



2ND YEAR

INTERIM REPORT



SRILEARN
EDUCATION

SCS2202
Group project
CS Group 56

Contents

Contents

01 Details of the Project Supervisor and Co-supervisor.....	4
02 Team Member	4
03 Introduction.....	5
3.1 Problem Identification.....	5
3.2 Solution	5
3.3 Project Goal	6
04.Scope of the Project	7
4.1 User Interfaces and Functionalities:	7
4.2 Additional Features:	7
4.3 Technical Details:.....	7
4.4 Deployment and Support:	7
4.5 Out of Scope:	8
05 Objectives of the Project	8
06 Feasibility Analysis	9
6.1 Social Feasibility.....	9
6.2 Technical Feasibility.....	10
6.3 Schedule Feasibility	12
6.4 Economical Feasibility.....	13
6.5 Legal and Ethical Feasibility	14
07 Project Deliverables.....	15
7.1 Requirements Documentation:.....	15
7.2 System Design Artifacts:	15
7.3 Front-End Development:	15
7.4 Back-End Development:.....	16
7.5 Database Implementation:.....	16
7.6 Core Functionality Implementation:	16
7.7 Testing and Quality Assurance:	16
7.8 Deployment and Release:	16
7.9 Documentation and Training:.....	16
7.10 Support and Maintenance:	17
08 Project Constraints and Assumptions.....	17
8.1 Constraints:	17
8.2 Assumptions:.....	17

09	Functional Requirements.....	18
9.1	Guest User:.....	18
9.2	Student:.....	18
9.3	Teacher:	19
9.4	Institute:	19
9.5	System Admin:	20
10	Non-Functional Requirements (Quality Attributes).....	20
11	Usecase diagram	21
11.1	All Users.....	23
11.2	Student.....	23
11.3	Teacher	23
11.4	Institute.....	24
11.5	System Admin	24
12	ER diagram	25
13	Component diagram	26
14	Use Case Narratives.....	28
15	Activity Diagram	69
16	Class Diagram	104
17	Technologies Used in the Project	105
18	Progress So Far	106
19	Next Step.....	107

01 Details of the Project Supervisor and Co-supervisor

- Proposed Project Supervisor**

Name of the Supervisor-Dr. H.N.D Thilini

Signature of the Supervisor-



.....
Date-2024.07.08

- Proposed Project Co-Supervisor**





Name of the Co-Supervisor-Ms E.B.M.R.K.W.K.D Ekanayake

Signature of the Co-Supervisor-



.....
Date-2024.07.08

02 Team Member

Index Number	Name of the student	Signature
22000097	ALLAM M.H.A.R	
22001842	SENARATHNA D.R.C.	
22000623	GUNARATHNA D.C.	
22001999	THALAGALAGE A.S	

03 Introduction

3.1 Problem Identification

In the realm of education, accessibility and organization are pivotal for creating conducive learning environments. However, current systems for accessing tuition classes are fragmented and inefficient. Students often struggle to find suitable classes tailored to their educational needs, while teachers and institutes face challenges in effectively managing and promoting their offerings. This fragmentation leads to a lack of centralized information, making it cumbersome for all stakeholders to navigate and utilize available educational resources effectively.

Our team has firsthand experience navigating these challenges during our O/Ls and A/Ls. We understand the frustration of searching across disparate platforms for quality tuition classes and the difficulty teachers encounter in promoting their services across multiple channels. This fragmentation not only hampers the learning process but also limits the opportunities available to aspiring teachers looking to enter the tuition sector.

3.2 Solution

To address these challenges, our project proposes the development of a comprehensive webbased platform designed specifically for students, teachers, and institutes involved in tuition classes. This platform will serve as a centralized hub, consolidating all relevant information and functionalities into a cohesive and user-friendly interface.

For students, the platform will streamline the process of searching for and enrolling in classes that align with their academic needs. They will have access to a comprehensive database of classes, enabling them to make informed decisions about their educational journey.

Teachers will benefit from enhanced tools for managing their classes, including scheduling, student management, and resource sharing. This will empower them to focus more on teaching and less on administrative tasks, thereby improving the overall quality of education delivered.

Institutes will gain capabilities to oversee and optimize their educational offerings through features such as class scheduling, teacher management, and promotional tools. This centralized approach will enable institutes to efficiently manage their operations and enhance their educational services.

By consolidating resources and fostering connections within the education sector, our platform aims to transform how tuition classes are managed and accessed. It will not only facilitate better educational outcomes for students but also create new opportunities for teachers to enter and thrive in the tuition field.

Through these initiatives, we aspire to contribute to a more efficient and effective educational ecosystem, where accessibility, organization, and quality education are prioritized and accessible to all stakeholders involved in the tuition sector.

3.3 Project Goal

Our goal is to launch a fully functional web platform designed to streamline operations for teachers, students, and institutes nationwide. This platform aims to simplify the management of tuition-related activities, providing a centralized solution that enhances efficiency and accessibility across the education sector.

Driven by the feedback and positive responses garnered from a comprehensive public survey, our project seeks to address the current challenges faced by students in finding suitable tuition classes, teachers in managing their classes effectively, and institutes in coordinating their educational offerings efficiently.

By leveraging modern technology and user-centric design principles, our platform will empower students to easily discover and enroll in classes that align with their academic needs. Simultaneously, it will equip teachers with tools to efficiently manage their classes and facilitate growth opportunities within the tuition sector. Institutes will benefit from enhanced administrative capabilities, enabling them to streamline operations and optimize educational resources.

Through this initiative, we aim to revolutionize the educational landscape by providing a unified platform that centralizes resources, fosters collaboration, and enhances the overall educational experience for all stakeholders involved.

04.Scope of the Project

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

4.1 User Interfaces and Functionalities:

- **For Students:**
 - Search and browse tuition classes based on subjects, location, and other criteria.
 - Enrollment and payment processing for classes.
 - Manage class schedules and fees through user profiles.
- **For Teachers:**
 - Class management tools for scheduling, tracking student records, and managing payments.
 - 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.

4.2 Additional Features:

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

4.3 Technical Details:

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

4.4 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.

4.5 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.

05 Objectives of the Project

The objectives of our project are to:

1. **Enhance Accessibility:** Provide a user-friendly web platform where students can easily search for and enroll in tuition classes based on their academic needs and preferences.
2. **Streamline Management:** Enable teachers to efficiently manage their classes by tracking student records, managing payments, and scheduling through the platform.
3. **Facilitate Administrative Efficiency:** Empower institutes to oversee and optimize their educational offerings by managing classes, teachers, and student records from a centralized administrative dashboard.
4. **Promote Educational Resources:** Offer a blog feature for sharing educational content and updates, fostering a community of learning among users.
5. **Support Marketing Efforts:** Enable teachers and institutes to market their classes effectively through targeted advertisements within the platform.
6. **Ensure Platform Stability and Security:** Develop a scalable and secure web application with robust database management and responsive design principles.
7. **Gather and Implement Feedback:** Implement feedback mechanisms to continuously improve the platform's functionality and user experience based on user input.
8. **Provide Training and Support:** Offer user manuals, documentation, and training sessions to ensure effective adoption and utilization of the platform by administrators, teachers, and students.

06 Feasibility Analysis

6.1 Social Feasibility

Social feasibility examines the acceptance and usability of the proposed web platform within the target user groups. Through a conducted survey, we gathered valuable insights that highlight strong support and preference for the functionalities our platform aims to offer:

User Preferences:

- **Teachers, Students, and Institutes:** The survey indicated a significant preference among teachers, students, and institutes for using a web-based platform to manage classes. Respondents expressed a need for centralized management of class-related activities, including scheduling, attendance tracking, and payment management.
- **Advertising Preference:** Teachers showed a keen interest in utilizing the platform for advertising their classes. They recognized the potential of targeted advertisements within the platform to reach a broader audience of prospective students effectively.
- **Real-time Communication:** There is a notable approval for the capability to chat with students in real-time through the platform. This feature is seen as beneficial for enhancing interaction and support between teachers and students outside of traditional classroom settings.
- **Online Learning Materials:** Students demonstrated a clear preference for accessing learning materials online directly through the platform. This aligns with modern educational trends towards digital content consumption and accessibility.

User Engagement and Adoption:

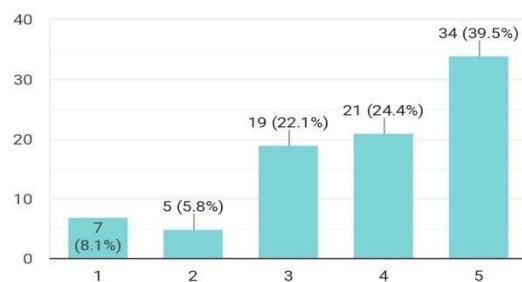
- The survey findings indicate a high likelihood of user engagement and adoption due to the platform's alignment with user preferences and needs identified in the survey.

Community Building:

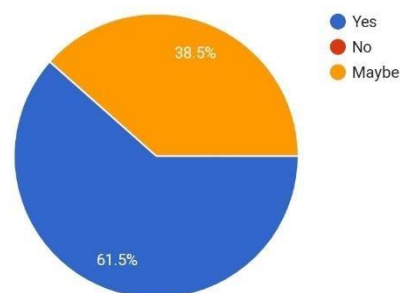
- The platform's blog feature, identified as a preferred medium for sharing educational content and updates, is anticipated to foster a sense of community among users, contributing to its social feasibility.

Overall, the survey results validate the social feasibility of our proposed web platform, affirming strong user interest and support for its intended functionalities and features.

How likely would you be to use a web application for tuition classes?



Do you find it valuable being able to instruct students real time, through a chat?



6.2 Technical Feasibility

Our technical approach for developing the web platform incorporates an open-source framework and follows the MVC architecture, utilizing the following technologies:

1. Front-End Development:

- **HTML and CSS:** Pure HTML and CSS are used for creating responsive and visually appealing user interfaces. This approach ensures compatibility across different browsers and devices while providing flexibility in design.

2. Back-End Development:

- **PHP:** PHP serves as the server-side scripting language responsible for processing user requests, managing application logic, and interacting with the MySQL database.
- **MySQL:** MySQL is employed as the relational database management system (RDBMS) for storing and managing data related to users, classes, schedules, payments, and other administrative records.

3. MVC Architecture:

- **Model-View-Controller (MVC):** The application is structured based on MVC architecture:
 - **Model:** Manages the data logic and database interactions (MySQL).
 - **View:** Presents data to users through HTML/CSS interfaces.
 - **Controller:** Handles user input, processes requests, and coordinates interactions between the model and view (PHP).

4. Version Control:

- **Git:** Git is utilized as the version control system to manage codebase changes, facilitate collaboration among team members, track project history, and deploy updates efficiently.

5. Scalability and Performance Considerations:

- The chosen technologies and architecture support scalability by allowing modular development and deployment of components. PHP's ability to handle server-side operations efficiently, coupled with MySQL's scalability options, ensures the platform can accommodate growth in user base and data volume.

6. Security Measures:

- Implementation of best practices in PHP (e.g., input validation, secure coding practices) and MySQL (e.g., parameterized queries, access control) helps mitigate security risks and protect user data.

7. Testing and Deployment:

- Utilization of local development environments and staging servers ensures thorough testing of application functionalities before deployment. This approach minimizes the risk of bugs and ensures a stable release.

Conclusion:

The technical feasibility of our project is underpinned by the strategic selection of open-source tools and technologies, adherence to MVC architecture, and robust version control with Git. These elements collectively support efficient development, deployment, scalability, and maintenance of the web platform, ensuring it meets both current and future requirements effectively.

6.3 Schedule Feasibility

Our project commenced in July, and we have completed the requirement gathering phase. Currently, we are in the process of analyzing and designing the web platform. Here's a breakdown of our planned schedule:

1. Development Phases:

- **Analysis and Design:** Ongoing (July to September)
 - During this phase, we are finalizing the system requirements, creating wireframes, and designing the architecture of the platform.
- **Front-End Development:** Planned start in October
 - This phase involves implementing the HTML/CSS interfaces based on the finalized designs and user interface specifications.
- **Back-End Development:** Following front-end development
 - PHP coding for business logic, database interactions (MySQL), and integration with the front-end interface.
- **Testing and Quality Assurance:** Parallel to development phases
 - Testing functionalities, performance, security, and usability to ensure a stable and reliable platform.
- **Deployment and Finalization:** February to March 2025
 - Final integration, deployment on production servers, user acceptance testing, and addressing any last-minute issues or adjustments.

2. Work Plan Details:

- **Working Hours per Week:** 14 hours ○ **Number of Group Members:** 4
- **Number of Weeks:** 36 (from July to March) ○ **Total Man Hours Calculation:**

$$36 \text{ weeks} \times 4 \text{ members} \times 14 \text{ hours/week} = 2016 \text{ man-hours}$$

3. Feasibility Assessment:

- The allocated 2016 man-hours align with the scope and complexity of the project, considering the phases and tasks outlined.
- Starting front-end development in October allows sufficient time for comprehensive design and analysis, ensuring quality in subsequent development phases. ○ Delivering the final platform by March 2025 is achievable, accounting for testing, deployment, and potential contingencies.

6.4 Economical Feasibility

Our project adopts a cost-effective approach by leveraging free and open-source tools for development. Here's a breakdown of the economical considerations:

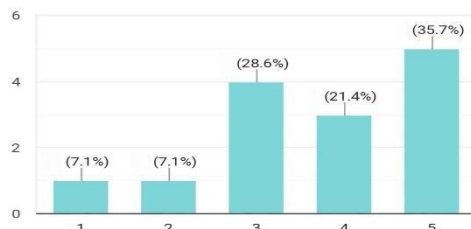
1. Cost Savings with Open-Source Tools:

- **Tools and Technologies:** Utilizing HTML/CSS, PHP, MySQL, and Git as open-source tools significantly reduces initial software licensing costs. These tools are widely supported, with extensive documentation and community resources available at no cost.
- **Cloud Storage:** While most tools are free, you plan to invest in a cloud server for storing materials. This expenditure is essential for scaling storage needs efficiently.
- **Payment Gateway:** During the testing phase, you'll explore free methods for handling transactions. However, for deployment, investing in a robust payment gateway ensures secure transactions and enhances user trust.

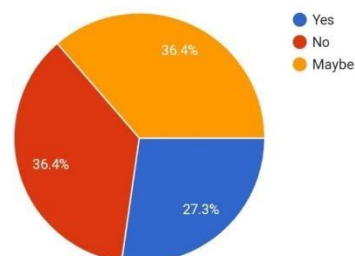
2. Revenue Generation:

- **Subscription Model:** Institutes and teachers will subscribe to operate within the platform. This subscription-based revenue model provides a steady income stream to cover operational expenses such as cloud storage and payment gateway fees.
- **Advertisement System:** Additionally, revenue from advertisements placed by institutes and teachers further supplements the subscription income. This dual revenue approach enhances financial stability and supports profitability.

How likely would be to advertise your classes on a platform mostly used by students?



Would you be willing to conduct classes under an institute?



3. Deployment Strategy:

- **Cost Optimization:** While deploying the platform, you aim to implement the best practices for cloud hosting and payment processing to optimize costs and ensure reliability.
- **Scalability:** Cloud services offer scalability, allowing you to adjust resources based on demand without incurring unnecessary costs during lower traffic periods.

4. Financial Viability:

- **ROI Calculation:** Projecting return on investment (ROI) based on subscription and advertisement revenues against operational costs ensures financial viability and sustainability.

5. **Risk Management:**

- **Budgeting:** Allocating funds for essential services like cloud storage and payment gateways ensures operational continuity and mitigates financial risks.

6. **Long-Term Sustainability:**

- **Profitability Forecast:** With a clear revenue model and prudent cost management, the platform is poised for long-term sustainability and potential expansion into additional features or services.

6.5 Legal and Ethical Feasibility

1. **Data Protection and Privacy Laws:**

- **Personal Data Protection Act (PDPA):** Sri Lanka is in the process of enacting comprehensive data protection legislation. It's essential to comply with PDPA requirements regarding the collection, processing, storage, and transfer of personal data. Ensure users' consent for data usage and implement robust security measures to protect sensitive information.
- **Privacy Concerns:** Respect users' privacy rights by clearly stating data handling practices in a privacy policy. Provide mechanisms for users to access, correct, or delete their personal information as per legal requirements.

2. **Intellectual Property Rights (IPR):**

- **Copyright and Trademarks:** Respect intellectual property rights when using content, logos, or trademarks belonging to others. Obtain necessary permissions or licenses for any copyrighted material used on the platform.
- **User-generated Content:** Establish guidelines for user-generated content to prevent infringement and ensure compliance with copyright laws. Implement mechanisms to handle complaints regarding intellectual property violations promptly.

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

- **Electronic Transactions Act:** Comply with the Electronic Transactions Act, which governs e-commerce activities, including online transactions and electronic contracts. Ensure transparency in pricing, terms of service, and refund policies for users' protection.
- **Consumer Rights:** Adhere to consumer protection laws by providing accurate product/service information, honoring subscription terms, and resolving disputes fairly and promptly.

4. **Ethical Considerations:**

- **Transparency and Accountability:** Maintain transparency in platform operations, including advertising practices, data usage policies, and user interactions. Ensure accountability for content shared and actions taken by users within the platform.
- **Fair Practices:** Avoid discriminatory practices based on race, gender, religion, or other protected characteristics. Promote inclusivity and diversity within the platform community.
- **User Safety:** Implement measures to safeguard users

from harassment, cyberbullying, or inappropriate content. Provide reporting mechanisms and take prompt action against violators of platform guidelines.

5. Compliance and Regulatory Framework:

- **Regulatory Compliance:** Stay informed about evolving regulatory requirements and adapt platform policies and practices accordingly. Engage legal counsel to ensure ongoing compliance with local laws and regulations affecting online platforms.
- **Corporate Governance:** Uphold ethical standards in corporate governance, including transparency in financial transactions, accountability to stakeholders, and responsible business practices.

07 Project Deliverables

7.1 Requirements Documentation:

- Detailed documentation outlining functional and non-functional requirements gathered from stakeholders, including students, teachers, and institutes.

7.2 System Design Artifacts:

- **System Architecture:** Diagrams and descriptions illustrating the overall structure of the web platform, including the MVC architecture, database schema, and integration points.
- **Wireframes and UI/UX Design:** Visual representations and design mockups of user interfaces (UI) and user experience (UX) elements for the platform's front-end.

7.3 Front-End Development:

- Implementation of HTML/CSS templates and interactive UI components based on approved designs.
- Responsive design ensuring compatibility across various devices and browsers.

7.4 Back-End Development:

- Development of server-side functionalities using PHP, including user authentication, session management, data processing, and integration with MySQL database.

7.5 Database Implementation:

- Setup and configuration of MySQL database to store and manage user profiles, class information, schedules, payments, and other relevant data.

7.6 Core Functionality Implementation:

- **User Management:** Registration, login/logout, profile management.
- **Class Management:** Creation, scheduling, enrollment, and payment processing.
- **Institute Management:** Administration tools for managing classes, teachers, and students.

7.7 Testing and Quality Assurance:

- **Unit Testing:** Testing individual components (front-end and back-end) to ensure functionality and reliability.
- **Integration Testing:** Verifying interactions between different modules and systems to validate end-to-end functionality.
- **User Acceptance Testing (UAT):** Involving stakeholders to evaluate the platform against agreed-upon requirements and usability criteria.

7.8 Deployment and Release:

- Deployment of the web platform on a production server or cloud infrastructure.
- Configuration of server settings, domain setup, and initial data migration.

7.9 Documentation and Training:

- **Technical Documentation:** Comprehensive documentation covering system architecture, installation instructions, and API references.
- **User Manuals:** Guides for administrators, teachers, and students on using the platform effectively.
- **Training Sessions:** Conducting training sessions or workshops to onboard users and administrators.

7.10 Support and Maintenance:

- Post-deployment support to address issues, bugs, and user inquiries.
- Ongoing maintenance to update features, enhance security, and optimize performance based on user feedback and platform usage.

08 Project Constraints and Assumptions

8.1 Constraints:

1. **Timeline:** The project is scheduled to be completed by March 2025, with a fixed timeline of 36 weeks from July 2024 for development and deployment phases.
2. **Budget:** The project operates within a predefined budget, which includes expenses for cloud services, payment gateway integration, and other essential resources.
3. **Resource Availability:** Limited availability of team members and external resources, impacting the speed and scope of development activities.
4. **Technology Stack:** The project is constrained to use HTML/CSS for front-end, PHP for back-end, MySQL for database management, and Git for version control, limiting flexibility in technology choices.
5. **Regulatory Compliance:** Adherence to Sri Lankan legal and regulatory requirements, particularly regarding data protection, electronic transactions, and intellectual property rights.

8.2 Assumptions:

1. **Stakeholder Support:** Assumption that stakeholders (students, teachers, institutes) will actively engage with the platform and adopt its functionalities as intended.
2. **Infrastructure Stability:** Assumption that selected cloud services and hosting providers will maintain high uptime and reliability, ensuring minimal disruption to platform availability.

3. **User Behavior:** Assumption that users will interact with the platform in accordance with expected patterns, such as class enrollment, payment processing, and engagement with educational content.
4. **Scalability:** Assumption that the platform architecture and infrastructure can scale effectively to accommodate increasing user traffic and data volume as the user base grows.

5. **Revenue Generation:** Assumption that the subscription model and advertisement system will generate sufficient revenue to sustain ongoing operational costs and support future platform enhancements.
6. **Technical Compatibility:** Assumption that integrations with third-party services (e.g., payment gateways) will proceed smoothly without significant technical or contractual issues.
7. **Security Measures:** Assumption that implemented security measures (e.g., data encryption, access controls) will adequately protect user data and mitigate risks associated with cyber threats.

