

**Class management system**

Learn, Grow and succeed together

**2ND YEAR**

# **FINAL REPORT**



**SCS2202**

**Group project**

**CS Group 56**

# Contents

01 Introduction .....	5
Domain Description .....	5
Current System and Limitations .....	5
Current System and Limitations .....	5
Goal & Objectives .....	6
Assumptions .....	6
02 Feasibility Study .....	7
Social Feasibility .....	7
Technical Feasibility .....	8
Schedule Feasibility .....	9
Economical Feasibility .....	9
Legal and Ethical Feasibility .....	11
03 Requirements .....	12
Stakeholders / Actors: .....	12
Functional Requirements .....	12
Guest User: .....	12
Student: .....	12
Teacher: .....	13
Institute .....	13
System Admin: .....	14
All Users .....	15
Student .....	16
Teacher .....	17
Institute .....	17
System Admin .....	18
Non-Functional Requirements (Quality Attributes) .....	18
Scope of the Project .....	19
Constraints and Limitations .....	20

04 System's Architecture .....	22
1. Model (Data Layer): .....	24
2. View (User Interface Layer): .....	25
3. Controller (Application Logic Layer): .....	25
Component Interactions: .....	25
05 System Design Diagrams .....	27
Use case diagram .....	27
Class diagram .....	28
ER diagram .....	29
Activity diagrams .....	30
Comments from the interim presentation.....	62
How the comments were addressed .....	63
(1) Verifying a NEW REGISTERING TEACHER through system admin.....	63
(2) RESTRICT students from VIEWING OTHERS' MARKS .....	64
(3) CATEGORIZE TEACHERS' SUBSCRIPTION PLANS according to specific needs.....	65
6. Completeness of the Project.....	66
• Functionalities completed.....	66
• Functionalities yet to complete .....	69
• Individual contribution of the team members.....	70
Test Cases .....	71
01. User Login .....	71
02 User profile update .....	73
03 User create blogs .....	75
04 Student filtering classes .....	76
05 Student joining classes.....	77
06 Student enrollment fees .....	79
07 Teacher creating a class .....	80
08 Teacher creating individual/institute class.....	81
09 Teacher vs other users signing up .....	82

10 Institute view classes .....	84
All the members contributed for the project .....	86

## 01 Introduction

### Domain Description

This project is focused on the education sector. The goal is to create an all-in-one website where students, teachers, and institutes can come together in one place. The website will help students find and join tuition classes easily. Unlike the current systems where students have to choose from a single institute, this platform allows them to find teachers from various institutes. Students can also discover individual teachers who might not have much visibility but can get attention through ads. Teachers can manage their classes and share materials, while institutes can recruit teachers, manage classes, and handle their operations more smoothly. The platform also solves the problem for institutes that struggle with managing physical money and materials.

### Current System and Limitations

Right now, our web application allows students to find classes, join them, and interact with teachers. Teachers can create classes and upload materials. Institutes can recruit teachers and manage classes. However, there are some limitations:

- **Payment System Issues:** The payment system is basic due to budget constraints. This means it may not always process payments smoothly or securely.
- **Storage Problems:** We have limited storage, so teachers and institutes may not be able to upload large amounts of content or files.
- **Scalability:** As the number of users increases, the system might struggle to handle heavy traffic efficiently.

We plan to address these issues as we continue developing the platform.

### Current System and Limitations

Right now, our web application allows students to find classes, join them, and interact with teachers. Teachers can create classes and upload materials. Institutes can recruit teachers and manage classes. However, there are some limitations:

- **Payment System Issues:** The payment system is basic due to budget constraints. This means it may not always process payments smoothly or securely.

- **Storage Problems:** We have limited storage, so teachers and institutes may not be able to upload large amounts of content or files.
- **Scalability:** As the number of users increases, the system might struggle to handle heavy traffic efficiently.

We plan to address these issues as we continue developing the platform.

## Goal & Objectives

The goal of this project is to build a platform that brings everything into one place, making it easier for students, teachers, and institutes to manage their activities. Our objectives are:

1. Make it easy for students to find and join classes online.
2. Help teachers manage their classes, materials, and payments more efficiently.
3. Allow institutes to manage teachers, classes, and administrative tasks easily.
4. Provide individual teachers with tools to advertise and grow their classes.
5. Ensure the platform is secure and easy to use.

## Assumptions

For this platform to succeed, we assume:

1. **Users** (students, teachers, and institutes) will actively engage with the system.
2. The platform will work well on **modern web browsers and devices**.
3. The platform will follow **data protection laws** and have **secure payments**.
4. Hall booking will be done by externally
5. There are no errors during payment processing.

## 02 Feasibility Study

### Social Feasibility

Social feasibility looks at how well the platform will be accepted and used by the target audience—students, teachers, and institutes. We conducted a survey to gather feedback, and the results showed strong support for the platform's features:

- **User Preferences:**
  - **Teachers, Students, and Institutes:** The survey showed that all groups prefer using a web-based platform to manage classes. People want a single place to handle tasks like class schedules, attendance, and payments.
  - **Advertising:** Teachers are interested in using the platform to advertise their classes. They see the benefit of reaching more students through targeted ads.
  - **Real-time Communication:** There was strong support for the ability to chat with students instantly. This feature helps teachers and students stay connected outside of regular class times.
  - **Online Learning Materials:** Students prefer to access learning materials online, which fits with current trends in digital learning.
- **User Engagement and Adoption:** The survey suggests that users are likely to engage with the platform because it meets their needs and preferences.
- **Community Building:** The blog feature, which allows users to share educational content and updates, is expected to help build a strong sense of community on the platform.

Overall, the survey results show that there is a lot of interest and support for the platform, making it socially feasible and well-received by potential users.



## Technical Feasibility

We are building the platform using open-source tools and the **MVC (Model-View-Controller)** architecture:

1. **Front-End:** We will use **HTML** and **CSS** to create a responsive and user-friendly design.
2. **Back-End:** The platform will use **PHP** for processing logic and **MySQL** for storing data like user profiles and class information.
3. **Architecture:** The **MVC** model separates the data (Model), user interface (View), and processing logic (Controller).
4. **Version Control:** We will use **Git** to manage and track code changes.
5. **Scalability:** PHP and MySQL ensure the platform can grow as the user base increases.
6. **Security:** Best practices in PHP and MySQL will protect user data.
7. **Testing:** Thorough testing will be done before deployment to ensure smooth functionality.

**Conclusion:** The technical plan uses reliable open-source technologies, ensuring the platform is scalable, secure, and easy to maintain.



## Schedule Feasibility

Our project started in July, and we are currently working on the design and planning. Here's the schedule:

### 1. Development Phases:

- **Design Phase:** July to September – Finalizing the design and requirements.
- **Front-End Development:** Starting in October – Building the user interface.
- **Back-End Development:** After front-end – Coding the logic and connecting the database.
- **Testing:** Happening throughout the development – Making sure everything works.
- **Final Deployment:** February to March 2025 – Completing and launching the platform.

### 2. Work Plan:

- Weekly Hours: 14 hours per person
- Team: 4 members
- Duration: 36 weeks (July to March)
- Total Man-Hours: 2016 man-hours

### 3. Feasibility:

- The 2016 man-hours are enough for the project.
- Starting front-end in October gives us enough time for design and development.
- The platform will be ready by March 2025, with time for testing and final adjustments.

## Economical Feasibility

Our project is cost-effective by using free and open-source tools. Here's a breakdown of the costs and revenue plan:

### 1. Cost Savings with Open-Source Tools:

- **Tools:** Using **HTML**, **CSS**, **PHP**, **MySQL**, and **Git** helps avoid software costs. These tools are free and well-supported.
- **Cloud Storage:** We'll pay for cloud storage to store materials and scale as needed.

- **Payment Gateway:** We'll test free options for payments, but will invest in a secure payment gateway for the final platform.
2. **Revenue Generation:**
- **Subscription Model:** Institutes and teachers will pay a subscription fee to use the platform, providing steady income to cover costs.
  - **Advertisements:** Additional income will come from ads placed by institutes and teachers, supporting the platform's finances.
3. **Deployment Strategy:**
- **Cost Optimization:** We will use efficient cloud hosting and payment methods to keep costs low.
  - **Scalability:** Cloud services will let us adjust costs based on demand, so we don't pay for more than needed.
4. **Financial Viability:**
- We will calculate the **return on investment (ROI)** to ensure the platform is profitable and sustainable.
5. **Risk Management:**
- We'll allocate money for essential services like cloud storage and payment processing to avoid financial risks.
6. **Long-Term Sustainability:**
- With the clear revenue model and cost management, the platform is expected to be profitable in the long run, with potential for future growth and new features.

## Legal and Ethical Feasibility

### 1. **Data Protection:**

- We will follow Sri Lanka's data protection laws, making sure to protect user data and get consent for using their information.
- Users will be able to access, update, or delete their personal data through the platform.

### 2. **Intellectual Property:**

- We will respect copyrights and trademarks, ensuring we have permission to use any content owned by others.
- We will have rules for user-uploaded content to prevent copyright issues and address any complaints quickly.

### 3. **E-commerce and Consumer Protection:**

- We will follow laws for online transactions, ensuring clear pricing, terms, and refund policies for users.
- We will protect consumer rights by being honest with information and resolving disputes fairly

### 4. **Ethical Considerations:**

- We will be transparent about how the platform works, including ads and data use.
- We will promote fairness and avoid discrimination based on race, gender, or other characteristics.
- We will have safety measures in place to protect users from harassment or inappropriate content.

### 5. **Regulatory Compliance:**

- We will stay updated on laws and make sure the platform follows all required regulations.
- We will practice good corporate governance with transparency and accountability.

## 03 Requirements

### Stakeholders / Actors:

1. **Students** – Users who search for, join, and interact with tuition classes.
2. **Teachers** – Users who create and manage classes, upload materials, and communicate with students.
3. **Institutes** – Organizations that manage teachers and classes, and post job vacancies.
4. **System Admin** – Manages the platform, monitors activities, and handles user accounts.

### Functional Requirements

#### Guest User:

- Register for the system.
- Purchase subscriptions to use premium features.
- View teachers, institutes, and blogs.
- Contact system admin for help.

#### Student:

- Log in to the system.
- View and edit profile.
- Browse teachers and institutes.
- View class details, teacher profiles, and institute profiles.
- Enroll in classes, view schedules, and access learning materials.
- Pay admission and class fees.

- Contact teachers.
- Rate and review teachers and institutes.
- Post and delete blog entries.
- Request old materials.
- View advertisements.
- Contact system admin for help.

#### Teacher:

- Log in (Normal Teacher or Premium Teacher).
- View and edit profile.
- Manage subscriptions (for Premium Teachers).
- Schedule individual classes (for Premium Teachers).
- Upload learning materials.
- View student information.
- Post advertisements.
- Request job offers from institutes.
- Request payroll (for Normal Teachers).
- Offer discounts and free cards (for Premium Teachers).
- Respond to student messages.
- View teachers, institutes, teacher profiles, and institute profiles.
- Post and delete blog entries.

#### Institute:

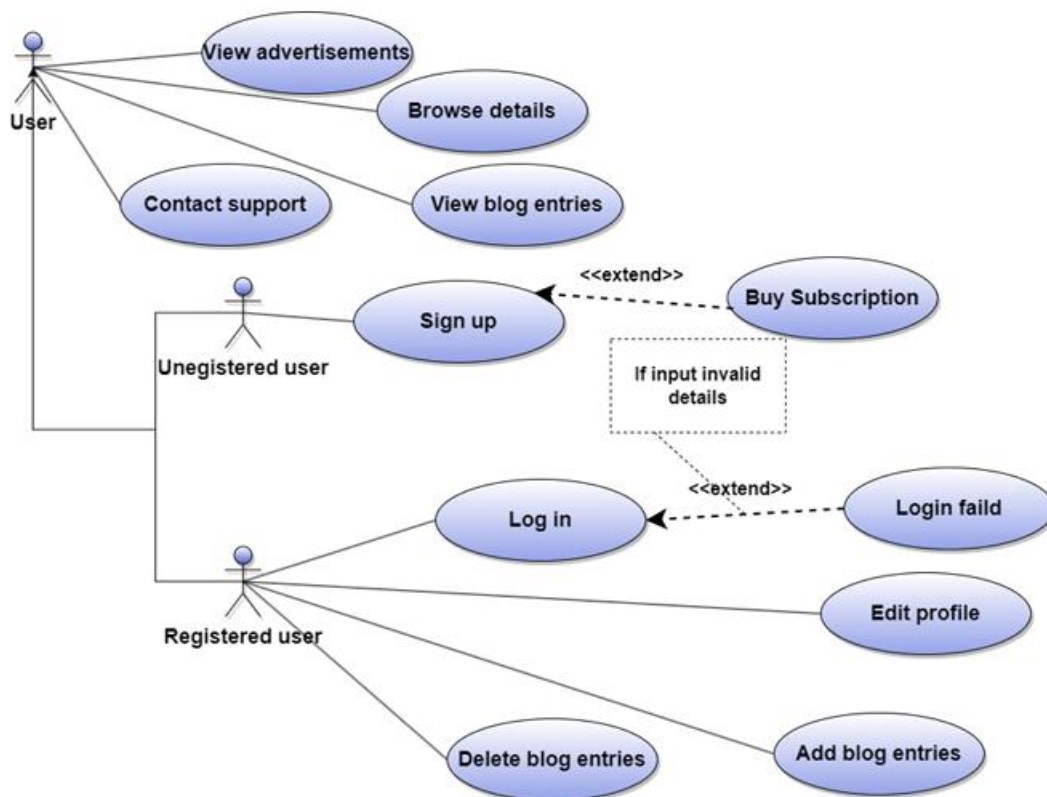
- Log in to the system.

- View and edit profile.
- Manage students and teachers.
- Schedule classes and hire teachers.
- Issue payroll.
- Manage subscription.
- Post advertisements.
- Offer discounts and free cards.

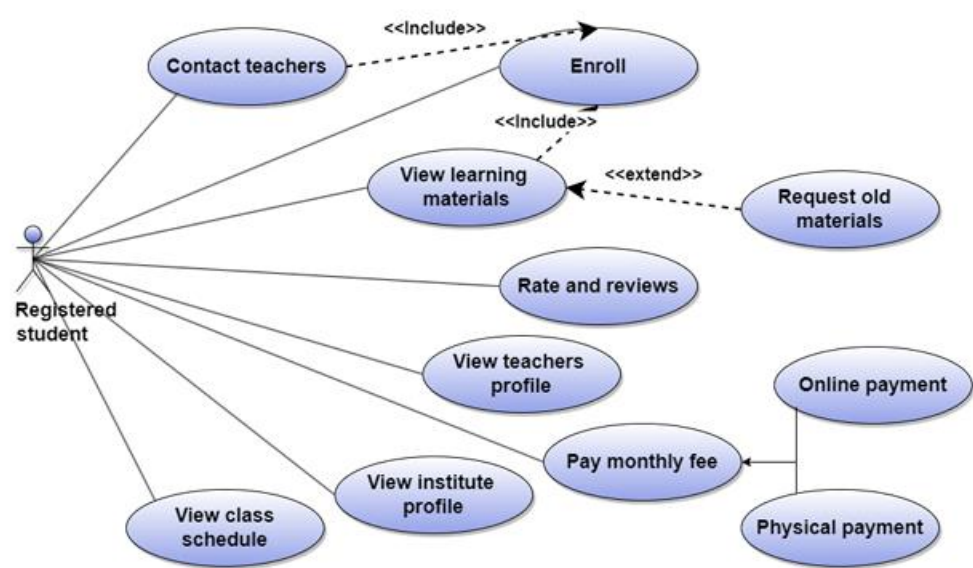
**System Admin:**

- Log in to the system.
- View and edit profiles (teachers, institutes, students).
- Respond to contact forms.
- Post system announcements.

All Users

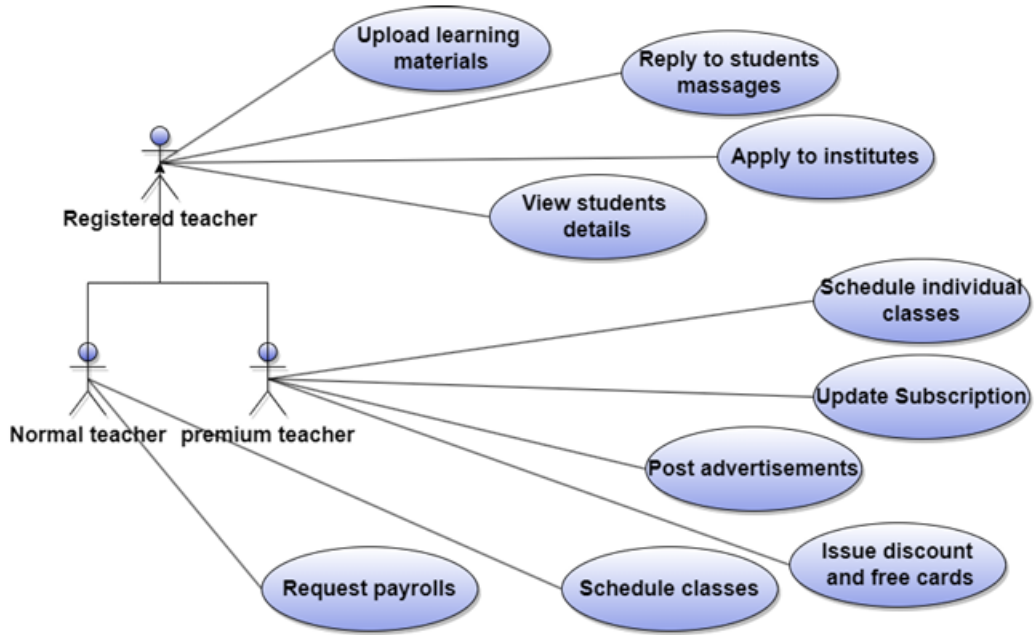


Student

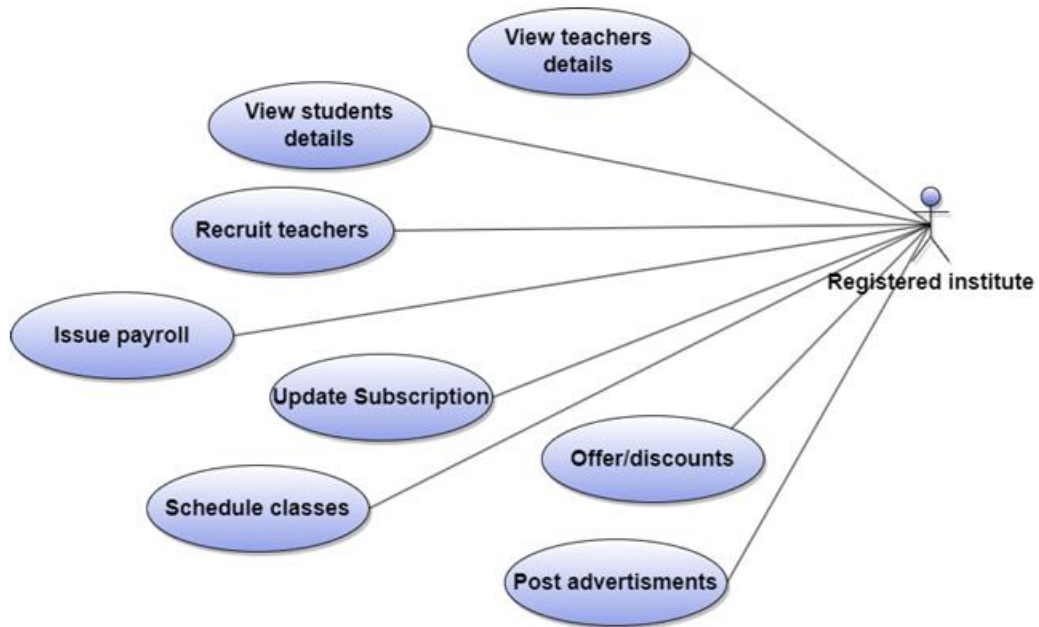




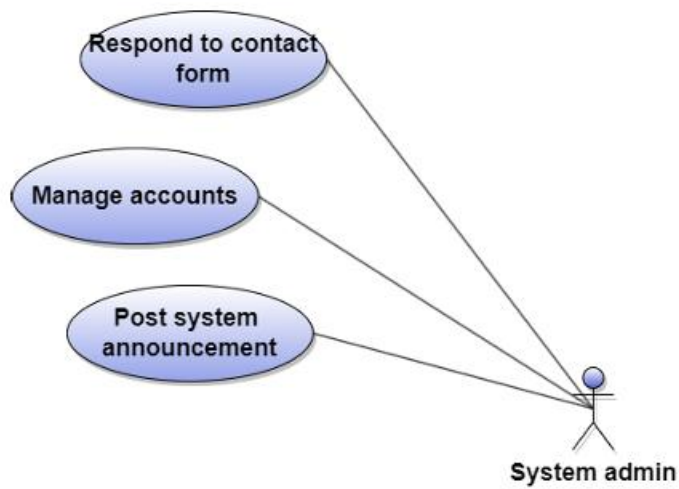
## Teacher



## Institute



## System Admin



## Non-Functional Requirements (Quality Attributes)

- **Performance:** Ensure pages load quickly o Handle multiple users simultaneously without slowing down.
- **Reliability:** Maintain high uptime (99.9% availability).
  - Quickly recover from failures to minimize downtime.
- **Security:** Protect user data with encryption and secure connections.
  - Use strong authentication and permissions to secure accounts.
- **Usability:** Design a user-friendly interface that is easy to navigate.
  - Ensure accessibility for users with disabilities.

- **Maintainability:** Organize code in a modular way for easier updates.
  - Follow coding standards and document changes.
- **Compliance:** Meet legal requirements for data protection and transactions.
  - Uphold ethical standards in user interactions and content.
- **Performance Metrics:** Test system under high loads to ensure it performs well.
  - Plan for enough server capacity to handle growth.
- **Integration:** Integrate smoothly with payment and messaging services.
  - Ensure third-party services work seamlessly with the platform.

## Scope of the Project

The scope of our project includes the development of a comprehensive web-based platform aimed at improving the management and accessibility of the education sector. Key functionalities include:

### User Interfaces and Functionalities:

#### For Students:

- Search and browse tuition classes based on subjects, location, and other criteria.
- Enroll in classes and process payments.
- Manage class schedules and fees through user profiles.

#### For Teachers:

- Tools for class management, scheduling, student tracking, and payment management.
- Profile management to showcase expertise and availability.

#### For Institutes:

- Administrative dashboard to manage classes, teachers, and student records.
- Tools for reporting and analytics to monitor class performance and enrollment trends.

#### **Additional Features:**

- Blog functionality available to all users for posting educational content and updates.
- Advertising tools for teachers and institutes to market their classes within the platform.

#### **Technical Details:**

- Development of a scalable and secure web application.
- Implementation of database systems to manage user profiles, class information, and administrative data securely.
- Responsive design to ensure compatibility across devices and browsers.

#### **Deployment and Support:**

- Comprehensive testing to ensure functionality and performance.
- Deployment on reliable hosting infrastructure with provisions for scalability and data backup.
- Ongoing maintenance and support to address user feedback and optimize performance.

#### **Out of Scope:**

- Mobile app development or support; the platform will only be accessible through web browsers.
- Multilingual support; the platform will be available in a single language.
- Live class sessions or real-time interactions; the platform will focus on providing learning materials and managing class logistics.

