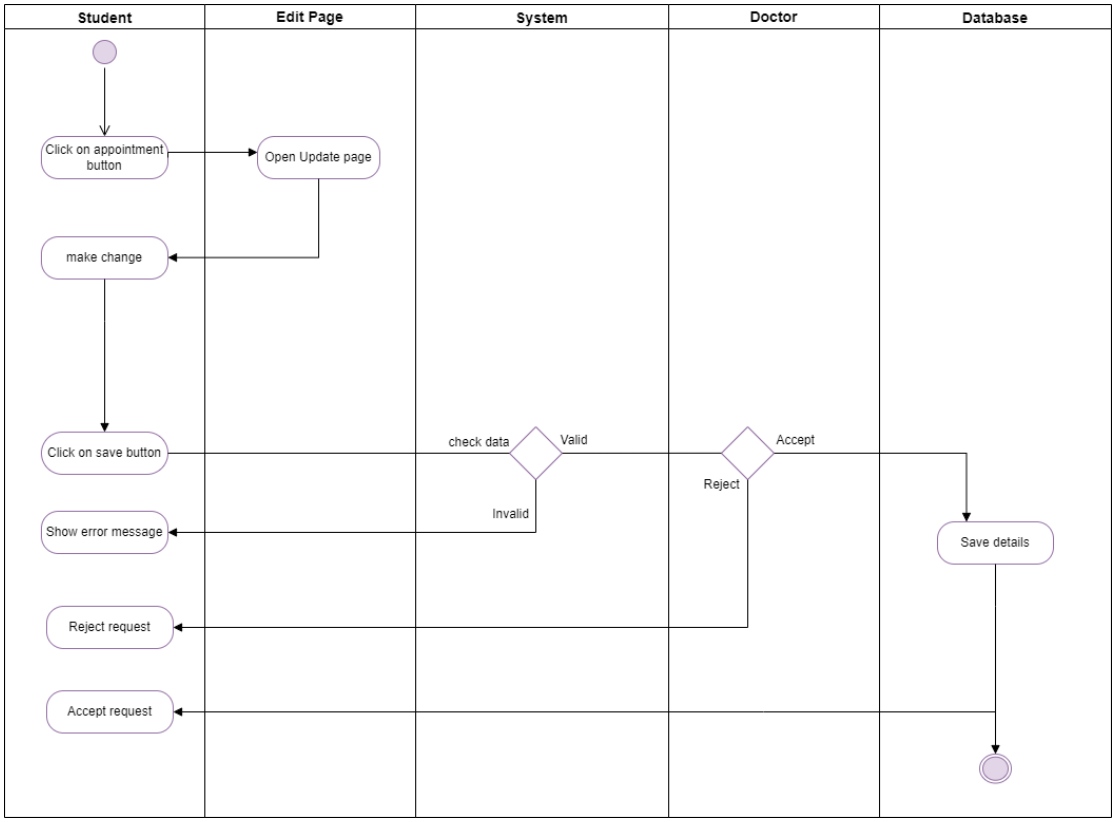
5.5.4 Appointments activity diagram



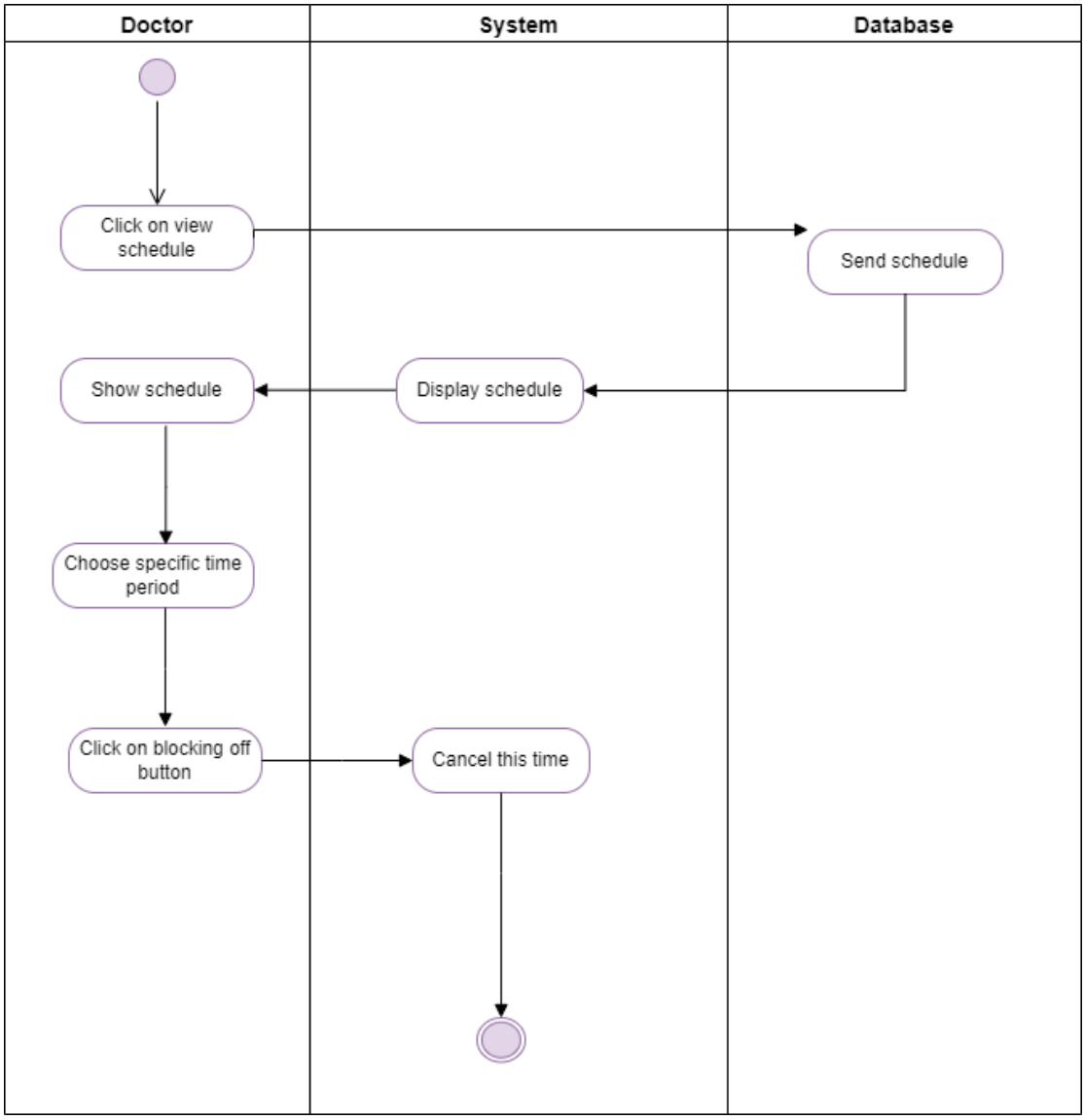
5.5.5 Cancel appointments activity diagram