09 Functional Requirements

9.1 Guest User:

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

9.2 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
- Rate and review teachers and institutes
- Request old materials
- View advertisements • Contact system admin for help

9.3 Teacher:

- 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
- Request Payrolls (Normal Teacher)
- Issue discounts and free cards (Premium Teacher)
- Respond to student messages
- View teachers
- View institutes
- View teacher profile
- View Institute profile
- View blog
- Post to the blog • Delete from the blog

9.4 Institute:

- 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 • Offer discounts and free cards

9.5 System Admin:

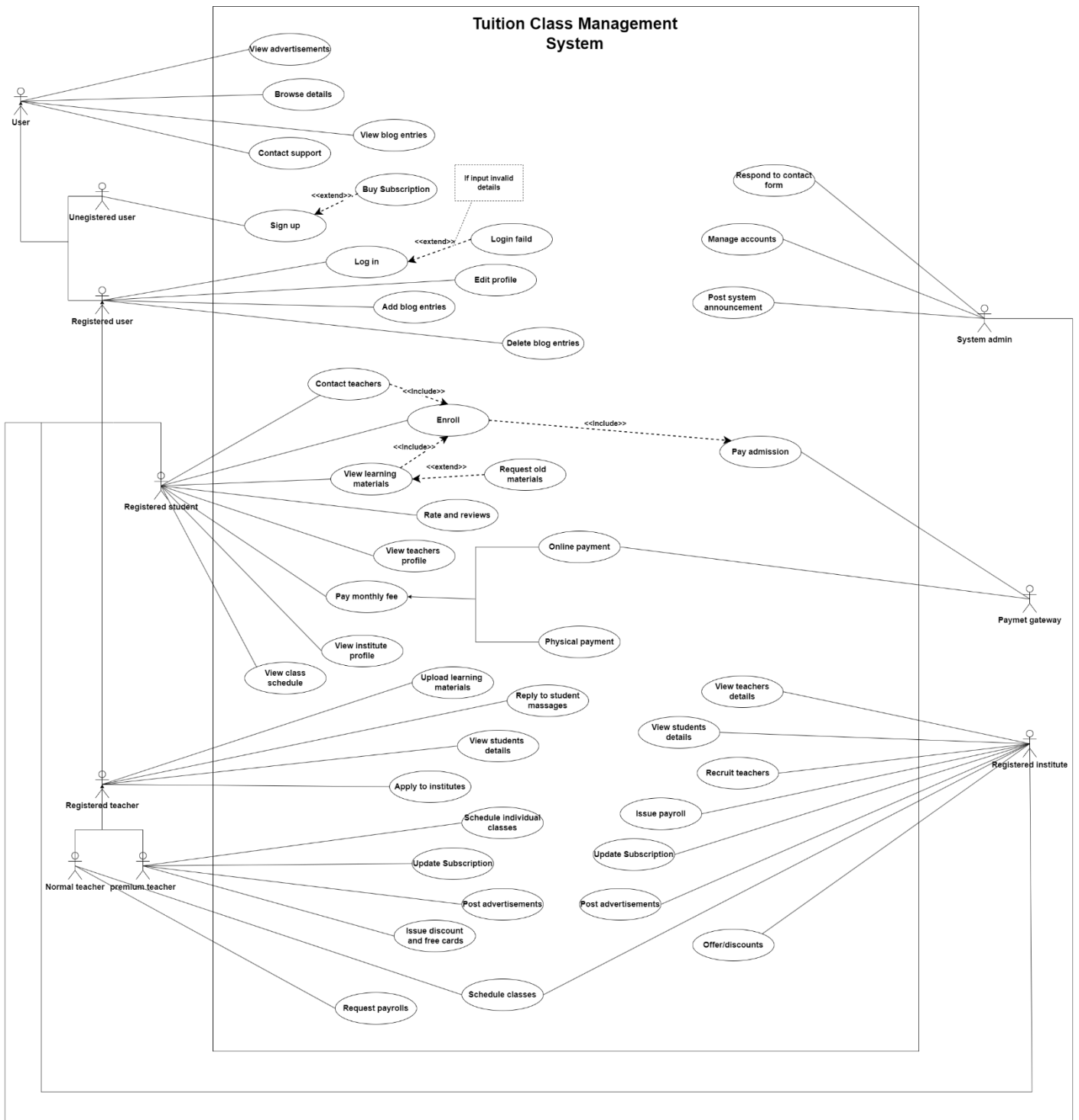
- Log in to the system
- View profile
- Edit profile
- View and edit teacher profile
- View and edit Institute profile
- View and edit student profile
- Respond to contact forms
- Post system announcements

10 Non-Functional Requirements (Quality Attributes)

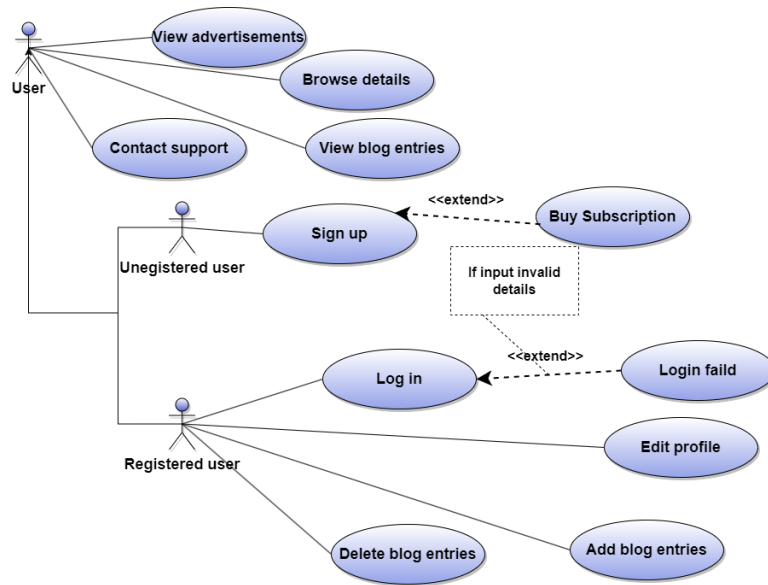
1. **Performance:**
 - Ensure pages load quickly
 - Handle multiple users simultaneously without slowing down.
2. **Reliability:**
 - Maintain high uptime (99.9% availability).
 - Quickly recover from failures to minimize downtime.
3. **Security:**
 - Protect user data with encryption and secure connections.
 - Use strong authentication and permissions to secure accounts.
4. **Usability:**
 - Design a user-friendly interface that is easy to navigate.
 - Ensure accessibility for users with disabilities.
5. **Maintainability:**
 - Organize code in a modular way for easier updates.
 - Follow coding standards and document changes.
6. **Compliance:**
 - Meet legal requirements for data protection and transactions.
 - Uphold ethical standards in user interactions and content.
7. **Performance Metrics:**
 - Test system under high loads to ensure it performs well.
 - Plan for enough server capacity to handle growth.
8. **Integration:**
 - Integrate smoothly with payment and messaging services.
 - Ensure third-party services work seamlessly with the platform.

11 Usecase diagram

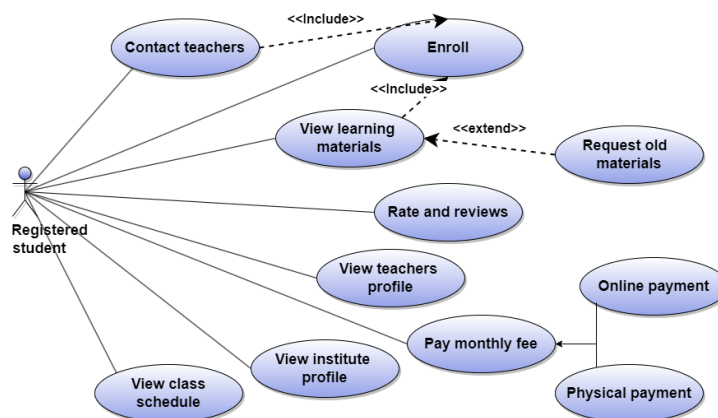
Tuition Class Management System



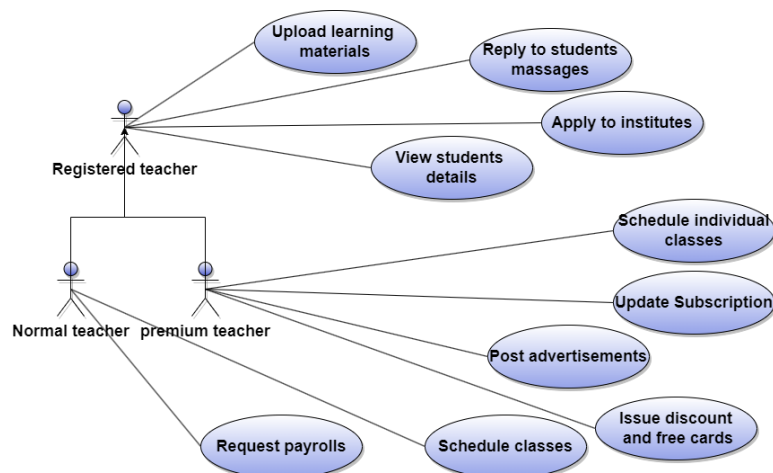
11.1 All Users



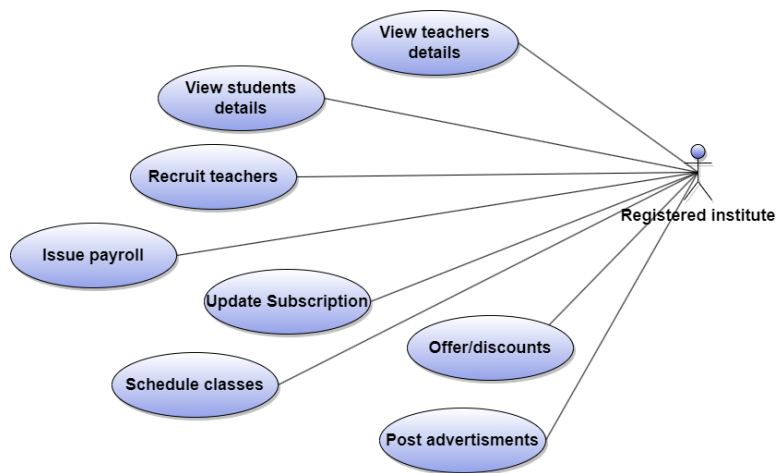
11.2 Student



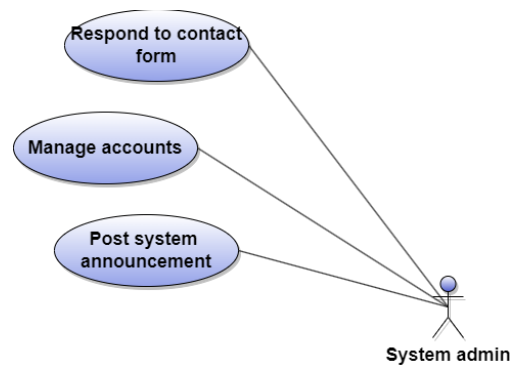
11.3 Teacher



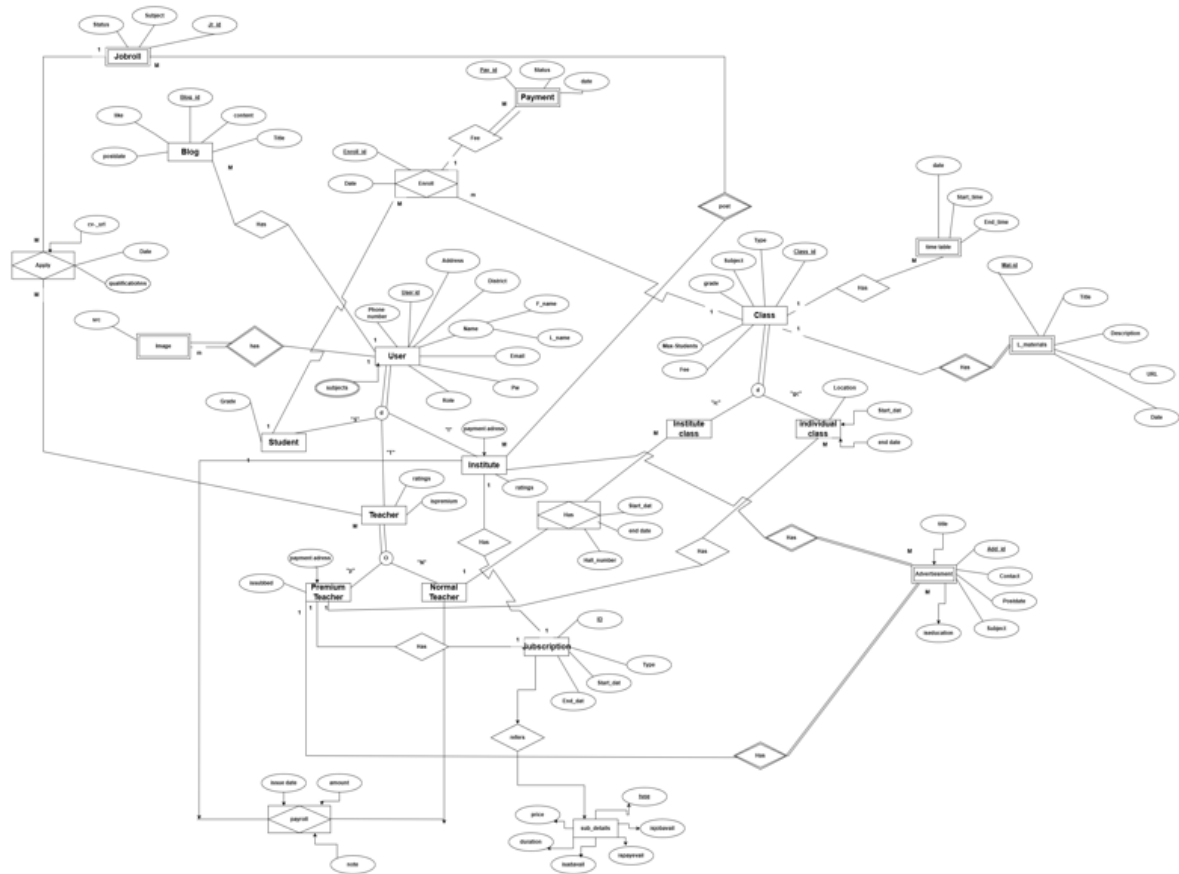
11.4 Institute



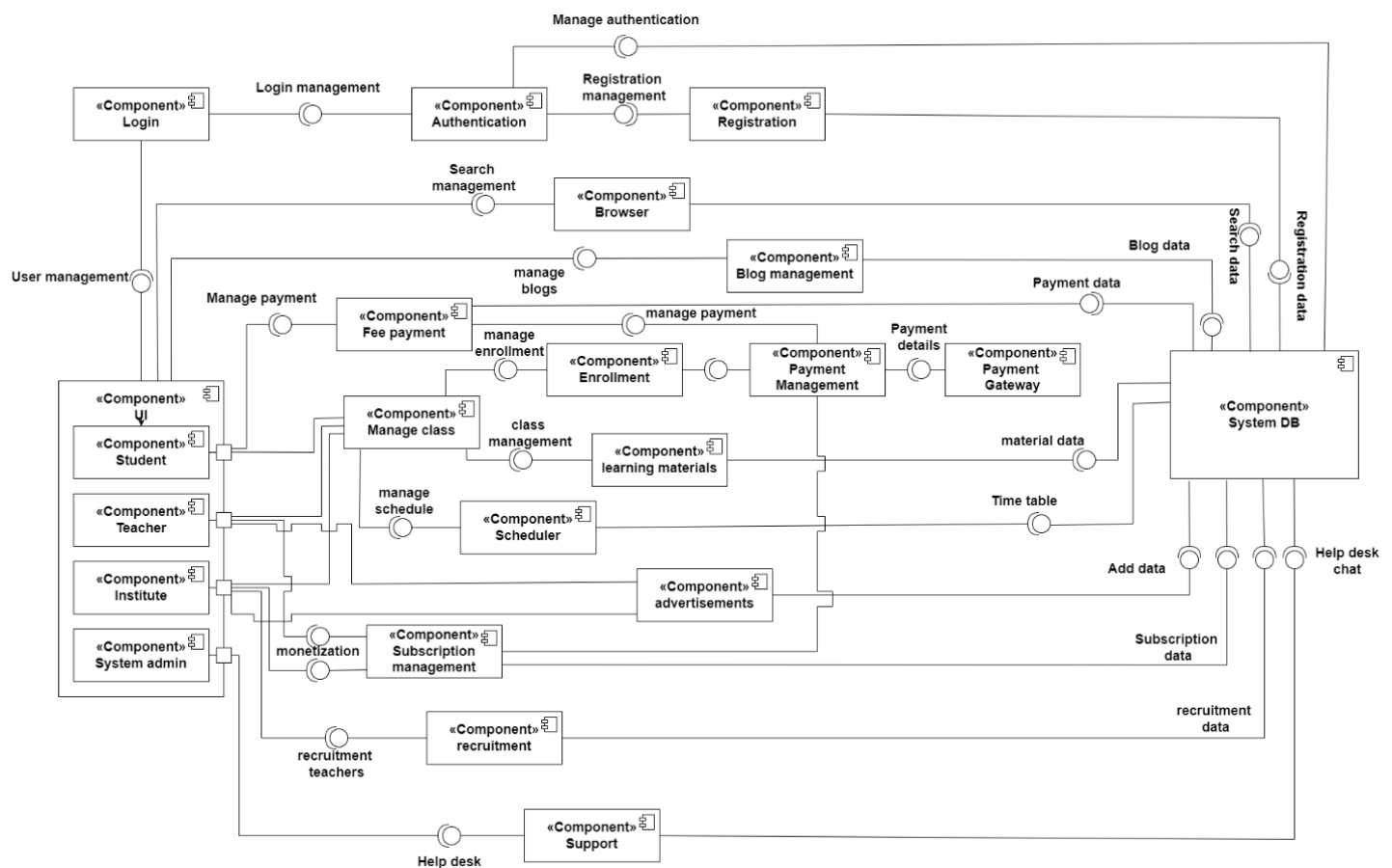
11.5 System Admin



12 ER diagram



Component diagram



- Registration management: Handles the registration of users and components.
- Login management: Responsible for user authentication and login related process.
- Components authentication: Manages the authentication of individual components within the system.
- Search management: Provides search functionality across the system.
- Components browser: Facilitates browsing and interacting with various components.
- Blog management: Enables the management of blog-related content and data.
- Manage payment: Handles payment-related operations, such as fee payment.
- Manage enrollment: Manages the enrollment of users in courses or classes.
- Class management: Responsible for managing class-related information and activities.
- Manage schedule: Allows users to manage their schedules within the system.

- Scheduler: Coordinates the scheduling of various system events and activities.
- Monetization: Handles the monetization aspects of the system, such as subscription management.
- Recruitment: Facilitates the recruitment of teachers and other personnel for the system.

These components interact with each other to provide a comprehensive web-based solution. For example, the Login management component handles user authentication, which is then used by other components like Components authentication and Registration management to ensure secure access to the system's features.

Regarding the MVC architecture, it is a software design pattern that separates an application's data, user interface, and control logic into three interconnected components: the Model, the View, and the Controller.

- The Model represents the application's data and the business logic. It is responsible for managing the data and enforcing the rules of the application.
- The View represents the user interface. It is responsible for displaying the data to the user and handling user input.
- The Controller acts as an intermediary between the Model and the View. It receives user input, manipulates the Model, and updates the View accordingly.

In the context of this web system, each component likely has its own set of Models, Views, and Controllers that work together to provide the desired functionality. For example, the Login management component would have a Model to manage user accounts, a View to display the login interface, and a Controller to handle the login process.

The use of the MVC architecture helps to promote modularity, maintainability, and testability of the web system, as the separation of concerns allows for easier development, debugging, and scalability of the various components.

Use Case Narratives

students

Use Case Rate and review.

Use-Case Name:	Rate teachers and institutes	
Use-Case ID:	UC-001	
Priority:	medium	
Trigger:	Using the rating system in the teacher/institute profile	
Primary Actor:	Student.	
Pre Conditions:	Logged in to the system. Should be enrolled to a class of the particular teacher or institute	
Post Conditions:	-	
Description:	Student will be able to rate their teachers and the institutes based on their experience	
Main Scenario:	1	User navigates to the "My Classes" page.
	2	User selects a particular class from their list.
	3	User chooses to view either the teacher's profile or the institute's profile associated with the class.
	4	User navigates to the ratings section on the profile page
	5	User adds ratings (e.g., star ratings, written feedback) for the teacher or institute.
	6	User submits the ratings.
	7	User exits from the profile page.

Use Case View teacher/institute profile.

Use-Case Name:	View teacher/institute profile	
Use-Case ID:	UC-002	
Priority:	medium	
Trigger:	Clicking view button in each table	
Primary Actor:	Student.	
Pre Conditions:	Logged in to the system.	
Post Conditions:	-	
Description:	Student will be able to check details about each and every teachers and institutes	
Main Scenario:	1	User navigates to the browsing page
	2	User selects a particular stream/subject of interest.
	3	User views the list of available teachers or institutes offering courses in the selected stream/subject.
	4	User chooses a specific teacher or institute from the list.
	5	User clicks on the "View Profile" button associated with the selected teacher or institute.
	6	User views the detailed profile information (e.g., bio, qualifications, courses offered) of the selected teacher or institute.
	7	User exits from the profile view

Use Case- Pay Class fees

Use-Case Name:	Pay class fees	
Use-Case ID:	UC-003	
Priority:	high	
Trigger:	Using the payment tab in my class page	
Primary Actor:	Student.	
Pre Conditions:	Logged in to the system. Should be enrolled to the class	
Post Conditions:	-	
Description:	Student will be able to check and pay their monthly fees to the relevant classes	
Main Scenario:	1	User navigates to the "My Classes" page.
	2	User selects a particular class from their list.
	3	User proceeds to the payment page for the selected class.
	4	User chooses between the options for payment (online payment or physical payment).
	5	If choosing online payment: User enters relevant payment details (e.g., credit card information). User confirms the payment details. User submits the payment.
	6	If choosing physical payment: User follows instructions to proceed with physical payment (e.g., bank transfer details, mailing address for check). User confirms the details. User submits the payment.

	7	User completes the payment process.
	8	User exits from the payment page.

Use Case –Contact Teacher.