### **Constraints and Limitations**

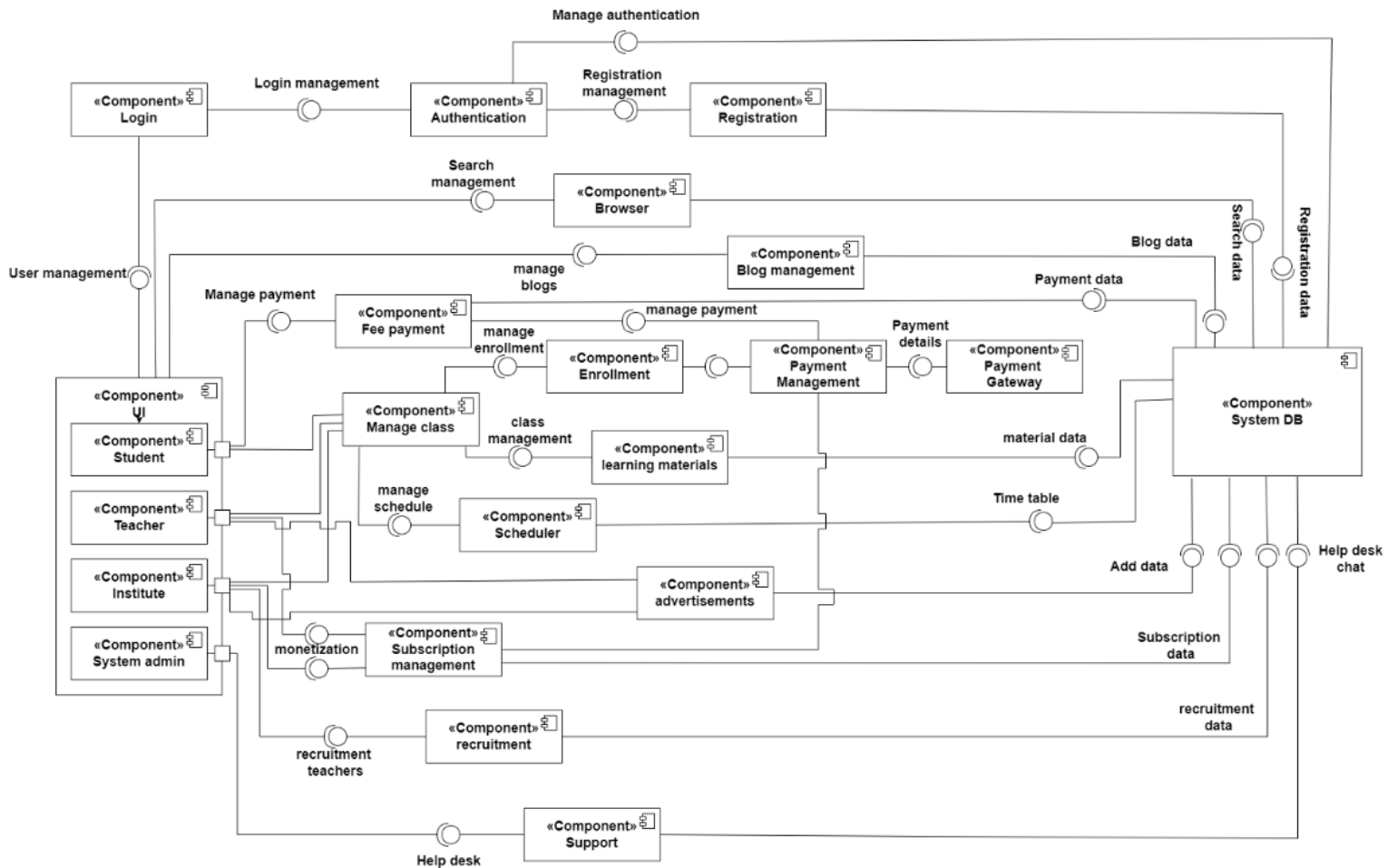
#### **Constraints:**

- **Timeline:** The project must be completed by March 2025, with a 36-week development and deployment period starting from July 2024.
- **Budget:** The project has a fixed budget for expenses like cloud services, payment gateway integration, and other necessary resources.
- **Resource Availability:** Limited team members and external resources may impact the speed and scope of development.
- **Technology Stack:** The project must use **HTML/CSS** for the front-end, **PHP** for the back-end, **MySQL** for the database, and **Git** for version control, limiting technology flexibility.
- **Regulatory Compliance:** The platform must follow Sri Lankan laws, especially regarding data protection, electronic transactions, and intellectual property.

#### *Limitations:*

- **Storage:** Limited storage capacity for user data and class materials.
- **Payment System:** The payment system will be basic during testing and may not handle high volumes of transactions securely until the final version.
- **Scalability:** As the user base grows, performance may be affected unless additional resources are added to the infrastructure.
- **Feature Set:** Some features, like mobile app support or live class sessions, are not included in the scope of this project.

## 04 System's Architecture



### Components and Their Functionalities:

- **User Interface (UI):**
  - **Functionality:** Provides a simple design where users (students, teachers, institutes) can interact with the system.
  - **Interaction:** This is the front-end where all users interact with the platform through browsers.
- **Authentication System:**

- **Functionality:** Manages user logins, registration, and profile management for students, teachers, institutes, and admins.
- **Interaction:** Communicates with the database to verify user credentials and manage user sessions.
- **Database:**
  - **Functionality:** Stores all user data, class details, schedules, payments, and other records securely.
  - **Interaction:** All data requests (like storing, retrieving, and updating information) are handled by the database.
- **Payment System:**
  - **Functionality:** Handles payments for class enrollment and monthly fees.
  - **Interaction:** Integrates with third-party payment gateways and updates payment status in the database.
- **Class Management System:**
  - **Functionality:** Allows teachers to create, manage, and schedule classes. Institutes can manage their teachers and class offerings.
  - **Interaction:** Interacts with the database to store class schedules, materials, and student enrollments.
- **Messaging System:**
  - **Functionality:** Enables students to communicate with teachers and admins, and for notifications like class updates.
  - **Interaction:** Interacts with the database to store and retrieve messages.
- **Admin Panel:**
  - **Functionality:** Allows system administrators to manage users, monitor platform performance, and handle user support requests.
  - **Interaction:** Communicates with all system components to manage user profiles and monitor system data.
- **UI and Authentication System:**

- The UI allows users to log in, register, and manage profiles, which interact with the authentication system to verify credentials.
- **UI and Database:**
  - The UI retrieves and displays data (like classes, teacher profiles, etc.) from the database and sends user input (like enrollment or updates) back to the database.
- **Database and Payment System:**
  - The database stores payment details, while the payment system processes payments and updates the database to reflect successful transactions.
- **Class Management System and Database:**
  - The class management system updates and retrieves class schedules, materials, and enrollment information from the database.
- **Messaging System and Database:**
  - The messaging system allows users to send and receive messages, which are stored in the database.
- **Admin Panel and Other Components:**
  - The admin panel interacts with all components to manage users, monitor activities, and ensure smooth operation of the platform.

The system follows an **MVC (Model-View-Controller)** architecture that we have developed. Below are the components and how they interact within the MVC framework:

#### 1. Model (Data Layer):

- **Functionality:** The Model is responsible for managing the data and the business logic. It interacts directly with the database to retrieve, store, and manipulate data such as user profiles, class details, schedules, and payments.
- **Interaction:**
  - When users enroll in classes or make payments, the Model updates the database with new information.



- It retrieves data from the database (like class schedules or teacher profiles) and passes it to the Controller for further processing.

## 2. View (User Interface Layer):

- **Functionality:** The View is the front-end of the system that users interact with. It displays the data provided by the Controller and allows users to interact with the system.
- **Interaction:**
  - The View presents the data from the Model to the user, such as displaying available classes, teacher profiles, or payment details.
  - User actions, such as clicking a button or submitting a form, are passed to the Controller for handling.

## 3. Controller (Application Logic Layer):

- **Functionality:** The Controller acts as the intermediary between the Model and the View. It processes user inputs (such as login or class enrollment requests), updates the Model, and selects the appropriate View to display.
- **Interaction:**
  - The Controller receives user input from the View (e.g., logging in or enrolling in a class) and updates the Model accordingly.
  - It communicates with the Model to fetch necessary data and then updates the View to display the data to the user.
  - For example, when a student enrolls in a class, the Controller updates the Model with enrollment data and tells the View to show the updated class schedule or confirmation.

## Component Interactions:

- **View and Controller:**
  - The View sends user actions (such as clicking a button or submitting a form) to the Controller.
  - The Controller processes these actions and calls the Model to update or retrieve data.

- **Controller and Model:**

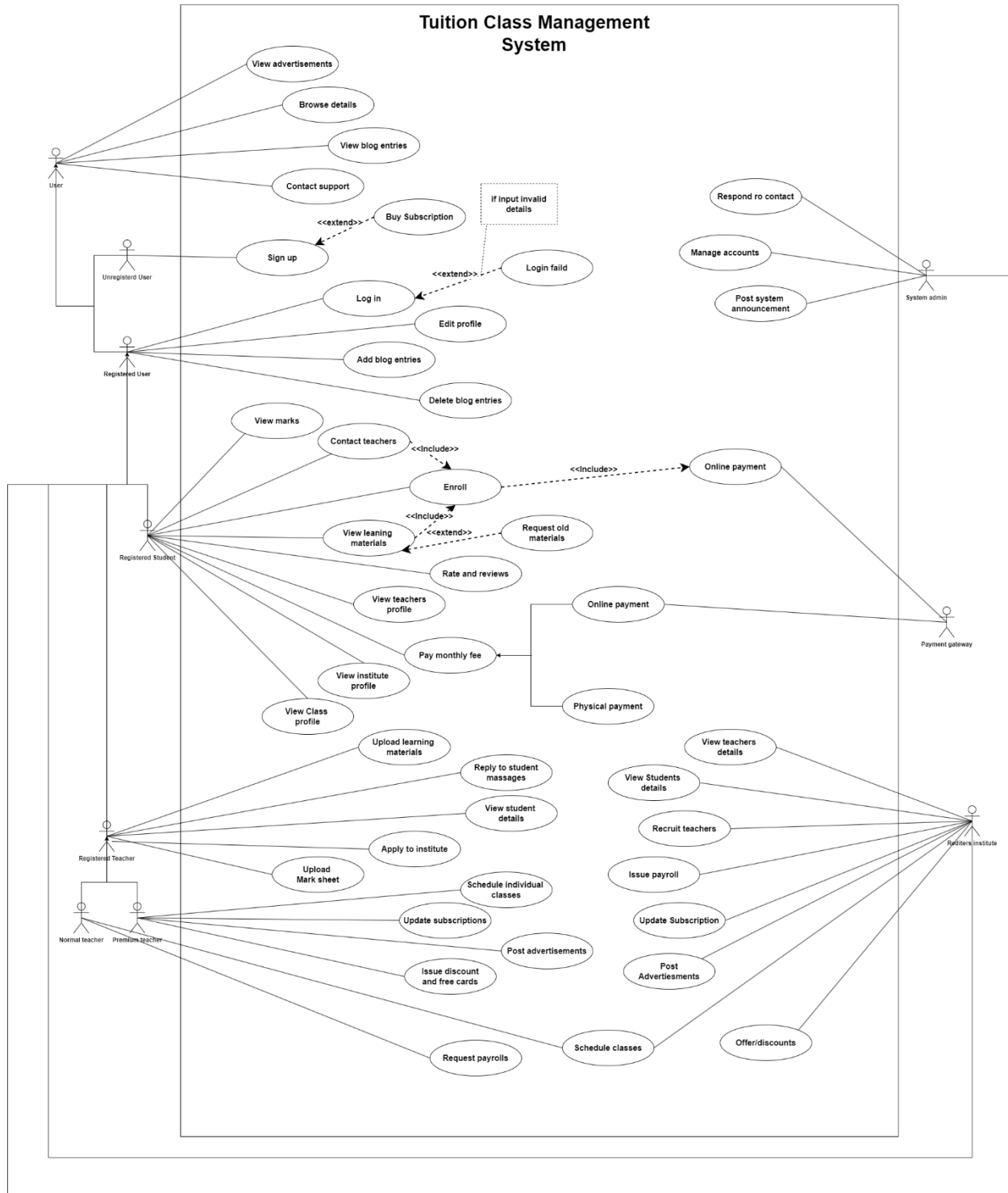
- The Controller updates the Model with data from the user (e.g., storing new class enrollments) and retrieves data from the Model (e.g., available classes, schedules).
- Once the Model is updated, the Controller selects the appropriate View to display the results.

- **Model and Database:**

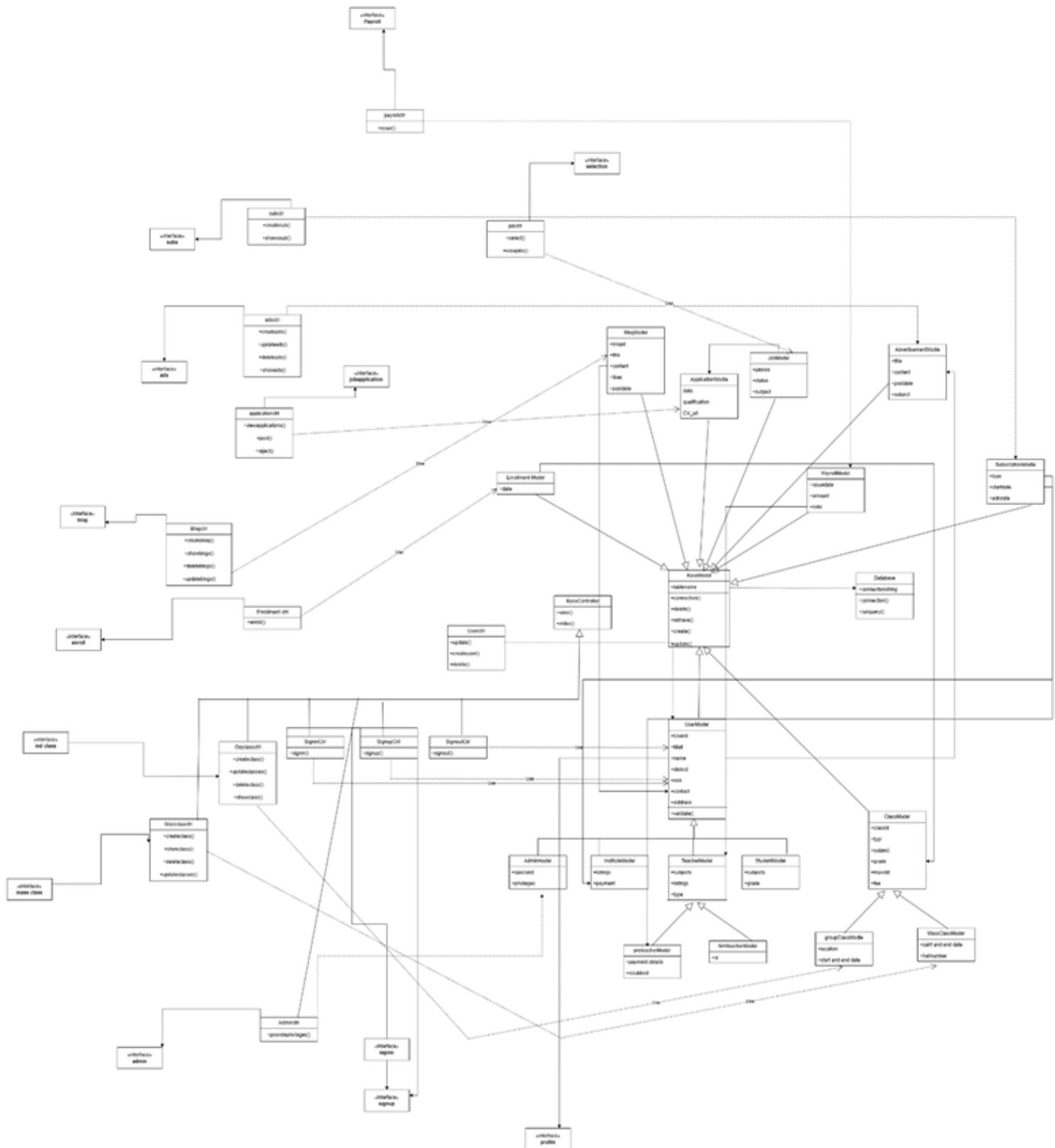
- The Model interacts directly with the database to fetch, store, or update data (e.g., user profiles, classes, schedules).
- It ensures that the data is correctly handled and passed back to the Controller when needed.