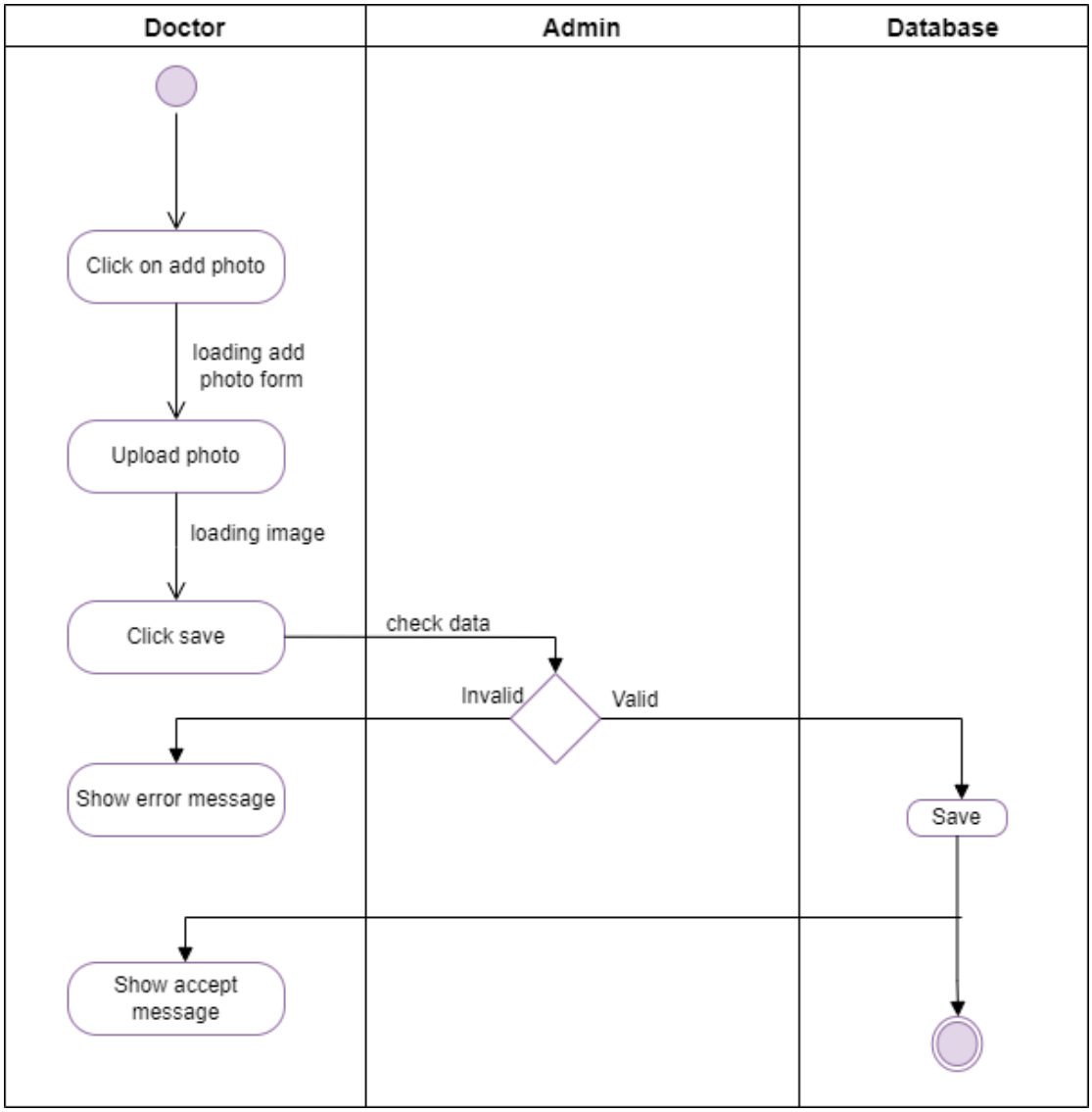
5.5.6 Reschedule appointments activity diagram



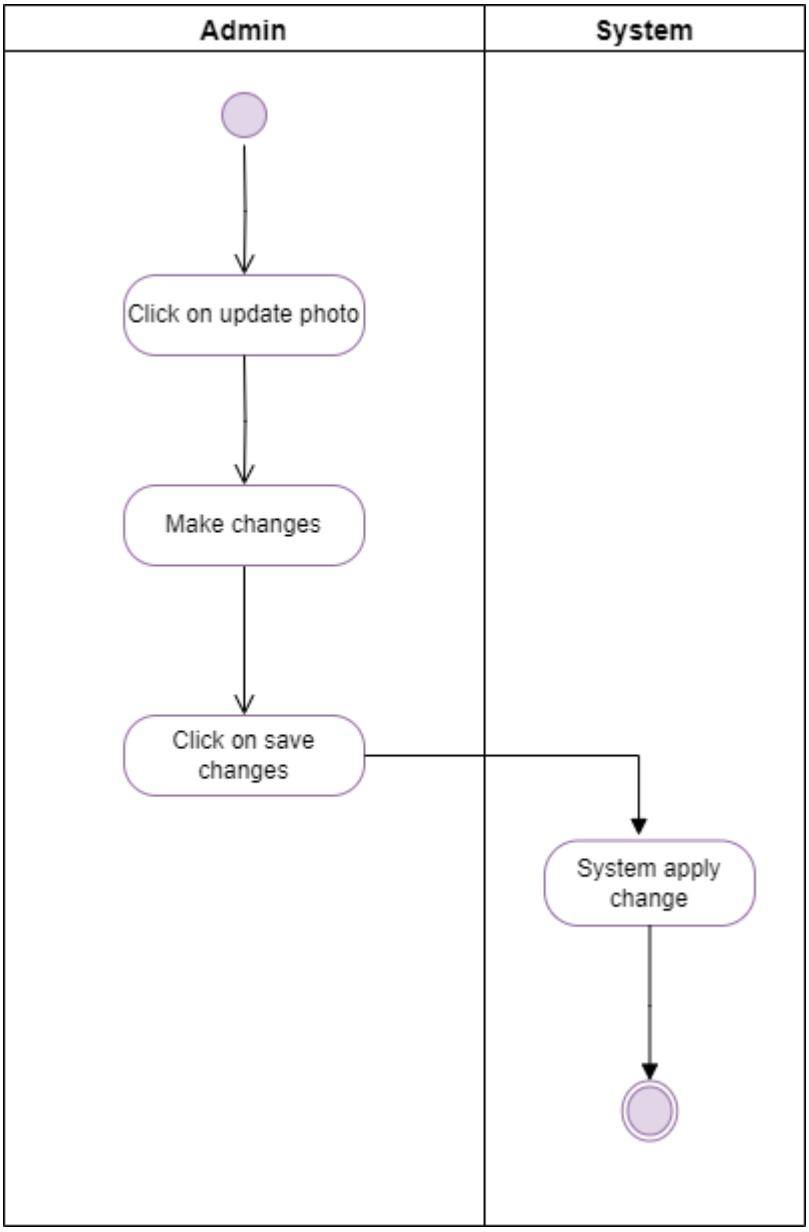
5.5.7 Blocking off specific time activity diagram



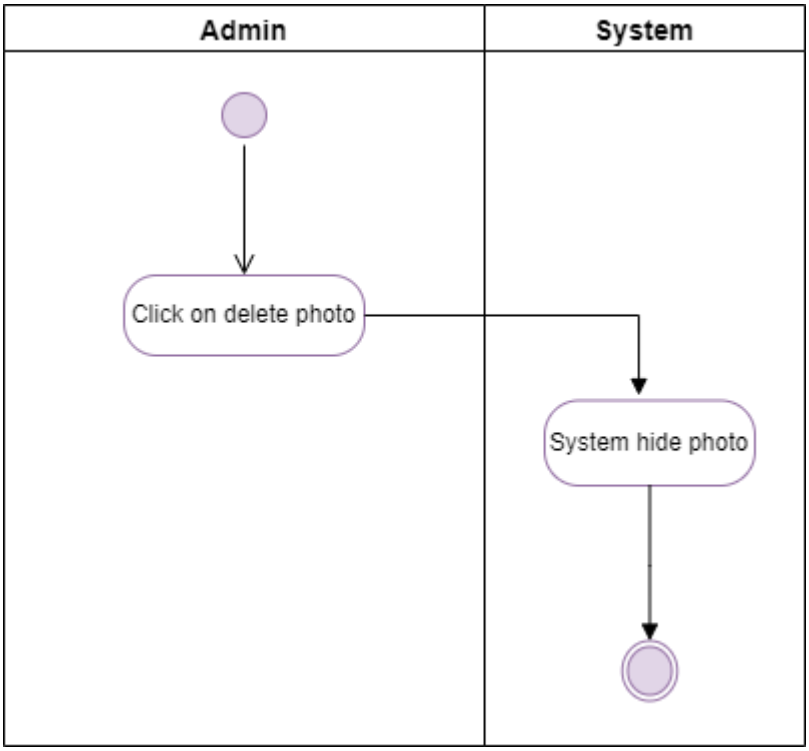
5.5.8 Add archive photo activity diagram