Use-Case Name:	Contact teacher.	
Use-Case ID:	UC-004	
Priority:	Medium.	
Trigger:	When clicked the button contact on teacher profile in My class page.	
Primary Actor:	Student.	
Pre Conditions:	Logged in to the system. Has enrolled to a class which is conducted by the selected teacher.	
Post Conditions:	-	
Description:	Student can send messages to the course teacher by using a simple form	
Main Scenario:	1	User navigates to the "My Classes" page.
	2	User selects a particular class from their list.
	3	User accesses the teacher's profile from the class details.
	4	User clicks on the "Contact Teacher" button on the teacher's profile.
	5	User fills out the contact form with the required details (e.g., name, email, message).
	6	User clicks the "Send" button to submit the contact form.

	7	User exits to the "My Classes" page after sending the message.
--	---	--

Use Case –Access learning materials.

Use-Case Name:	Access learning materials	
Use-Case ID:	UC-005	
Priority:	High	
Trigger:	After clicking the view materials button in the class tab in my classes page	
Primary Actor:	Student.	
Pre Conditions:	Logged in to the system. Should be enrolled to the class Should have paid monthly fee	
Post Conditions:	-	
Description:	Students can access learning materials of the class they have enrolled in (only the materials of current month)	
Main Scenario:	1	User navigates to the "My Classes" page.
	2	User selects a particular class from their list.

	3	User clicks the "View Learning Materials" button for the selected class.
	4	User accesses and views the learning materials provided for the class.

Use-Case Name:	View class schedules	
Use-Case ID:	UC-006	
Priority:	medium	
Trigger:	Clicking timetable button on the particular class on my classes page	
Primary Actor:	Student.	
Pre Conditions:	Logged in to the system. Should be enrolled to the particular class	
Post Conditions:	-	
Description:	Student will be able to check the schedule of classes after enrolling to a class	
Main Scenario:	1	User navigates to the "My Classes" page.
	2	User selects a particular class from their list.
	3	User clicks on the "View Timetable" button or link for the selected class.
	4	User views the timetable for the selected class.
	5	User exits from the timetable view

Teacher

Use Case- Reply to Students forms

Use-Case Name:	Reply to students	
Use-Case ID:	UC-007	
Priority:	medium	
Trigger:	Clicking chats in manage classes page	
Primary Actor:	Teacher	
Pre Conditions:	Logged in to the system. Should have a class which is taught by him/her	
Post Conditions:	-	
Description:	Teacher will be able to reply to students messages	
Main Scenario:	1	User navigates to the "Manage Classes" page.
	2	User selects a particular class from the list of classes they manage.
	3	User navigates to the inbox or messages section for the selected class.
	4	User selects the message from the list of received messages.
	5	User clicks on the "Reply" button or link associated with the selected message.
	6	User types the reply message in the text box provided.
	7	User clicks the "Submit" or "Send" button to send the reply.
	8	User exits from the inbox or messages page.

Use Case- View student info

Use-Case Name:	View student info	
Use-Case ID:	UC-008	
Priority:	high	
Trigger:	clicking view students button in manage classes page	
Primary Actor:	Teacher	
Pre Conditions:	<p>Logged in to the system.</p> <p>Should have a class which is taught by him/her Should have students enrolled to that particular class</p>	
Post Conditions:	-	
Description:	Teacher will be able to check student info	
Main Scenario:	1	User navigates to the "Manage Classes" page.
	2	User selects a particular class from the list of classes they manage.
	3	User selects "View Student Table" or a similar option to access the student information for the selected class.
	4	User locates the specific student whose information needs to be updated.
	5	User updates the necessary information (e.g., contact details, attendance records) for the selected student.
	6	User clicks the "Submit" or "Update" button to save the changes made to the student's information.
	7	User exits from the student information page.

Use Case- Apply to institute

Use-Case Name:	Apply to institutes	
Use-Case ID:	UC009	
Priority:	medium	
Trigger:	Click apply in the particular job role in job advertisement page	
Primary Actor:	Teacher	
Pre Conditions:	Logged in to the system.	
Post Conditions:	-	
Description:	Teacher will be able to apply for jobs in a institute	
Main Scenario:	1	User navigates to the "Job Advertisement" page
	2	User selects a particular job role from the list of available job advertisements.
	3	User clicks on the "Apply" button associated with the selected job role.
	4	User fills out the application form with required details (e.g., personal information, resume/CV upload, cover letter).
	5	User reviews the filled-out form to ensure accuracy and completeness.
	6	User clicks the "Submit" or "Apply Now" button to submit the job application.
	7	User exits from the job application page.

*Usecase - Upload
learning materials*

Use-Case Name: •	Upload learning materials	
Use-Case ID:	UC-010	
Priority:	High	
Trigger:	Navigating to learning material page	
Primary Actor:	Teacher	
Pre Conditions:	Logged in to the system. Should have a class which is taught by him/her	
Post Conditions:	-	
Description:	Teacher will be able to upload learning materials to classes	
Main Scenario:	1	User navigates to the "Manage Classes" page.
	2	User selects a particular class from the list of classes they manage.
	3	User navigates to the learning materials page or section for the selected class.
	4	User clicks on the "Upload Materials" button or link.
	5	User selects the files or drag-and-drops files to upload as learning materials.
	6	User verifies or enters details related to the uploaded materials (e.g., description, tags).
	7	User clicks the "Submit" or "Upload" button to upload the materials.
	8	User exits from the upload materials page.

*Use Case- Schedule
Classes-T*

Use-Case Name:	Schedule classes	
Use-Case ID:	UC-011	
Priority:	High	
Trigger:	Click schedule timetable button in my classes page	
Primary Actor:	Teacher	
Pre Conditions:	Logged in to the system. Should have classes Should have bought a subscription plan	
Post Conditions:	-	
Description:	Teacher will be able to schedule his own classes	
Main Scenario:	1	User navigates to the "Manage Classes" page.
	2	User selects "My Private Classes" or a similar option to access their private classes.
	3	User identifies the specific private class for which the timetable needs to be updated.
	4	User clicks on the "Update Timetable" or similar option for the selected private class.
	5	User fills out the timetable update form with the new schedule details (e.g., date, time, location).
	6	User reviews the filled-out form to ensure accuracy.
	7	User clicks the "Submit" or "Update" button to save the updated timetable.
	8	User exits from the timetable update page.

*Usecase Update
subscription-T*

Use-Case Name:	Update subscription-T	
Use-Case ID:	UC-012	
Priority:	High	
Trigger:	Clicking my subscription plan in home page	
Primary Actor:	Teachers with premium subscription	
Pre Conditions:	Logged in to the system. Should have a premium subscription.	
Post Conditions:	Payment should be done	
Description:	Teacher should be able to update the subscription plan to enjoy an uninterrupted service	
Main Scenario:	1	User navigates to the home page of the platform.
	2	User selects "My Subscription Plans" or a similar option to view their subscription details.
	3	User clicks on the option to view their subscription details or plans.
	4	If the current subscription plan is expired, user clicks on the "Update Plan" or similar button.
	5	User proceeds to the payment page to renew or update the subscription plan.
	6	User selects the payment method and enters relevant payment details (e.g., credit card information).
	7	User confirms the payment details.
	8	User clicks the "Submit" or "Pay Now" button to complete the payment process.
	9	User exits from the subscription update and payment pages.

Post advertisement-T

Use-Case Name:	Post advertisement-T	
Use-Case ID:	UC-013	
Priority:	medium	
Trigger:	Navigating to advertisement page and clicking post advertisement	
Primary Actor:	Teachers with premium subscription	
Pre Conditions:	<p>The teacher is logged into their account on the platform.</p> <p>The platform provides functionality for teachers to post advertisements or announcements.</p> <p>The teacher has permissions to post advertisements based on their role and settings.</p>	
Post Conditions:	-	
Description:	This use case describes the process of a teacher posting an advertisement or announcement through an online platform.	
Main Scenario:	1	Teacher logs into their account on the platform.
	2	Teacher navigates to the dashboard or relevant section for posting advertisements.
	3	Teacher clicks on the option to "Post Advertisement" or "Create Announcement."
	4	<p>Teacher fills out the advertisement details:</p> <p>Advertisement title</p> <p>Advertisement content (description, requirements, contact information)</p> <p>Optional: Attachments or images related to the advertisement</p>
	5	Teacher reviews the filled-out advertisement to ensure accuracy.
	6	Teacher clicks the "Submit" or "Post Advertisement" button to publish the advertisement.
	7	Teacher exits from the advertisement posting page.

offer discounts and free cards-T

Use-Case Name:	Offer discounts and free cards	
Use-Case ID:	UC-014	
Priority:	medium	
Trigger:	Clicking offer a discount button in student table	
Primary Actor:	Teacher	
Pre Conditions:	Logged in to the system. Have a class under the particular teacher Teacher have a subscription plan	
Post Conditions:	-	
Description:	Teacher will be able to provide discounts and free cards for students	
Main Scenario:	1	Teacher logs into their account on the platform.
	2	Teacher navigates to the "My Classes" or "Manage Classes" section.
	3	Teacher selects the specific class they are managing.
	4	Teacher accesses the student list or manages student details for the selected class.
	5	Teacher identifies the student(s) to whom they want to apply a discount.
	6	Teacher applies the discount by adjusting the pricing or offering a coupon code to eligible students.
	7	Teacher confirms the discount application and notifies the students (if necessary).
	8	Teacher exits from the class management or student details page.

Request payroll-T

Use-Case Name:	Request Payroll	
Use-Case ID:	UC-015	
Priority:	medium	
Trigger:	Requesting a payroll from the institute	
Primary Actor:	Teacher	
Pre Conditions:	Logged in to the system. Registered under an institute	
Post Conditions:	-	
Description:	teacher can request a payroll from the institute	
Main Scenario:	1	Teacher logs into their account on the platform.
	2	Teacher navigates to the "My Institutes" or "Institute Dashboard" section.
	3	Teacher selects the relevant institute or organization where they are employed.
	4	Teacher accesses the payroll request form or section within the selected institute's profile.
	5	Teacher fills out the payroll request form with necessary details:
	6	Teacher reviews the filled-out payroll request form to ensure accuracy and completeness.
	7	Teacher clicks the "Submit" or "Request Payroll" button to submit the payroll request.
	8	Teacher receives confirmation of the payroll request submission.
	9	Teacher exits from the payroll request page.

User

Use Case – View Advertisement

Use-Case Name:	View Advertisements	
Use-Case ID:	UC-016	
Priority:	Medium	
Trigger:	When a user navigates to the advertisements section or views an advertisement on the main dashboard.	
Primary Actor:	User	
Pre Conditions:	The user must be logged into the system (except for Guest Users).	
Post Conditions:	The user views the advertisements. The system may track advertisement views for analytics.	
Description:	This use case describes the steps a user takes to view advertisements through the system.	
Main Scenario:	1	The user navigates to the "Advertisements" section from the main menu or dashboard.
	2	The system displays a list of available advertisements.
	3	The user can search particular advertisements.
	4	The user clicks on an advertisement to view its details.
	5	The user reviews the advertisement details.
	6	If interested in enrolling in or searching for a class, the user clicks on the "Enroll" or "Search Class" button within the advertisement.
	7	The user reviews the advertisement details
	8	The user can close the tab

Use Case – Browse details

Use-Case Name:	Browse Details	
Use-Case ID:	UC-017	
Priority:	Medium	
Trigger:	When a user selects to view details about a specific user, teacher, or institute.	
Primary Actor:	User	
Pre Conditions:	<p>The user must be logged into the system (except for Guest Users).</p> <p>The system must have details available for the selected teacher, or institute.</p>	
Post Conditions:	<p>The user views the detailed information about the selected teacher, or institute.</p> <p>The system may track user interactions with the details for analytics purposes.</p>	
Description:	This use case describes the steps a user takes to browse detailed information about a specific teacher, or institute through the system.	
Extensions :	"Users can choose to browse details by teacher or institute."	
Main Scenario:	1	The user logs into the system using their credentials (if required).
	2	The user navigates to the appropriate section to browse details:
	3	The system displays a search interface or options to find the desired teacher or institute.
	4	The user enters search criteria or selects from available options to locate the desired profile.
	5	The system retrieves and displays detailed information about the selected teacher or institute
	6	The user reviews the displayed information.

	7	The user can interact with the displayed details, such as contacting the teacher or institute directly from the details page.
	8	The system may log the user's interaction with the details and any subsequent actions for analytics purposes.

Use Case – View blogs entries

Use-Case Name:	View Blog Entries	
Use-Case ID:	UC-018	
Priority:	Medium	
Trigger:	When a user selects to view blog entries authored by a specific user.	
Primary Actor:	User	
Pre Conditions:	The user must be logged into the system and system must have blog entries authored by the selected user available	
Post Conditions:	The user views the blog entries authored by the selected user and system may track user interaction	
Description:	This use case describes the steps a user takes to view blog entries authored by a specific user.	
Main Scenario:	1	The user log in to the system
	2	The user navigates to the section for viewing blogs
	3	The user selects an option to view blog entries
	4	The system retrieves and displays blog entries
	5	The user can click blog entries to view its contents
	6	The user can navigate back to the list of blog entries

	7	User can exit and close the tab
--	---	---------------------------------

Use Case – Contact Support

Use-Case Name:	Contact Support	
Use-Case ID:	UC-019	
Priority:	Medium	
Trigger:	When a user initiates a request for support or assistance.	
Primary Actor:	User	
Pre Conditions:	The user must be logged into the system (except for Guest Users).	
Post Conditions:	The user's support request is submitted to the system. The system acknowledges receipt of the support request. A support agent or system administrator responds to the user's request.	
Description:	his use case describes the steps a user takes to contact support for assistance or to report an issue within the system.	
Main Scenario:	1	The user navigates to the "Support" or "Contact Us" section from the main menu or dashboard.

	2	The system displays options for contacting support, such as a contact form, email address, or direct messaging system.
	3	The user selects the preferred method to contact support
	4	The system presents a form or interface for the user to enter details about their support request
	5	The user fills out the contact form with the required information.
	6	The user reviews the entered information
	7	The user submits the support request by clicking on the "Submit" or "Send" button.
	8	The system confirms receipt of the support request

Use Case – Sign Up

Use-Case Name:	Sign Up
Use-Case ID:	UC-020
Priority:	High
Trigger:	When a user initiates the sign-up process to create a new account
Primary Actor:	Unregisterd user
Pre Conditions:	The user is not logged into the system.
Post Conditions:	The user successfully creates a new account and is logged into the system.
Extensions :	When user sign up to the system, user can chose buy subscription, it is not mandataray (Optional)

Description:	This use case describes the steps a guest user takes to sign up and create a new account within the system.	
Main Scenario:	1	The guest user navigates to the sign-up or registration page.
	2	The system displays a registration form with fields such as Name, Email, Password like that
	3	The guest user fills out the registration form with valid information.
	4	The system validates the entered information
	5	If any validation errors occur, the system displays error messages and prompts the user to correct the information.
	6	Once the information is validated, the guest user submits the registration form by clicking on the "Sign Up" or "Create Account" button.
	7	The system creates a new user account using the provided information and assigns a unique user ID.
	8	The system logs the user into the newly created account

Use Case – Log in

Use-Case Name:	Log In
Use-Case ID:	UC-021
Priority:	High
Trigger:	When a user initiates the login process to access their account
Primary Actor:	Registered User
Pre Conditions:	The user is not logged into the system.
Post Conditions:	The user successfully logs into their account and gains access to system features.

Description:	This use case describes the steps a user takes to log into their existing account within the system.	
Main Scenario:	1	The user navigates to the login page or interface.
	2	The system displays a login form with fields such as Username password
	3	The user enters their username and password.
	4	The user submits the login form by clicking on the "Log In" or "Sign In" button.
	5	The system verifies the entered credentials against the stored user database.
	6	If the entered credentials are correct : The system authenticates the user's identity And the system logs the user into their account.
	7	If the entered credentials are incorrect :The system displays an error message indicating invalid credentials and the user is prompted to re-enter their username and password.
	8	Upon successful login, the system redirects the user to their dashboard, main menu, or the last accessed page.

Edit Profile

Use-Case Name:	Edit Profile	
Use-Case ID:	UC-022	
Priority:	Medium	
Trigger:	When a user initiates the process to edit their profile information.	
Primary Actor:	Registered User	
Pre Conditions:	The user must be logged into the system.	
Post Conditions:	The user successfully updates their profile information.	
Description:	This use case describes the steps a user takes to edit their profile information within the system.	
Main Scenario:	1	The user navigates to the profile or account settings section.
	2	The system displays options for editing profile information, such as personal details, Contact info, educational info, Preferences like that
	3	The user selects the section or field they wish to edit.
	4	The system presents a form or interface populated with the current information.
	5	The user makes changes to the relevant fields.
	6	The system validates the entered information for accuracy and completeness.
	7	If any validation errors occur: The system displays error messages next to the respective fields. The user corrects the information as needed.
	8	The user saves the changes by clicking on the "Save" and confirms the successful update

Add Blog Entries

Use-Case Name:	Add Blog Entries	
Use-Case ID:	UC-023	
Priority:	Medium	
Trigger:	When Registered User clicks the "Add Blog Entry" button on the blog page.	
Primary Actor:	Registered student , Registered Teacher, System admin	
Pre Conditions:	The Registered user must be logged into the system.	
Post Conditions:	<p>A new blog entry is created and visible to other users.</p> <p>The Registered user receives a confirmation that the blog entry has been successfully posted.</p>	
Description:	This use case describes the steps a Registered user takes to add a blog entry through the system.	
Main Scenario:	1	The Registered user logs into the system using their credentials.
	2	The Registered user navigates to the "Blog" section.
	3	The system displays a list of existing blog entries. And the student clicks the "Add Blog Entry" button.
	4	The system displays a form for creating a new blog entry.
	5	The Registered user enters the title, content, and any other required details and user clicks the "Submit" button.
	6	The system saves the new blog entry to the database.
	7	The system displays the new blog entry in the list of blog entries.
	8	The user receives confirmation that the blog entry has been successfully posted

Delete Blog Entries

Use-Case Name:	Delete Blog Entries	
Use-Case ID:	UC-024	
Priority:	Medium	
Trigger:	When the student clicks the "Delete" button on one of their blog entries.	
Primary Actor:	Registered student , Registered Teacher, System admin	
Pre Conditions:	The student must be logged into the system. The student must have at least one blog entry created.	
Post Conditions:	The selected blog entry is removed from the system. The student receives a confirmation that the blog entry has been successfully deleted.	
Description:	This use case describes the steps a student takes to delete a blog entry through the system..	
Main Scenario:	1	The student logs into the system using their credentials.
	2	The student navigates to the "Blog" section.
	3	The system displays a list of existing blog entries, including those created by the student.
	4	The student locates the blog entry they wish to delete.
	5	The student clicks the "Delete" button associated with the selected blog entry.
	6	The system prompts the student to confirm the deletion and confirms deletion
	7	The system removes the blog entry from the database.
	8	The student receives a confirmation that the blog entry has been successfully deleted.

Use Case – Post Advertisements

Use-Case Name:	Post Advertisements	
Use-Case ID:	UC-025	
Priority:	Medium	
Trigger:	The institute admin clicks on the "Post Advertisement" button.	
Primary Actor:	Registered Institute	
Pre Conditions:	<ul style="list-style-type: none"> · The institute is registered in the system. · The institute admin is logged into the system. · Advertisement posting permissions are granted to the institute. 	
Post Conditions:	The advertisement is successfully posted, and the details are updated in the system.	
Description:	This use case allows a registered institute to post advertisements for events, new courses, or other relevant information. The process includes entering advertisement details, reviewing, and posting the advertisement.	
Main Scenario:	1	The institute admin logs into the system.
	2	The system verifies the admin's credentials and grants access.
	4	The admin clicks on the "Post Advertisement" button.
	5	The system displays a form for entering advertisement details (e.g., title, description, date, and time).
	6	The admin fills in the required advertisement details.
	7	The admin reviews the advertisement details for accuracy.
	8	The admin submits the advertisement for posting.
	9	The system verifies the entered information and posts the advertisement.
	10	The system updates the advertisement section with the new post.
	11	The admin can view the posted advertisement or log out of the system.

Use Case – schedule classes

Use-Case Name:	Scheduling classes	
Use-Case ID:	UC-026	
Priority:	High	
Trigger:	The institute admin clicks on the "Schedule Classes" button.	
Primary Actor:	Registered Institute	
Pre Conditions:	<ul style="list-style-type: none"> · The institute is registered in the system. · The institute admin is logged into the system. · Teacher and student details are available in the system. 	
Post Conditions:	Classes are successfully scheduled, and the schedule is updated in the system.	
Description:	This use case allows a registered institute to schedule classes. The process includes selecting the class, assigning a teacher, specifying the date and time, and updating the schedule in the system.	
Main Scenario:	1	The institute admin logs into the system.
	2	The system verifies the admin's credentials and grants access.
	3	The admin navigates to the "Classes" section.
	4	The admin clicks on the "Schedule Classes" button.
	5	The system displays a form for entering scheduling details (e.g., class name, teacher, date, time, and location).
	6	The admin fills in the required scheduling details.
	7	The admin reviews the scheduling details for accuracy.
	8	The admin submits the schedule for confirmation.

	9	The system verifies the entered information and schedules the class.
	10	The system updates the class schedule and notifies the relevant teachers and students.

Use Case – Offer/discounts

Use-Case Name:	Offer/discounts	
Use-Case ID:	UC-027	
Priority:	low	
Trigger:	The institute admin clicks on the "Offer Discounts" button	
Primary Actor:	Registered Institute	
Pre Conditions:	<ul style="list-style-type: none"> · The institute is registered in the system. · The institute admin is logged into the system. · Discount criteria and policies are defined in the system. 	
Post Conditions:	The discount offer is successfully created, and the details are updated in the system.	
Description:	This use case allows a registered institute to offer discounts on courses or services. The process includes entering discount details, setting the criteria for the discount, and updating the discount offers in the system.	
Main Scenario:	1	The institute admin logs into the system.
	2	The system verifies the admin's credentials and grants access.
	3	The admin navigates to the "Discounts" section.
	4	The admin clicks on the "Offer Discounts" button.
	5	The system displays a form for entering discount details (e.g., discount name, percentage, applicable courses, start date, end date, and criteria).
	6	The admin fills in the required discount details.