## 05 System Design Diagrams

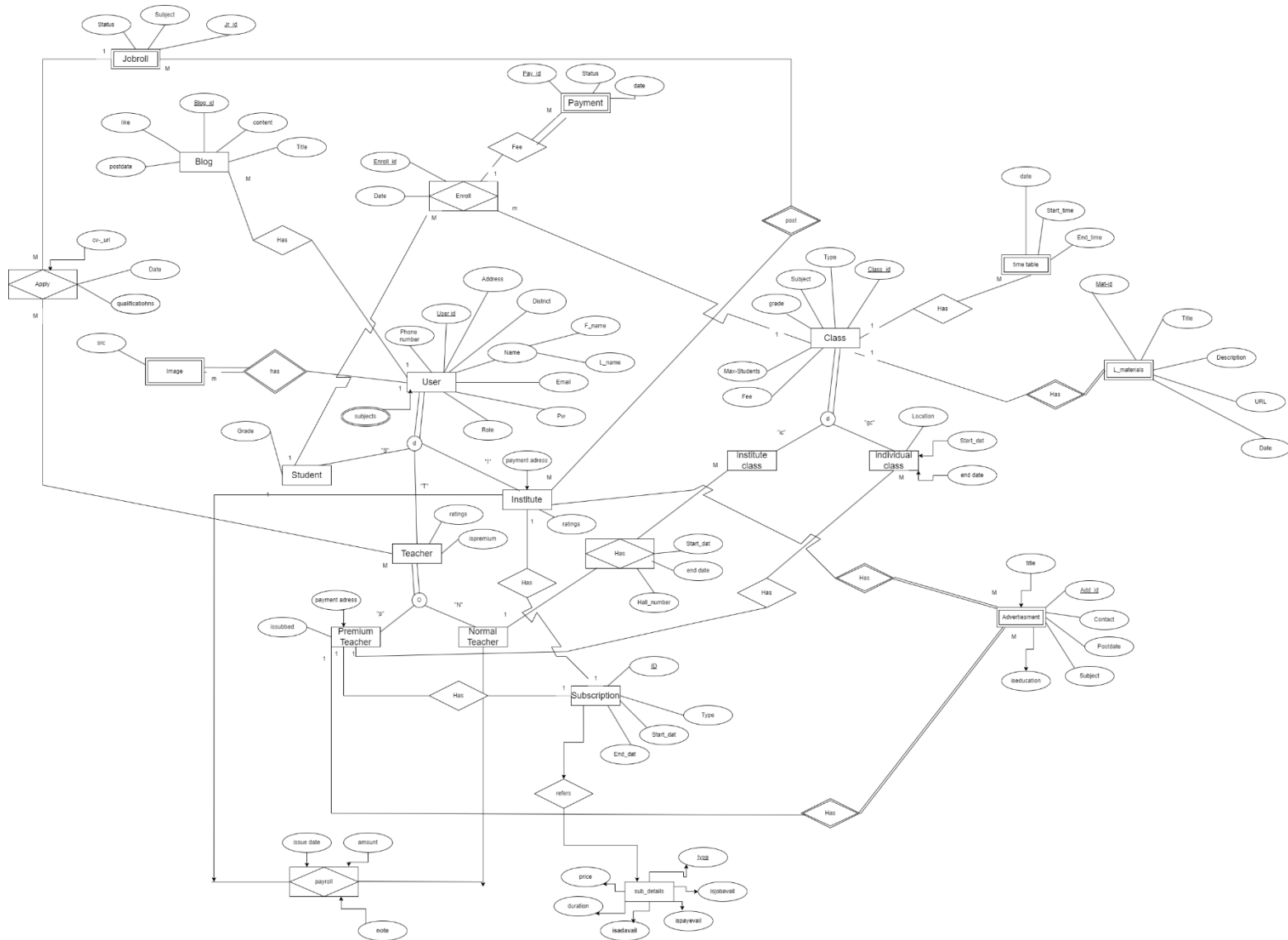
### Use case diagram



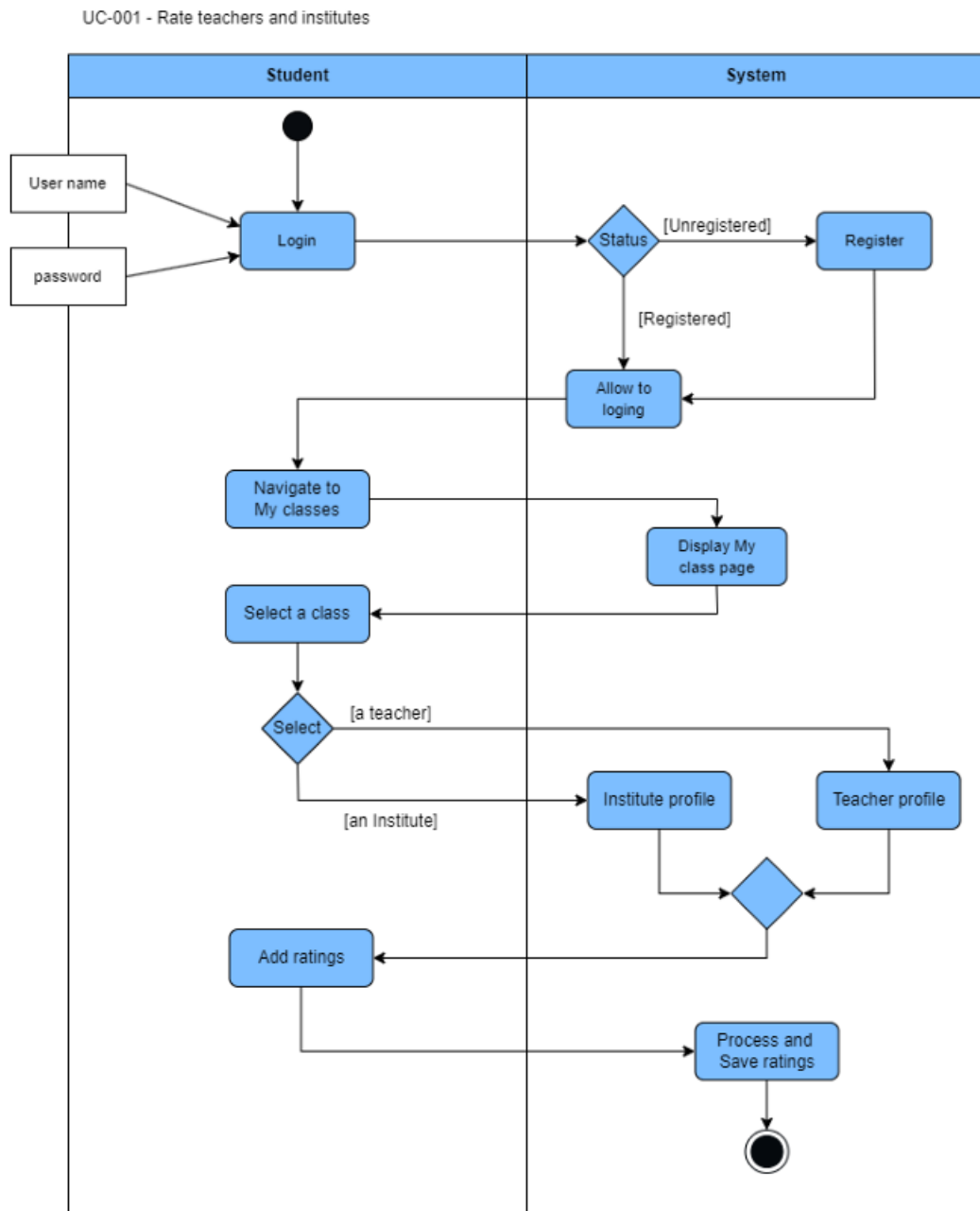
## Class diagram



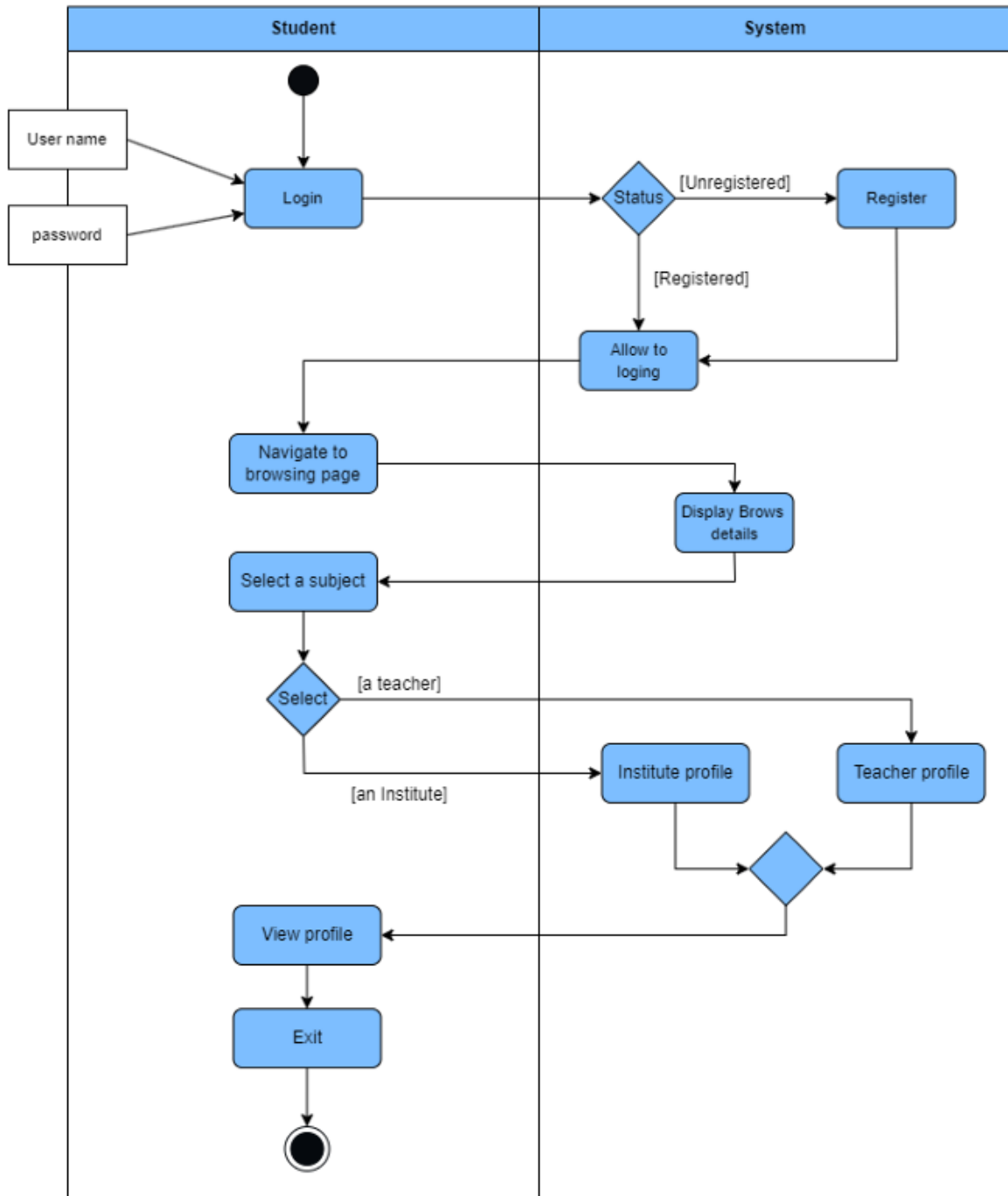
## ER diagram



## Activity diagrams



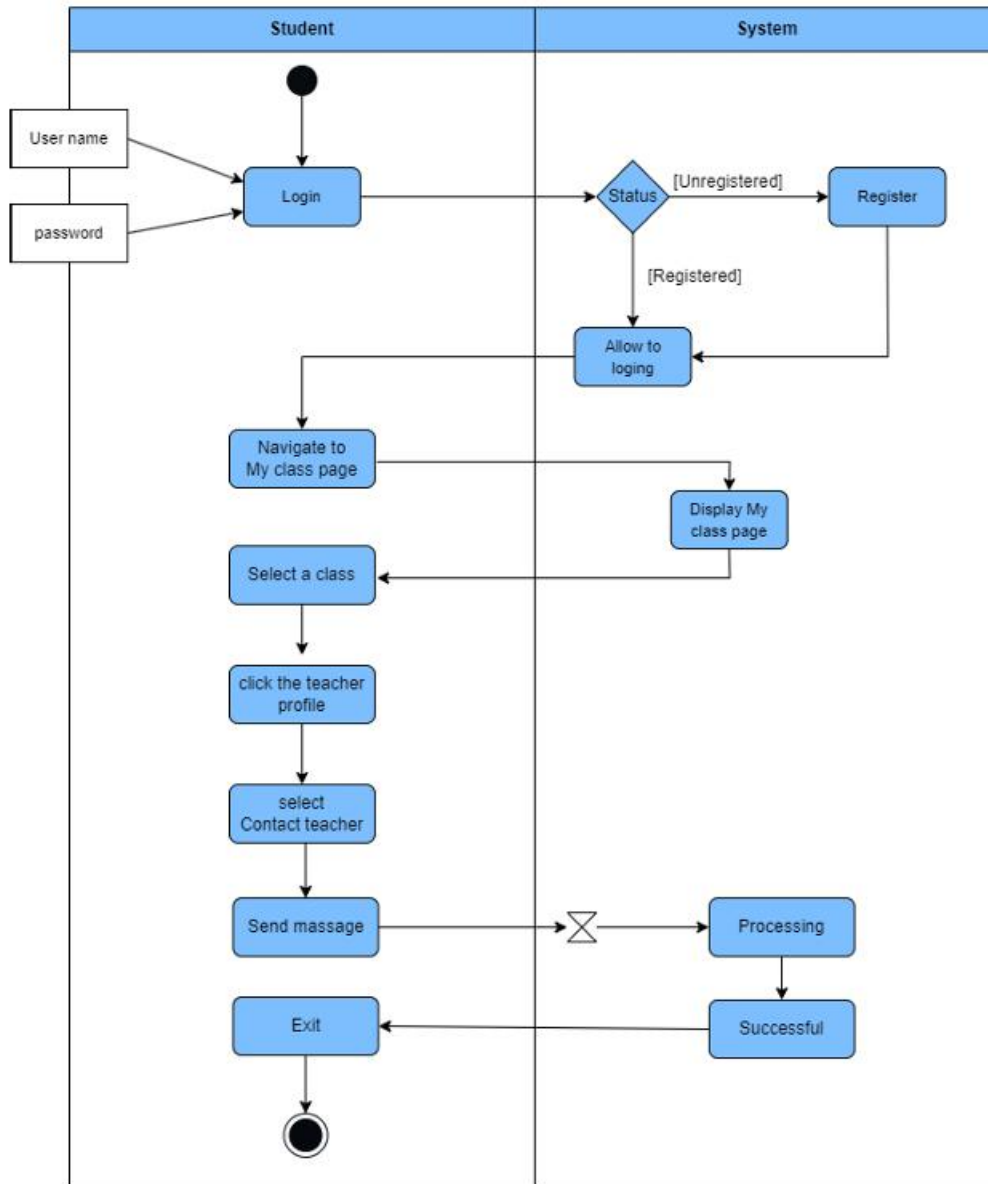
UC-002 - View teacher/institute profile

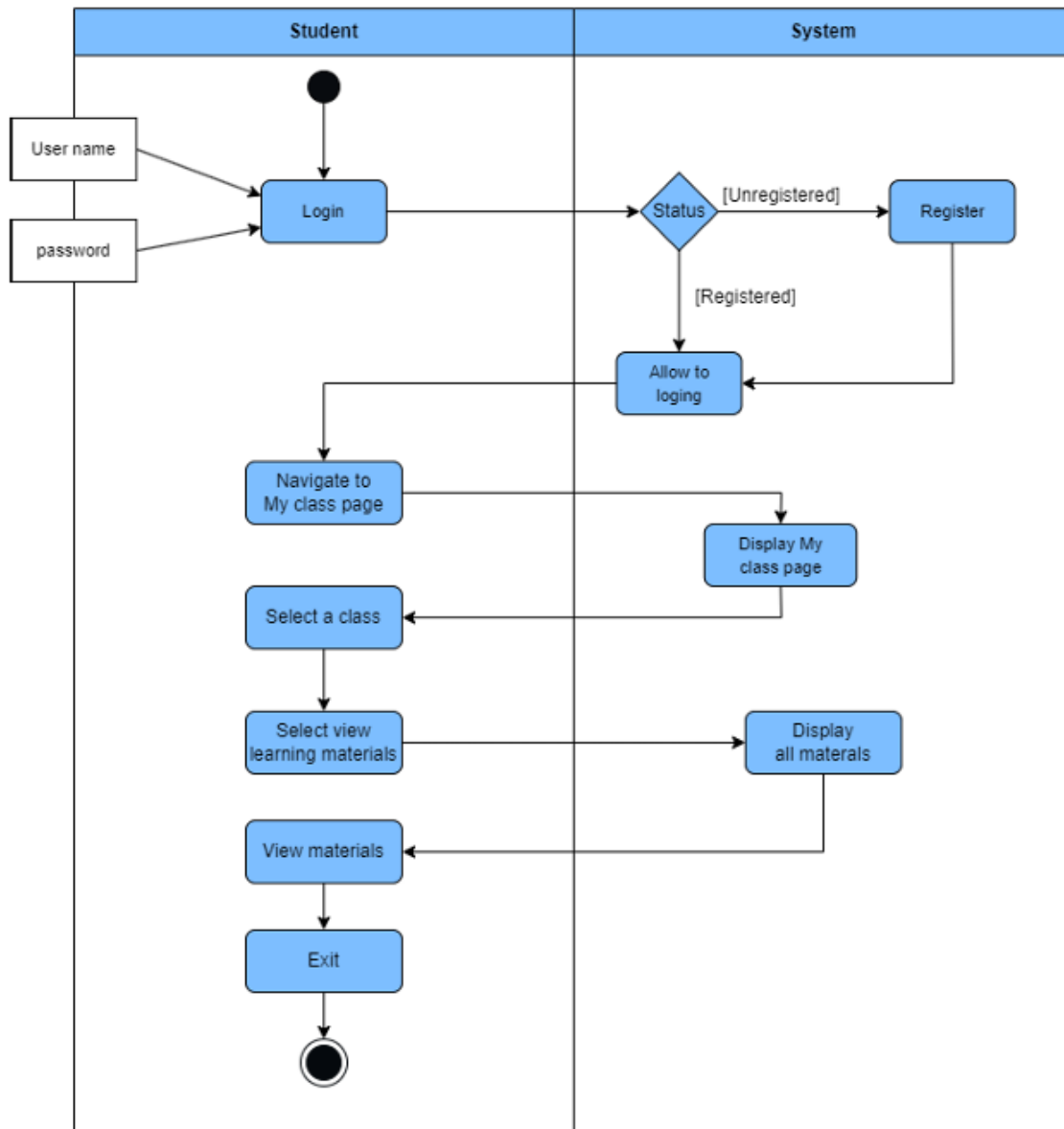




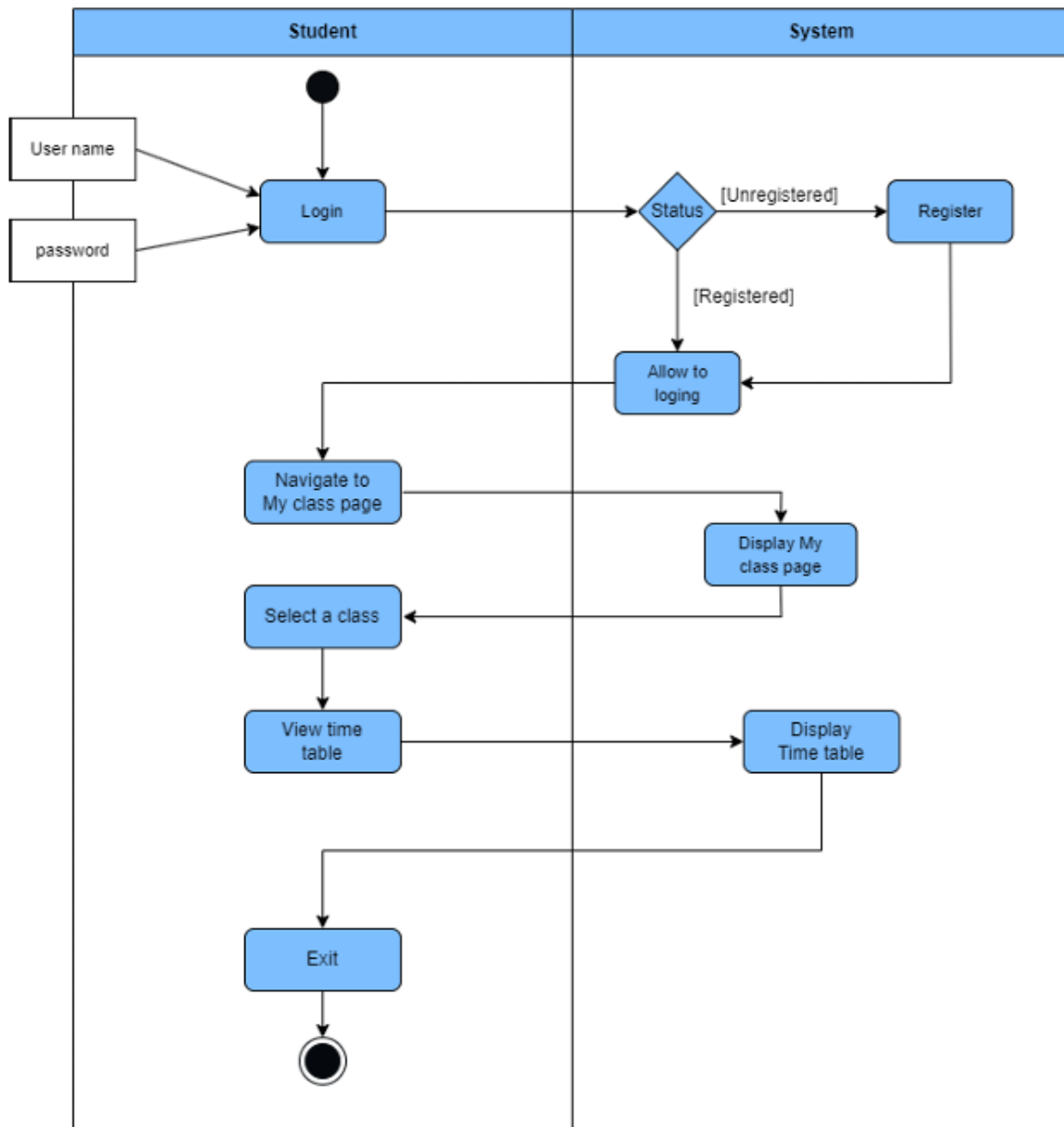


UC-004 - Contact teacher

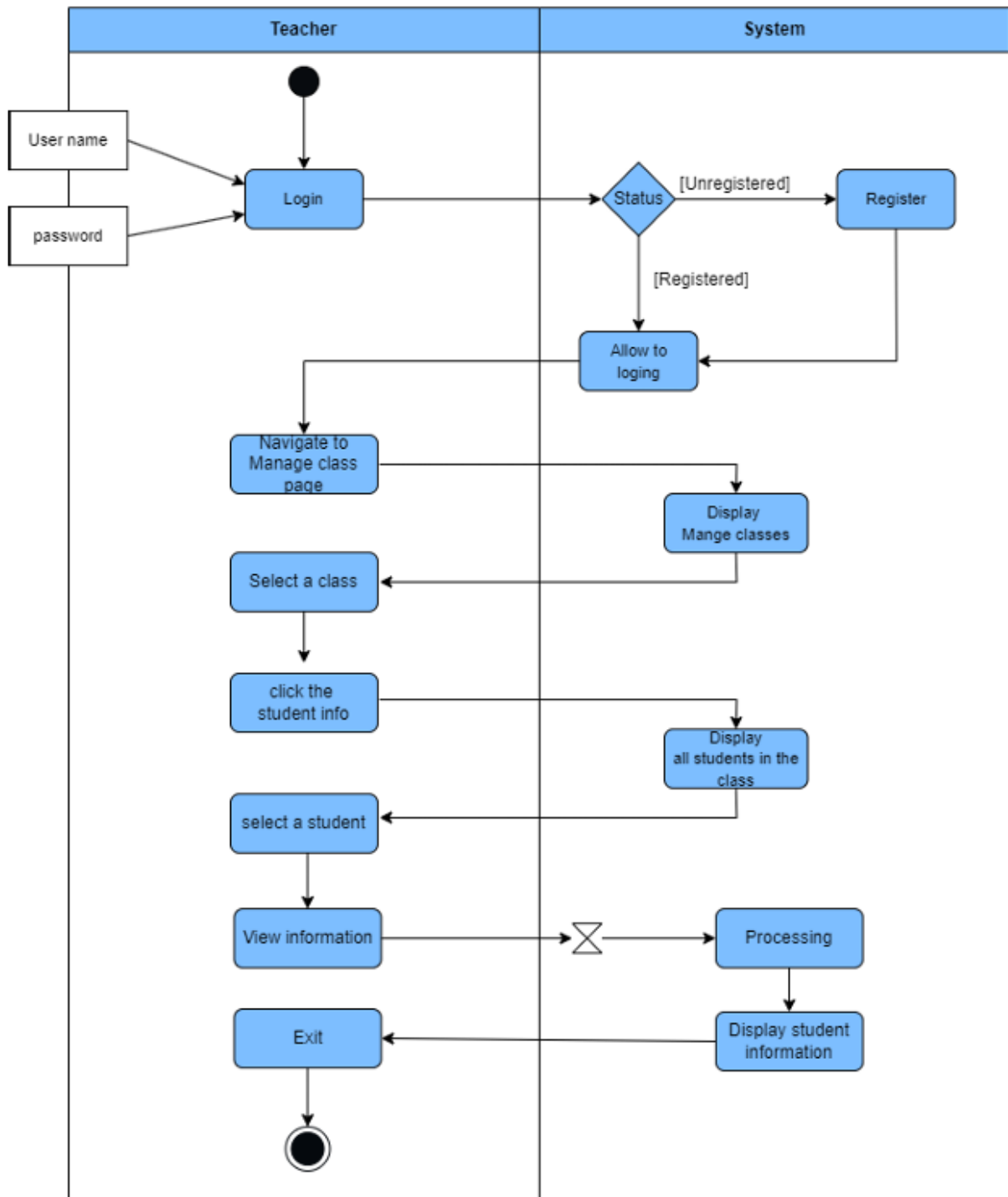




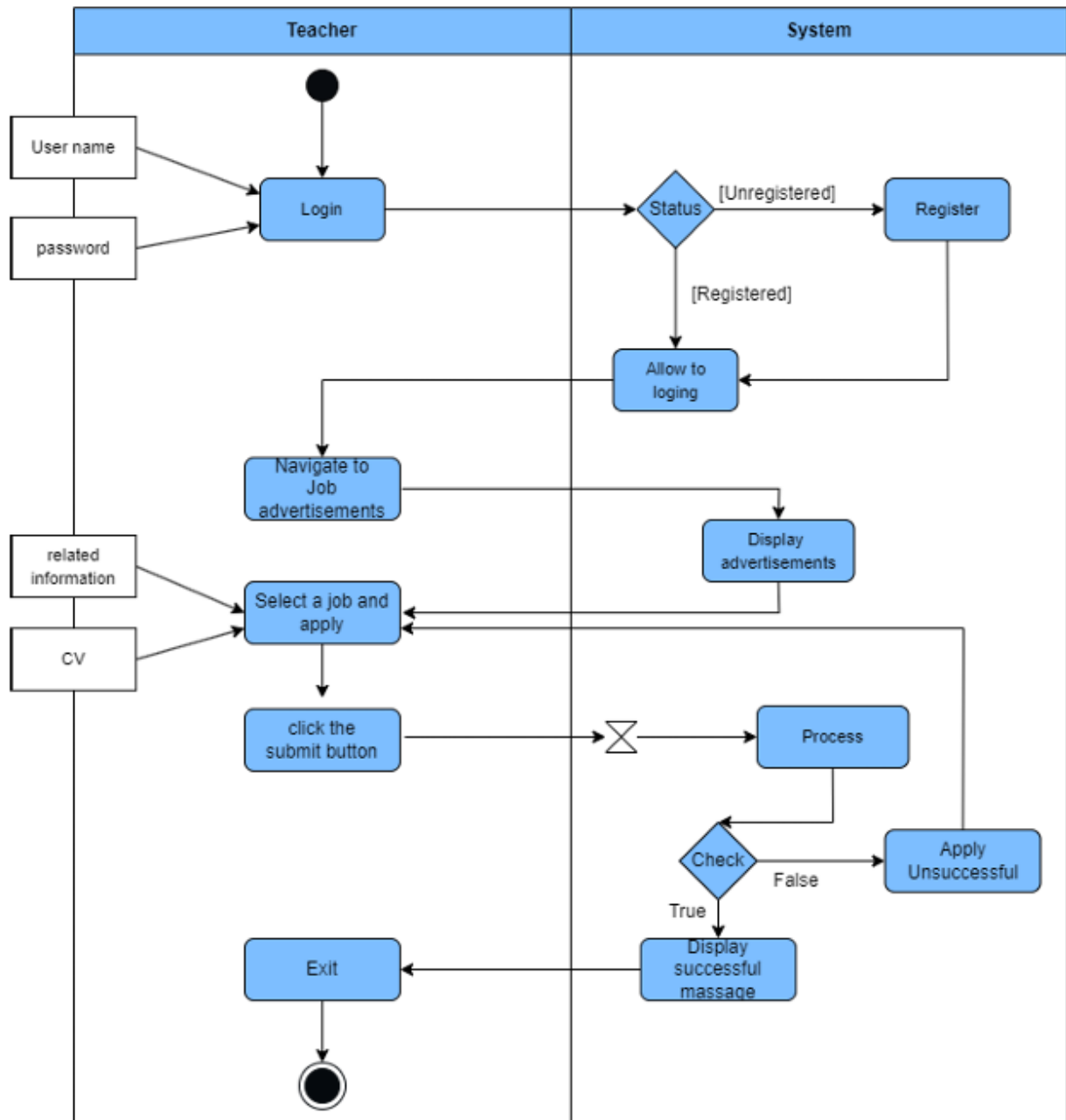
UC-006 - View class schedule



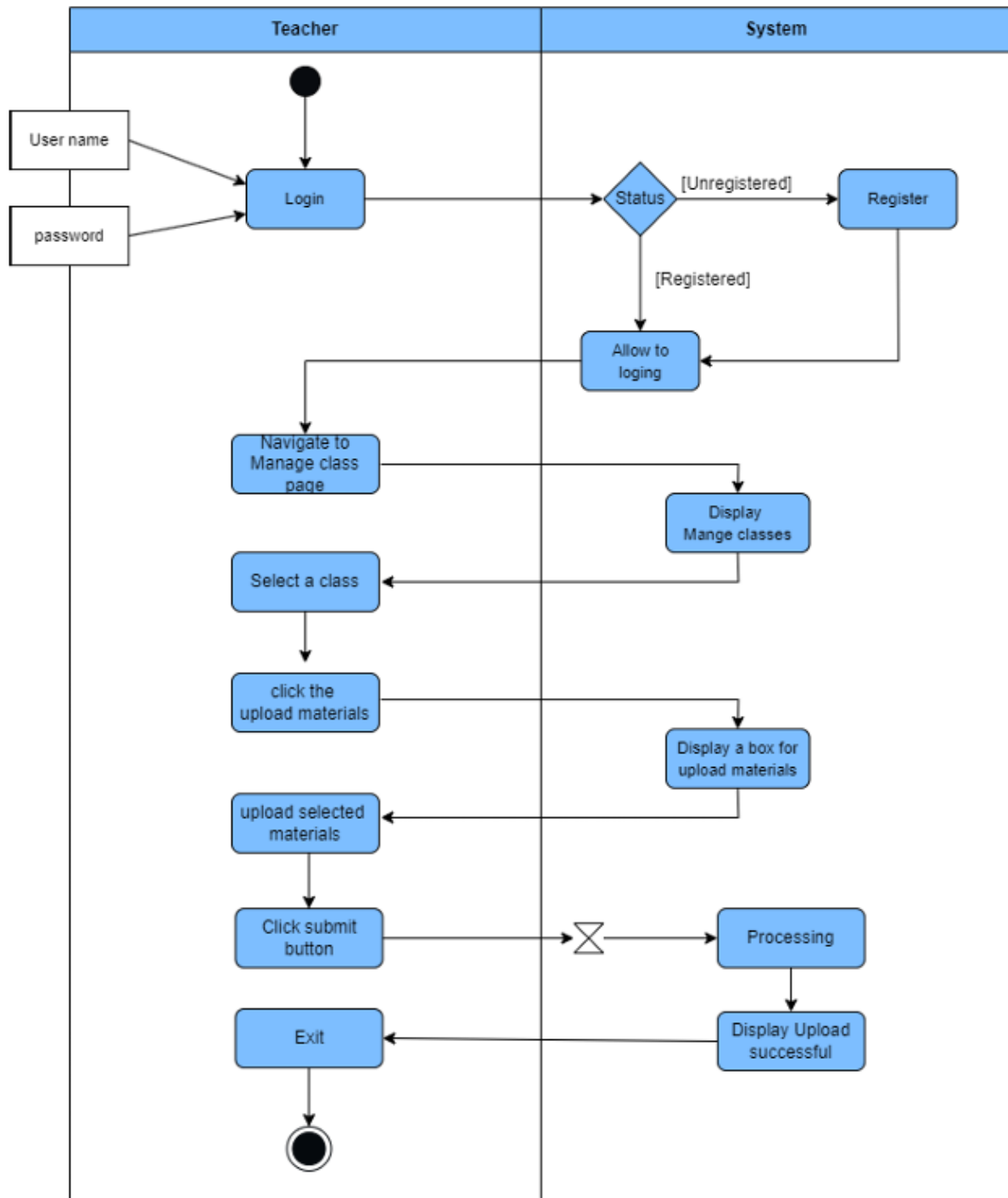
UC-008 - View students information



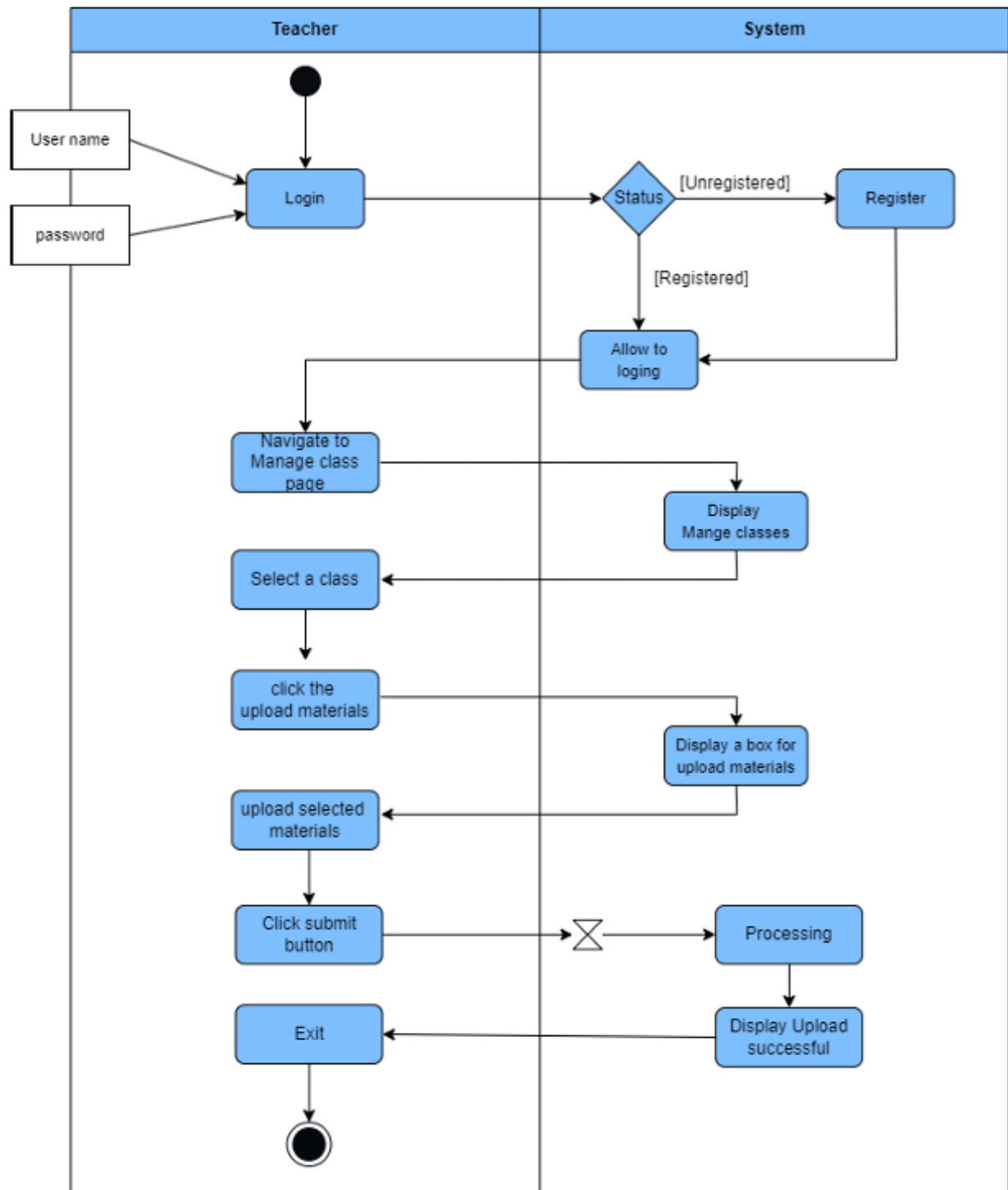
UC-009 - apply to institute

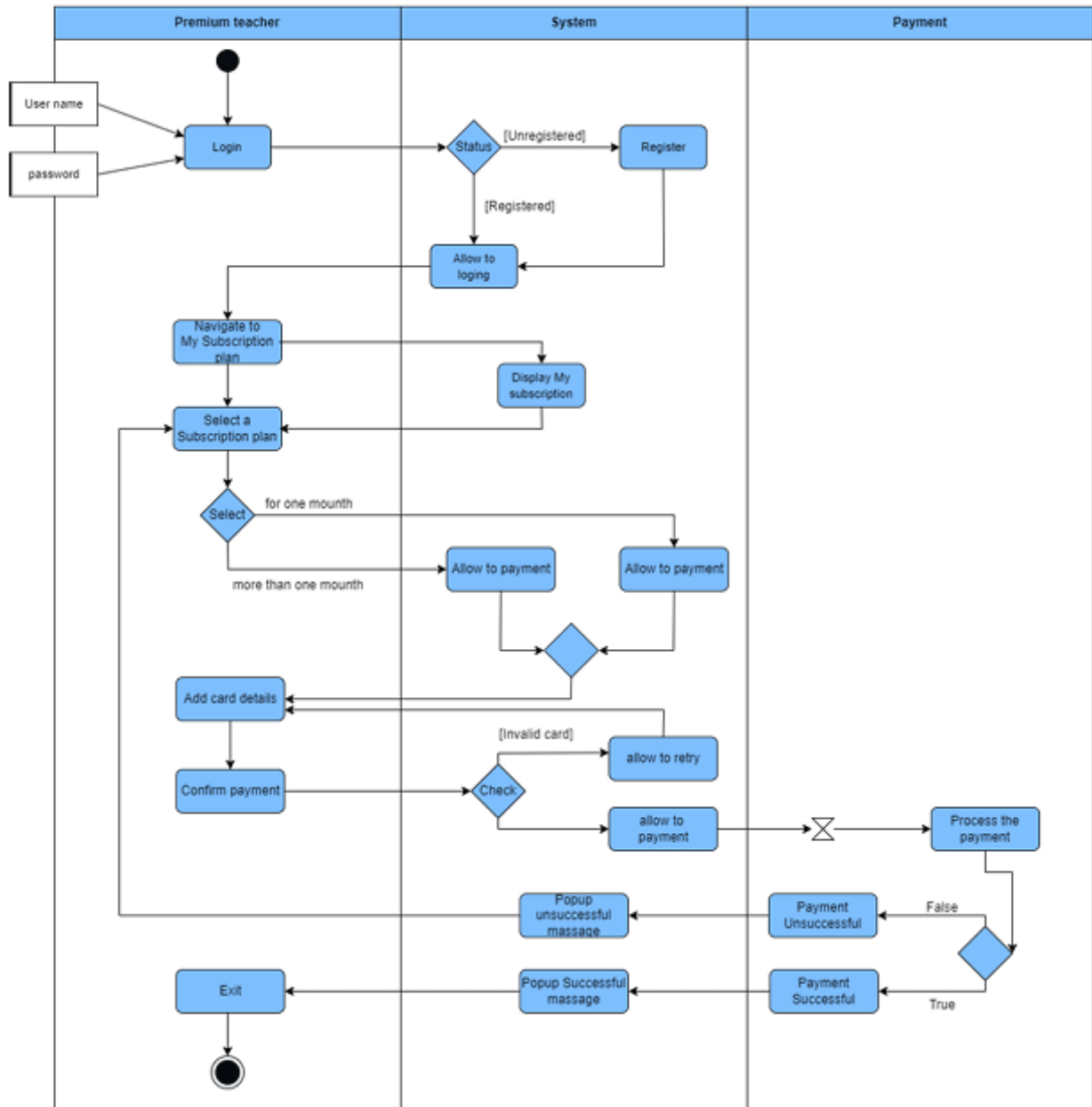


UC-010 - Upload learning materials



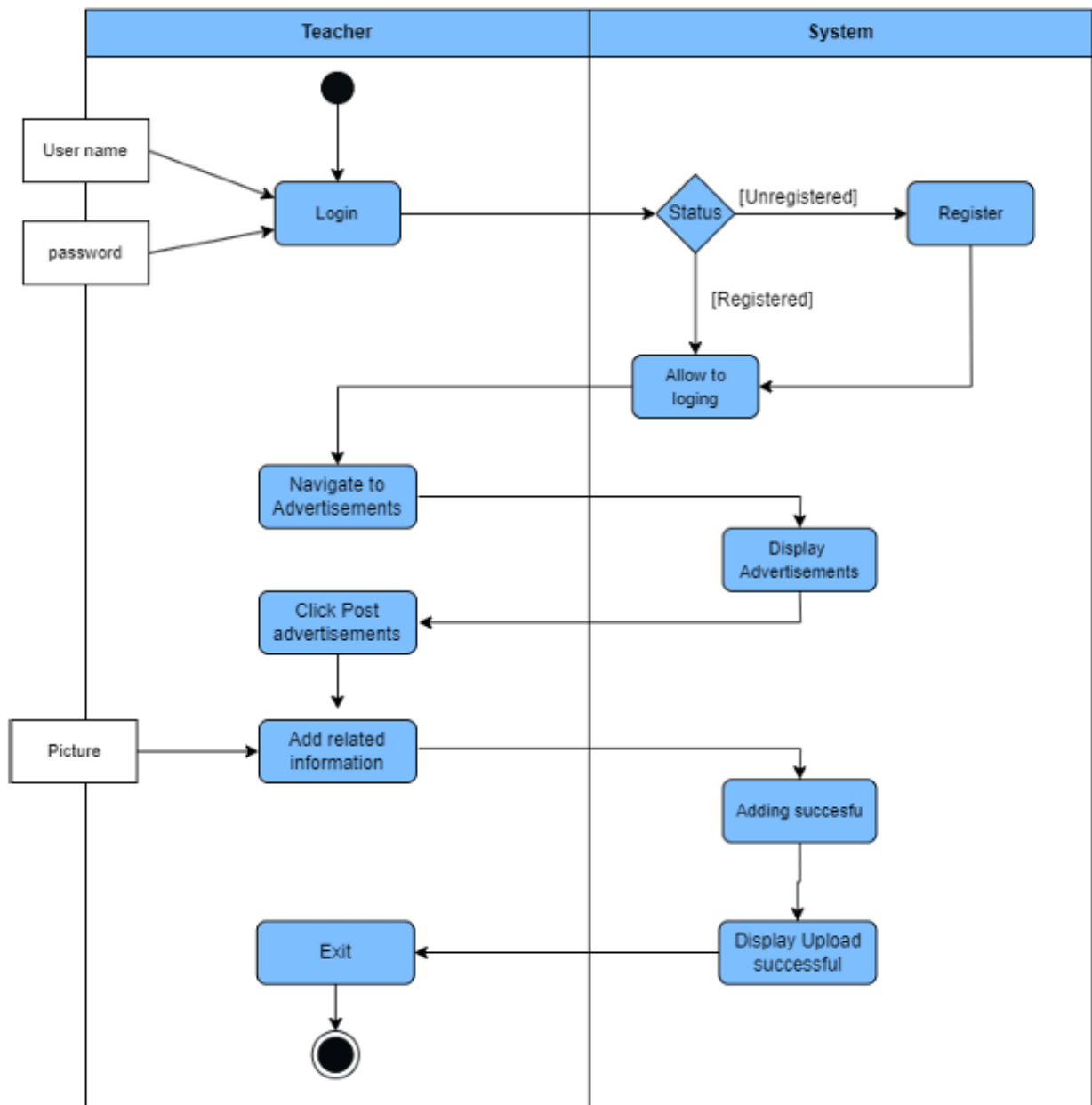
UC-011 - Schedule classes



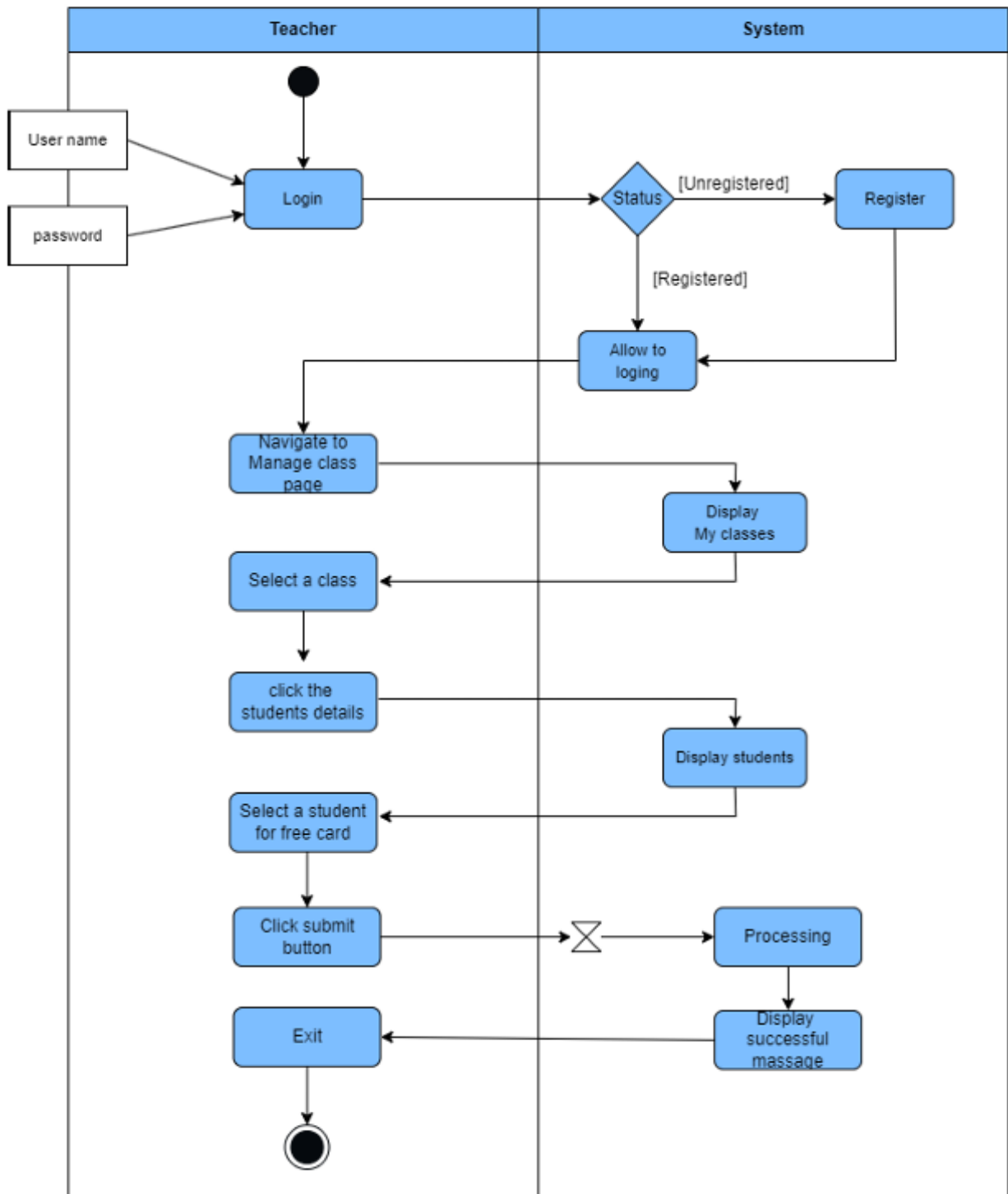




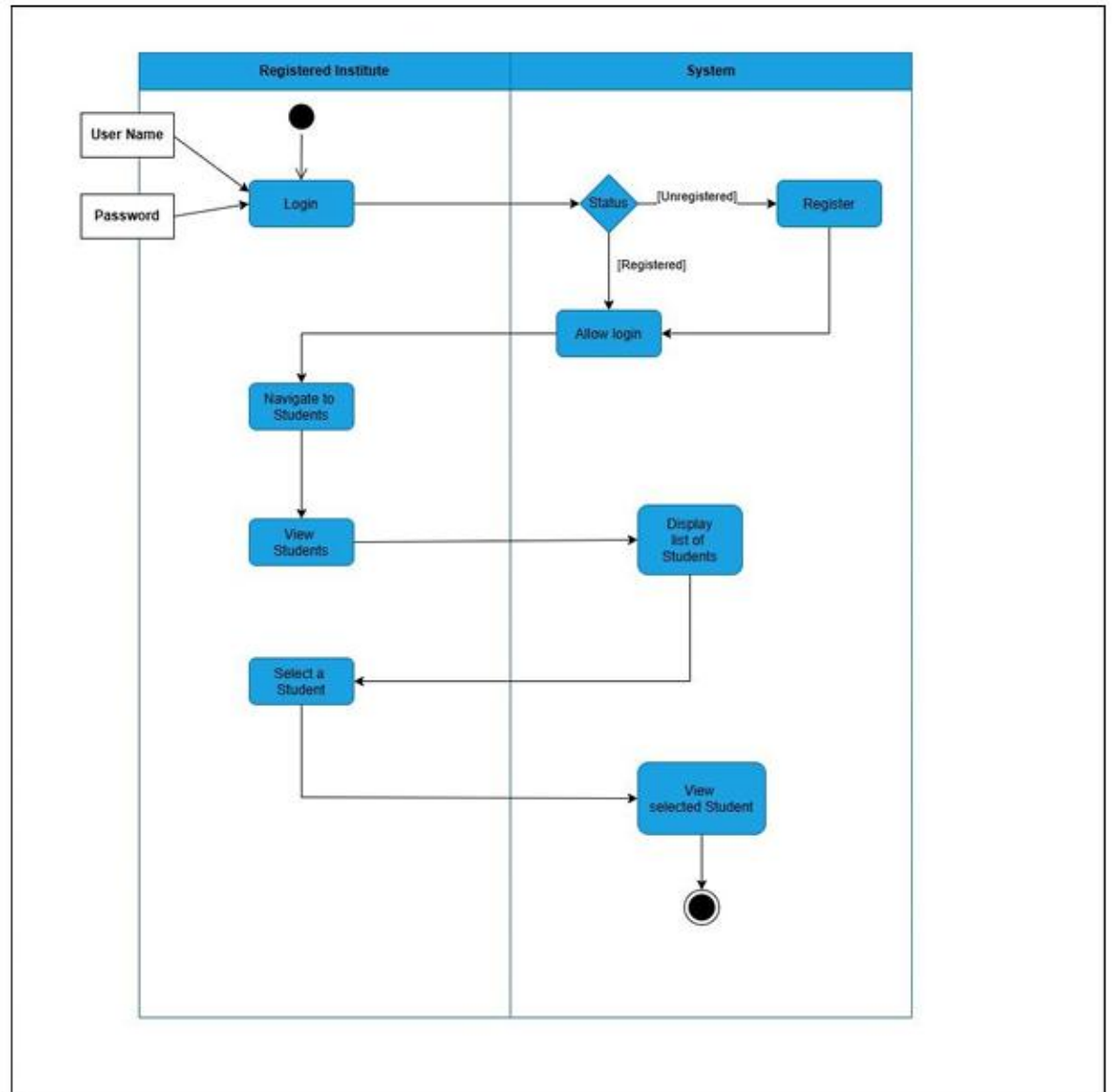
UC-013 - Post advertisement



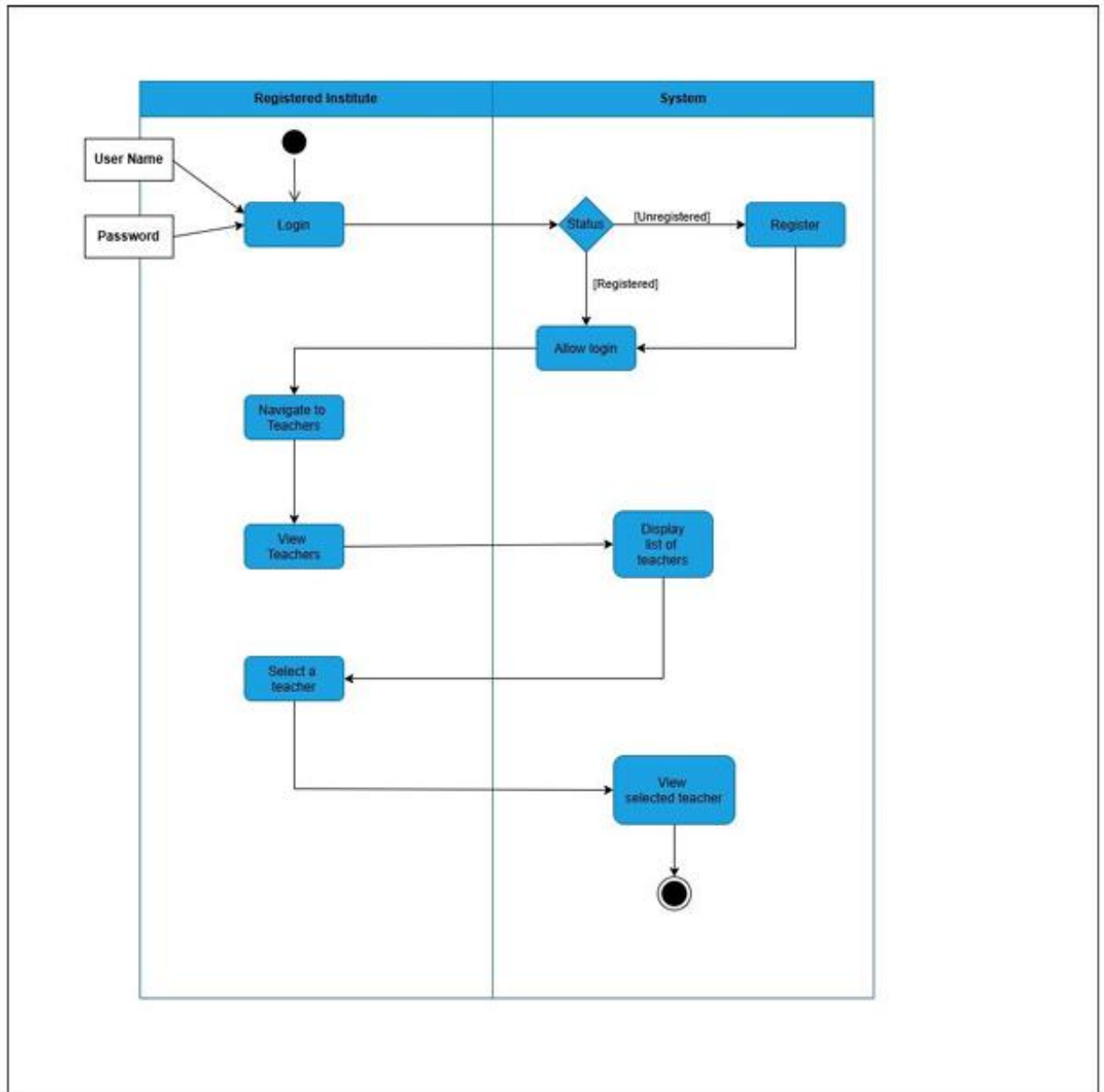
UC-014 - Offer discounts and free cards

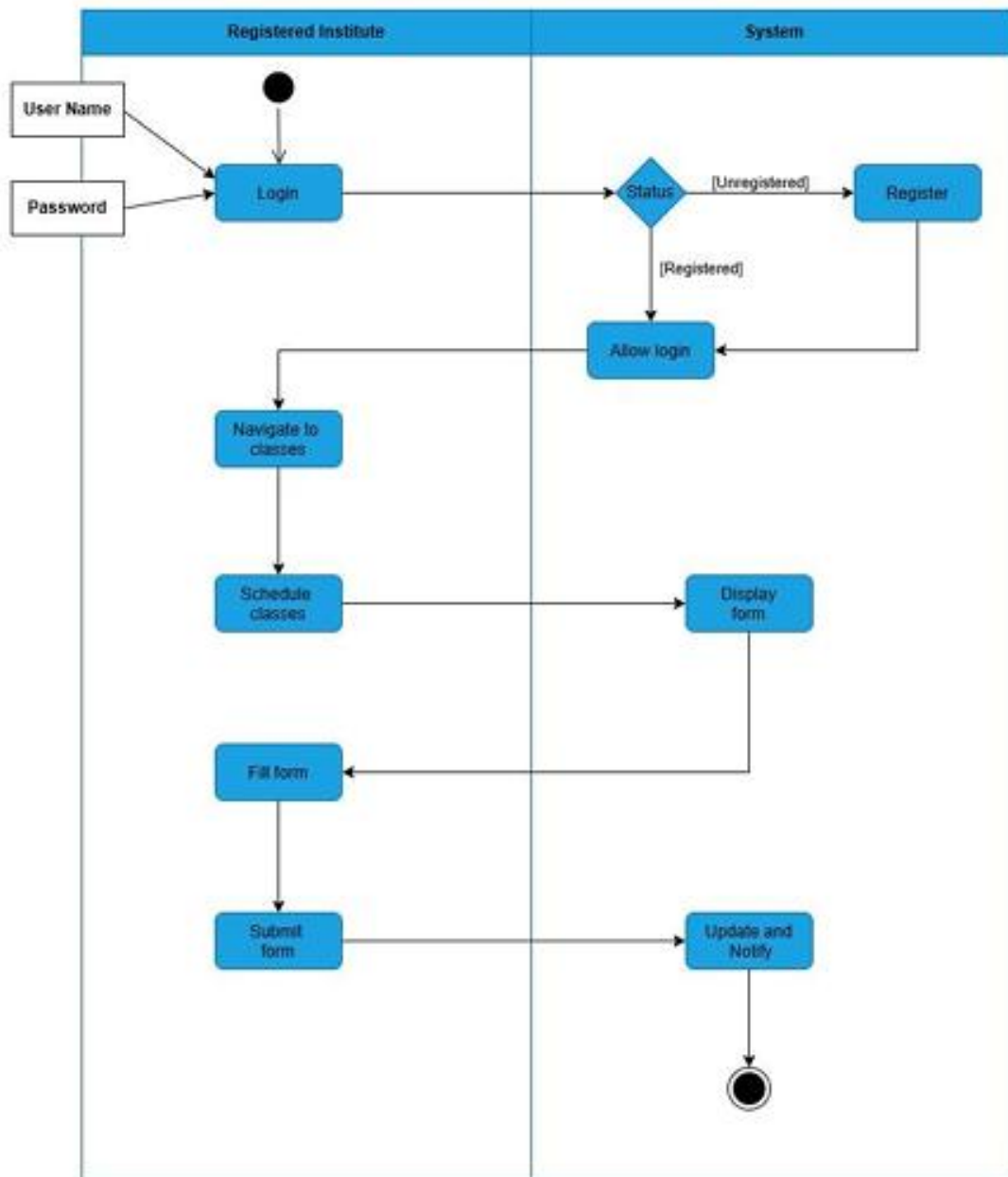


## View Students



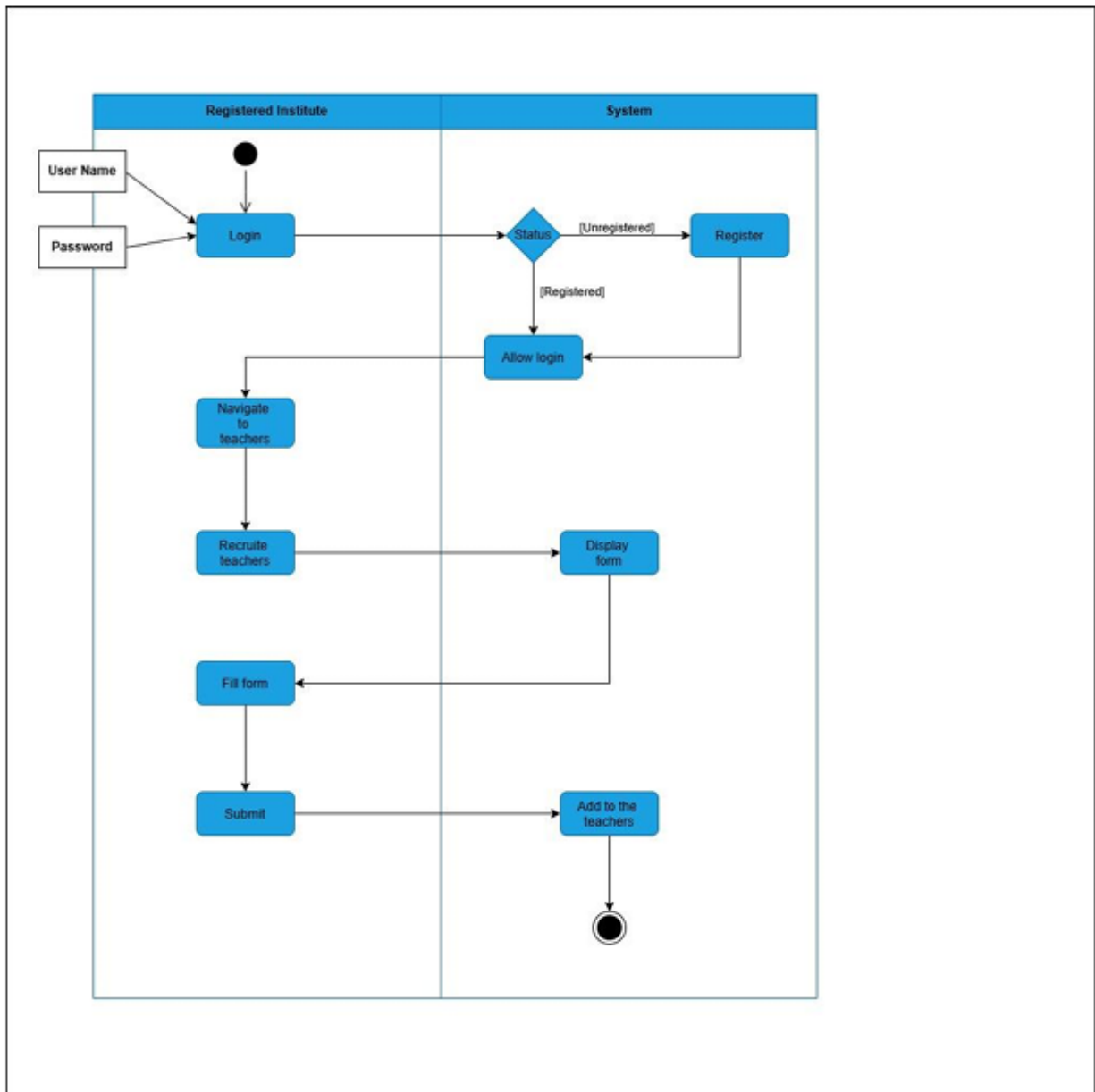
## View teachers



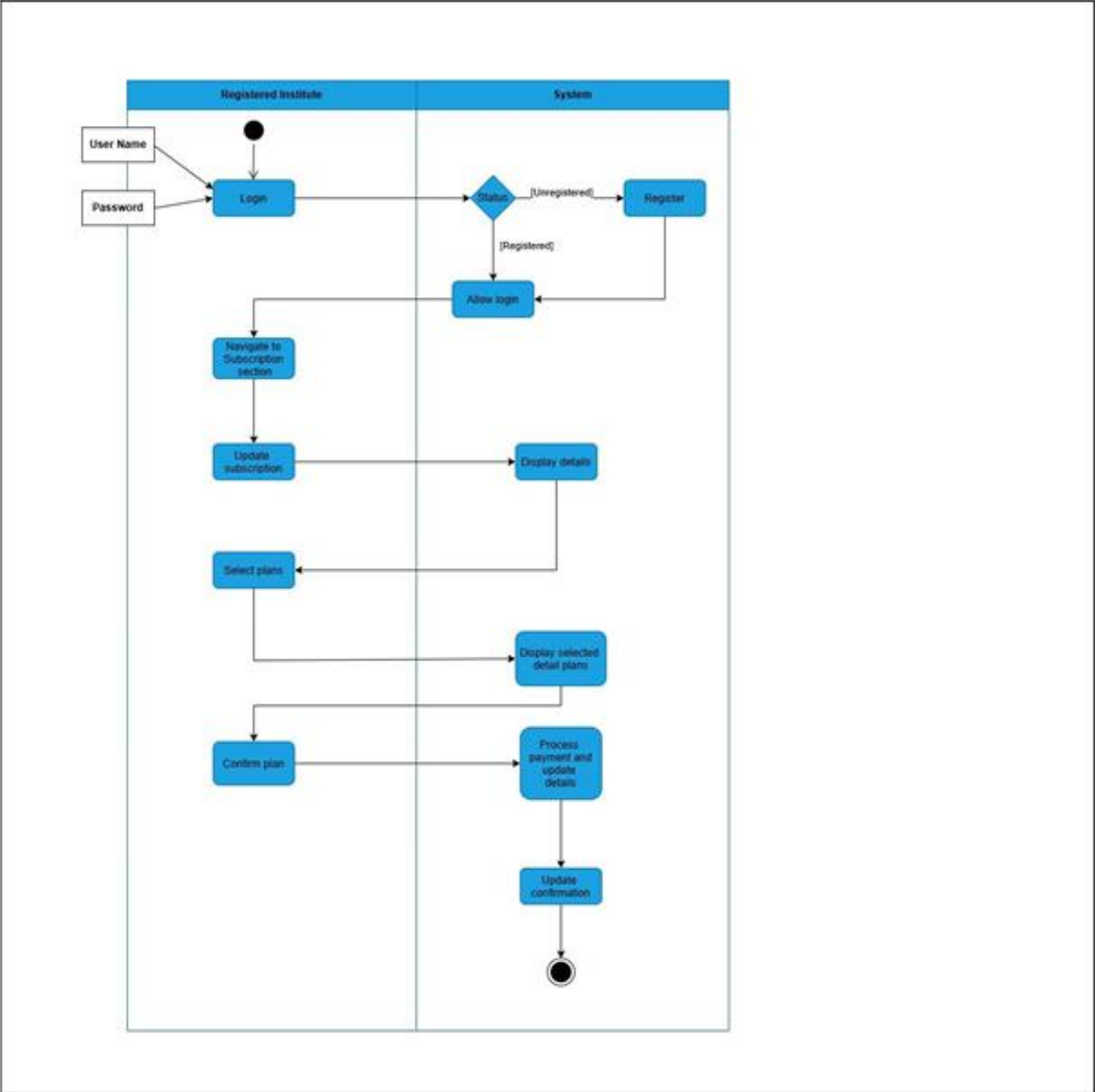


## Recruite teachers

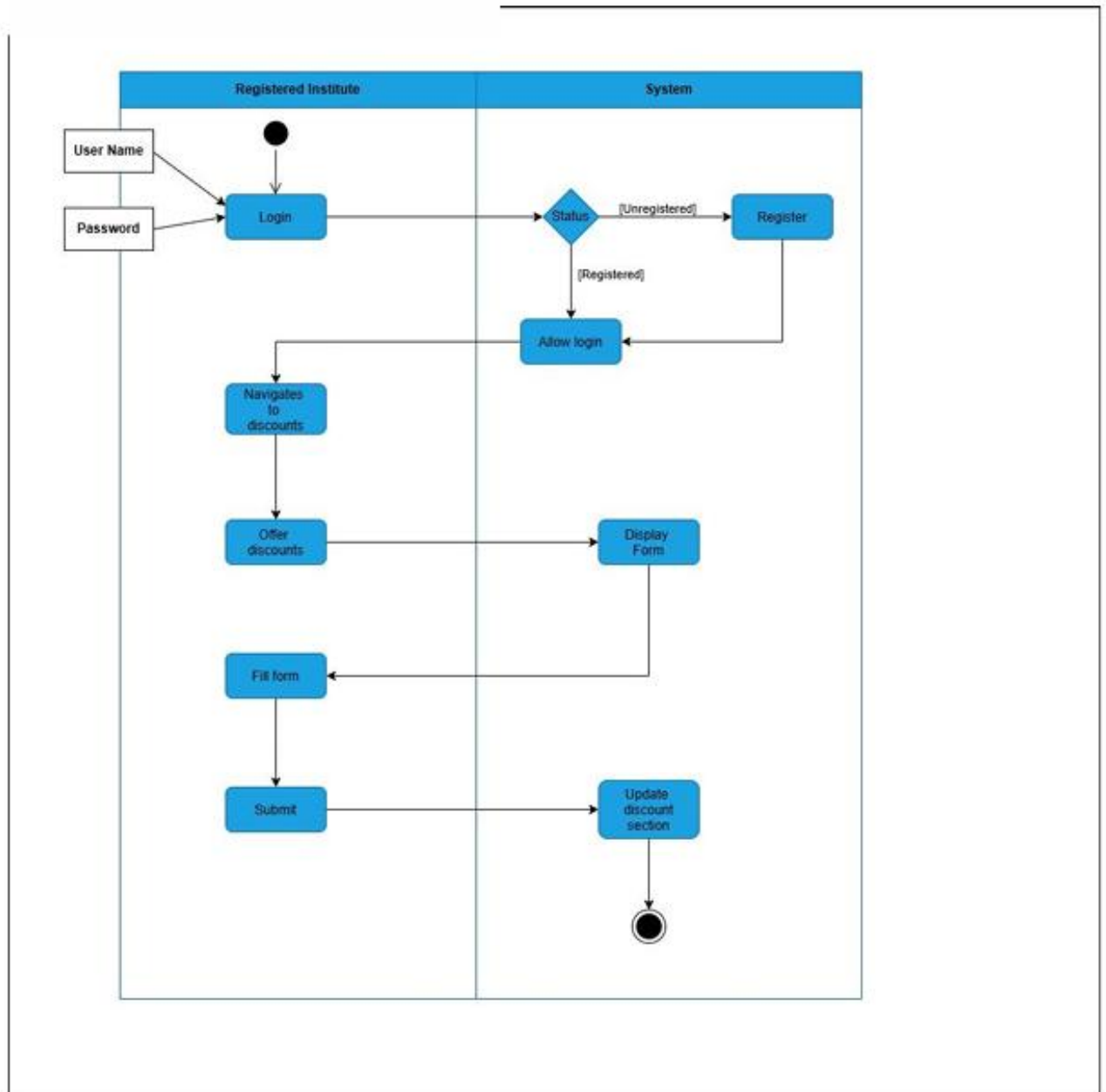
8



Buy subscription

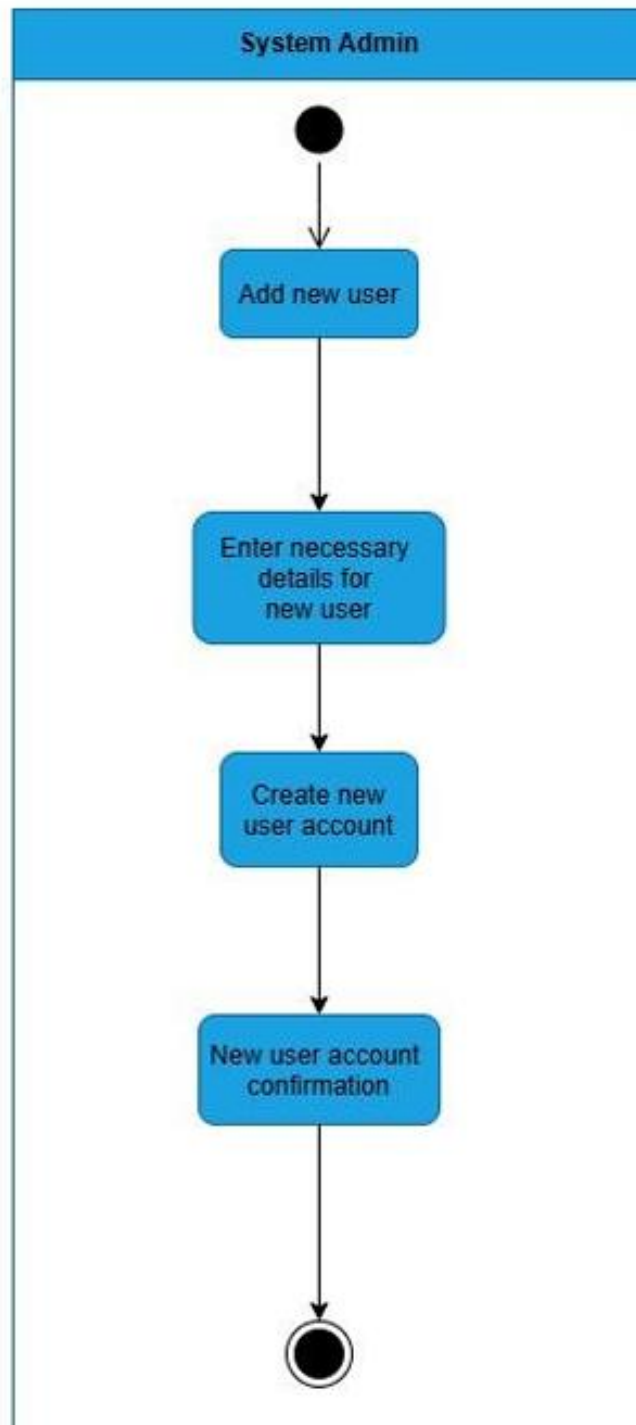


## Issue discount for students

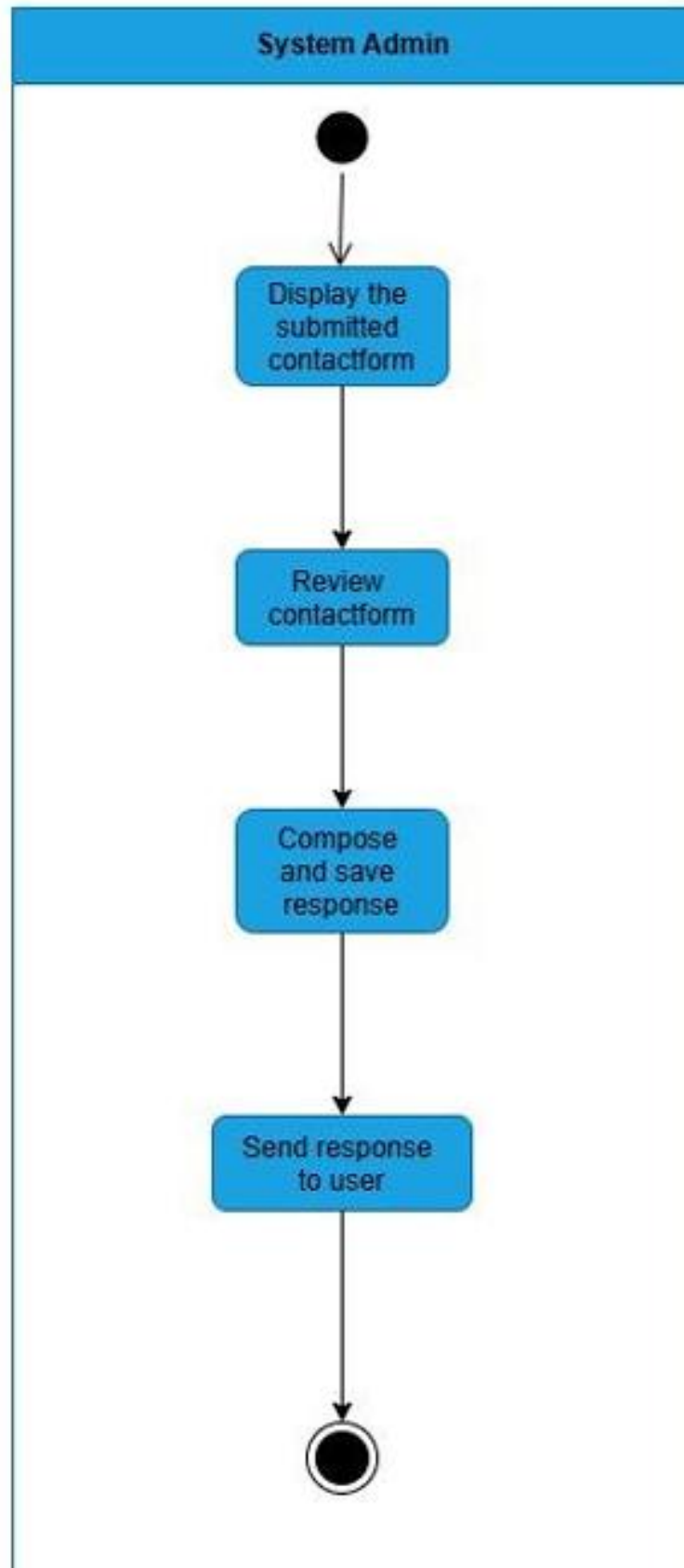




## Add new user

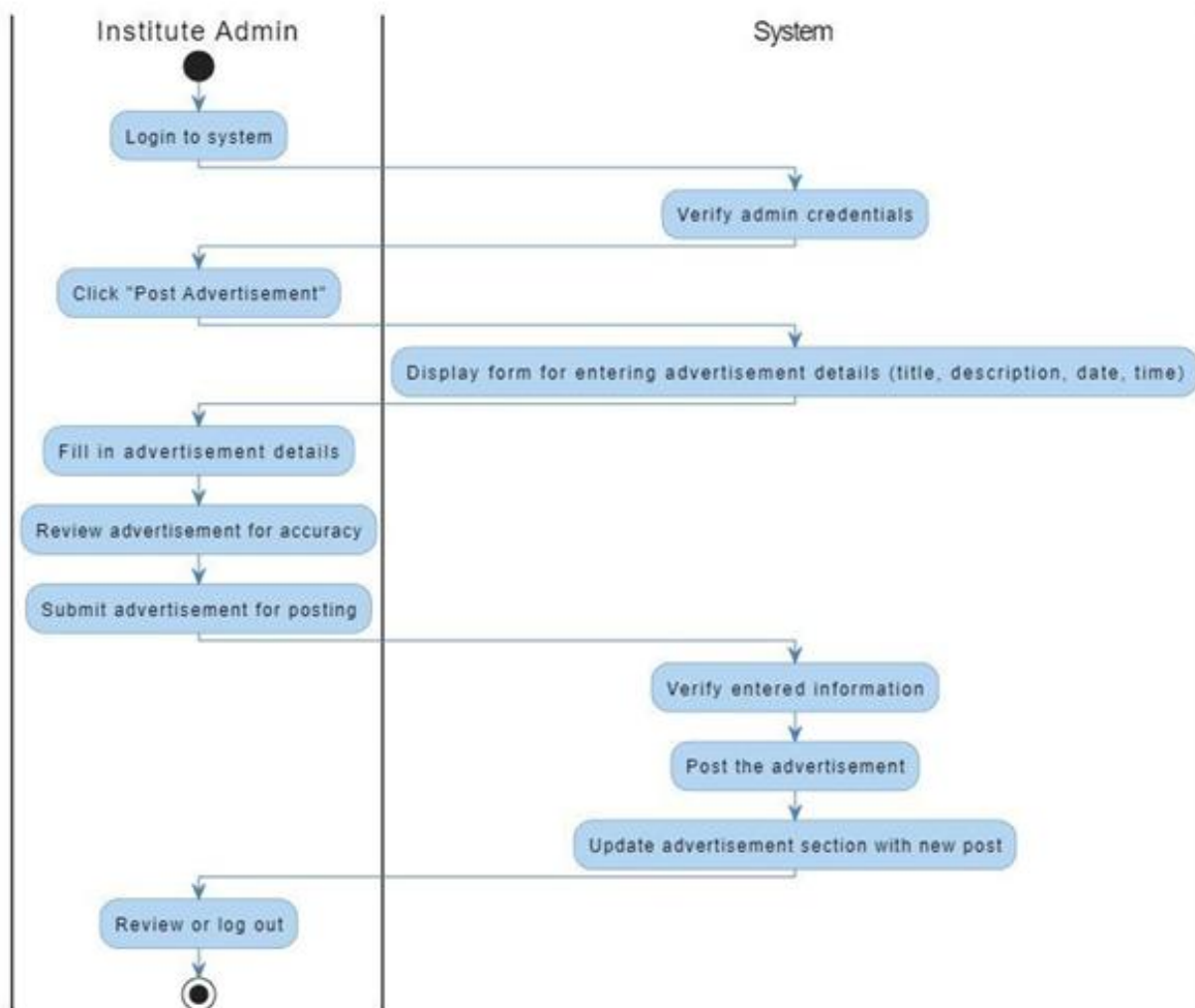


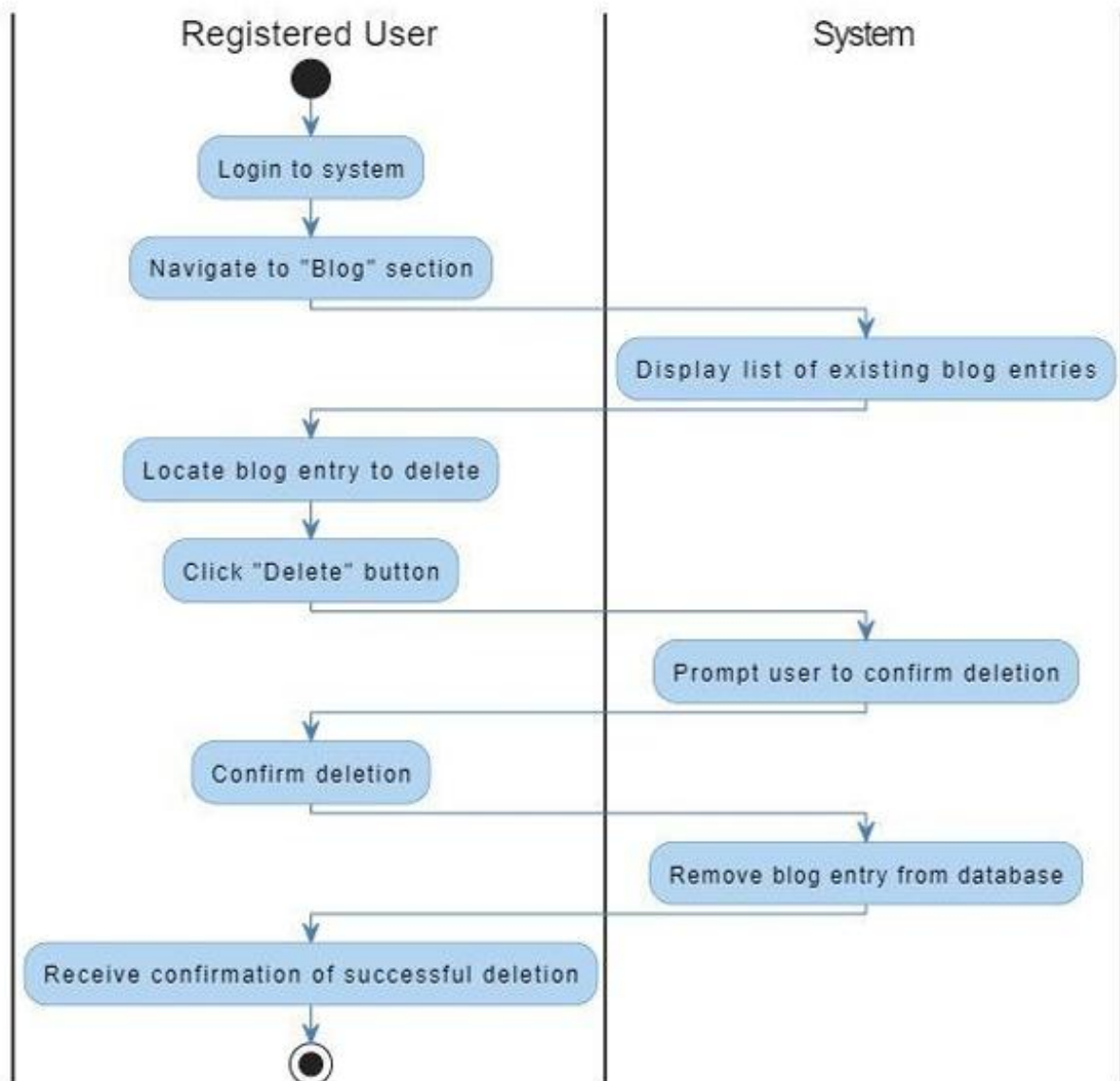
View submitted contact form by  
admin

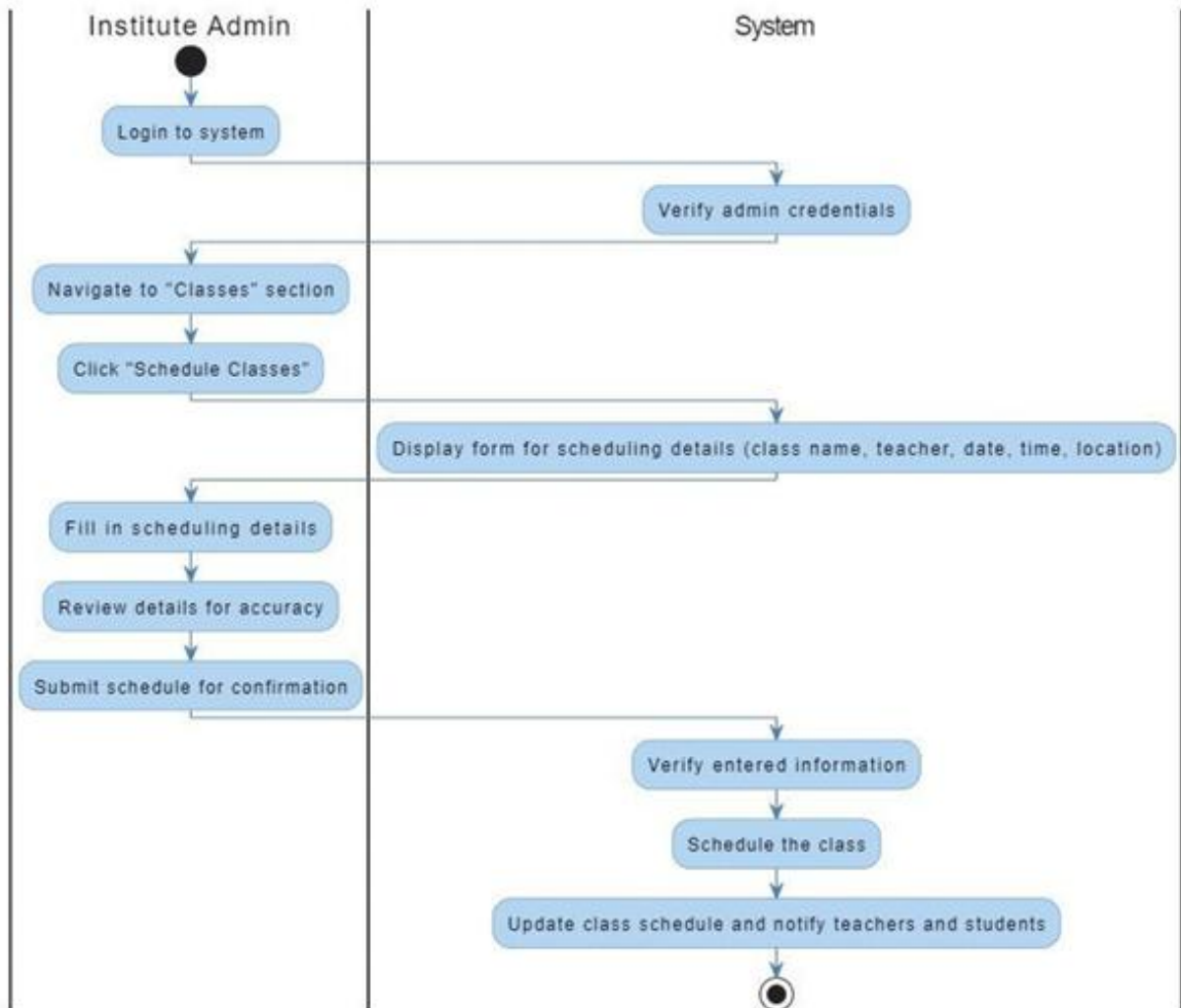




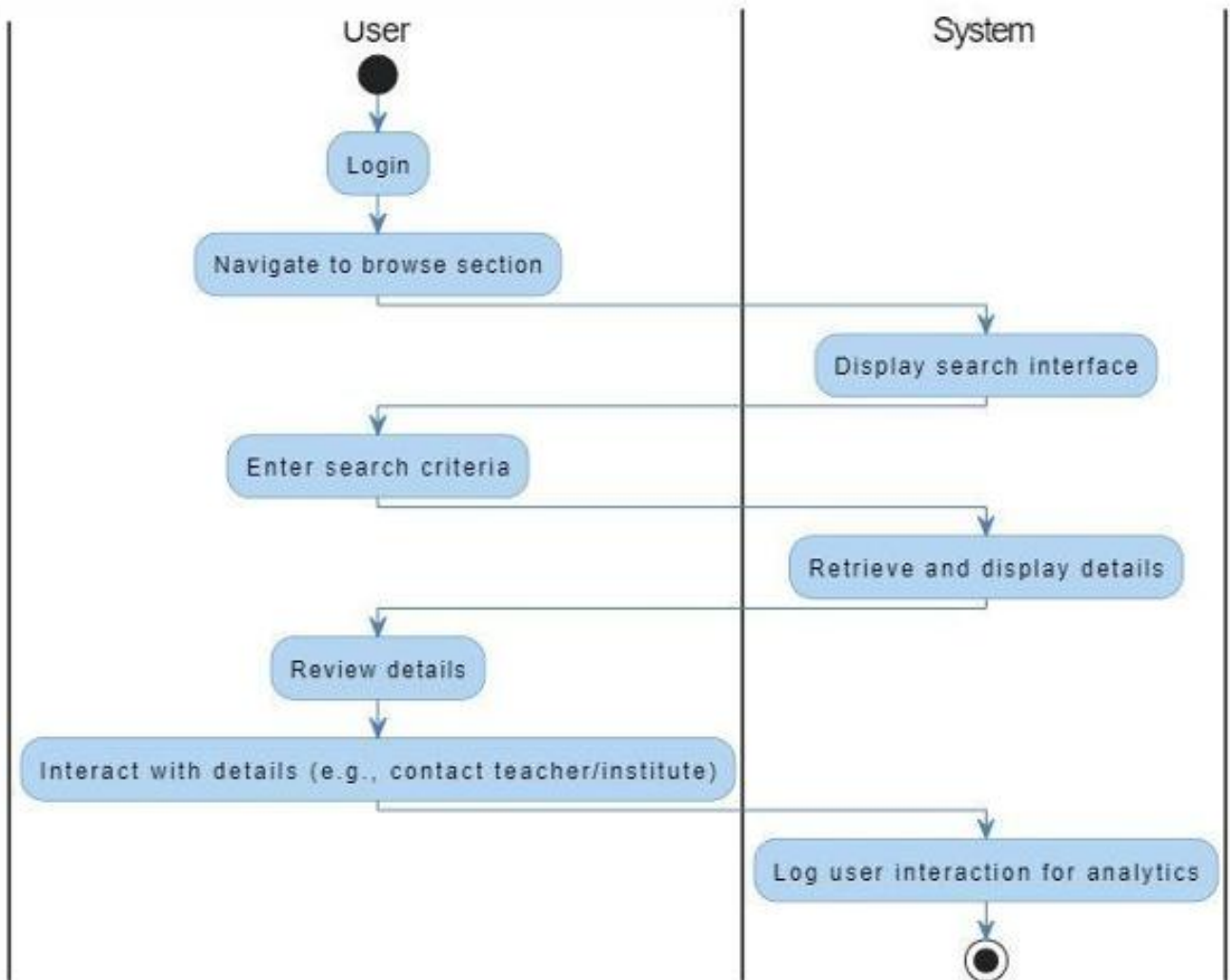
## View Advertisements

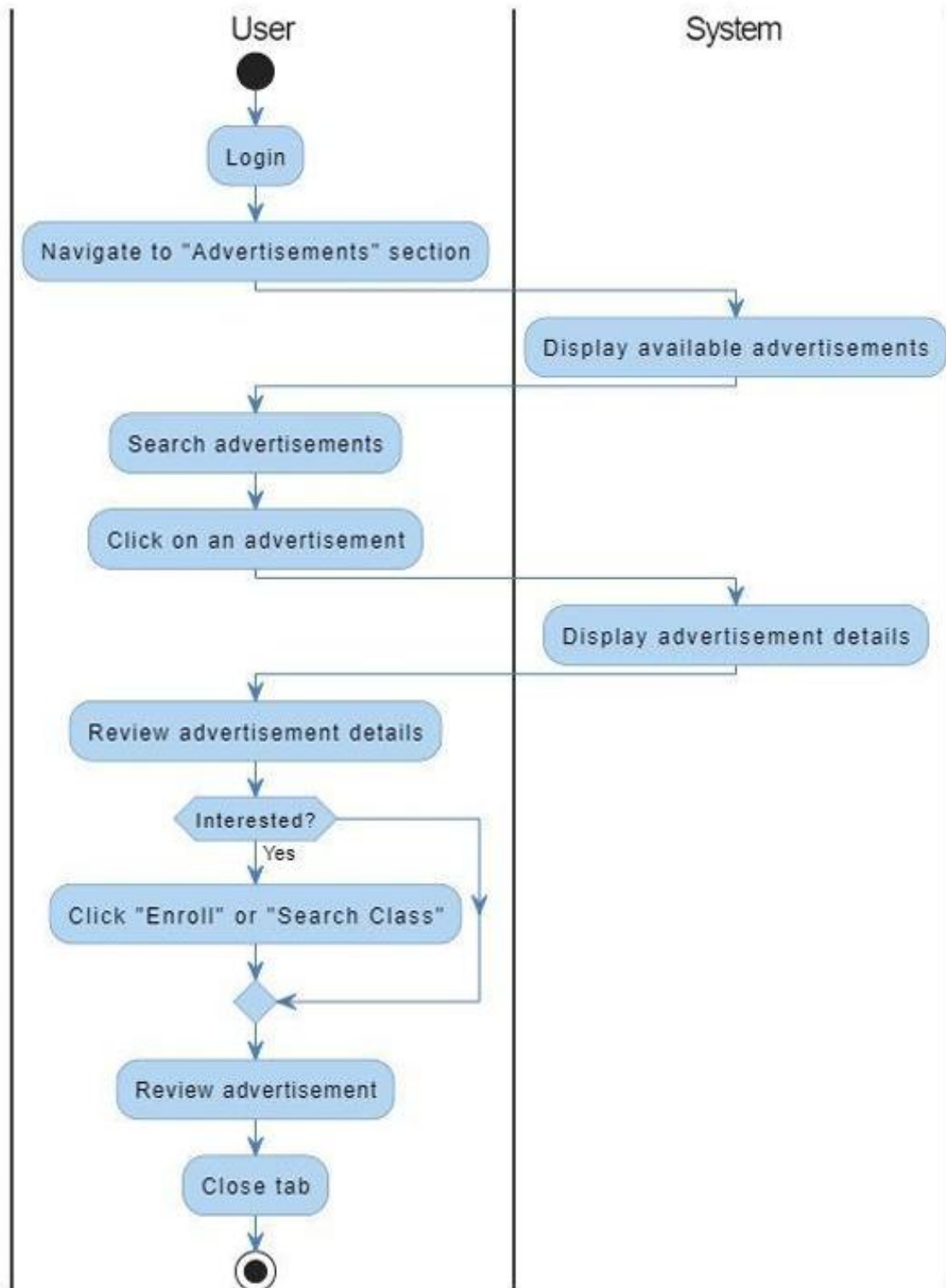


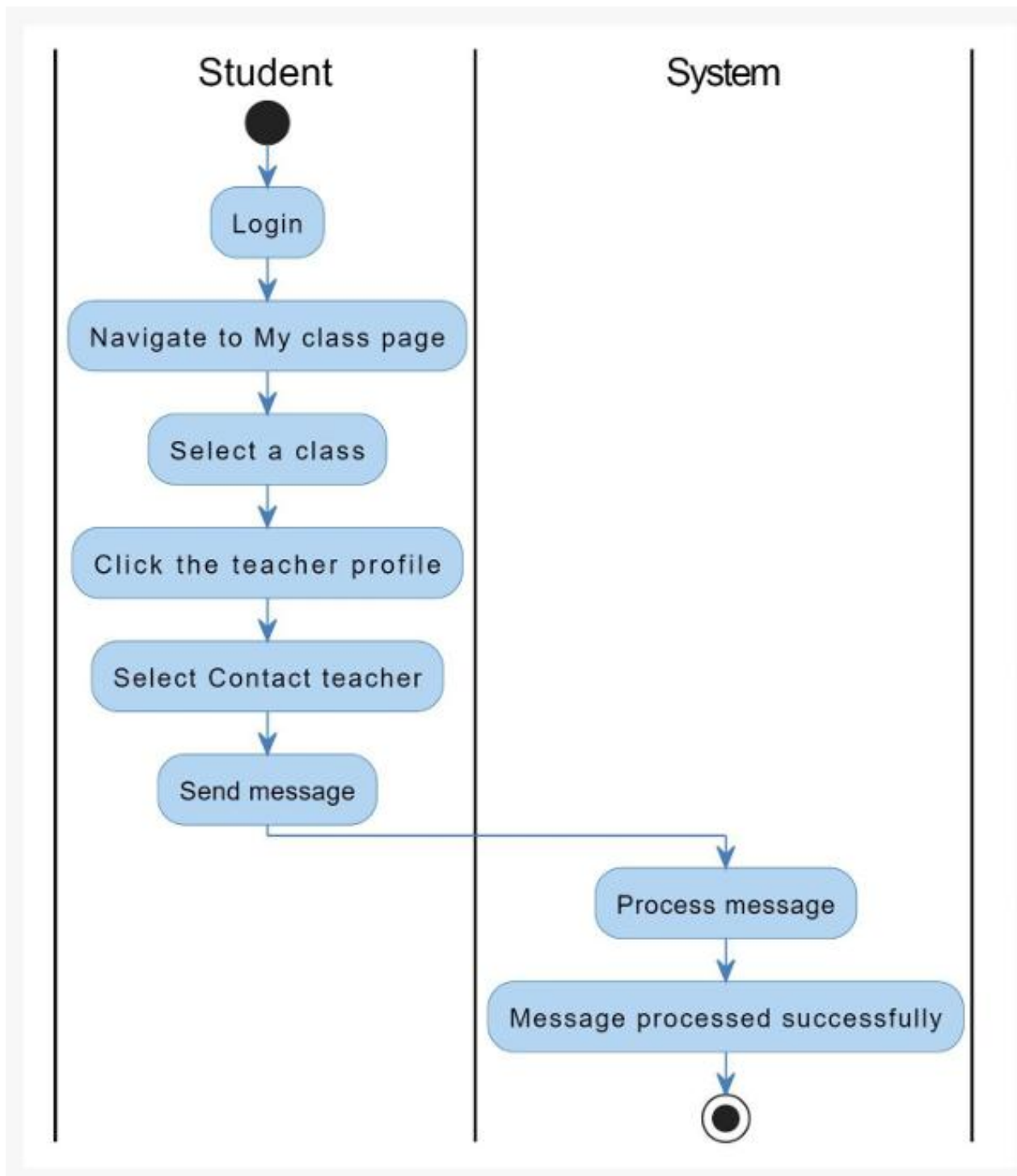