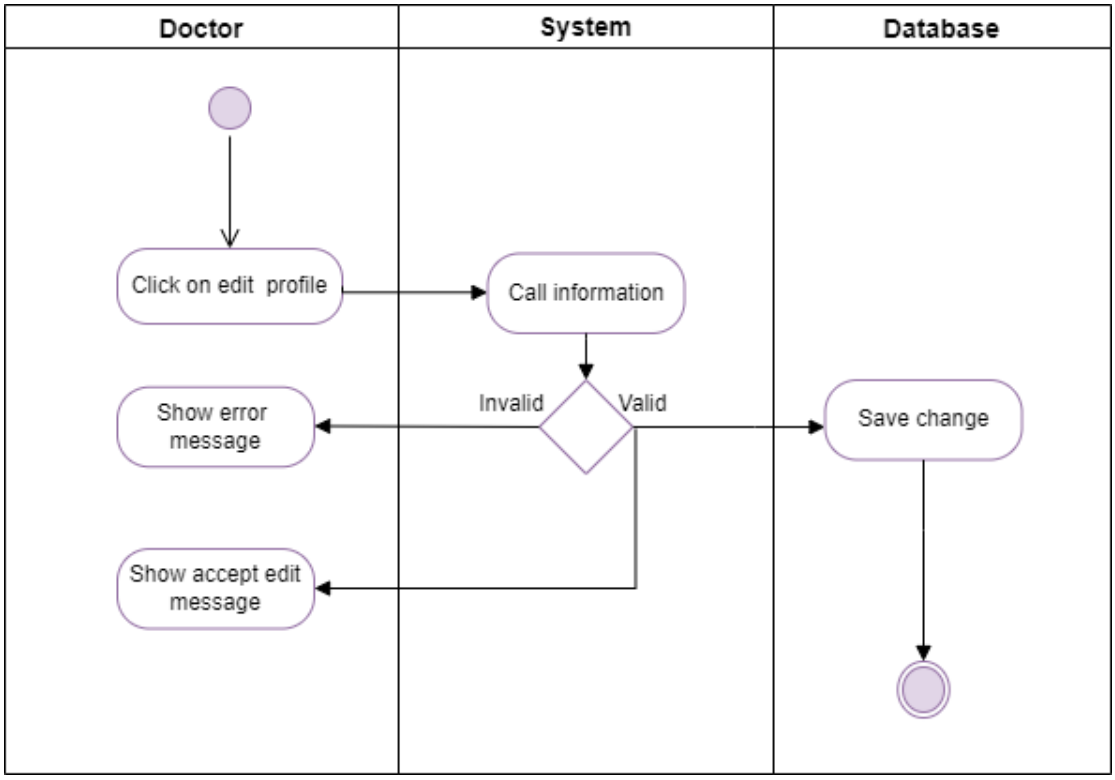
5.5.9 Update photo archive activity diagram



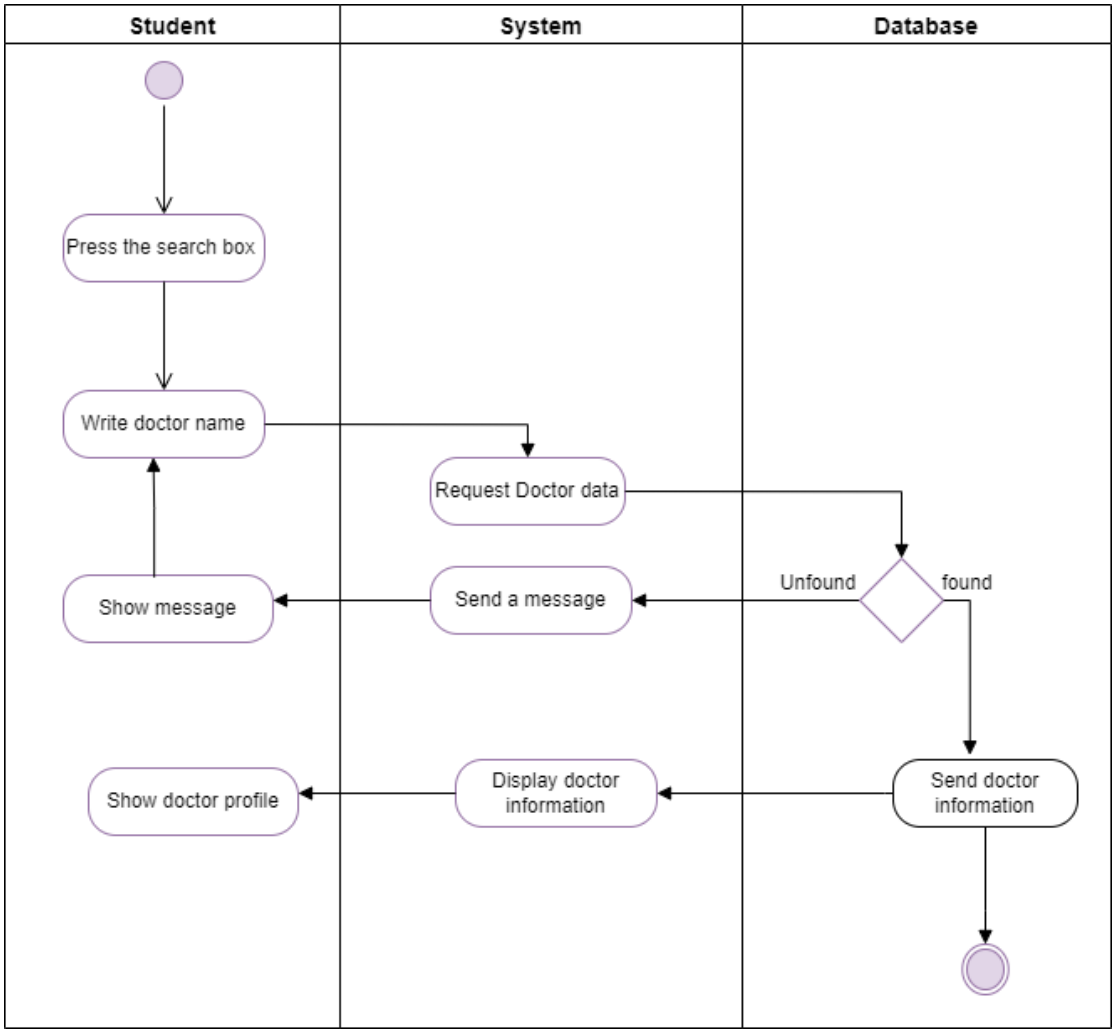
5.5.10 Delete Photo archive activity diagram



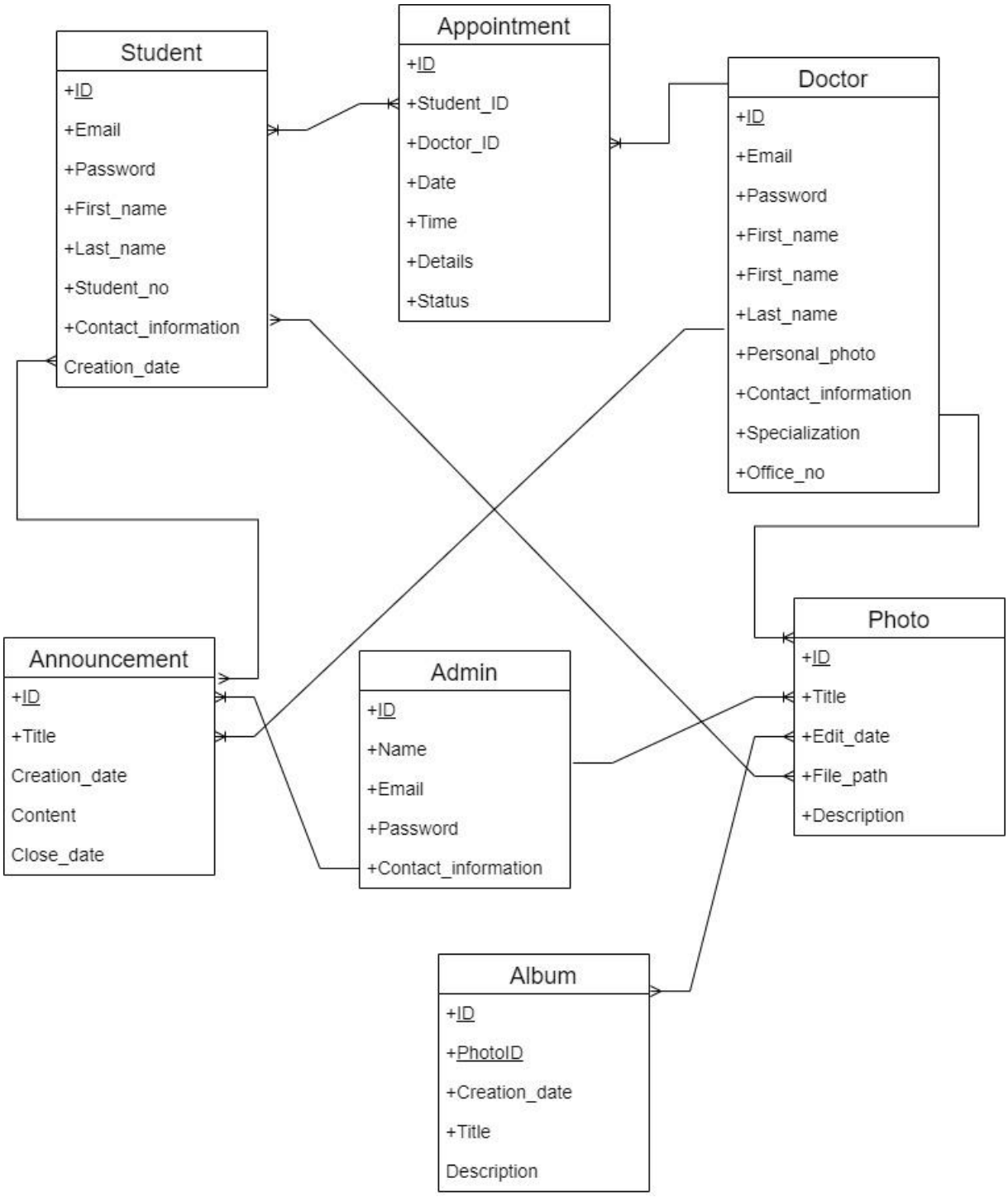
5.5.11 Update personal doctor information activity diagram