	7	The admin reviews the discount details for accuracy.
	8	The admin submits the discount offer for confirmation.
	9	The system verifies the entered information and applies the discount offer.
	10	The system updates the discount section with the new offer.

Use Case –

Recruit Teachers

Use-Case Name:	Recruit Teachers	
Use-Case ID:	UC-028	
Priority:	High	
Trigger:	The institute admin clicks on the "Recruit Teachers" button.	
Primary Actor:	Registered Institute	
Pre Conditions:	<ul style="list-style-type: none">· The institute is registered in the system.· The institute admin is logged into the system.	
Post Conditions:	The institute admin successfully recruits a new teacher and the teacher's details are added to the system.	
Description:	This use case allows a registered institute to recruit new teachers for their classes. The process includes providing the necessary details about the new teacher, verifying the information, and adding the teacher to the system.	
Main Scenario:	1	The institute admin logs into the system.
	2	The system verifies the admin's credentials and grants access.
	3	The admin navigates to the "Teachers" section.
	4	The admin clicks on the "Recruit Teachers" button.
	5	The system displays a form for entering the new teacher's details.

Use Case –

	6	The admin fills in the required details (e.g., name, qualifications, contact information, subjects to be taught).
	7	The admin submits the form.
	8	The system verifies the entered information for completeness and correctness.
	9	The system adds the new teacher's details to the list of teachers.
	10	The admin can view the updated list of teachers or log out of the system.

Issue Payroll

Use Case –

Use-Case Name:	Issue Payroll	
Use-Case ID:	UC-029	
Priority:	High	
Trigger:	The institute admin clicks on the "Issue Payroll" button	
Primary Actor:	Registered Institute	
Pre Conditions:	<ul style="list-style-type: none"> · The institute is registered in the system. · The institute admin is logged into the system. Teacher salary details are configured in the system. 	
Post Conditions:	Payroll is successfully issued to the teachers, and payment records are updated in the system.	
Description:	This use case allows a registered institute to issue payroll to their teachers. The process includes verifying the salary details, processing the payments, and updating the payment records in the system.	
Main Scenario:	1	The institute admin logs into the system.
	2	The system verifies the admin's credentials and grants access.
	3	The admin navigates to the "Payroll" section.
	4	The admin clicks on the "Issue Payroll" button.
	5	The system displays a list of teachers along with their salary details.
	6	The admin reviews and verifies the salary details for each teacher.
	7	The admin confirms the issuance of payroll.
	8	The system processes the payments and updates the payment records.
	9	The system generates payroll reports for recordkeeping.
	10	The admin can view the payroll reports or log out of the system.

Use Case –

Update Subscription

Use-Case Name:	Update Subscription	
Use-Case ID:	UC-030	
Priority:	High	
Trigger:	The institute admin clicks on the "Update Subscription" button.	
Primary Actor:	Registered Institute	
Pre Conditions:	<ul style="list-style-type: none"> · The institute is registered in the system. · The institute admin is logged into the system. · The current subscription details are available. 	
Post Conditions:	The institute's subscription is successfully updated, and the new subscription details are reflected in the system.	
Description:	This use case allows a registered institute to update their subscription plan. The process includes selecting a new subscription plan, confirming the update, and processing any necessary payments.	
Main Scenario:	1	The institute admin logs into the system.
	2	The system verifies the admin's credentials and grants access.
	3	The admin navigates to the "Subscription" section.
	4	The admin clicks on the "Update Subscription" button.
	5	The system displays the current subscription details and available plans.
	6	The admin selects a new subscription plan from the available options.
	7	The system displays the details and cost of the selected plan.
	8	The admin confirms the update to the new subscription plan.
	9	The system processes any necessary payments and updates the subscription details.

Use Case –

	10	The system generates a confirmation of the updated subscription.
--	----	--

Use Case – View Teachers Details

Use-Case Name:	View teachers details																
Use-Case ID:	UC-031																
Priority:	Medium																
Trigger:	The institute admin clicks on the "View Teachers" button.																
Primary Actor:	Registered Institute																
Pre Conditions:	The institute is registered in the system. The institute admin is logged into the system.																
Post Conditions:	The institute admin successfully views the details of the teachers.																
Description:	This use case allows a registered institute to view details about the teachers who are associated with the classes they manage. The details may include name, qualifications, contact information, subjects taught, and schedules.																
Main Scenario:	<table> <tr> <td>1</td><td>The institute admin logs into the system.</td></tr> <tr> <td>2</td><td>The system verifies the admin's credentials and grants access.</td></tr> <tr> <td>3</td><td>The admin navigates to the "Teachers" section.</td></tr> <tr> <td>4</td><td>The admin clicks on the "View Teachers" button.</td></tr> <tr> <td>5</td><td>The system displays a list of teachers associated with the institute.</td></tr> <tr> <td>6</td><td>The admin selects a teacher from the list to view details.</td></tr> <tr> <td>7</td><td>The system displays detailed information about the selected teacher.</td></tr> <tr> <td>8</td><td>The admin can choose to view details of other teachers or log out of the system.</td></tr> </table>	1	The institute admin logs into the system.	2	The system verifies the admin's credentials and grants access.	3	The admin navigates to the "Teachers" section.	4	The admin clicks on the "View Teachers" button.	5	The system displays a list of teachers associated with the institute.	6	The admin selects a teacher from the list to view details.	7	The system displays detailed information about the selected teacher.	8	The admin can choose to view details of other teachers or log out of the system.
1	The institute admin logs into the system.																
2	The system verifies the admin's credentials and grants access.																
3	The admin navigates to the "Teachers" section.																
4	The admin clicks on the "View Teachers" button.																
5	The system displays a list of teachers associated with the institute.																
6	The admin selects a teacher from the list to view details.																
7	The system displays detailed information about the selected teacher.																
8	The admin can choose to view details of other teachers or log out of the system.																

Use Case – View Student Details

Use-Case Name:	View Student Details	
Use-Case ID:	UC-032	
Priority:	Medium	
Trigger:	The institute admin clicks on the "View Students" button.	
Primary Actor:	Registered Institute	
Pre Conditions:	<ul style="list-style-type: none"> · The institute is registered in the system. · The institute admin is logged into the system. · There are students enrolled in the institute. 	
Post Conditions:	The institute admin successfully views the details of the students.	
Description:	This use case allows a registered institute to view details about the students who are enrolled in the classes they manage. The details may include name, age, contact information, enrolled courses, and attendance records.	
Main Scenario:	1	The institute admin logs into the system.
	2	The system verifies the admin's credentials and grants access.
	3	The admin navigates to the "Students" section.
	4	The admin clicks on the "View Students" button.
	5	The system displays a list of students enrolled in the institute.
	6	The admin selects a student from the list to view details.
	7	The system displays detailed information about the selected student.
	8	The admin can choose to view details of other students or log out of the system.

System admin

Use Case Respond to the contact form

Use-Case Name:	Respond to the contact form	
Use-Case ID:	UC-033	
Priority:	Medium	
Trigger:	A contact form is submitted by a user.	
Primary Actor:	System Administrator	
Pre Conditions:	<p>The contact form submission must be received and accessible to the System Administrator.</p> <p>The System Administrator must be logged into the system.</p>	
Post Conditions:	<p>The contact form is responded to and saved in the system.</p> <p>The response is sent to the user who submitted the contact form.</p>	
Description:	This use case describes the process by which the System Administrator responds to a contact form submitted by a user through the tuition management system	
Main Scenario:	1	The System Administrator receives a notification or accesses the contact form submission through the system dashboard.
	2	The System Administrator reviews the content of the contact form, including the user's query or message.
	3	The System Administrator composes a response to the contact form, addressing the user's query or providing necessary information.
	4	The System Administrator saves the response in the system, associating it with the original contact form submission
	5	The system sends the response to the user via email or updates the user through the system's messaging functionality.
	6	The use case ends once the response is successfully sent and saved.

Use Case -Post system announcement

Use-Case Name:	POST SYSTEM ANNOUNCEMENT	
Use-Case ID:	UC-034	
Priority:	Medium	
Trigger:	System Administrator initiates a new system announcement	
Primary Actor:	System Administrator	
Pre Conditions:	<ul style="list-style-type: none"> The System Administrator is logged into the tuition management system. The System Administrator has appropriate permissions to post announcements. 	
Post Conditions:	The announcement is successfully posted and visible to relevant users. Users are notified of the new announcement	
Description:	This use case describes the process of creating and publishing announcements within the tuition management system. Announcements can be used to inform users about important updates, events, or changes related to the system.	
Main Scenario:	1	The System Administrator selects the option to create a new announcement
	2	The System Administrator enters the title and content of the announcement
	3	Optionally, the System Administrator selects the target audience or groups for the announcement (e.g., all users, specific user roles).
	4	The System Administrator sets any additional parameters such as start and end dates for the announcement visibility.
	5	The system verifies the entered information and processes the announcement
	6	The announcement is saved and displayed on the system's announcement board or dashboard.
	7	Users within the designated audience receive notifications about the new announcement.
	8	The use case ends once the announcement is successfully posted and notifications are sent.

Use Case – Manage accounts

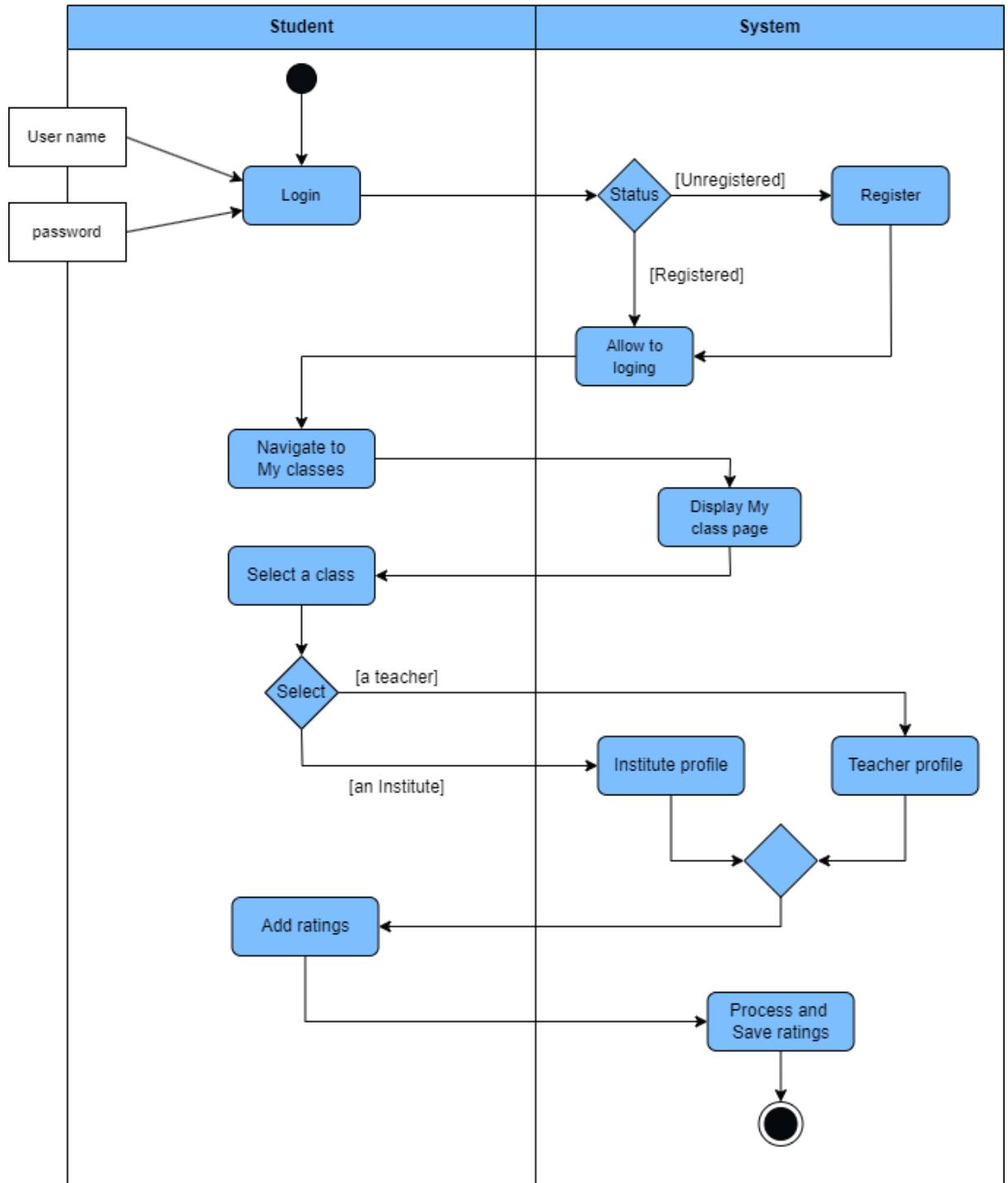
Use-Case Name:	MANAGE ACCOUNTS
Use-Case ID:	UC-035
Priority:	High
Trigger:	System Administrator initiates account management tasks
Primary Actor:	System Administrator
Pre Conditions:	<ul style="list-style-type: none"> • The System Administrator is logged into the tuition management system. • Access to user account management functionalities is granted to the System Administrator.
Post Conditions:	<ul style="list-style-type: none"> • User accounts are updated according to the actions performed (added or removed). • System records reflect the changes made to user accounts.
Description:	This use case encompasses the actions related to managing user accounts within the tuition management system by the System Administrator. It includes adding new users and removing existing users.

Main Scenario:	1	<p>ADD USERS:</p> <ul style="list-style-type: none"> • The System Administrator selects the option to add a new user. • The System Administrator enters the necessary details for the new user (e.g., username, password, role). • The system verifies the entered information and creates a new user account. • Confirmation of the new user account creation is displayed to the System Administrator.
----------------	---	--

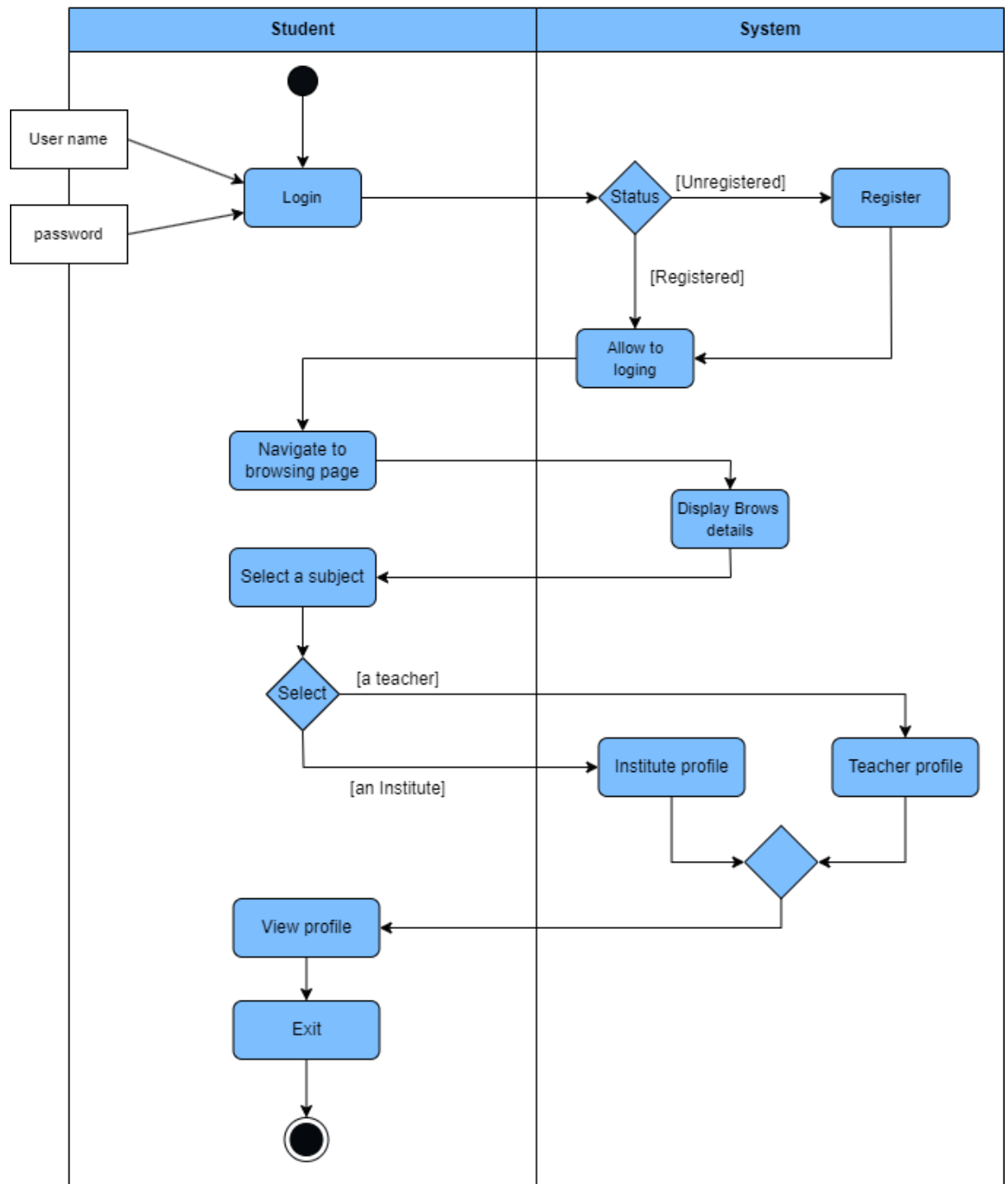
	2	<p>REMOVE USERS:</p> <ul style="list-style-type: none"> • The System Administrator selects the option to remove a user. • The System Administrator identifies the user to be removed from the system. • The system prompts the System Administrator to confirm the removal. • Upon confirmation, the system removes the user account from the system. • Confirmation of the user account removal is displayed to the System Administrator.
	3	<p>Extensions:</p> <ul style="list-style-type: none"> • Error Handling: If there are errors in adding or removing users (e.g., invalid user details, dependencies on the user account), appropriate error messages are displayed, and the System Administrator is prompted to correct the issue. • Undo: Depending on system requirements, an option to undo the removal of a user account could be provided to the System Administrator within a specified timeframe.