## Contact support

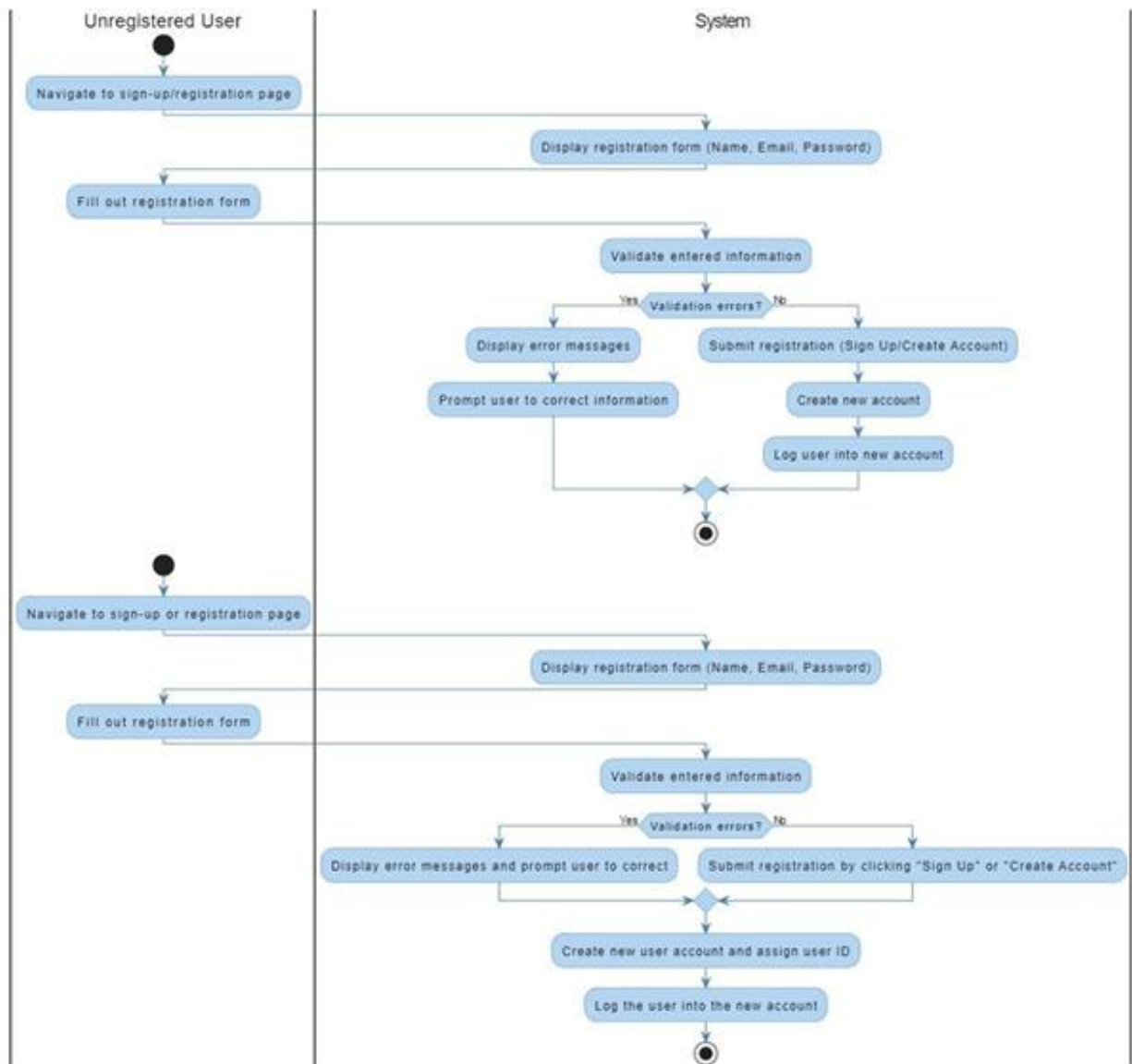


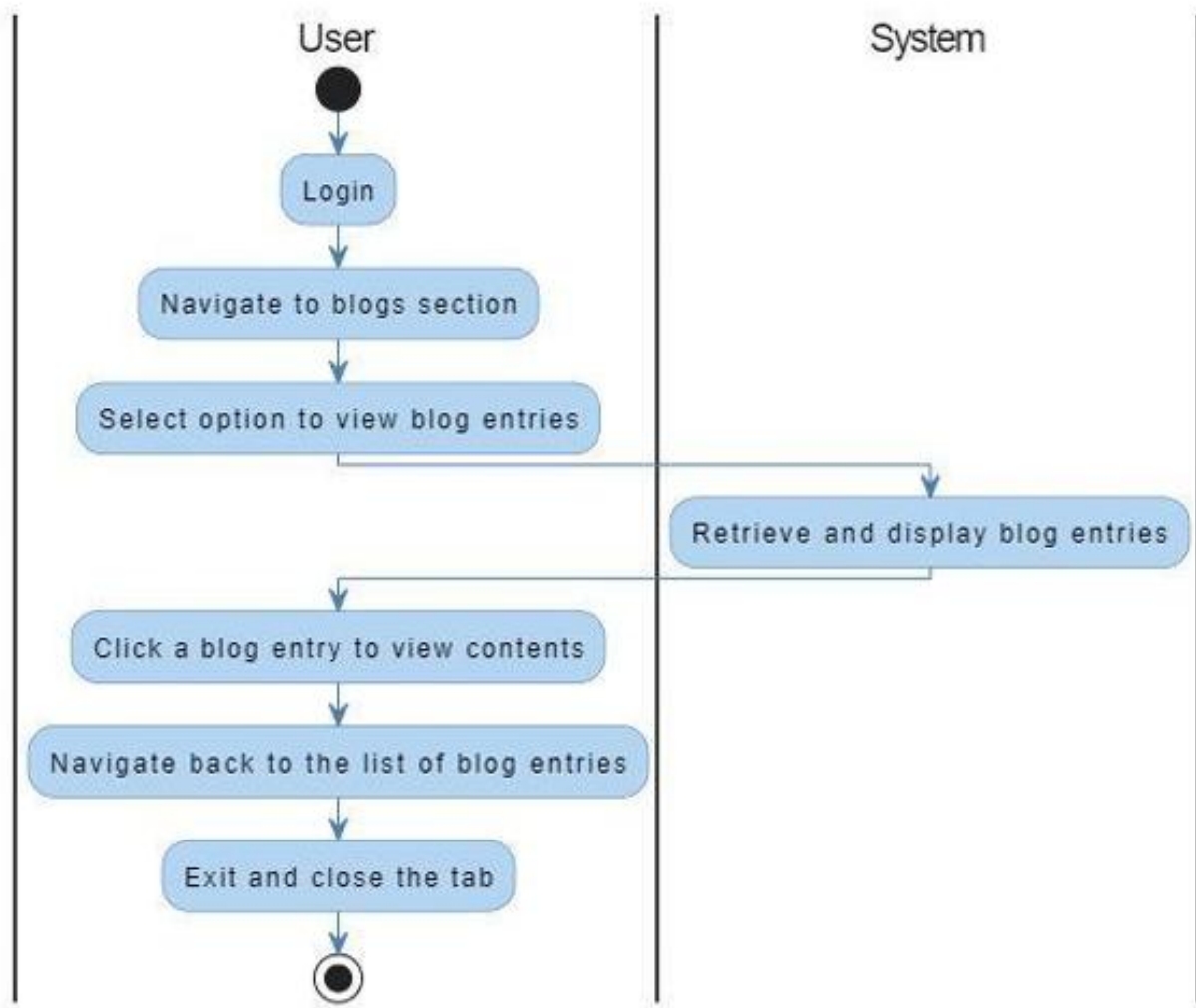




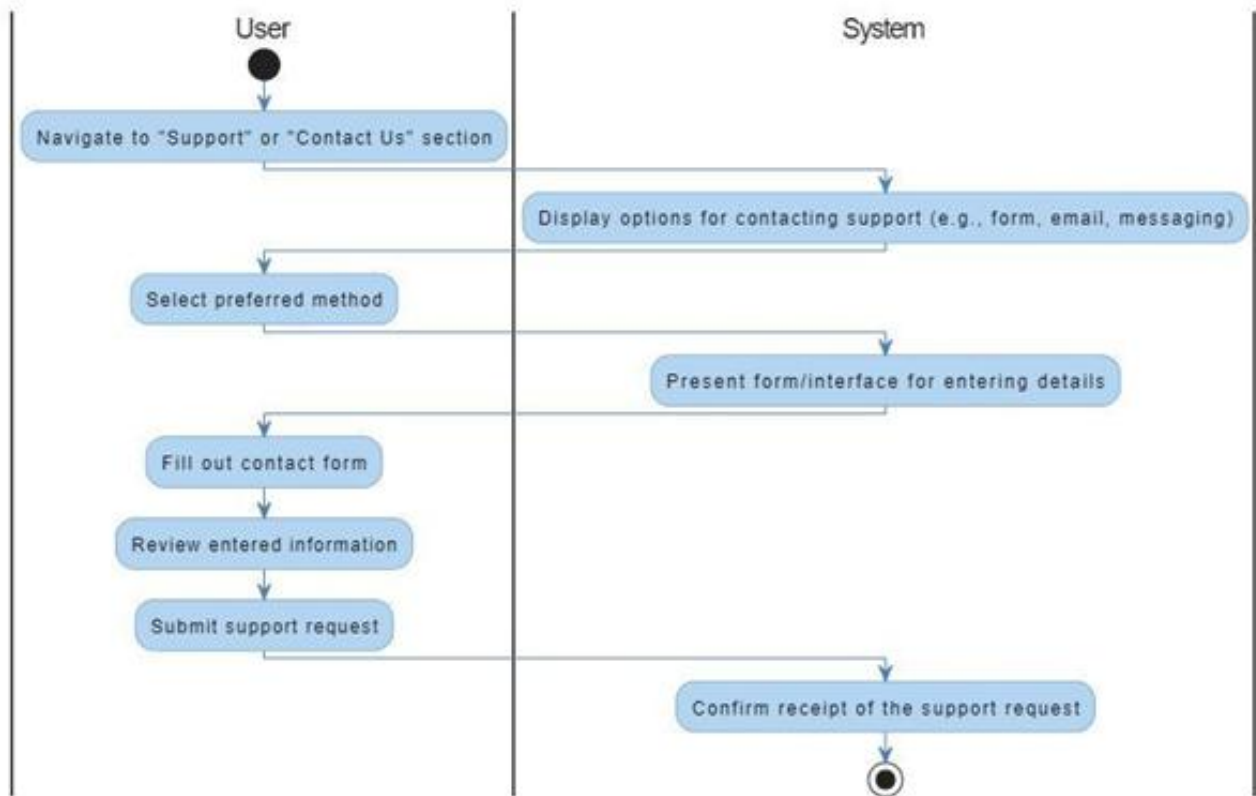


## Sign up and login

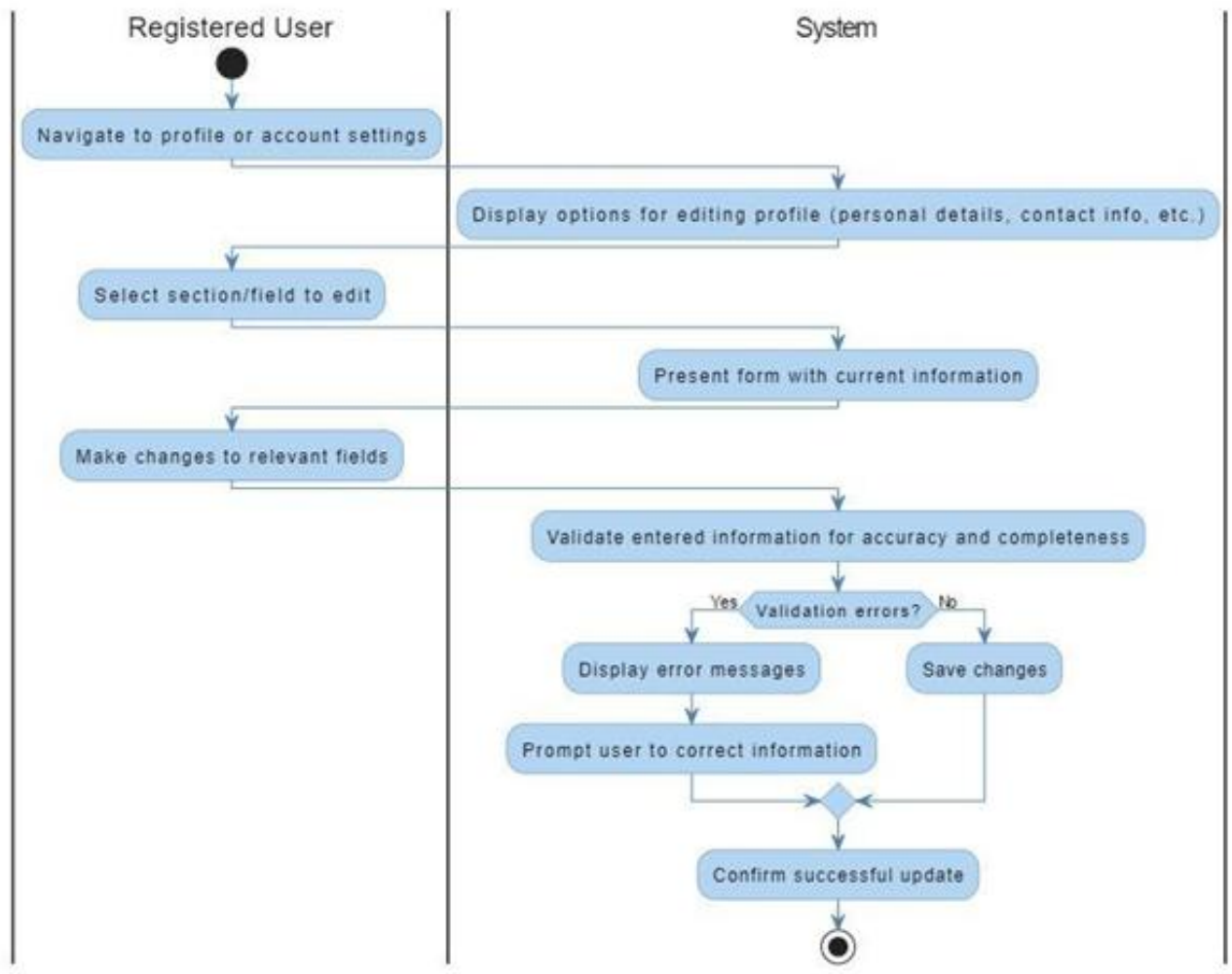




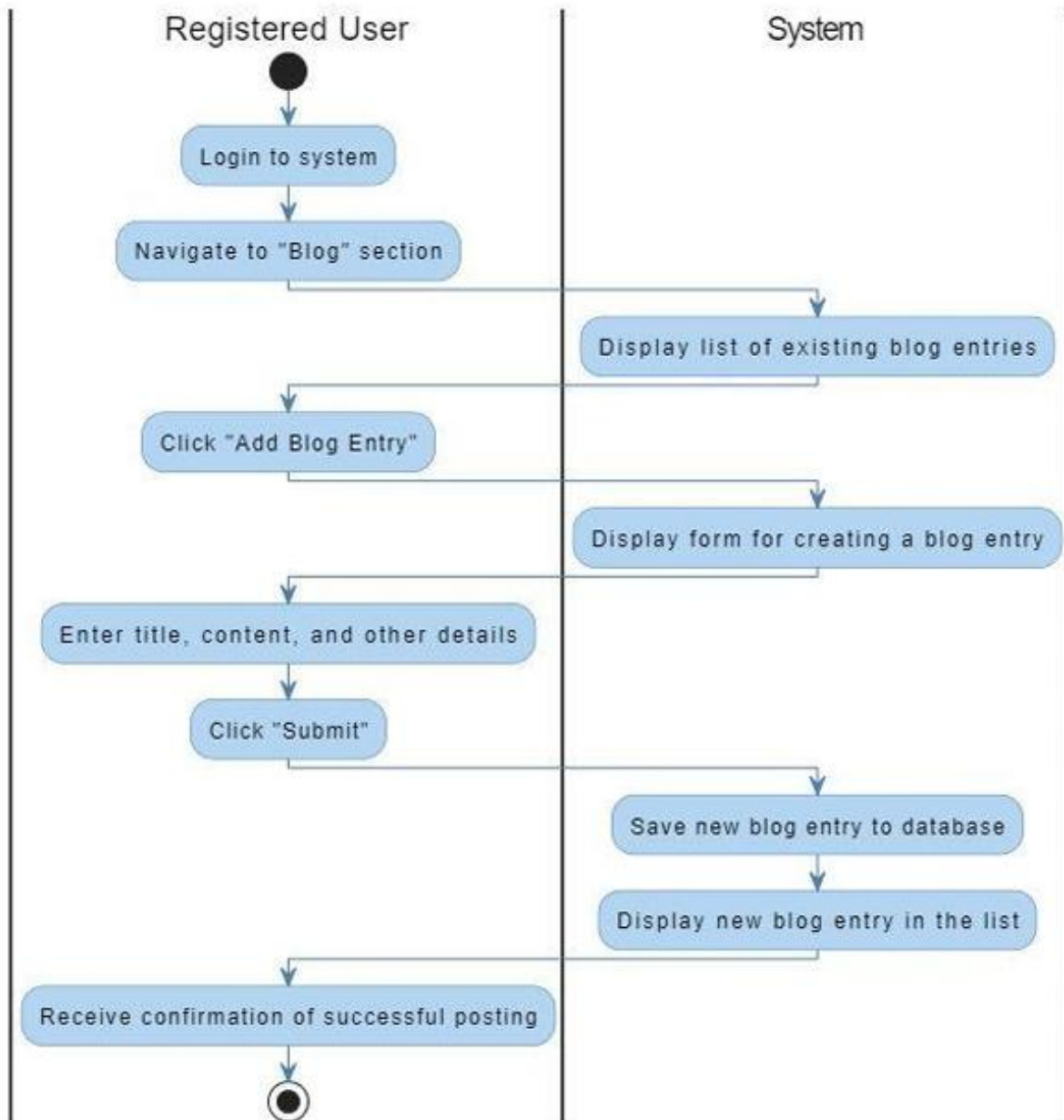
## Contact information



## Edit profile

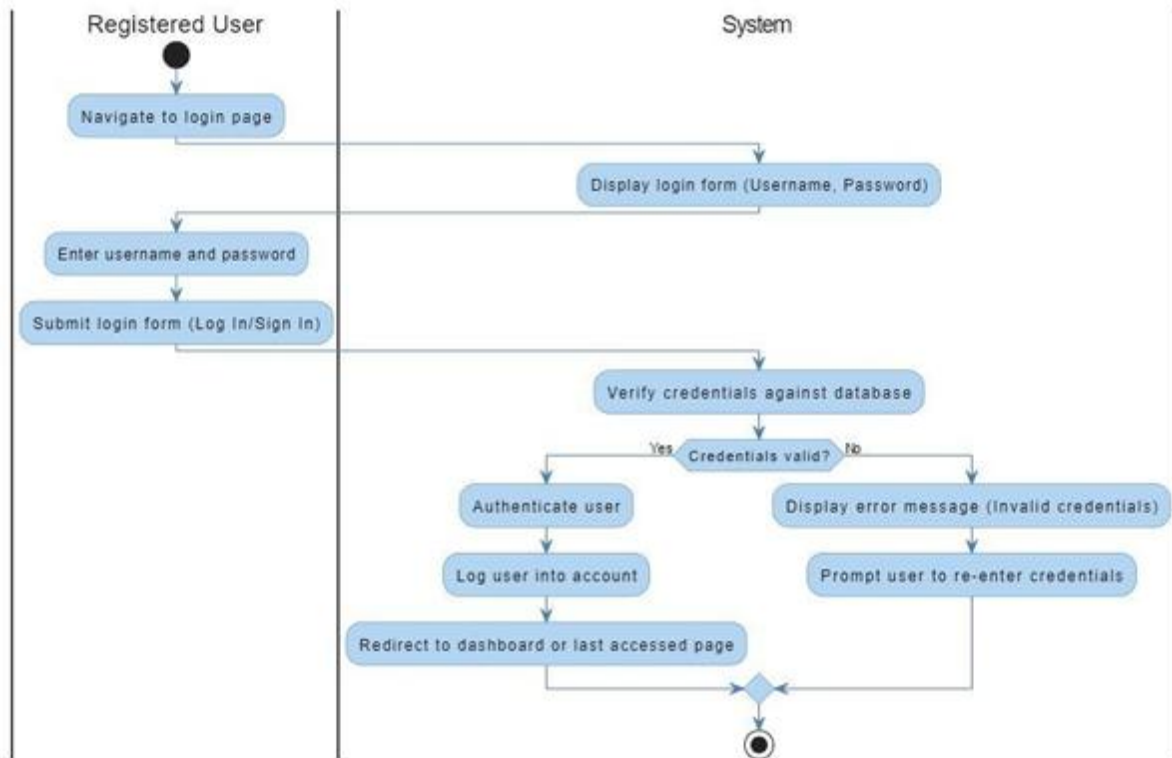


## Add blog entries





## USER LOGIN



## Comments from the interim presentation

- (1) Verifying a NEW REGISTERING TEACHER through system admin
- (2) RESTRICT students from VIEWING OTHERS' MARKS
- (3) CATEGORIZE TEACHERS' SUBSCRIPTION PLANS according to specific needs

## How the comments were addressed

### (1) Verifying a NEW REGISTERING TEACHER through system admin

District:

Ampara

▼

Role:

Teacher

▼

Provide a Valid Link with Required Documents:

Select Subject(s):

Select Subject(s):

☐ Accounting

☐ Bio Systems Technology

☐ Business Studies

☐ Commerce

☐ Engineering Technology

☐ History

☐ Physics

☐ Tamil

☐ Agriculture

☐ Biology

☐ Catholicism

☐ Drama and Theatre

☐ Geography

☐ ICT

☐ Science

☐ Art

☐ Buddhism

☐ Civic Education

☐ English

☐ Health & Physical Education

☐ Mathematics

☐ Sinhala

Request Signup

To ensure that only verified teachers can access the system, we implemented a **teacher verification process**.

When a new teacher registers, their information is first reviewed by the system administrator. Only after the admin manually verifies and approves the details (such as qualifications and identity proof), the teacher account is activated and allowed to access the platform.

## (2) RESTRICT students from VIEWING OTHERS' MARKS

We modified the system permissions so that **students can only view their own marks.**