5.5.12 Browse and search doctor's activity diagram

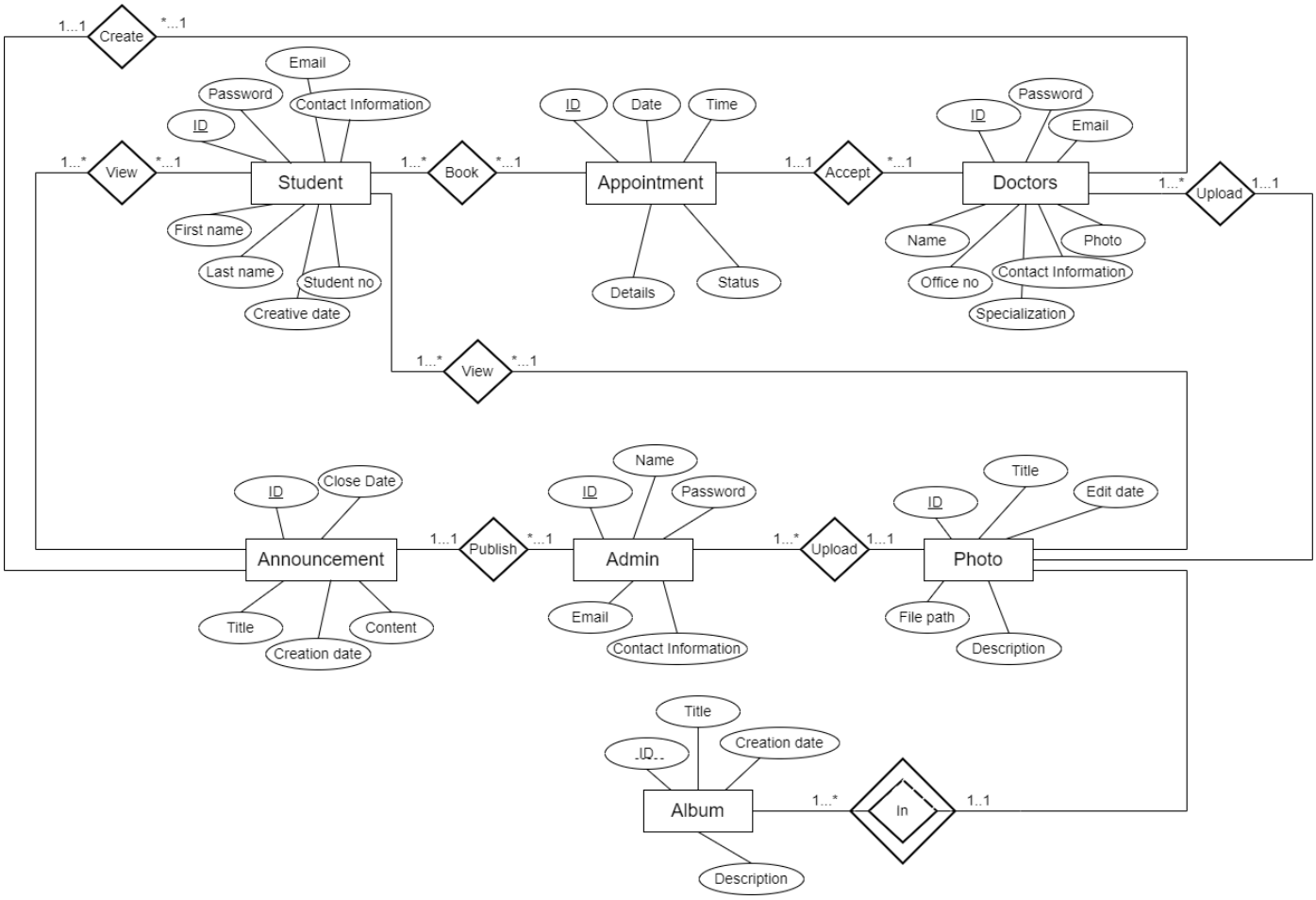


5.6 Class Diagram



5.7Database Diagrams

5.7.1 ER diagram



5.7.2 Relation Database Schema diagram

Admin Announcement

Announcement (**ID** ,Close_data,Title,creation date,content,**Doctor_ID**)

Admin (**ID**,Name,Password,Email,Content info)

Publish (**Announcement_ID** ,Admin_ID)

Announcement (**ID** ,Close_data,Title,creation date,content,Doctor_ID, Admin_ID)

Doctor Announcement

Doctor (**ID** , Password,Email,Name office no,Photo,contact info,Specialization)

Announcement (**ID** ,Close_data,Title,creation date,content)

Create (Doctor_ID,**Announcement_ID**)

Announcement (**ID** ,Close_data,Title,creation date,content,Doctor_ID)

Student Announcement

Student (**ID**,Password,Email,Content_info,First Name,Last Name,Student no,creation date)

Announcement (**ID** ,Close_data,Title,creation date,content)

View_Announcement(**Student_ID**,**Announcement_ID**)

Admin Photo

Admin (**ID**,Name,Password,Email,Content info)

Photo (**ID** ,Title,Edit date,File Path,Description,Doctor_ID)

Upload (**Admin_ID**,Photo_ID)

Admin (**ID**,Name,Password,Email,Content info,Photo_ID)

Doctor Photo

Doctor (**ID** , Password,Email,Name office no,Photo,contact info,Specialization)

Photo (**ID** ,Title,Edit date,File Path,Description)

Upload (Doctor_ID,**Photo_ID**)

Photo (**ID** ,Title,Edit date,File Path,Description,**Doctor_ID**)

Student Photo

Student (**ID**,Password,Email,Content_info,First Name,Last Name,Student no,creation date)

Photo (**ID** ,Title,Edit date,File Path,Description)

View_Photo(**Student_ID**,**Photo_ID**)

Photo Album

Photo (**ID** ,Title,Edit date,File Path,Description,Doctor_ID)

Album (**ID** ,Title,creation date,Description,**Photo_ID**)

Student Appointment

Student (**ID**,Password,Email,Content_info,First Name,Last Name,Student no,creation date)

Appointment (**ID,**Date_Time,Details,Status)

Book (**student_ID**,**Appointment_Id**)