Activity

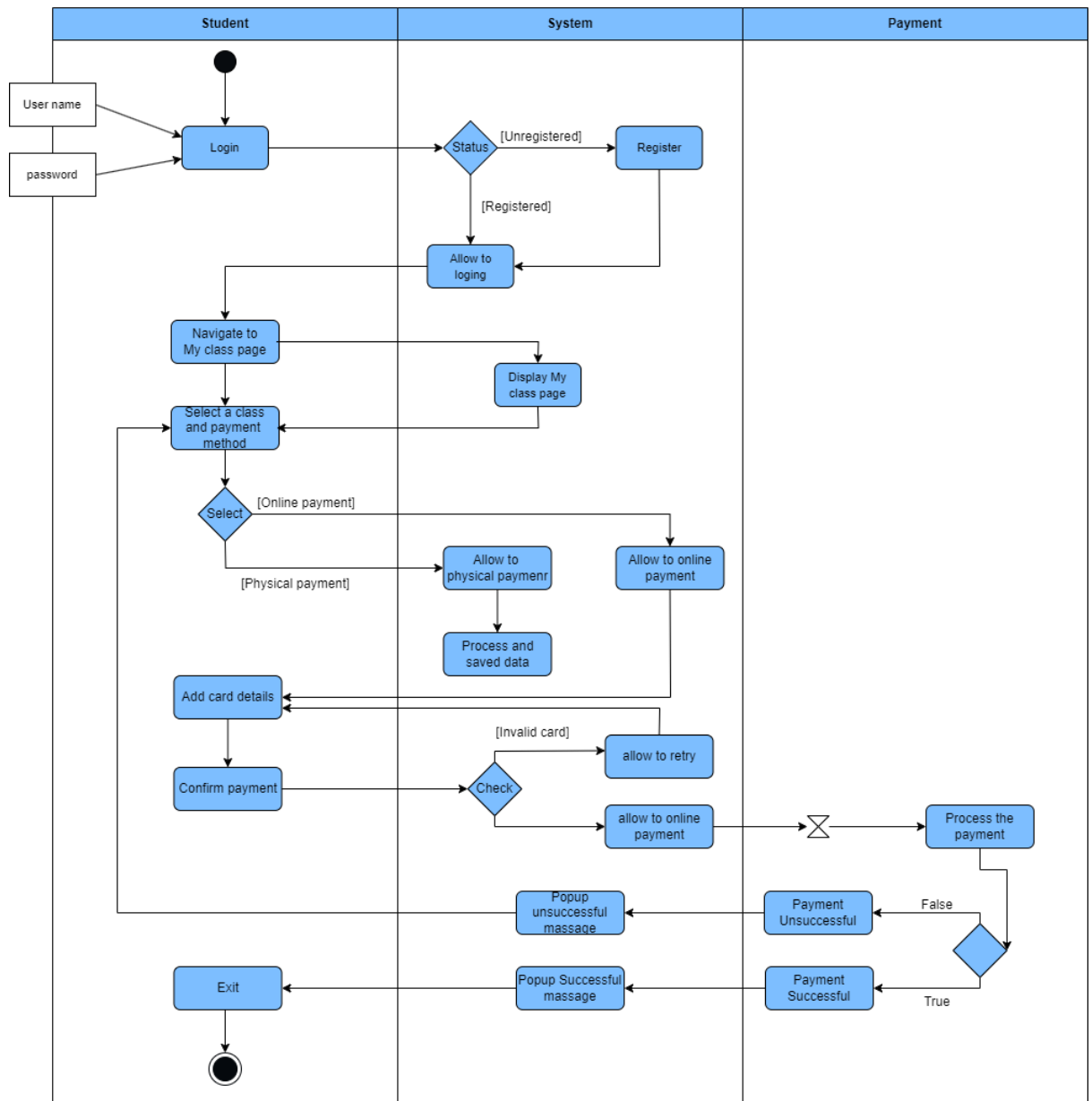
UC-001 - Rate teachers and institutes



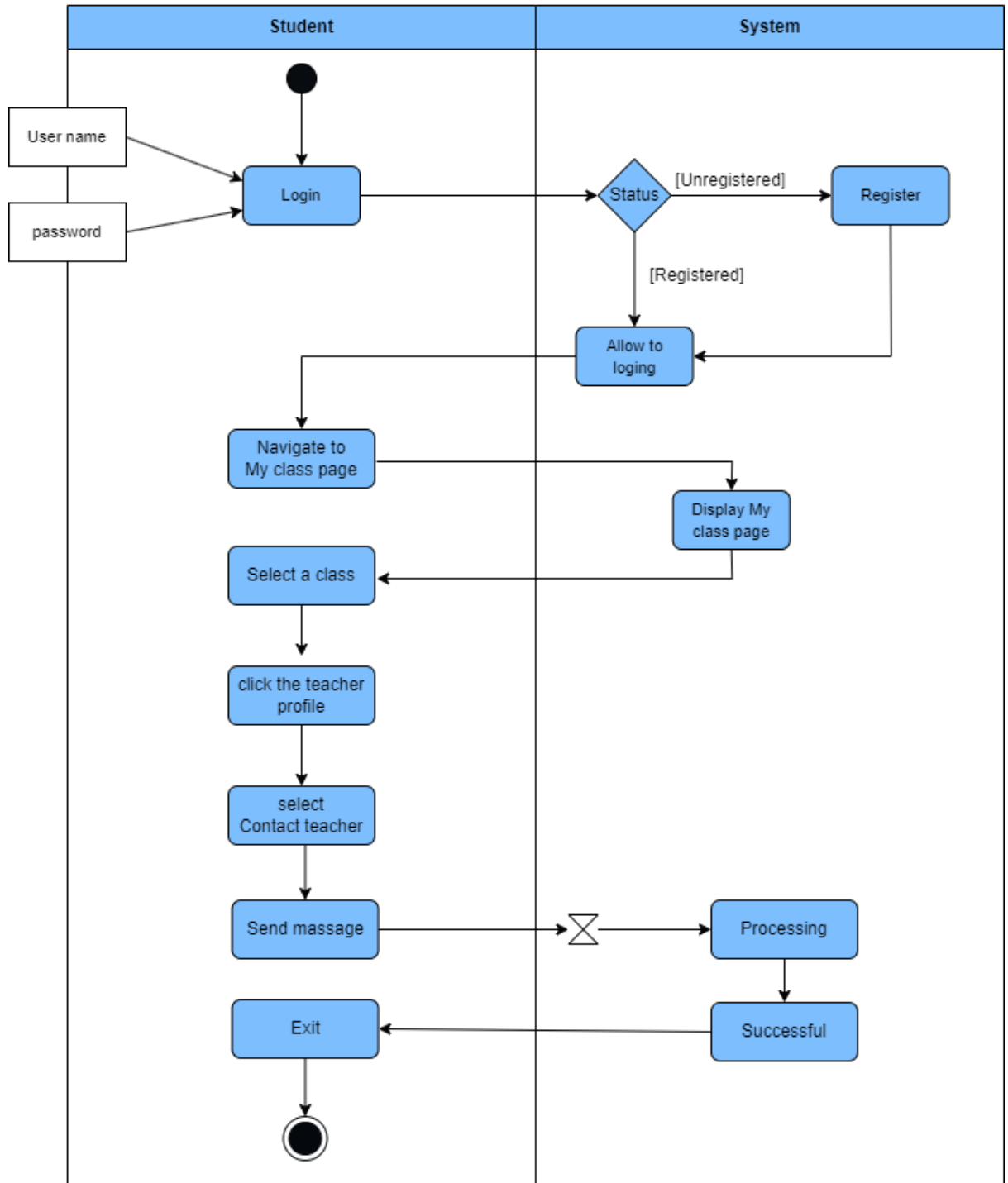
UC-002 - View teacher/institute profile



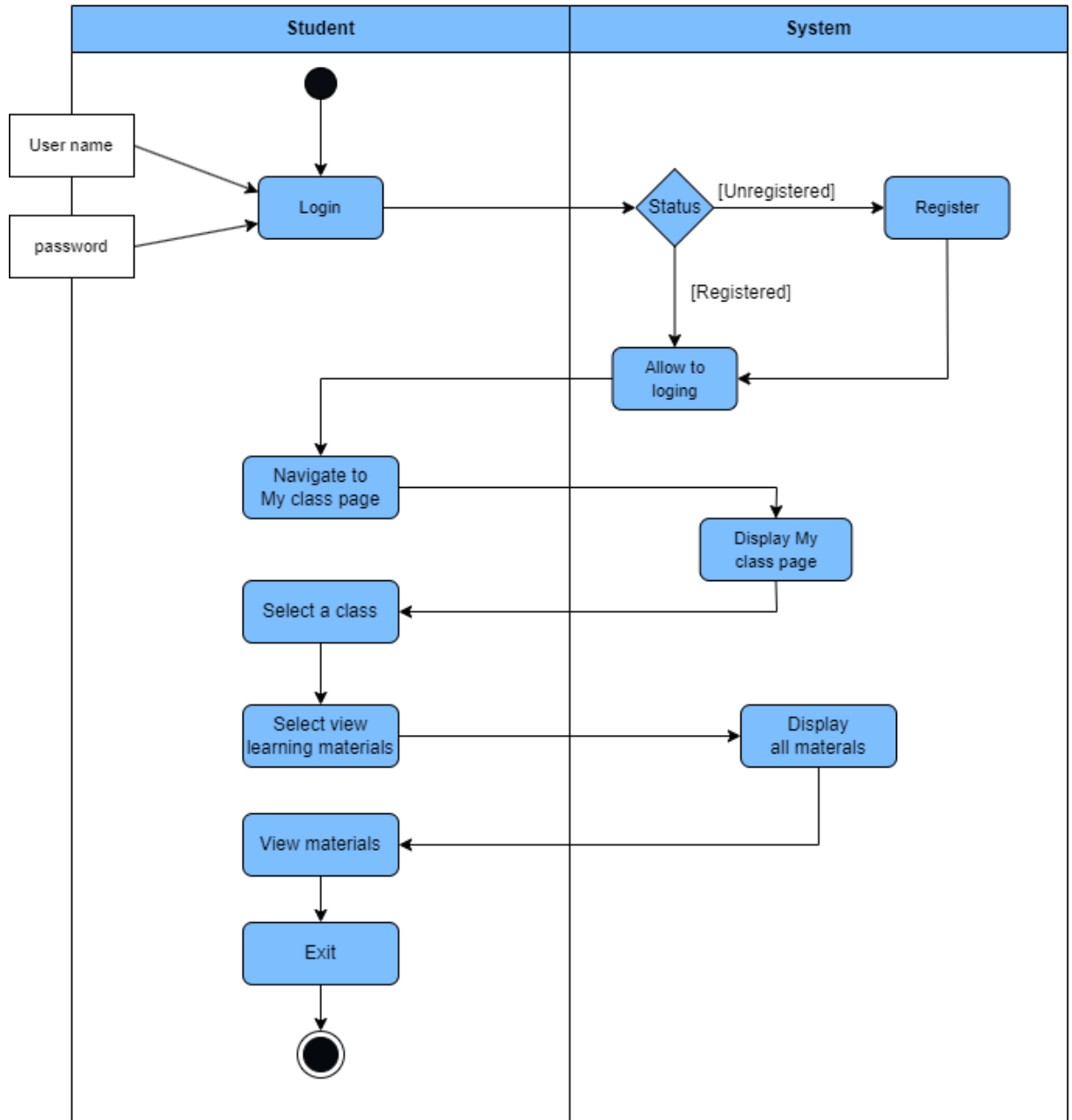
UC-003 - Pay class fees



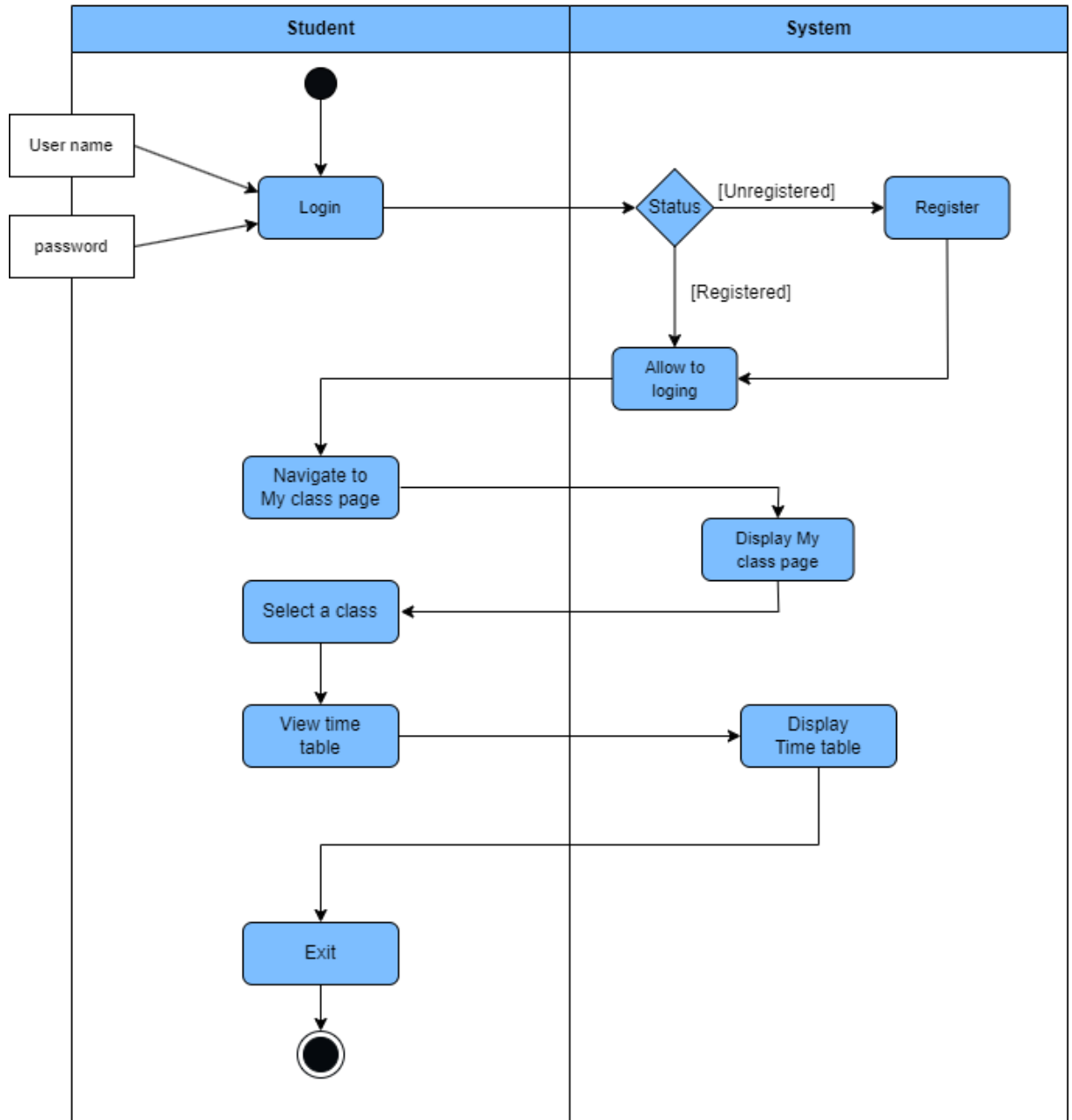
UC-004 - Contact teacher



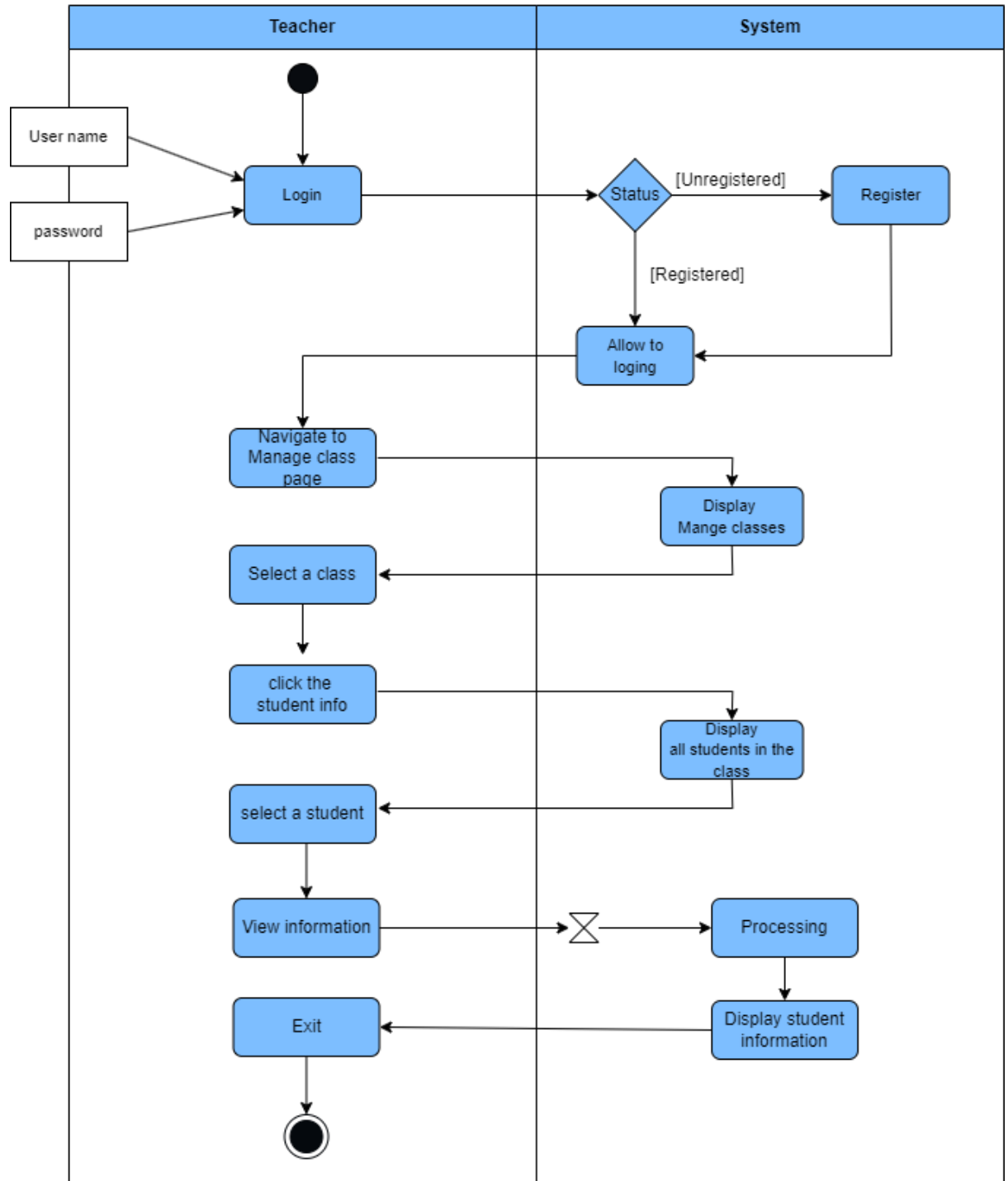
UC-005 - Access learning materials



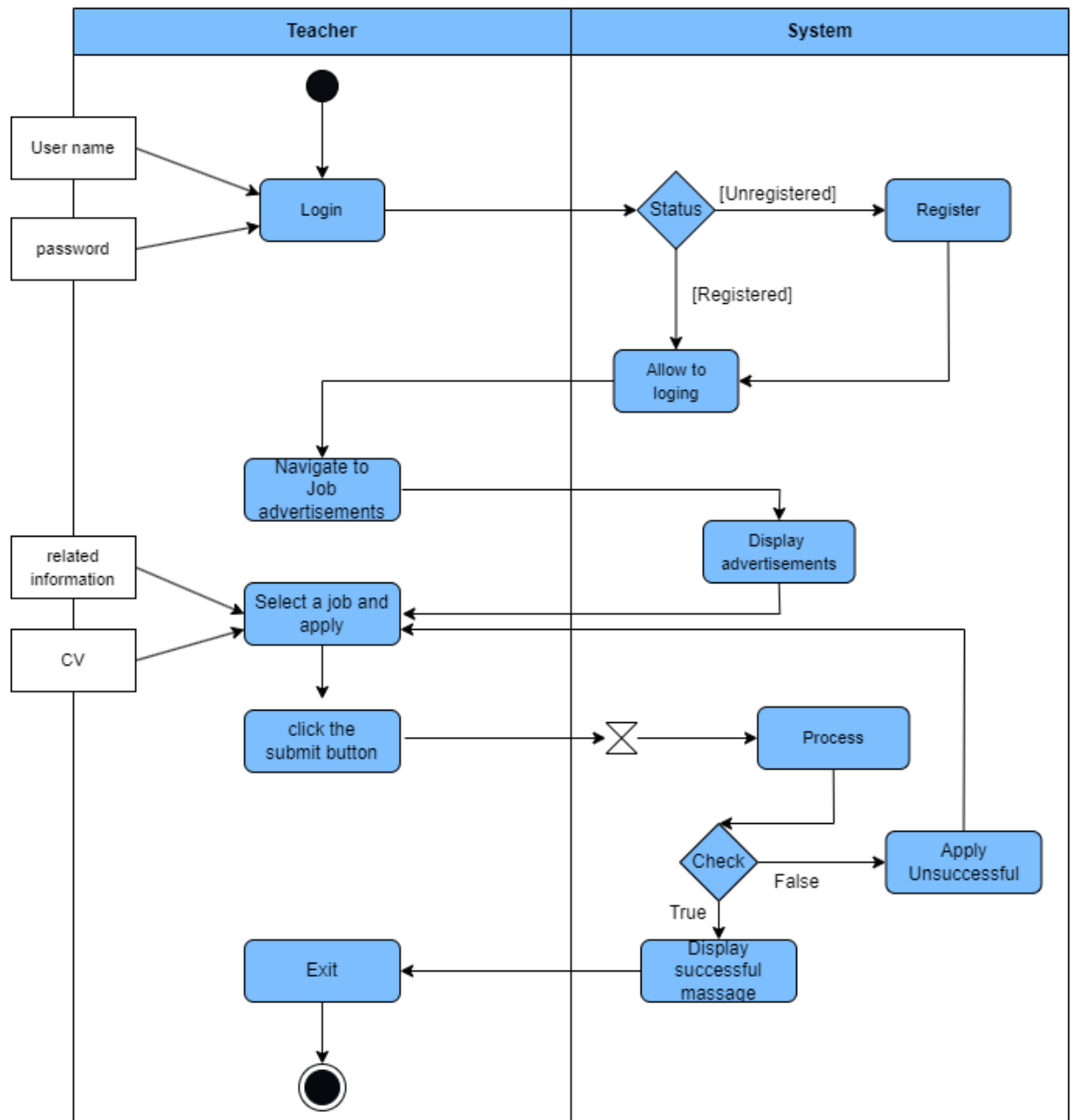