Marks are now linked directly to individual student accounts, and access is restricted based on user session data. Students cannot see other students' grades under any circumstance, ensuring privacy and data protection.

Teachers have permission to view the marks of all students

Student Marks <span>×</span>			
STUDENT ID	NAME	MARKS	UPDATE MARKS
42	Nimal Perera	75	<button>Update Marks</button>
43	Sachini Fernando	55	<button>Update Marks</button>
44	Thisara Silva	68	<button>Update Marks</button>
46	Harsha Samarasinghe	97	<button>Update Marks</button>



student can **only see their own marks**

## All Assignments

Ass1

View Marks

**Student Details**  
Student ID: 42  
Name: Nimal Perera  
Marks: 75


### (3) CATEGORIZE TEACHERS' SUBSCRIPTION PLANS according to specific needs

We have **added subscription plan categorization** based on the needs of teachers. Different plans (e.g., Basic, Standard, Premium) are now available, offering customized features such as material uploads, student limits, and course visibility options. Teachers can select a plan that matches their specific teaching goals and class size requirements.

**Teacher Plans**

Lite


\$10



3 months

Pro


\$15



6 months

Ultimate

\$20



12 months

## 6. Completeness of the Project

- Functionalities completed

### **Guest User:**

- Register to the system
- View teachers
- View institutes
- View blogs
- Contact system admin for help

### **Student**

- Log in to the system
- View profile
- Edit profile
- View teachers
- View institutes
- View teacher profile
- View Institute profile
- View classes
- Enroll to classes
- View class schedule
- View learning materials
- Pay admission fees
- Pay class monthly fees

- Contact teacher
- View blog
- Post to the blog