Doctor Appointment

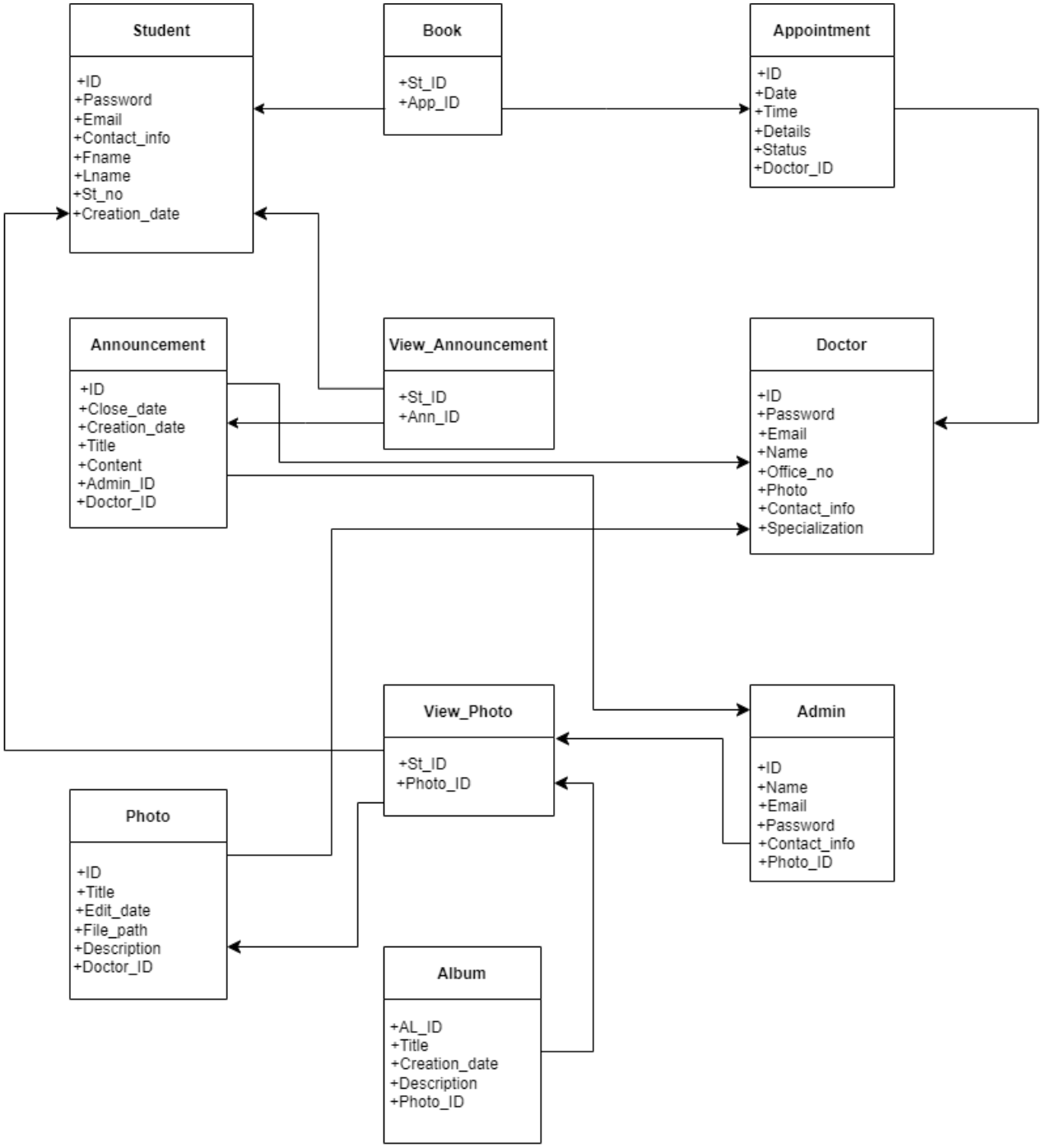
Appointment (**ID,**Date_Time,Details,Status)

Doctor (**ID**,Password,Email,Content_info, Name,Photo,office_ no,creation date,specialization)

Accept (**Appointment_Id** ,Doctor_ID)

Appointment (**ID,**Date_Time,Details,Status,Doctor_ID)