UC-006 - View class schedule



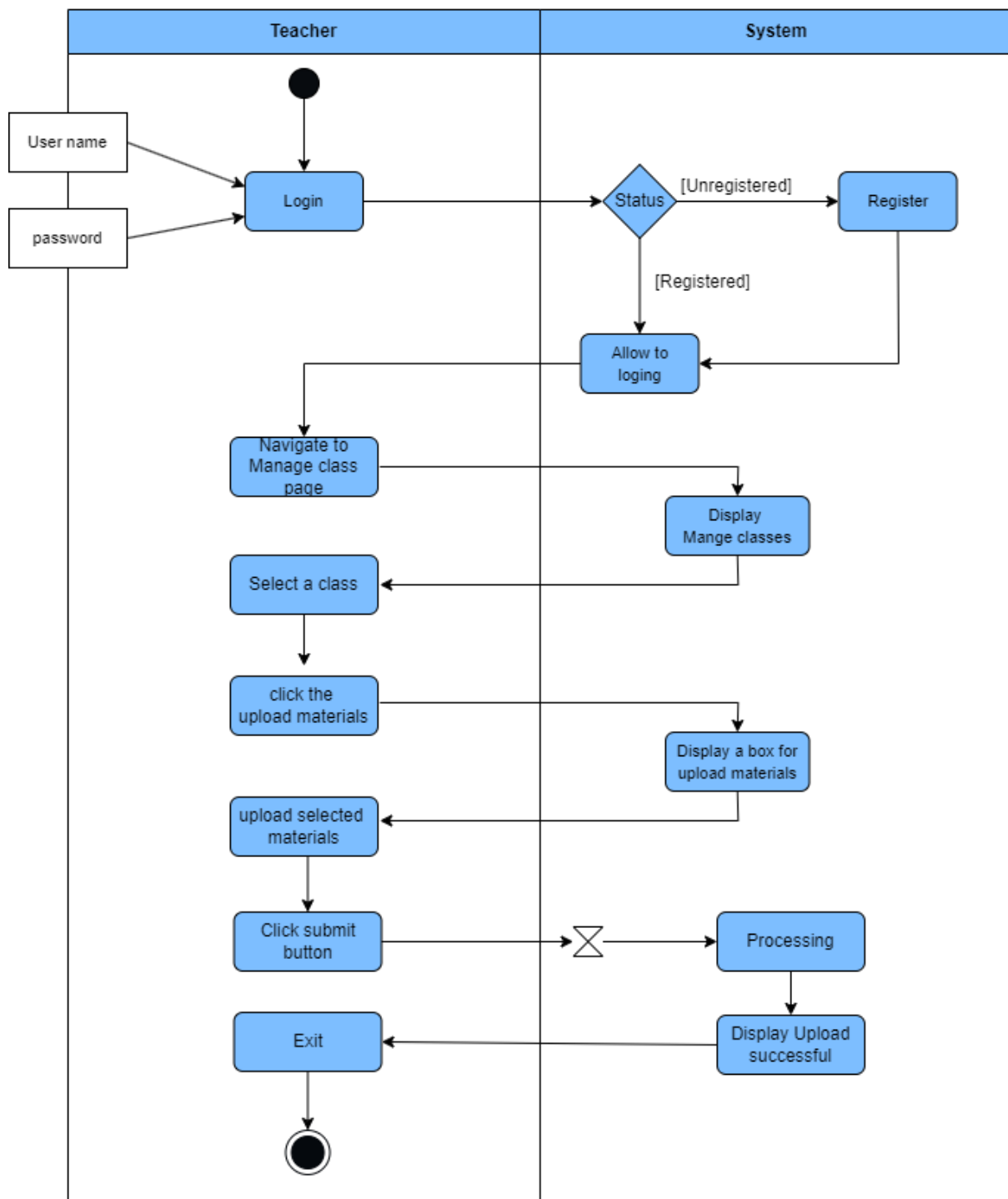
UC-008 - View students information



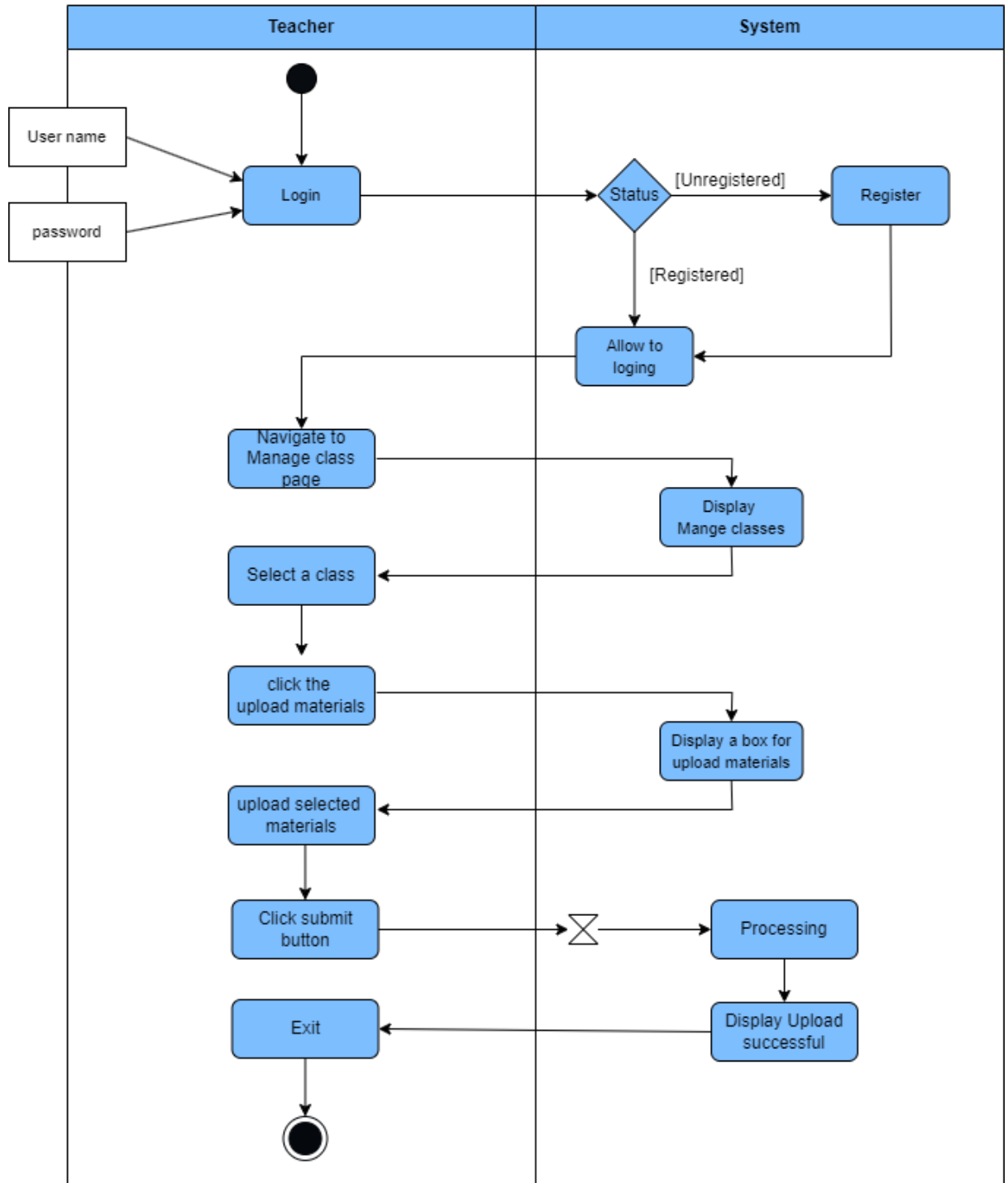
UC-009 - apply to institute



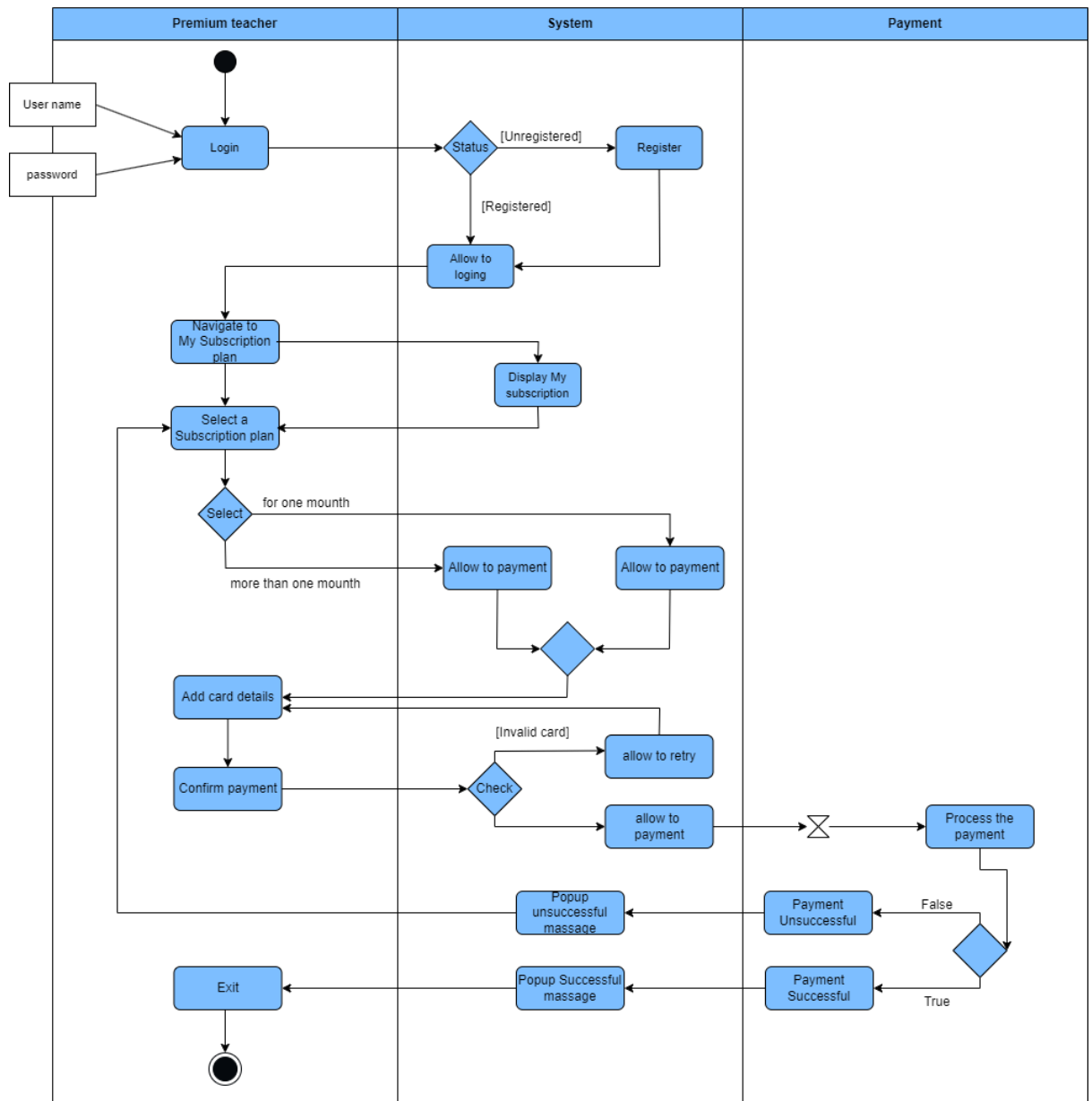
UC-010 - Upload learning materials



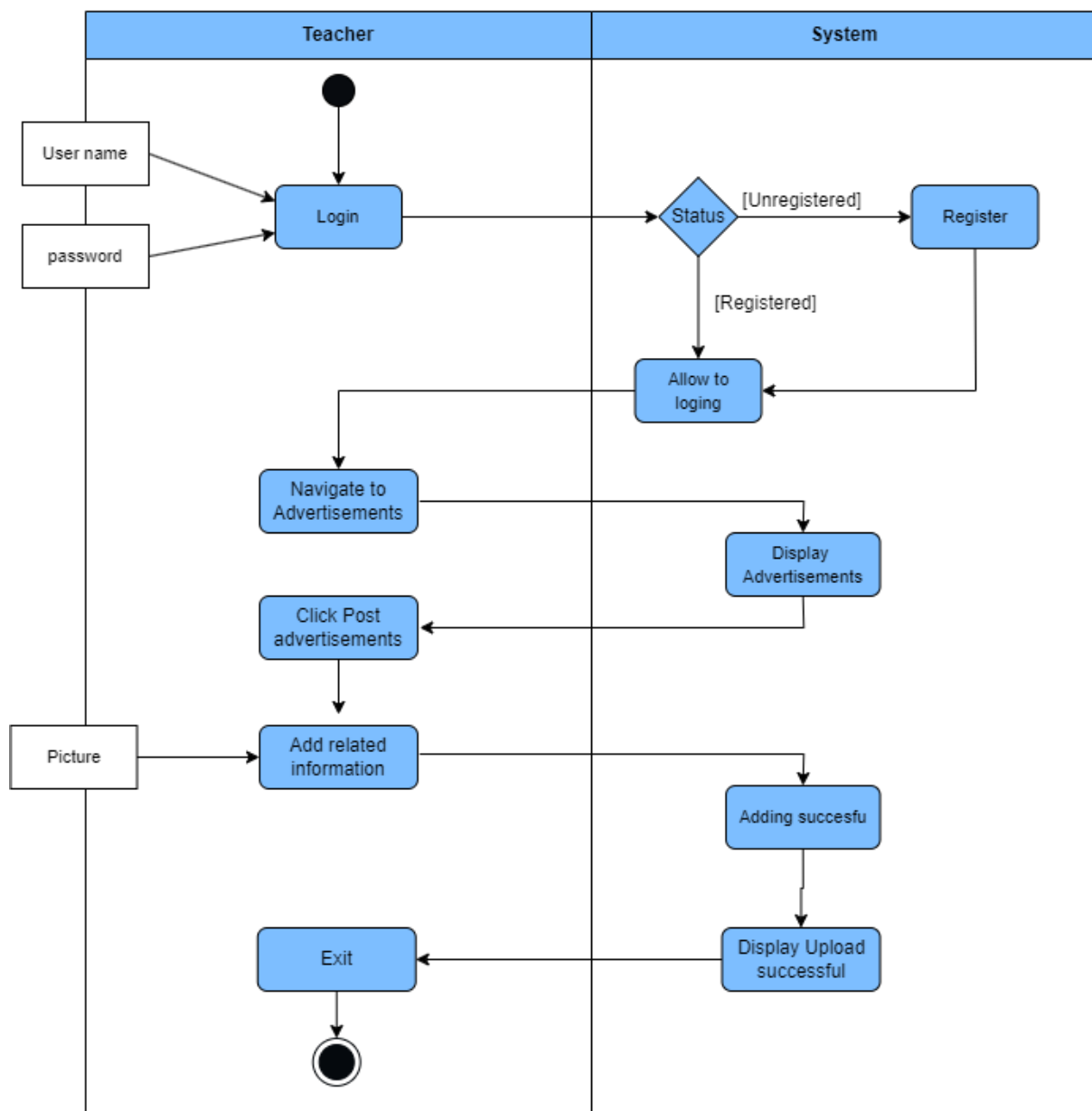
UC-011 - Schedule classes



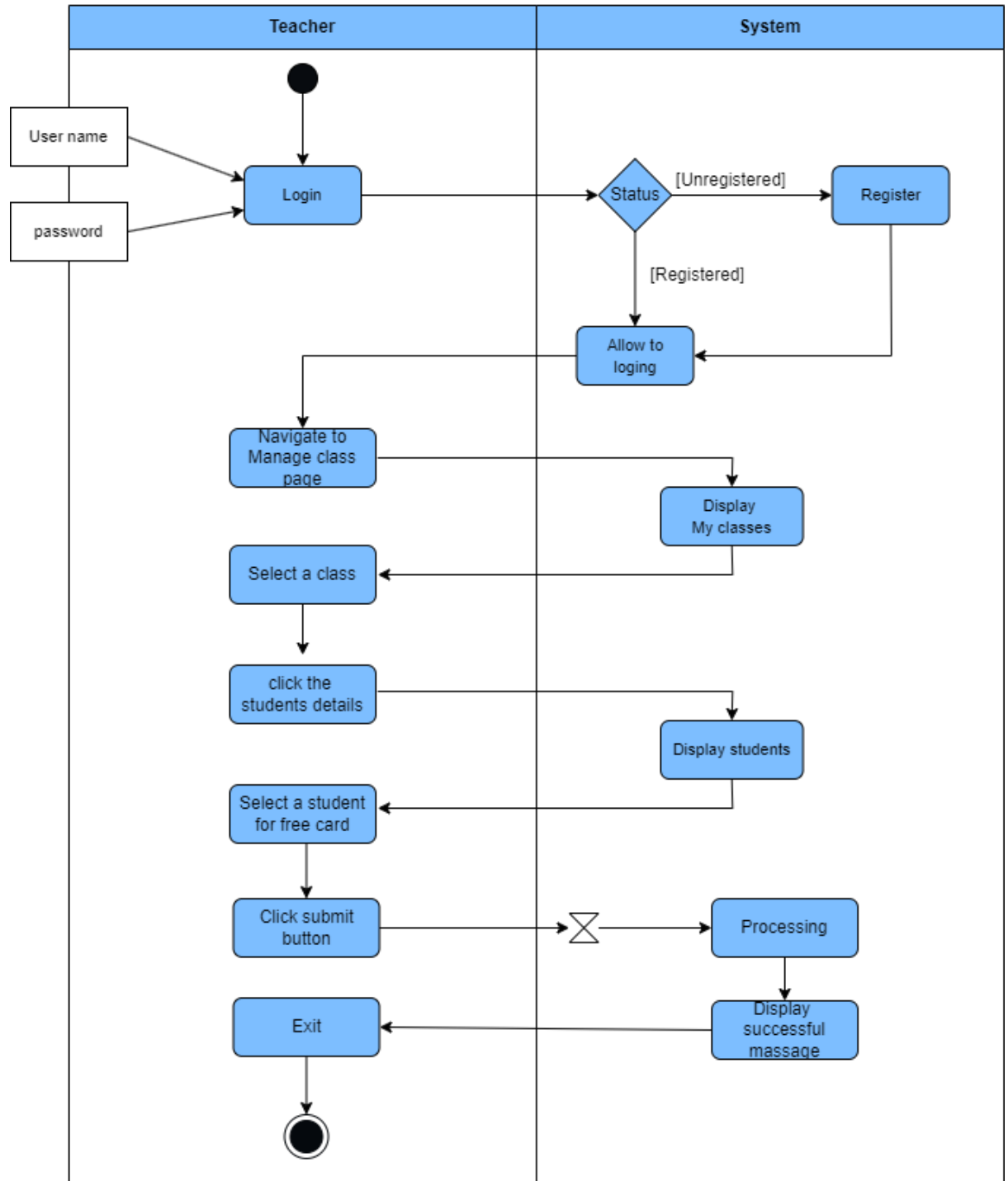
UC-012 - Update subscription



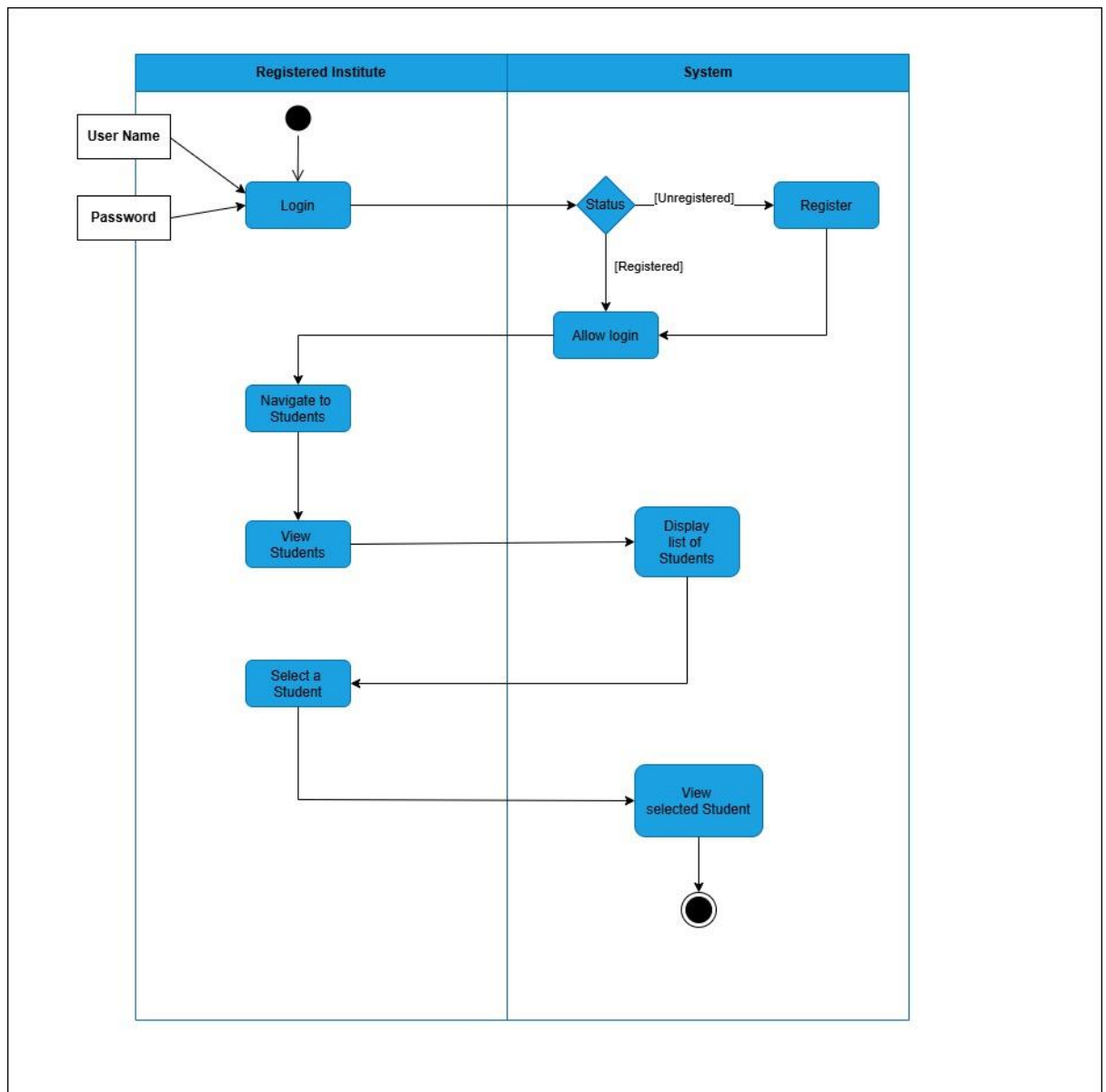