- Delete from the blog
- Request old materials
- View advertisements
- Contact system admin for help

## **Teacher**

- Purchase subscriptions
- Log in to the system (Normal Teacher / Premium Teacher)
- View profile
- Edit profile
- Update Subscriptions (Premium Teacher)
- Schedule Individual Classes (Premium Teacher)
- Upload learning materials
- View student info
- Post advertisements
- Request job offers from institutes
- Issue discounts and free cards
- Respond to student messages
- View teachers
- View institutes
- View teacher profile

- View Institute profile
- View blog
- Post to the blog
- Delete from the blog

## **Institute**

- Purchase subscriptions
- Log in to the system
- View profile
- Edit profile
- View teachers
- View institutes
- View teacher profile
- View Institute profile
- Manage students
- Manage teachers
- Schedule classes
- Hire teachers
- Issue payroll
- Manage subscription
- Post advertisements

## System Admin

- Log in to the system
- View profile
- Edit profile
- View and edit teacher profile
- View and edit Institute profile
- View and edit student profile
- Respond to contact forms
- Post system announcements
  
- Functionalities yet to complete

We need to make **responsive** for better viewing on mobile and tablet devices.

We need to **implement the "Rate and Review Teacher"** feature, allowing students to rate teachers.

- Individual contribution of the team members

## **Raheem 22000097-2022cs009**

- All Payment Handling
- Subscription Handling
- Admin Features
- Chat System
- Mail System

## **Randila 22001842 - 2022cs184**

- All Class Management
- Profile Image Upload
- All Class Options
- Job Roll Features

## **Dulshan 22000623- 2022cs062**

- All Blogs Management