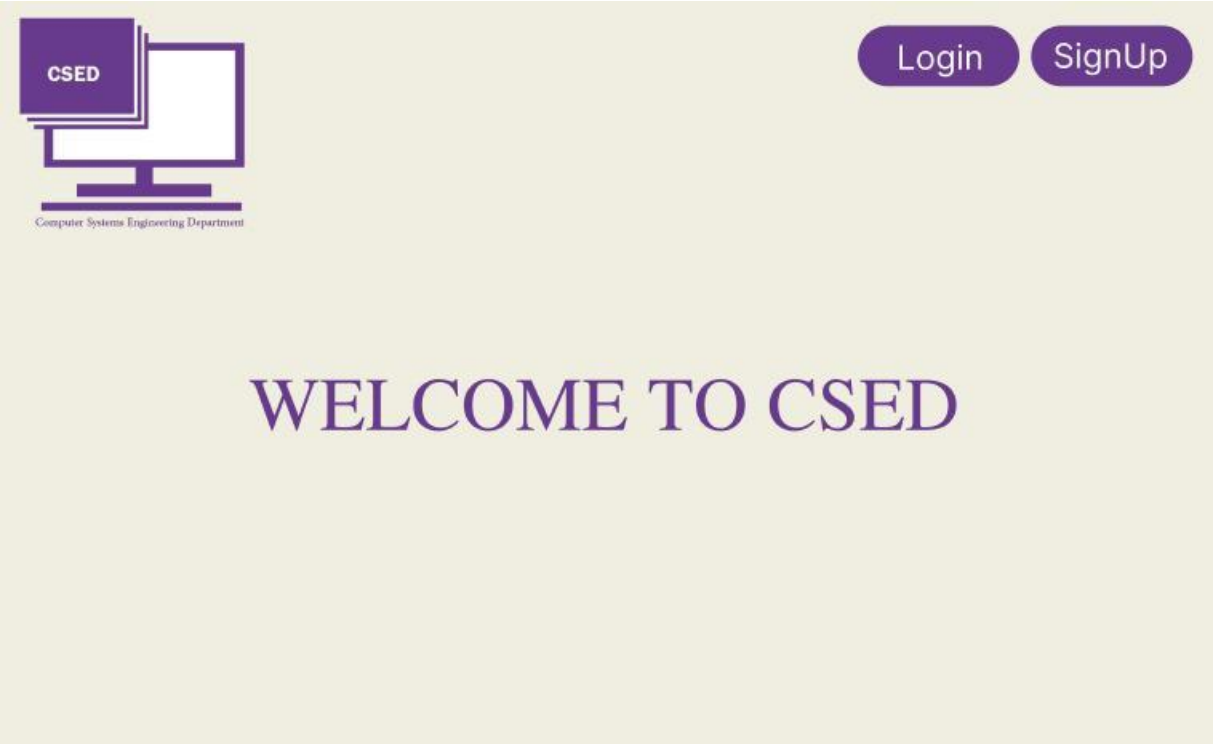
5.7.3Database Schema diagram



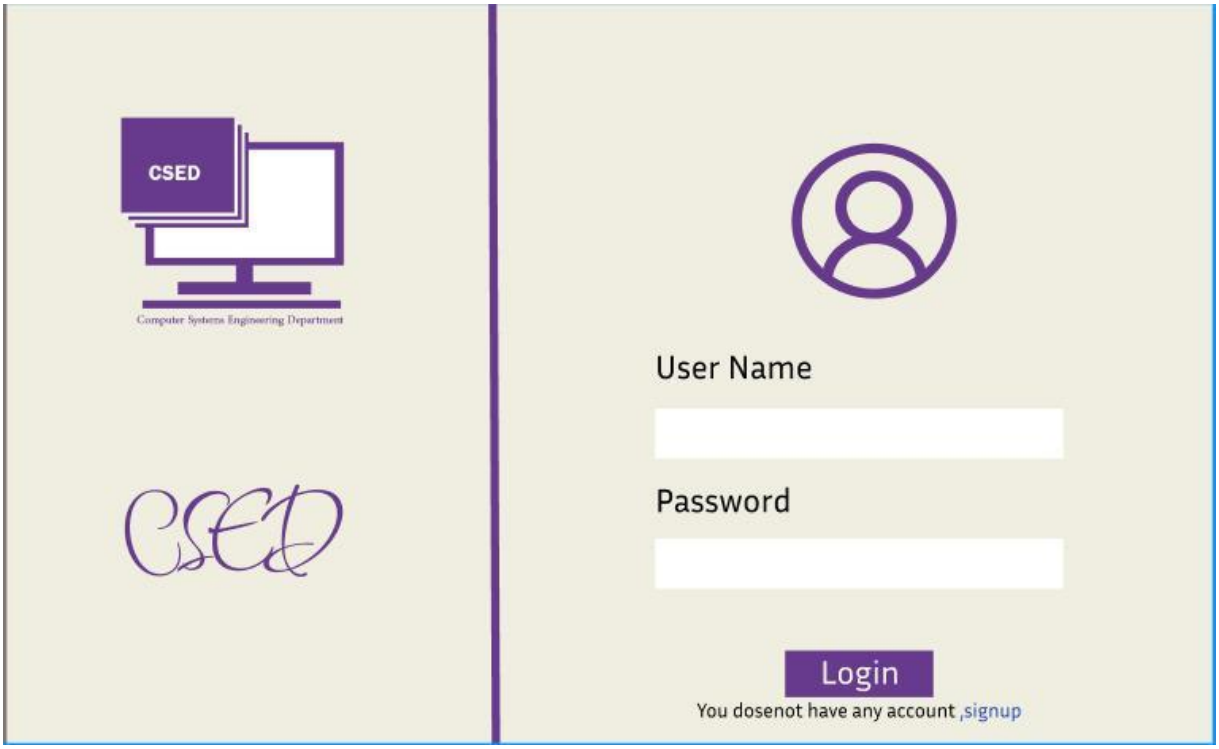
CHAPTER 6

USER INTERFACE

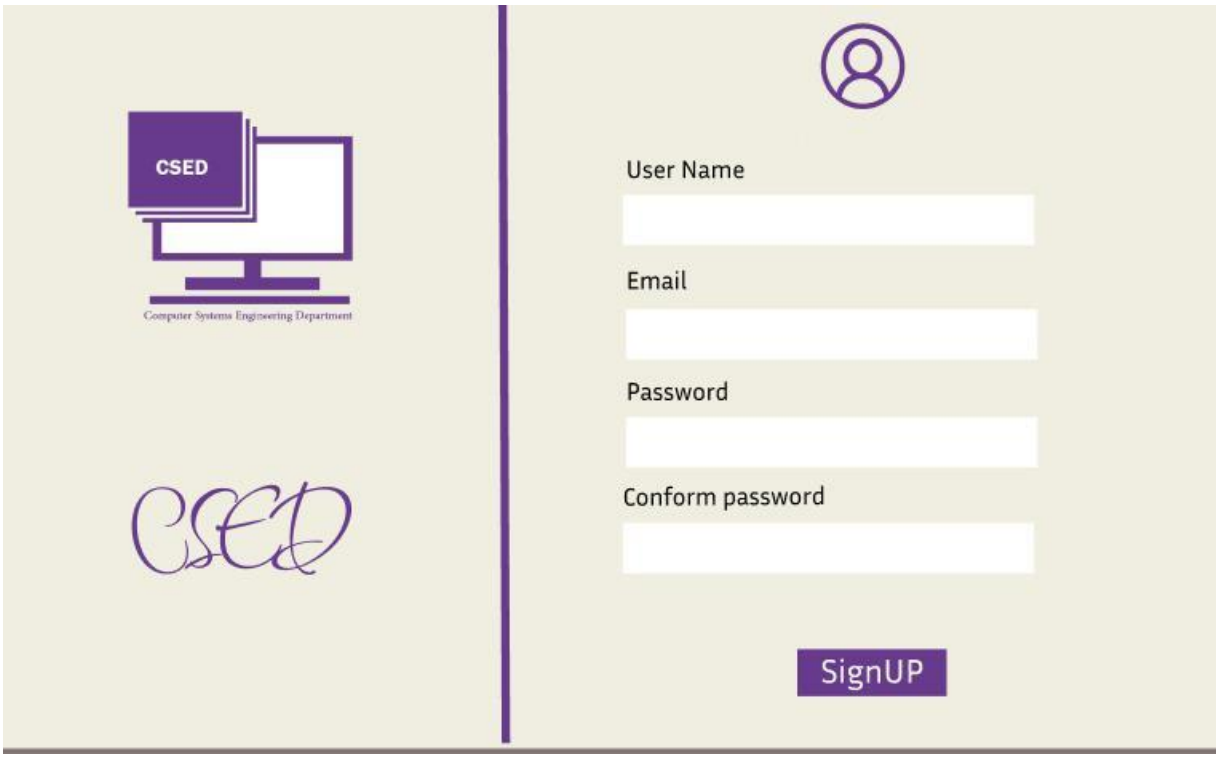
6.1 Cover Page



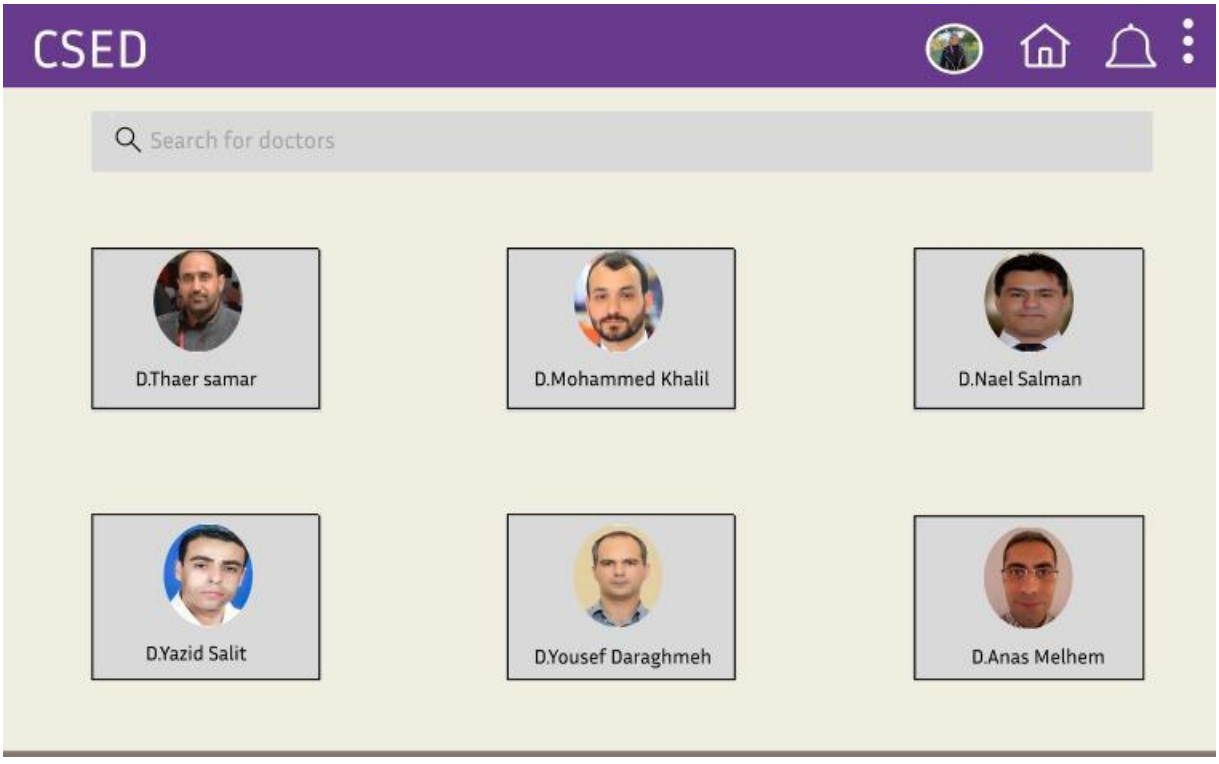
6.2 Sign in Page



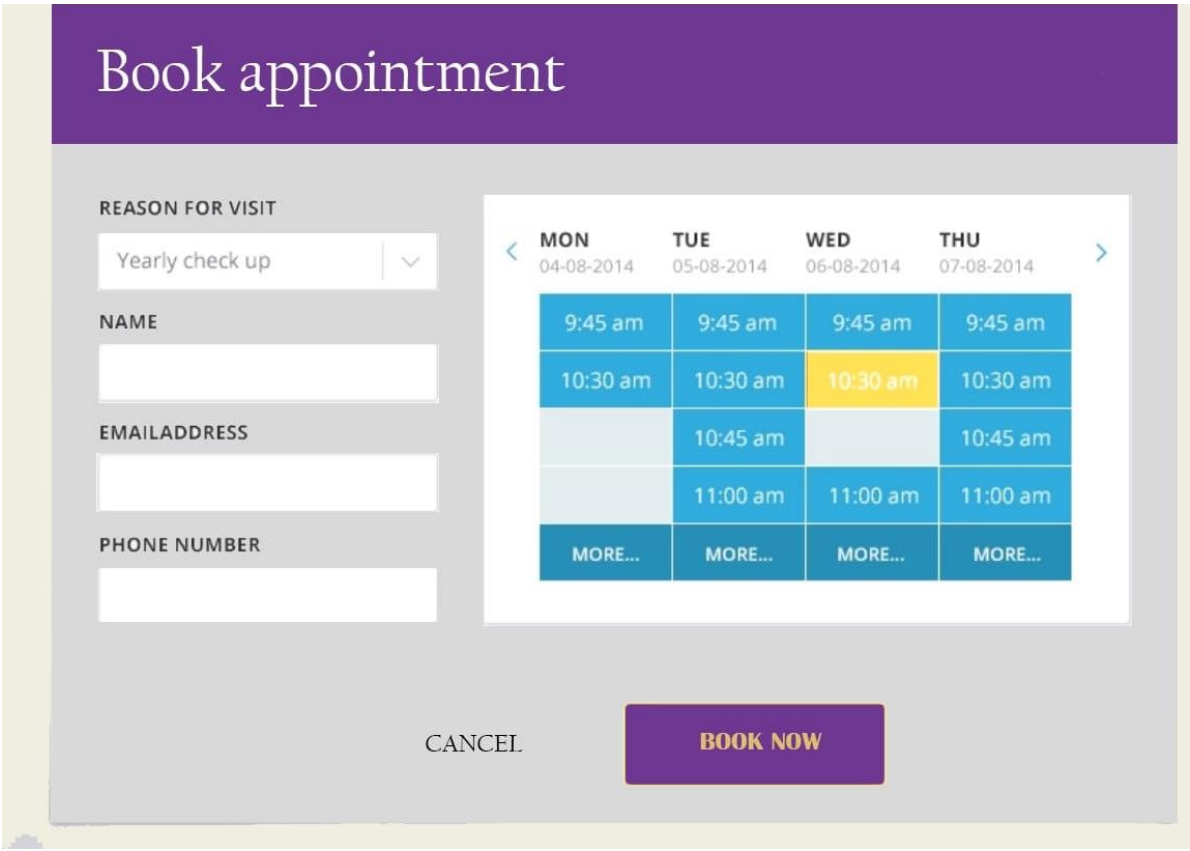
6.3 Sign up Page



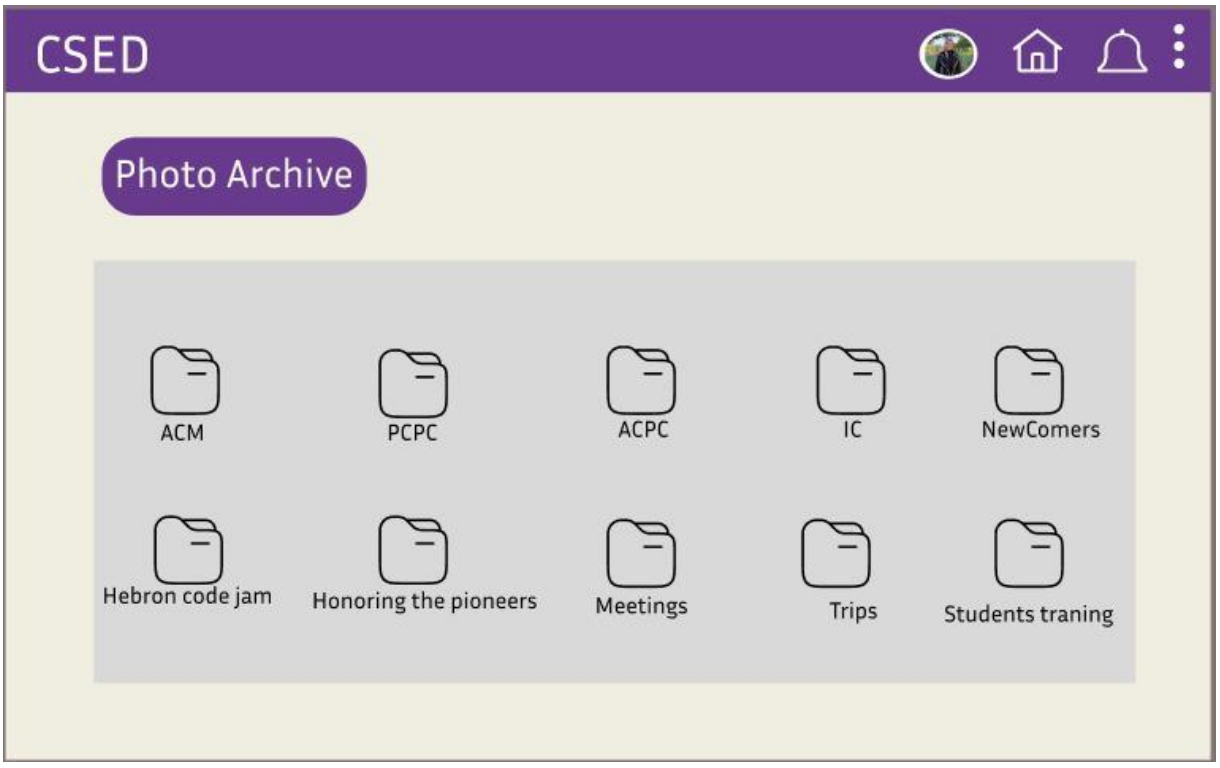
6.4 Doctors Profile Page



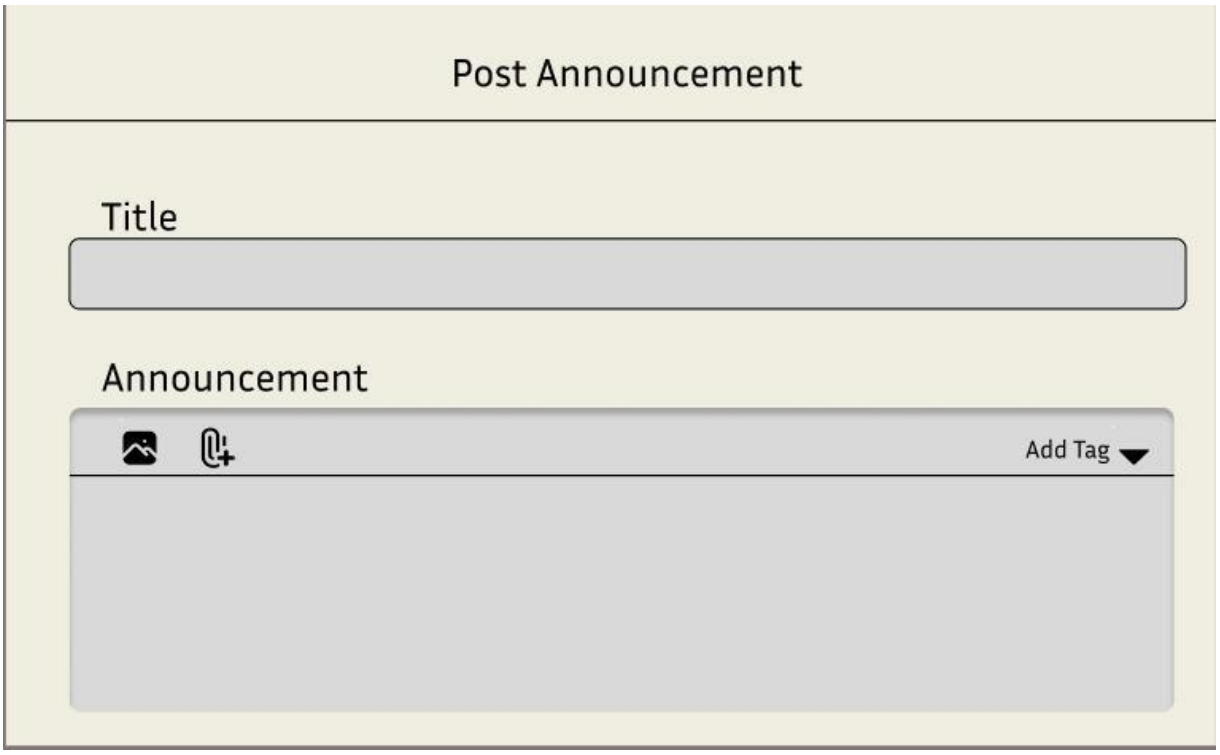
6.5 Book Appointment Page



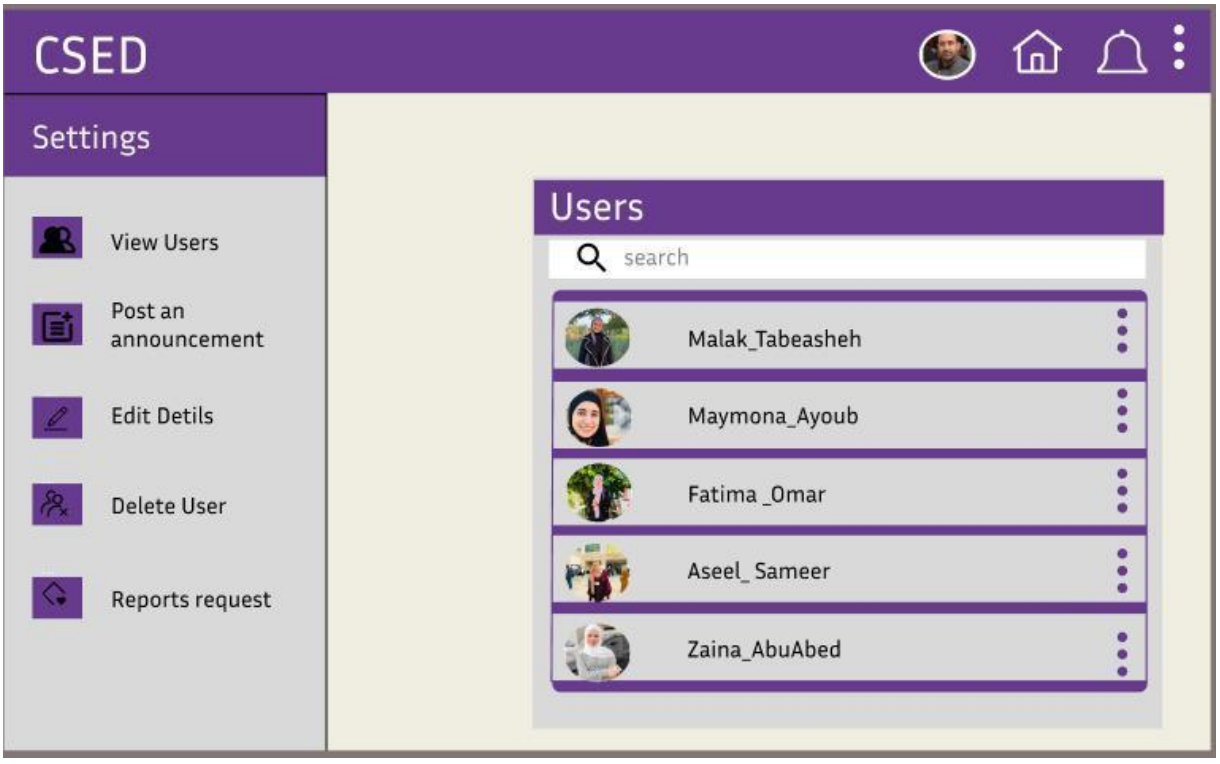
6.6 Photo Archive Page



6.7 Post Announcement Page



6.8 Admin Page



CHAPTER 7

CONCLUSION

Overall Evaluation: Based on the implementation and the design of the system, it can be concluded that the system can effectively fulfill the main purpose of providing a common platform for students, doctors, and administrators for managing appointments, handling communication, and accessing important academic resources.

Potential Impact: The project could lead to significant improvements in the efficiency of scheduling and communication within the university, providing a more streamlined, user-friendly, and centralized way for users to manage their schedules and academic resources.

Future Direction: The project lays the groundwork for future expansion and additional features. While the current scope of the system is limited to appointment scheduling, announcement posting, and resource sharing, it could potentially be expanded to include additional modules like online consultations, real-time availability updates, or even integrations with other university systems like grading or course management.

Importance of User-Centered Design: One key take away from this project is the importance of a user-centered approach in the design and development of such systems. Ensuring that the system is easy to use and meets the needs of all types of users is crucial for its adoption and overall success.