UC-013 - Post advertisement



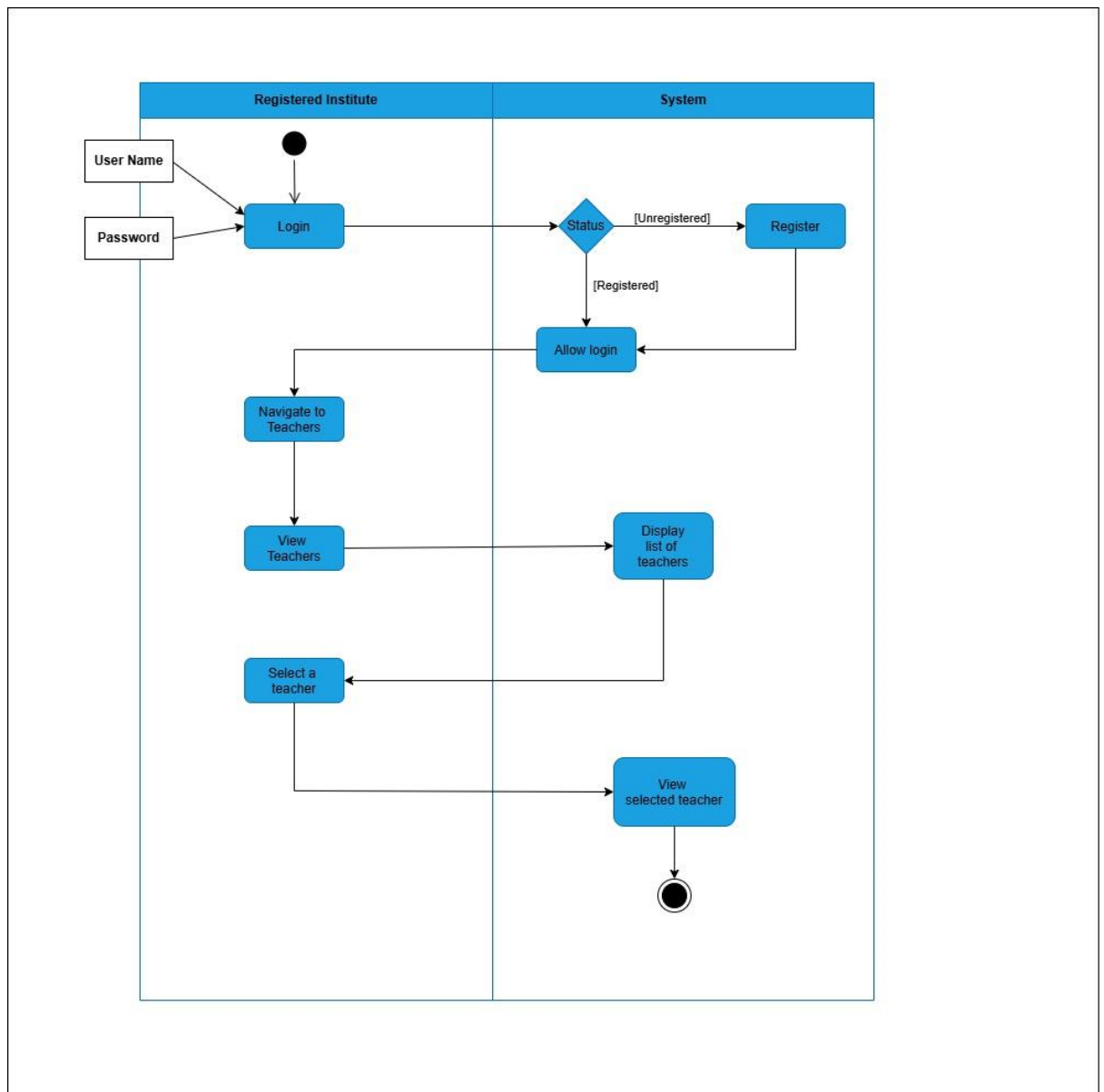
UC-014 - Offer discounts and free cards



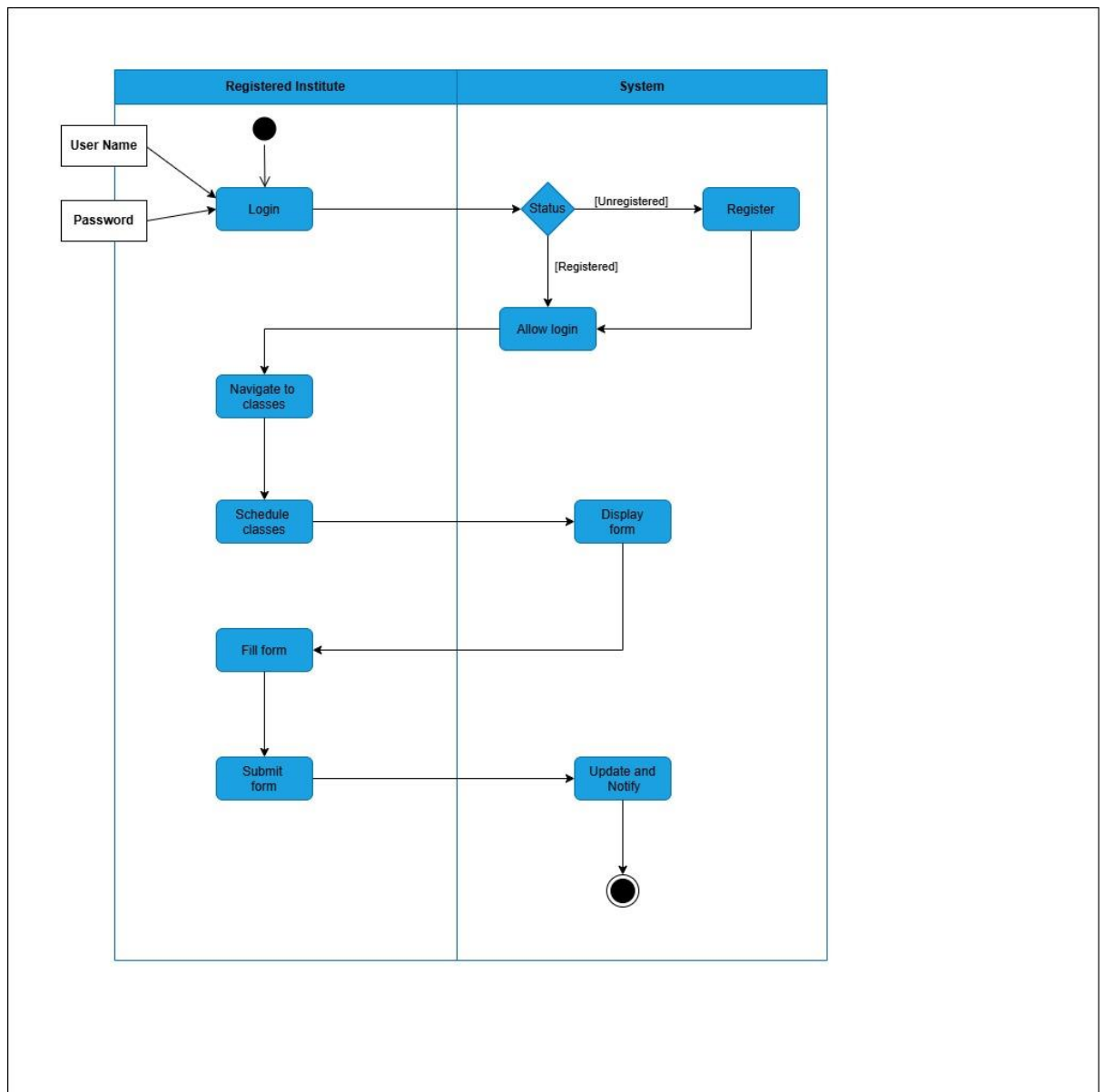
View Students



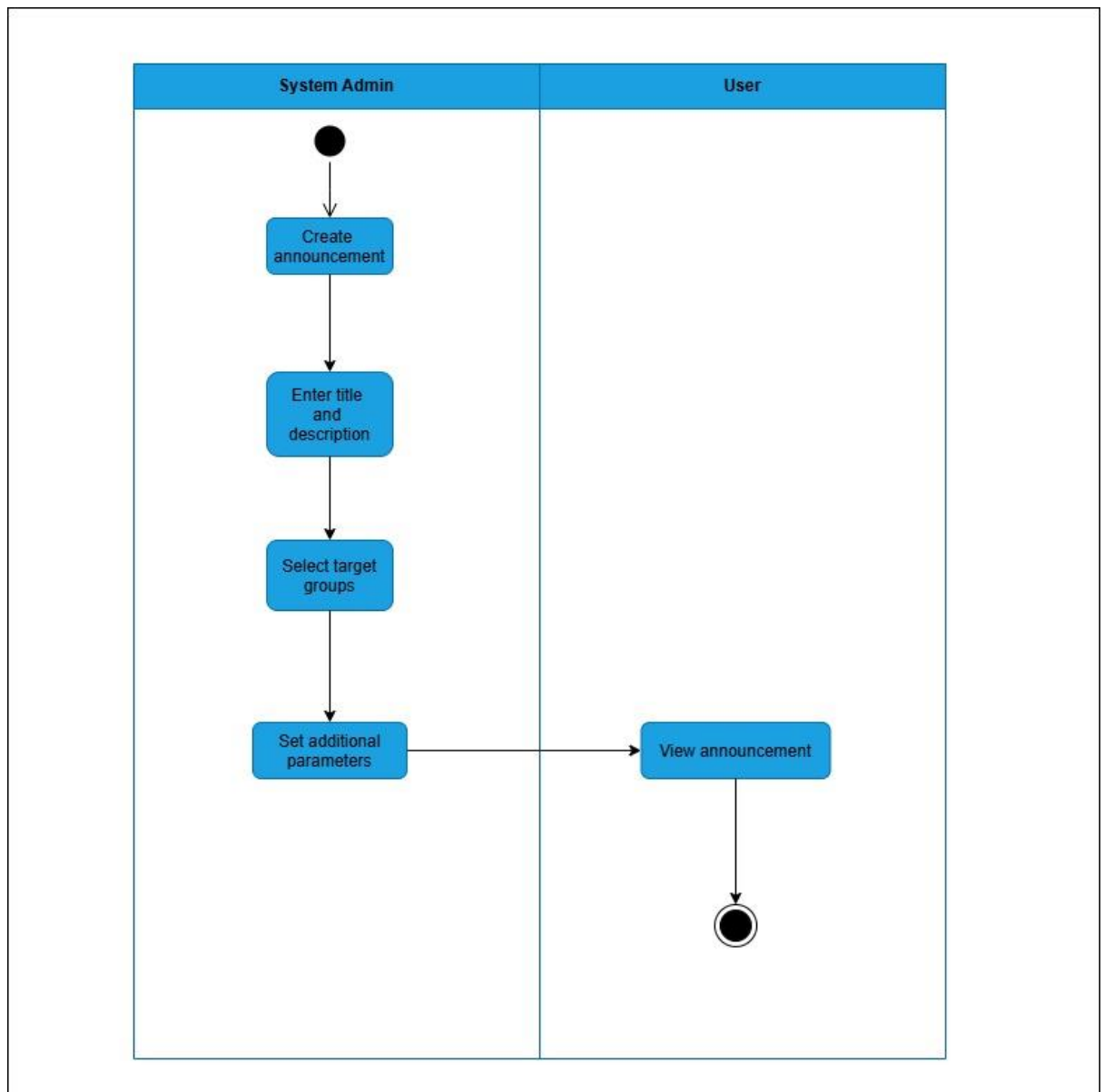
View teachers



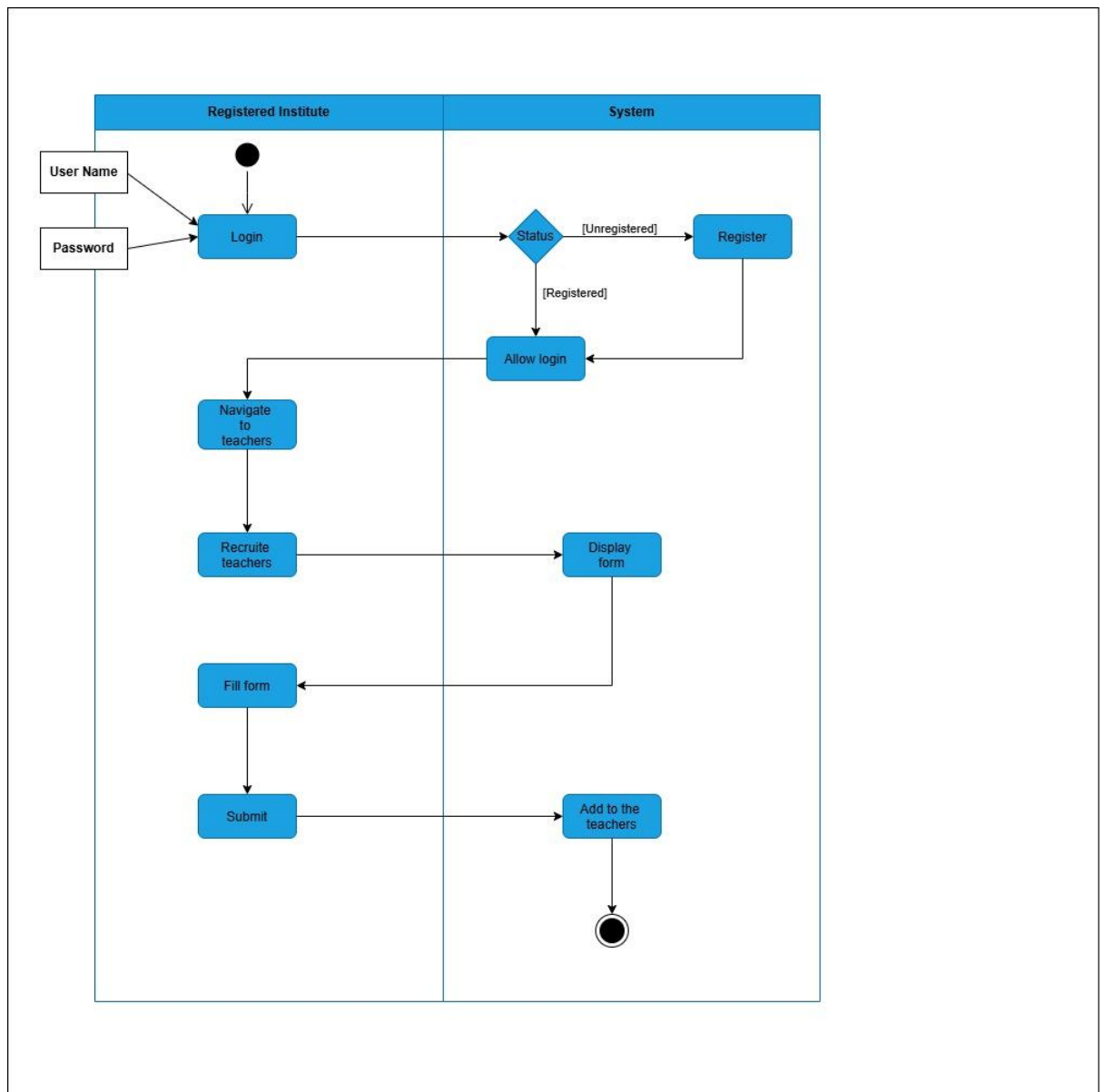
Schedule Classes



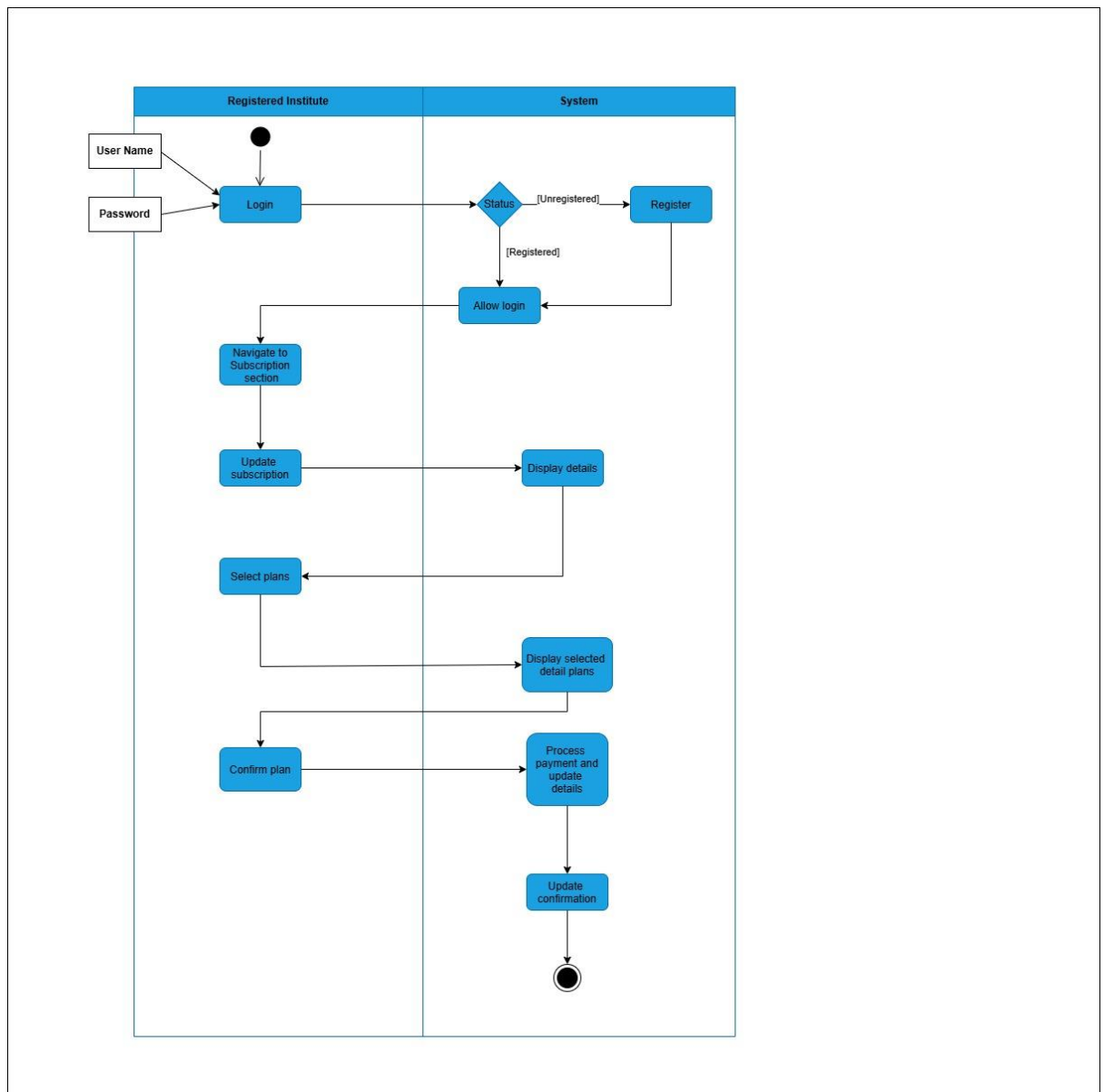
Create Announcements



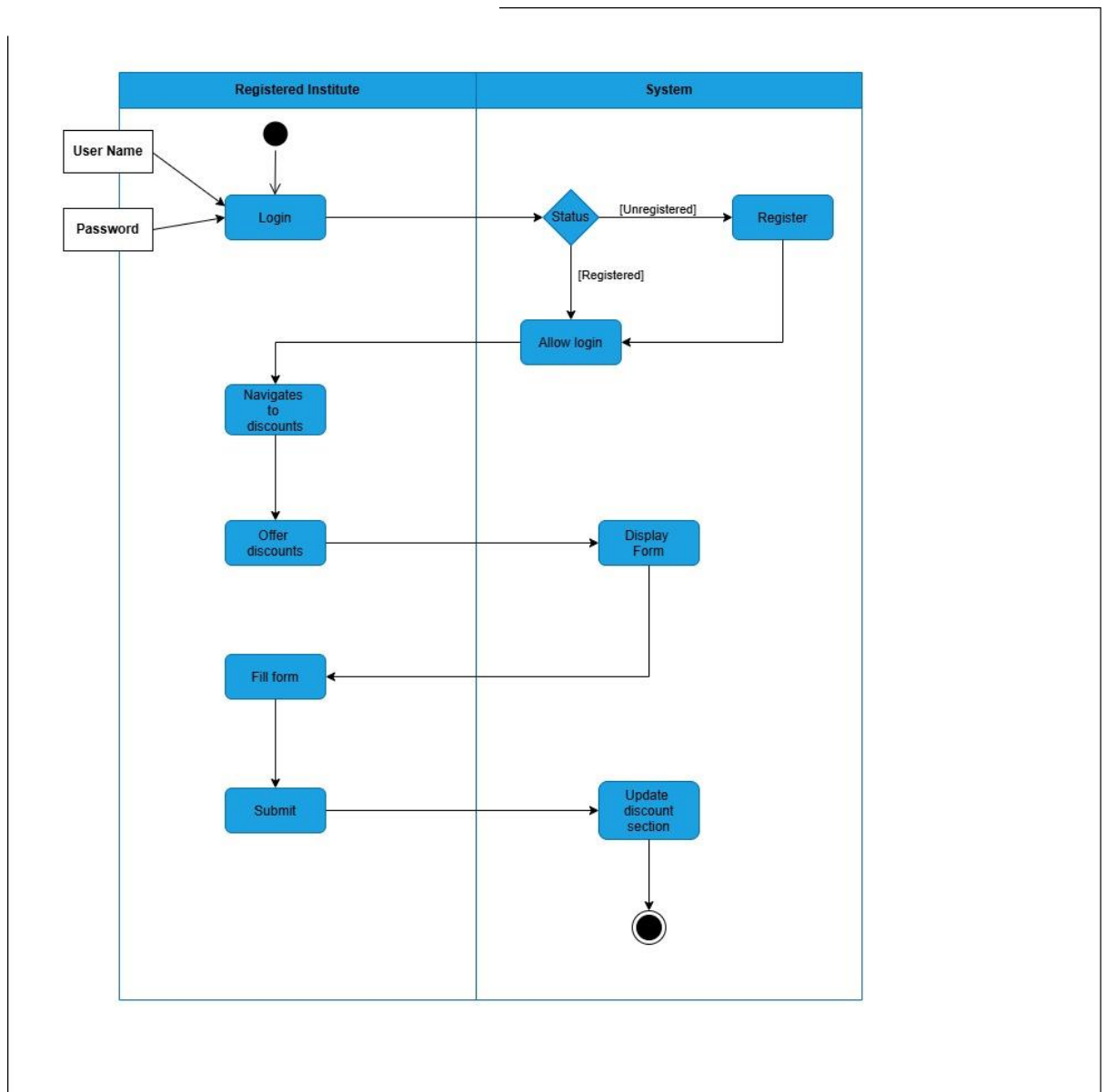
Recruite teachers