- Announcement Management
- Advertisement Management

## **Saranga 22001999 - 2022cs199**

- Student Enrollment
- Class Schedule Scene

- View Classes by Teacher
- View Classes by Institute
- View Classes by Class Name

## Test Cases

### 01. User Login

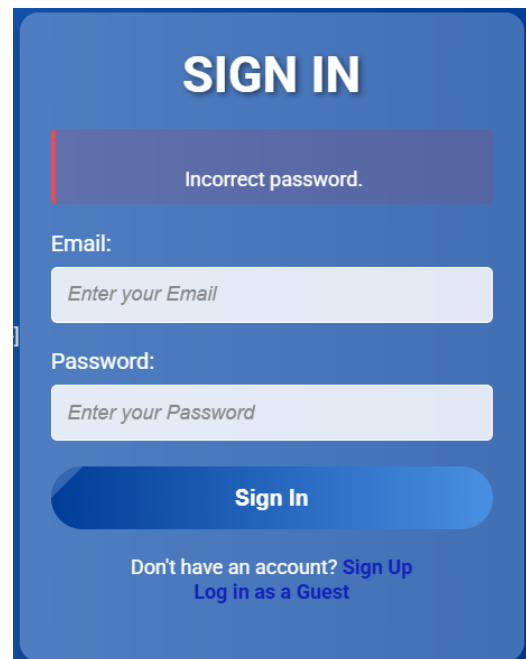
Scenario ID	Scenario Description	Test Case ID	Pre-Condition	Steps to execute	Expected Result	Actual Result	Status
1	Login with valid user	1	SriLearn auth page is loaded	Enter registered mail and its pwd	Successfully signing in to the relevant dashboard	Signs into the dashboard	pass
	Login with invalid user	2	SriLearn auth page is loaded	Enter not registered mail and some pwd	Displays "email not found or incorrect"	Displays "email not found or incorrect"	pass
	Login with invalid mail	3	SriLearn auth page is loaded	Enter random word for mail and a pwd	Displays "email not found or incorrect"	Displays "email not found or incorrect"	pass
	Login with invalid password	4	SriLearn auth page is loaded	Enter registered mail and a wrong pwd	Displays "incorrect password"	Displays "incorrect password"	pass

## 1 A valid email and password



A sign-in form with a blue background and rounded corners. At the top, the text "SIGN IN" is displayed in white, bold, uppercase letters. Below this, the label "Email:" is followed by a light blue input field containing the text "randila@example.com". Underneath, the label "Password:" is followed by a light blue input field with four dots representing a masked password. A dark blue button with the text "Sign In" in white is positioned below the password field. At the bottom, the text "Don't have an account? Sign Up" is shown, with "Sign Up" as a link, followed by "Log in as a Guest" on the next line.