Need for Continuous Improvement: As with any software system, it is concluded that continuous monitoring and improvement will be necessary as users begin to use the system and provide feedback. Future work should focus on addressing any issues or challenges that arise, as well as implementing improvements and new features based on user feedback and needs.

Future Vision

If the project delivers the expected results, the next steps may include:

Expanding to the entire university: If successful in the pilot phase, the system could be implemented across all departments, benefiting a larger portion of the university's community.

Adding more features: Based on user feedback and needs, the system could evolve to incorporate more functionalities, such as assignment tracking, grading systems, or forums for academic discussions.

Integrating with other university systems: The project could be linked with other existing university systems, like the library database or a university-wide calendar, creating a holistic academic management tool.

Developing a mobile application: With the rise of mobile technology, a mobile app version of the system could make it even more accessible and convenient for users.

This project aims to revolutionize the academic experience at the university. The ultimate success of this project will largely depend on user engagement and feedback, which will guide further improvements and enhancements.

7.1 References

1. [1-1] The questionnaire
   https://docs.google.com/forms/d/1IxAVWc69wlEU0v2YePcOYbxTnvEU__8IwBA-DBJEx1Q/edit
2. [2-1] Moodle (LMS)
   https://lms.ptuk.edu.ps/
3. [2-2] Edugate (PTUK)
   https://edugate.ptuk.edu.ps/
4. [2-3] Canvas
   https://www.canvas.net/