## 2



A sign-in form identical in layout to the first one, but with an error message. At the top, "SIGN IN" is in white, bold, uppercase letters. A dark blue error banner with a red vertical bar on the left contains the text "Incorrect password." in white. Below this, the "Email:" label is followed by a light blue input field with the placeholder text "Enter your Email". The "Password:" label is followed by a light blue input field with the placeholder text "Enter your Password". A dark blue button with "Sign In" in white is below the password field. At the bottom, the text "Don't have an account? Sign Up" is shown, with "Sign Up" as a link, followed by "Log in as a Guest" on the next line.



3

**SIGN IN**

Email not found or incorrect.

Email:

Enter your Email

Password:

Enter your Password

**Sign In**

Don't have an account? [Sign Up](#)  
[Log in as a Guest](#)

4

**SIGN IN**

Email not found or incorrect.

Email:

Enter your Email

Password:

Enter your Password

**Sign In**

Don't have an account? [Sign Up](#)  
[Log in as a Guest](#)

## 02 User profile update


Scenario ID	Scenario Description	Test Case ID	Pre-Condition	Steps to execute	Expected Result	Actual Result	Status
2	User fills all fields according to logic	1	Update profile page is loaded	Keep no field empty before clicking update button	Page refreshes and shows the new info	Page refreshes and shows the new info	pass
	User adds an invalid phone number	2	Update profile page is loaded	Enter a phone number with differed length	"phone number must be exactly 10 digits" alert	"phone number must be exactly 10 digits" alert	pass

	User keeps a field empty before update	3	Update profile page is loaded	Keeping field empty	“the field cannot be empty” alert	“the field cannot be empty” alert	pass
--	--	---	-------------------------------	---------------------	-----------------------------------	-----------------------------------	------

1

localhost says  
Address cannot be empty.  
OK

User Profile



Upload Image

First Name:  
RandilaCC

Last Name:  
Chathuranga

Phone Number:

Address:


District:  
Galle

EDITUPDATE

2

localhost says  
Phone number must be exactly 10 digits.  
OK

User Profile



First Name:  
RandilaCC


Last Name:  
Chathuranga

Phone Number:  
75

3

localhost says  
Address cannot be empty.  
OK

User Profile



Upload Image

First Name:  
RandilaCC

Last Name:  
Chathuranga

Phone Number:

Address:

District:  
Galle

EDITUPDATE

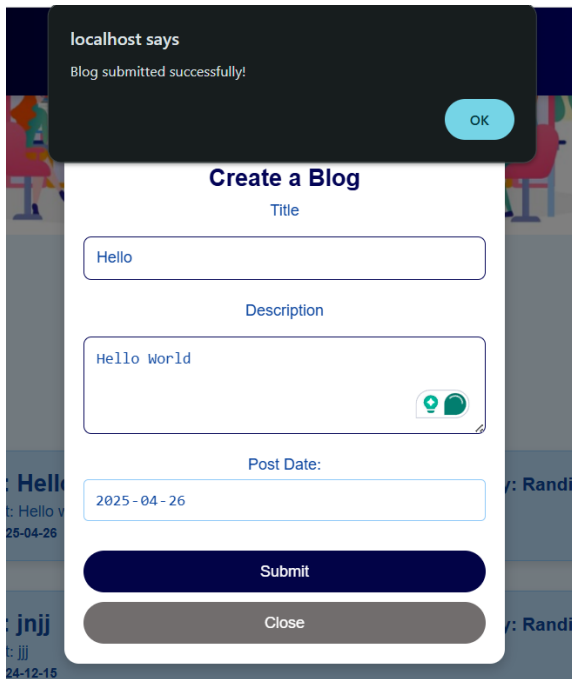
74

### 03 User create blogs

Scenario ID	Scenario Description	Test Case ID	Pre-Condition	Steps to execute	Expected Result	Actual Result	Status
3	User fills all fields in the create a blog page	1	Create a blog page is loaded	Keep no field empty before clicking submit button	Load the created blog	Load the created blog	pass
	User keeps empty fields in the create a blog page	2	Create a blog page is loaded	Keeping a field empty in the create a blog page	"Please fill out this field" alert	"please fill out this field" alert	pass

1

1



localhost says  
Blog submitted successfully!

**Create a Blog**

Title  
Hello


Description  
Hello World

Post Date:  
2025 - 04 - 26

Submit

Close

2



**Create a Blog**

Title  
Please fill out this field.

Description

Post Date:  
2025 - 04 - 26

Submit

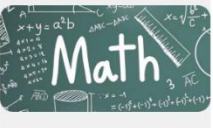
Close

## 04 Student filtering classes

Scenario ID	Scenario Description	Test Case ID	Pre-Condition	Steps to execute	Expected Result	Actual Result	Status
4	Student filters classes by entering both grade and subject	1	Classes page is loaded	Select both subject and grade on Select fields	Shows the matching classes	Shows the matching classes	pass
	Student filters class by entering only one field	2	Classes page is loaded	Select only one field from subject and grade	"No classes available for the selected filters."	"No classes available for the selected filters."	pass

1

Select Grade:



**Subject: Mathematics**  
**Teacher: RandilaCC Chathuranga**  
**Grade: 12**  
 Type: Individual  
 Address: No. 45, Galle Road,  
 Colombo 03, Western Province,  
 Sri Lanka.  
**Fee: 2500**

### Filter Classes

Select Subject:

Mathematics

Select Grade:

--Select Grade--

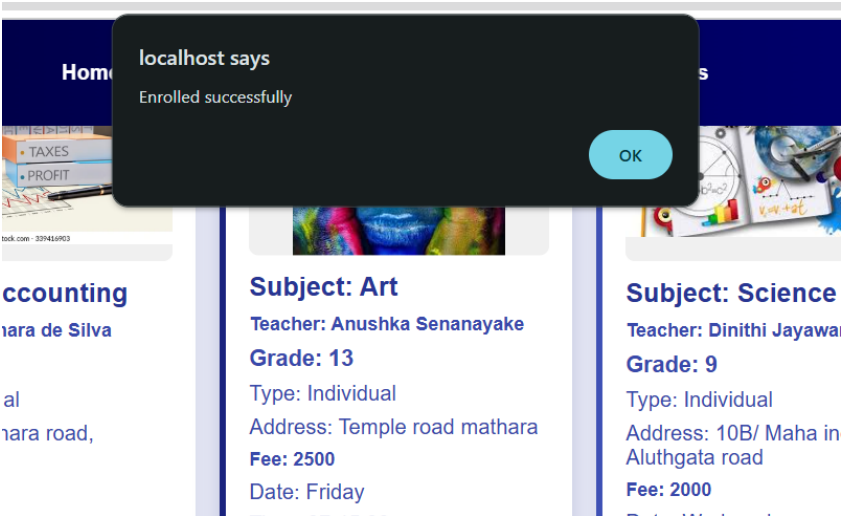
Select both grade and subject or No classes available for the selected filters.

## 05 Student joining classes

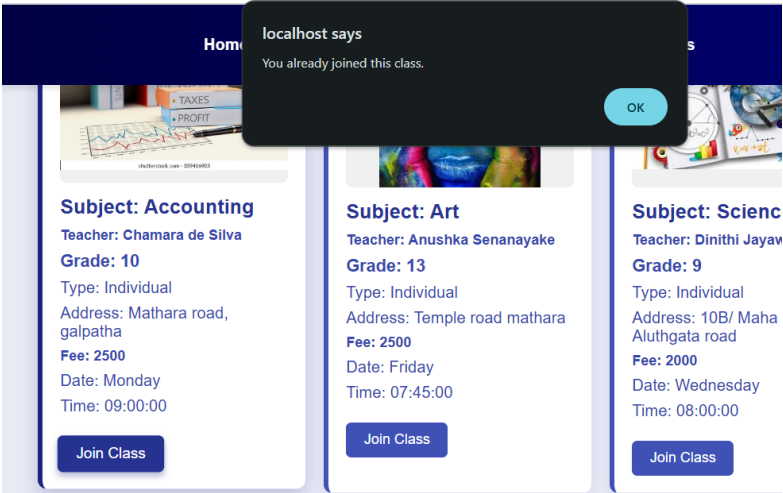
Scenario ID	Scenario Description	Test Case ID	Pre-Condition	Steps to execute	Expected Result	Actual Result	Status
5	Student clicks join the class button	1	Classes page is loaded	Click join class button on a not enrolled class	"joined class successfully" alert	"joined class successfully" alert	pass

	Student clicks join the class button on a class he already enrolled in	2	Classes page is loaded	Click join class button on already enrolled class	"already enrolled in class" alert	"already enrolled in class" alert	pass
--	--	---	------------------------	---	-----------------------------------	-----------------------------------	------

1



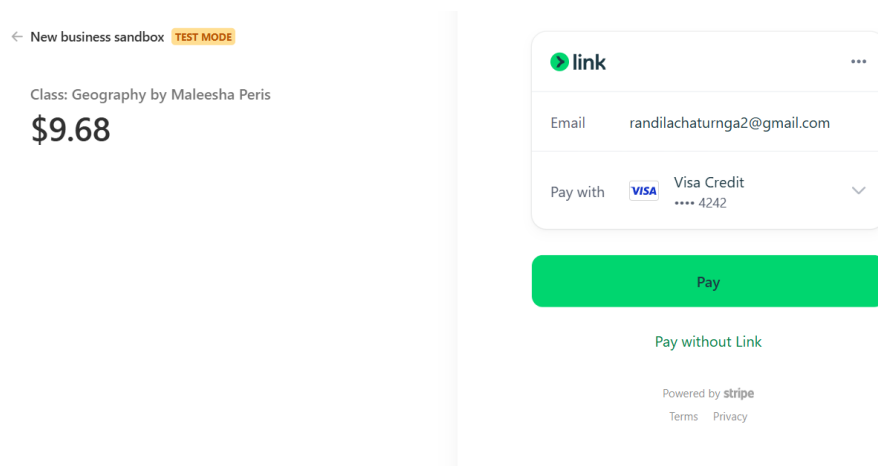
2



## 06 Student enrollment fees

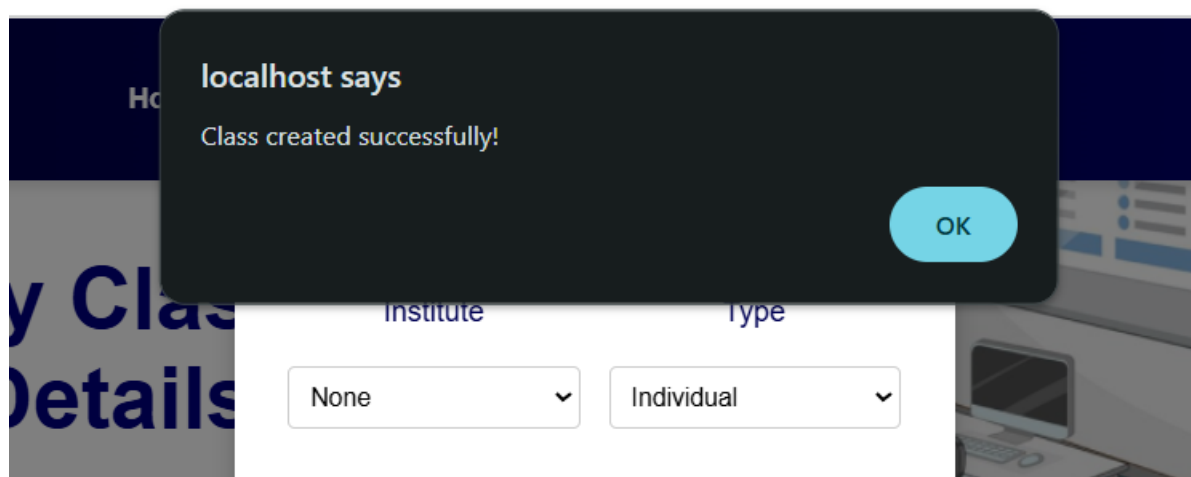
Scenario ID	Scenario Description	Test Case ID	Pre-Condition	Steps to execute	Expected Result	Actual Result	Status
6	Student clicks join the class button	1	Classes page is loaded	Click join class button on a lite subscription class	"joined class successfully" alert	"joined class successfully" alert	pass
	Student clicks join the class button	2	Classes page is loaded	Click join class button on a premium subscription class	Navigates to payment page	Navigates to payment page	pass

### Navigate to the payment page



## 07 Teacher creating a class

Scenario ID	Scenario Description	Test Case ID	Pre-Condition	Steps to execute	Expected Result	Actual Result	Status
7	Teacher fills all the fields in the create a class form	1	Create a class page is loaded	Fill all the fields in the form	"schedule created successfully" alert	"schedule created successfully" alert	pass
	Teacher leaves empty fields in the create a class form	2	Create a class page is loaded	Fill all the fields in the form	"please fill this field" alert	"please fill this field" alert	pass





## 08 Teacher creating individual/institute class

Scenario ID	Scenario Description	Test Case ID	Pre-Condition	Steps to execute	Expected Result	Actual Result	Status
8	Teacher selects "institute" in the Type list	1	Create a class page is loaded	Teacher selects "institute" in the Type list	Class type = institute	Class type = institute	pass
	Teacher selects "individual" in the Type list	2	Create a class page is loaded	Teacher selects none in the institute list and "individual" in the Type list	Class type = individual	Class type = individual	pass

localhost says  
Institute name should be 'None' for individual classes and Institute name should not be None for institute classes.

OK

My Class Details

Filter by Type: All

Class Details Form:

- Type:  (dropdown)
- Institute:  (dropdown)
- Hall number:
- Grade:
- Max Students:
- Fee:
- Address:
- Start Date:  (calendar icon)
- End date:  (calendar icon)
- Class Date:
- Class Time:  (clock icon)

Submit

Close

Institute name should be None for individual classes and Institute name should not be None for Institute class

## 09 Teacher vs other users signing up

Scenario ID	Scenario Description	Test Case ID	Pre-Condition	Steps to execute	Expected Result	Actual Result	Status
9	Choosing role teacher when signing up	1	Sign up page is loaded	User choose teacher role in the sign up form	Sign up form filled	Sign up form filled	pass
	Choosing role student/institute when signing up	2	Sign up page is loaded	User chooses student/institute role in the sign up form	Drop down section for teachers to enter proof	Drop down section for teachers to enter proof	pass

District:

Ampara

Role:

Teacher

Provide a Valid Link with Required Documents:

Select Subject(s):

Select Subject(s):

- |   |  |  |
|---|--|--|
| <input type="checkbox"/> Accounting             | <input type="checkbox"/> Agriculture       | <input type="checkbox"/> Art                         |
| <input type="checkbox"/> Bio Systems Technology | <input type="checkbox"/> Biology           | <input type="checkbox"/> Buddhism                    |
| <input type="checkbox"/> Business Studies       | <input type="checkbox"/> Catholicism       | <input type="checkbox"/> Civic Education             |
| <input type="checkbox"/> Commerce               | <input type="checkbox"/> Drama and Theatre | <input type="checkbox"/> English                     |
| <input type="checkbox"/> Engineering Technology | <input type="checkbox"/> Geography         | <input type="checkbox"/> Health & Physical Education |
| <input type="checkbox"/> History                | <input type="checkbox"/> ICT               | <input type="checkbox"/> Mathematics                 |
| <input type="checkbox"/> Physics                | <input type="checkbox"/> Science           | <input type="checkbox"/> Sinhala                     |
| <input type="checkbox"/> Tamil                  |  |  |

Request Signup

## 10 Institute view classes

Scenario ID	Scenario Description	Test Case ID	Pre-Condition	Steps to execute	Expected Result	Actual Result	Status
10	Institute clicks view students button	1	Our classes page is loaded	clicks view students button	Navigates to student details page	Navigates to student details page	pass
	Institute clicks view schedules button	2	Our classes page is loaded	clicks view schedules button	Navigates to class schedule page	Navigates to class schedule page	pass
	Institute clicks view payments button	3	Our classes page is loaded	clicks view payments button	Navigates to all payment requests page	Navigates to all payment requests page	pass

1

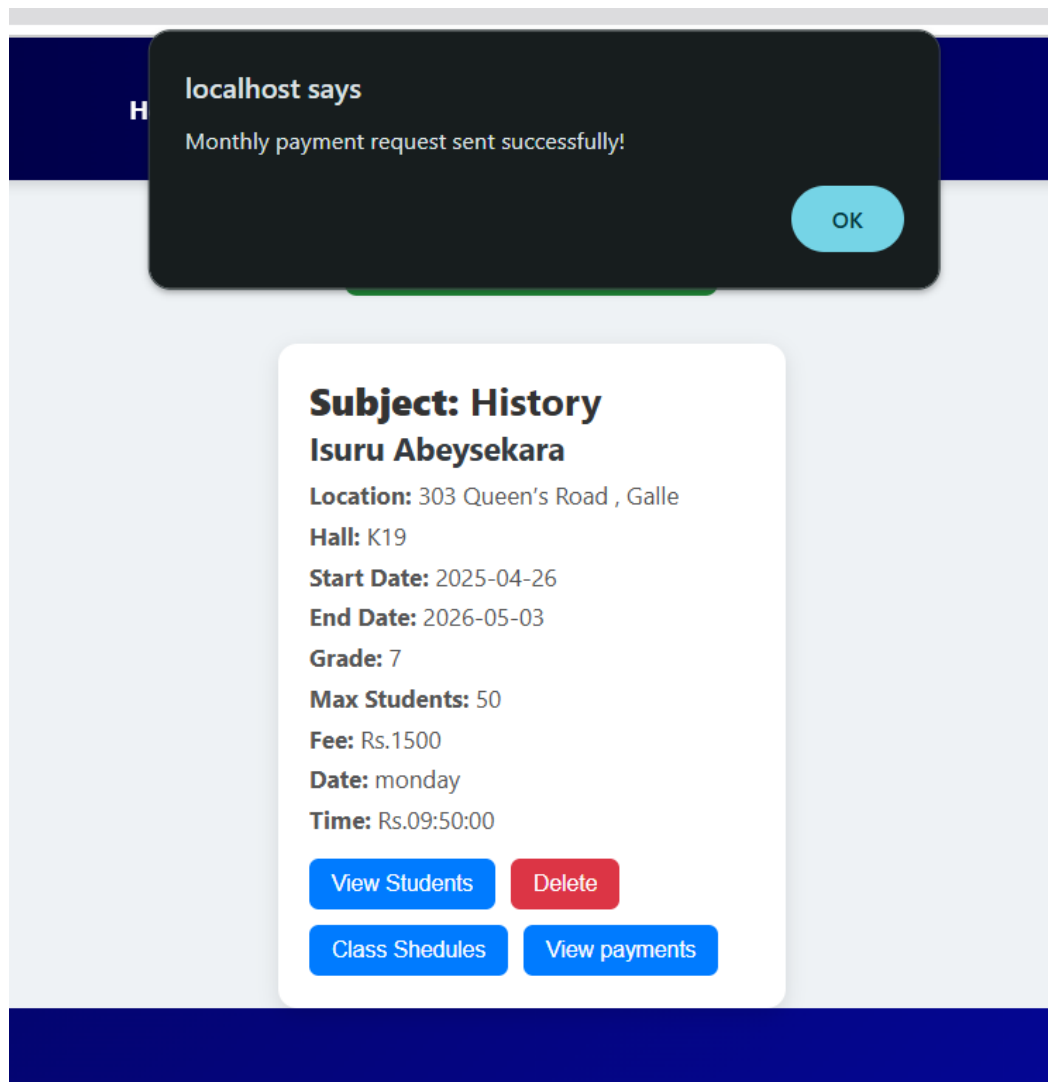
### Students Details

Student ID	Name	Email	District	Phone Number	Address	Action	Chat
42	Nimal Perera	nimal.perera@email.com	Galle	752658759	45/2 Galle road, Galle	<input type="button" value="Remove"/>	<input type="button" value="Send Message"/>

2

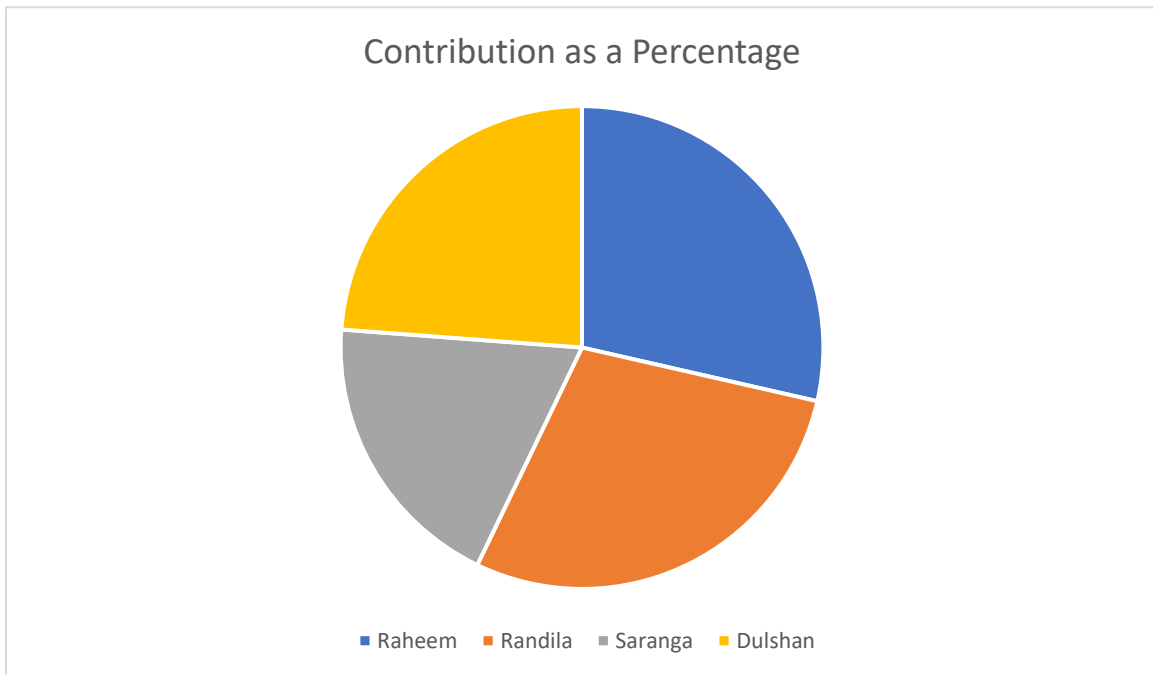
## Class Schedule

Start Time	End Time	Date
No classes found for this teacher.		



All the members contributed for the project

M.H.A.R Allam 22000097-2022cs009	30 %
D. R. C. Senarathna 22001842 - 2022cs184	30 %
A. S. Thalagalage 22000623- 2022cs062	20 %
D. C. Gunarathna 22001999 - 2022cs199	20 %



## Admin sign up requests

Scenario ID	Scenario Description	Test Case ID	Pre-Condition	Steps to execute	Expected Result	Actual Result	Status
11	Approving a teacher sign up request	1	Sign up requests page is loaded	Admin clicks approve sign up request button	"sign up request successfully approved" alert	"sign up request successfully approved" alert	pass
	Rejecting a teacher sign up request	2	Sign up requests page is loaded	Admin clicks approve sign up request button	"sign up request successfully rejected" alert	"sign up request successfully rejected" alert	pass

## Admin posting announcements

Scenario ID	Scenario Description	Test Case ID	Pre-Condition	Steps to execute	Expected Result	Actual Result	Status
12	Admin filling all fields in the post announcement form	1	Post announcements page is loaded	Admin filling all fields in the post announcement form	Navigate to system announcements page	Navigate to system announcements page	pass
	Admin leaving a field empty in the post announcement form	2	Post announcements page is loaded	leaving a field empty in the post announcement form	"the field cannot be empty" alert	"the field cannot be empty" alert	pass