5. [2-4] Doodle
   https://doodle.com/en/
6. [4-1] Agile
   https://www.wrike.com/project-management-guide/faq/what-is-agile-methodology-in-project-management/
7. [4-2] Front-end web development
   https://frontendmasters.com/guides/front-end-handbook/2018/what-is-a-FD.html
8. [4-5] React
   https://kruschecompany.com/react-framework-library/
9. [4-6] Why Should You Use React?
   https://www.freecodecamp.org/news/why-use-react-for-web-development/
10. [4-7] React MUI
    https://mui.com/material-ui/getting-started/overview/

11. [4-8] Back-end web development
https://www.coursera.org/articles/back-end-developer

12. [4-9] ASP.NET Framework
https://en.wikipedia.org/wiki/ASP.NET#

13. [4-10]Why we will use ASP.NET Framework?
https://programmers.io/advantages-of-asp-net-development/

14. [4-11] MVC
https://learn.microsoft.com/en-us/aspnet/mvc/overview/older-versions-1/overview/asp-net-mvc-overview

15. [4-12] SQL Server
https://www.guru99.com/sql-server-introduction.html

16. [4-13] ORM
https://www.freecodecamp.org/news/what-is-an-orm-the-meaning-of-object-relational-mapping-database-tools/

17. [4-14]IDE
https://en.wikipedia.org/wiki/Integrated_development_environment

18. [4-15] VS Code
https://code.visualstudio.com/docs/editor/whyvscode

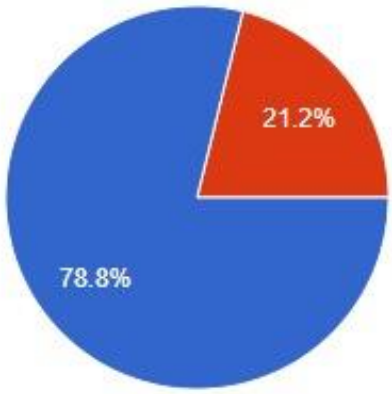
19. [4-16] VS
https://en.wikipedia.org/wiki/Visual_Studio

7.2 Appendix

The questionnaire results

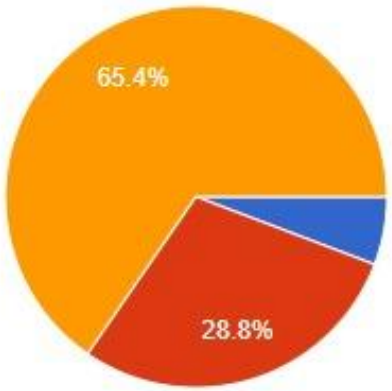
هل ستستخدم ميزة لعرض واستخدام أرشيف صور للمحتوى الأكاديمي؟ , صور المسابقات والفعاليات والخ

52 ردًا



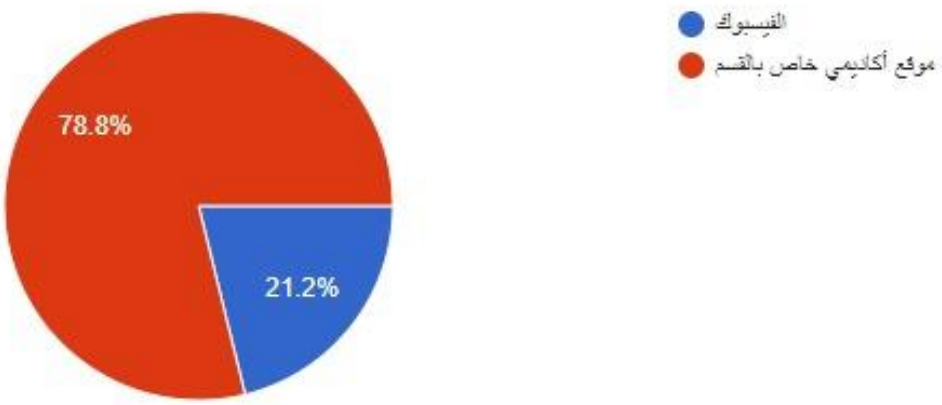
ما مدى أهمية نظام حجز المواعيد مع الأكاديميين في رأيك؟

52 ردًا



أين تفضل أن تكون الاعلانات الخاصة بالقسم والأمور الأكاديمية ؟

52 ردًا



هل تشعر بالحاجة إلى نظام أساسي منفصل للتواصل والتفاعلات الخاصة بكل قسم؟

52 ردًا