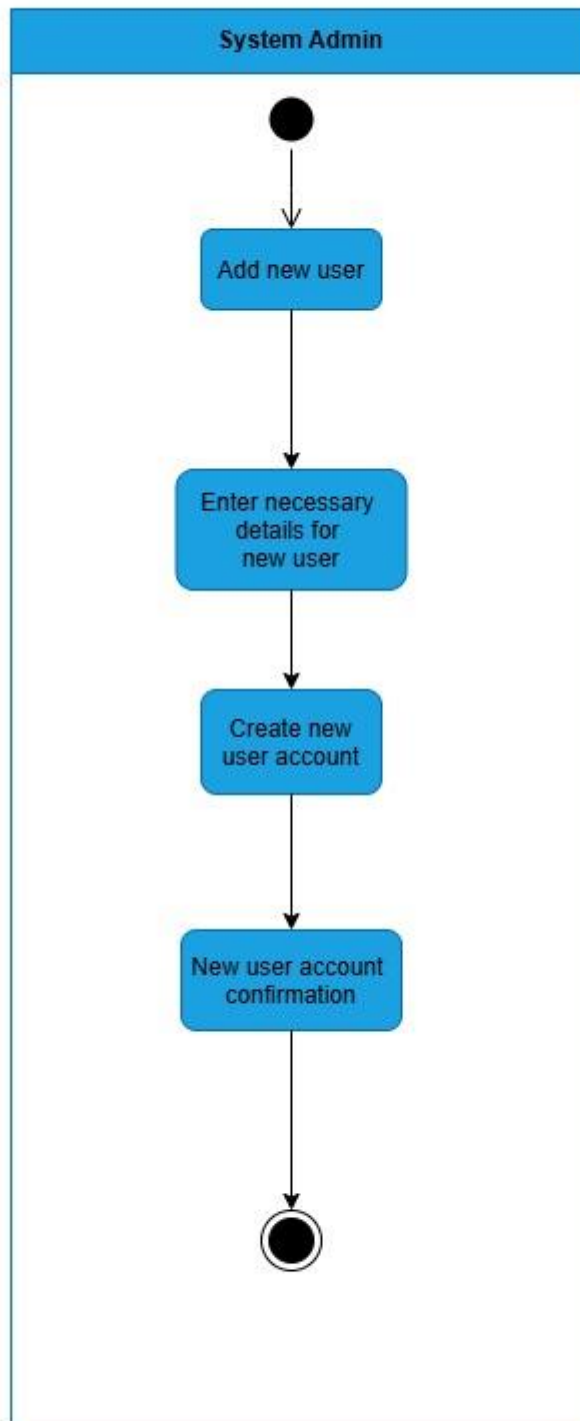
Buy subscription



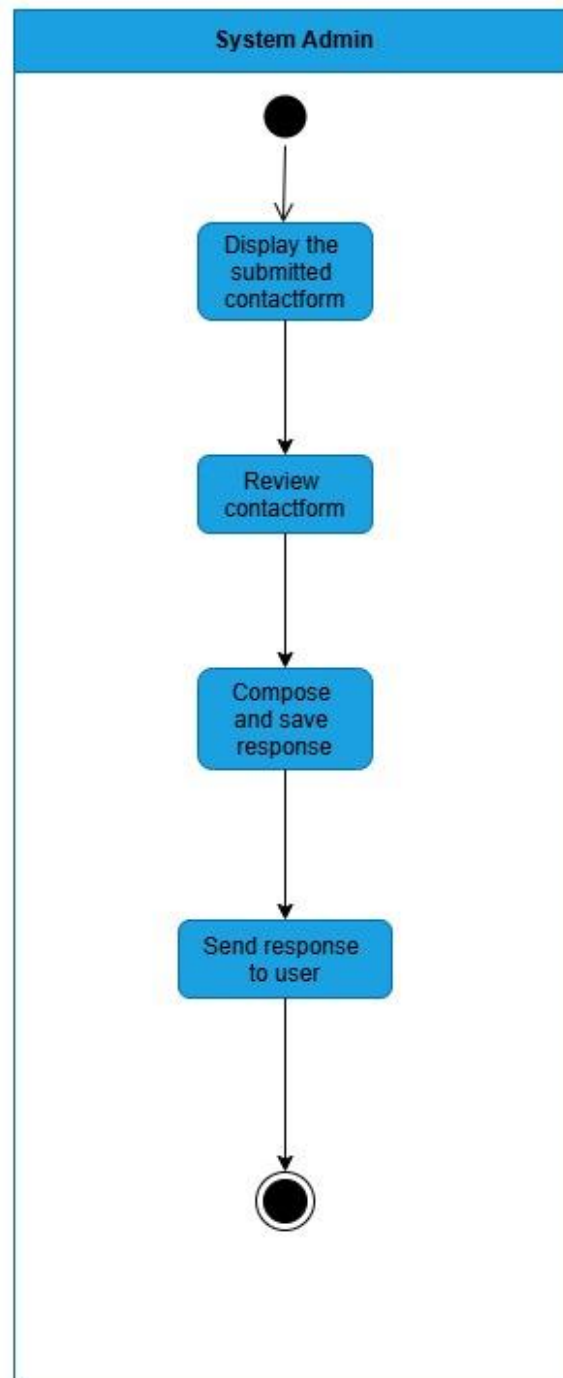
Issue discount for students



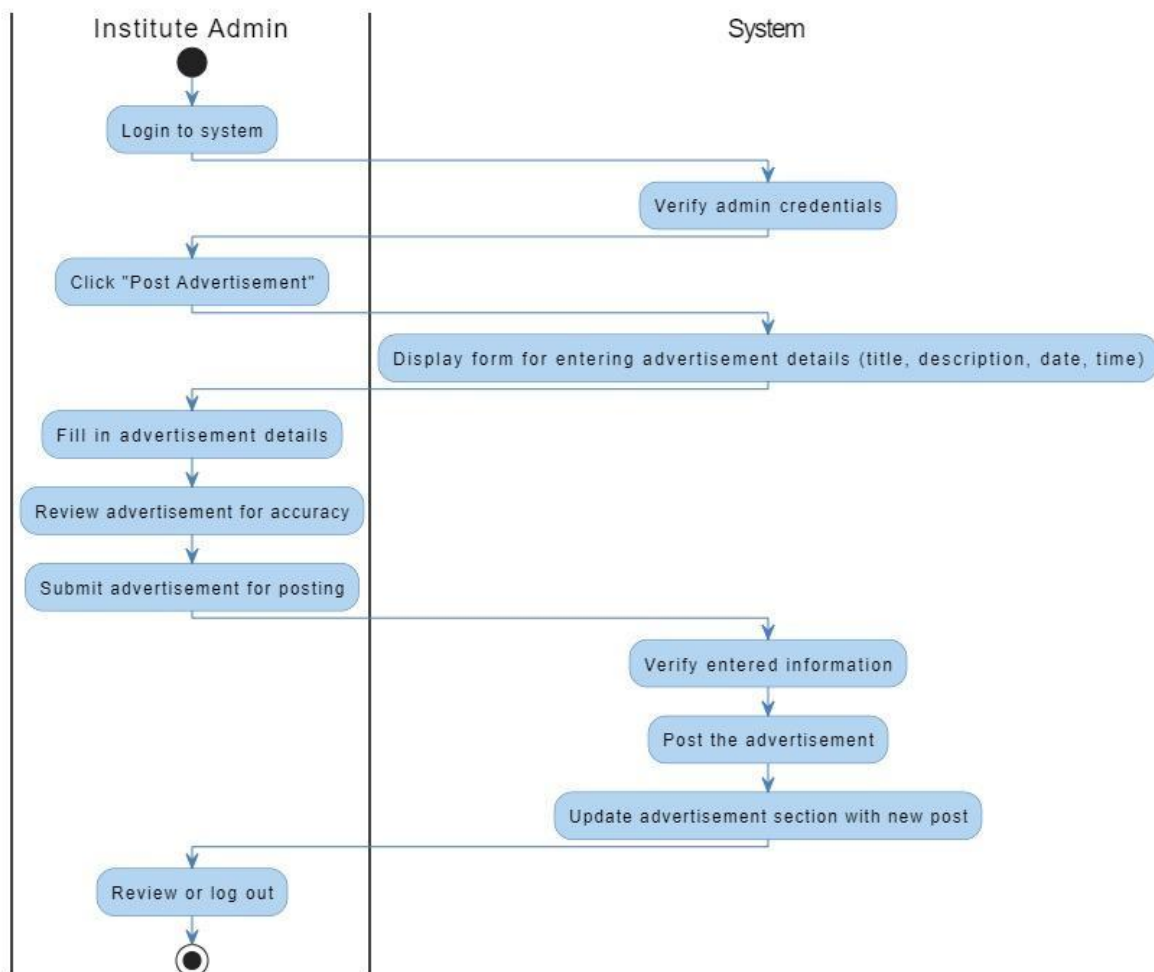
Add new user



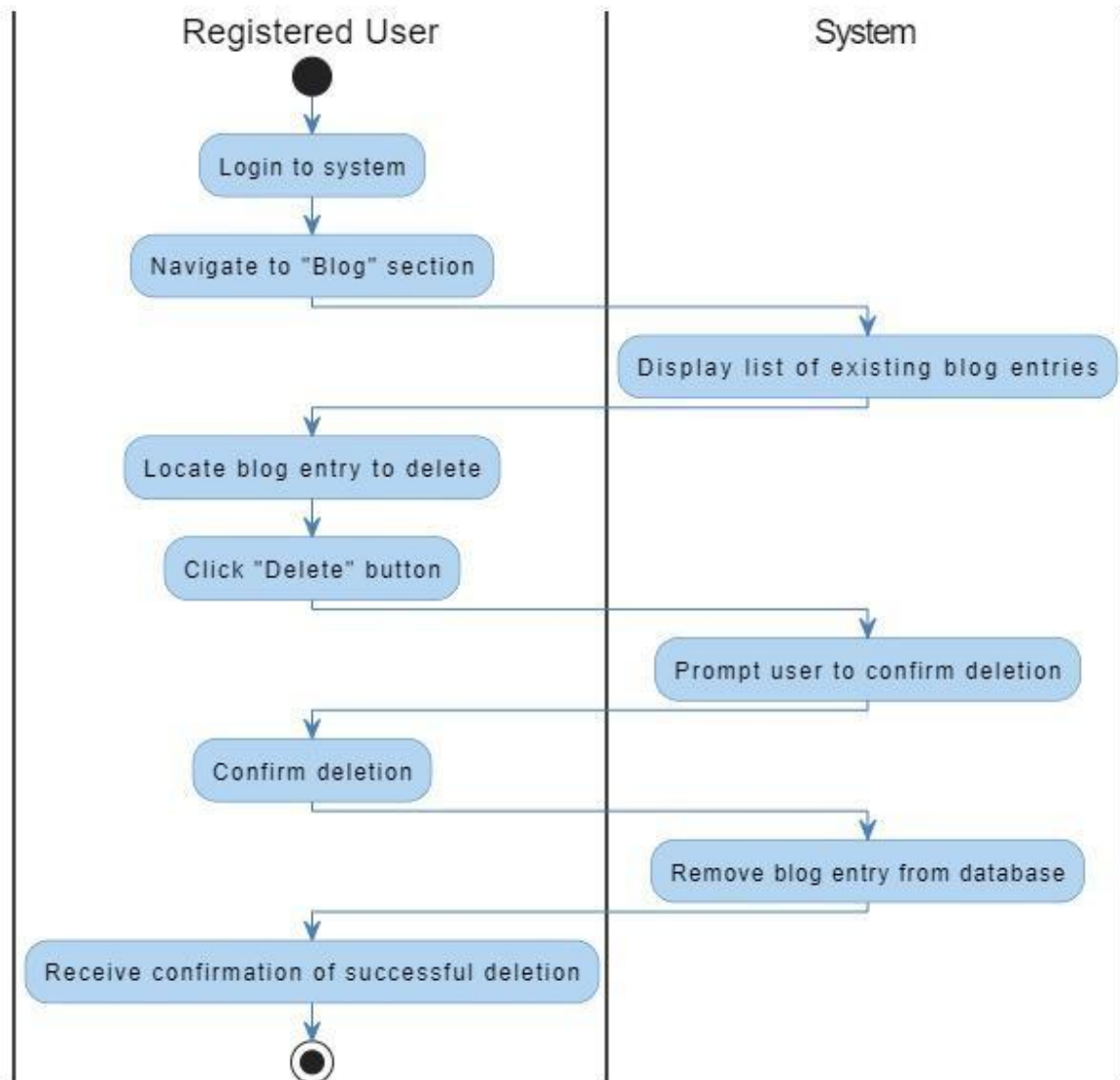
View submitted contact form by
admin



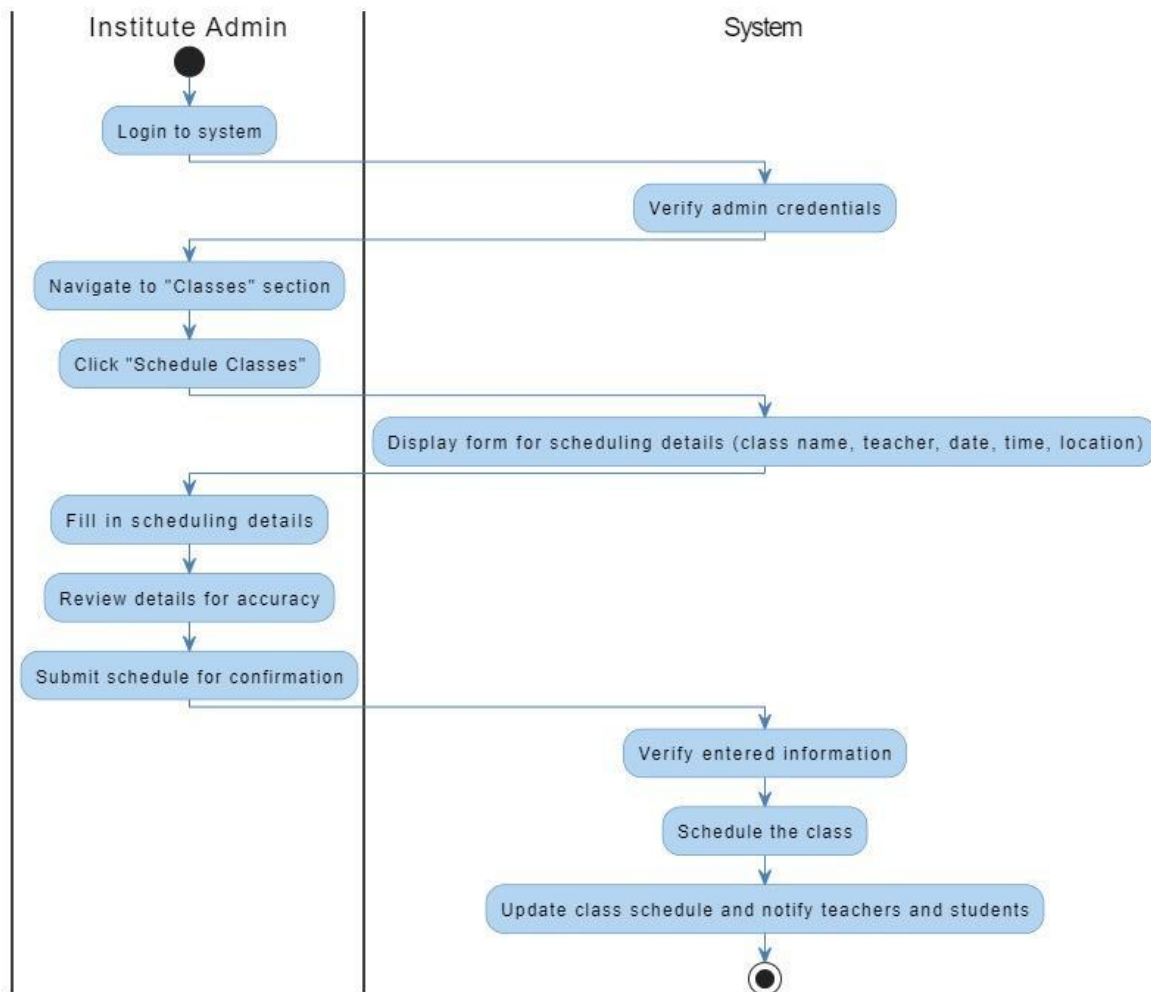
View Advertisements



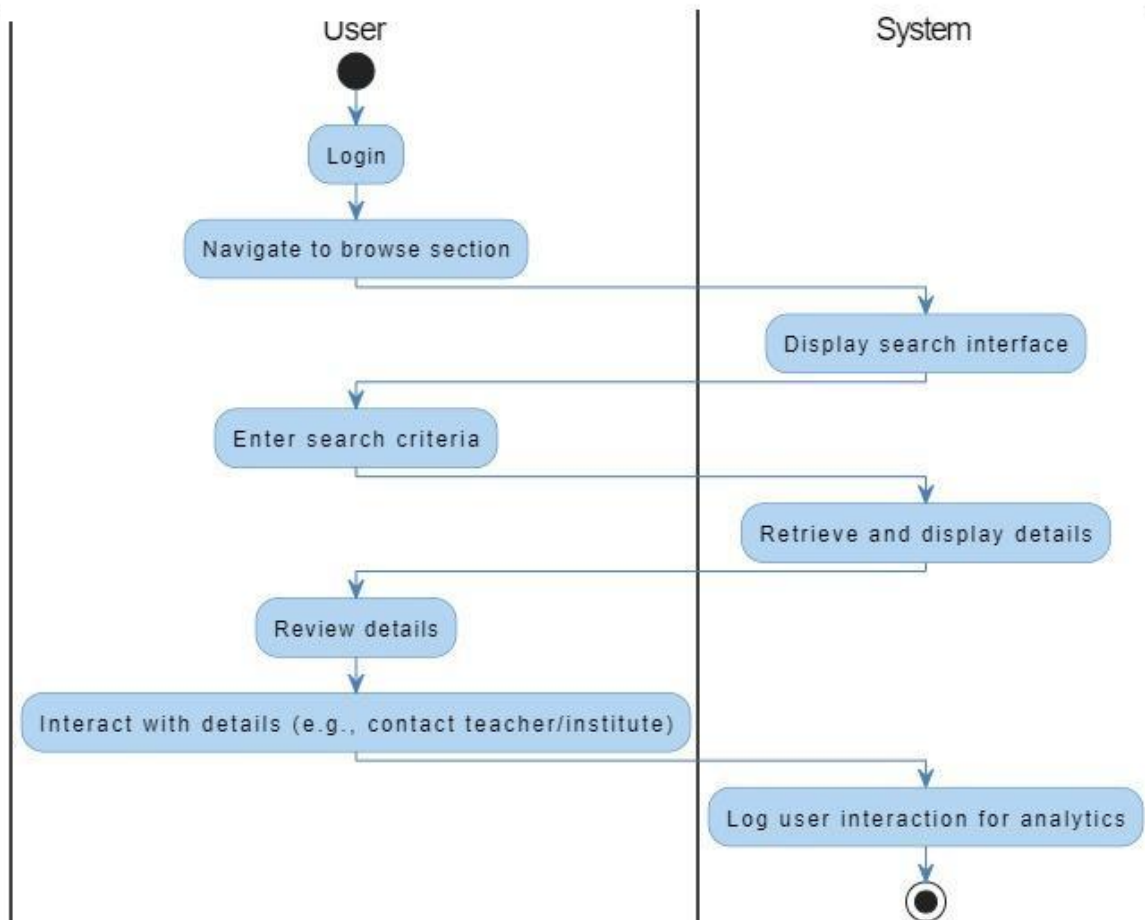
Browse Details



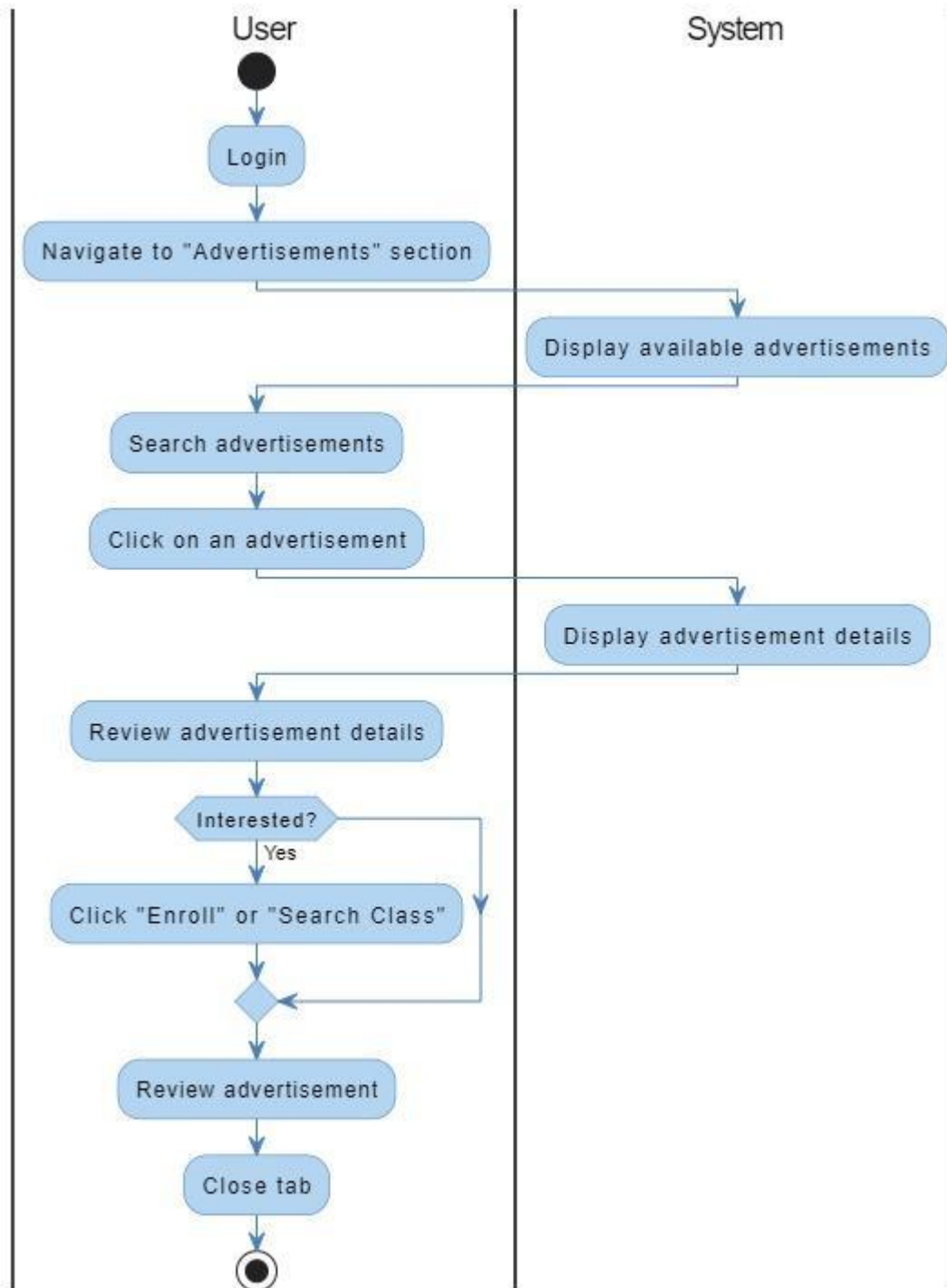
View Blog entries



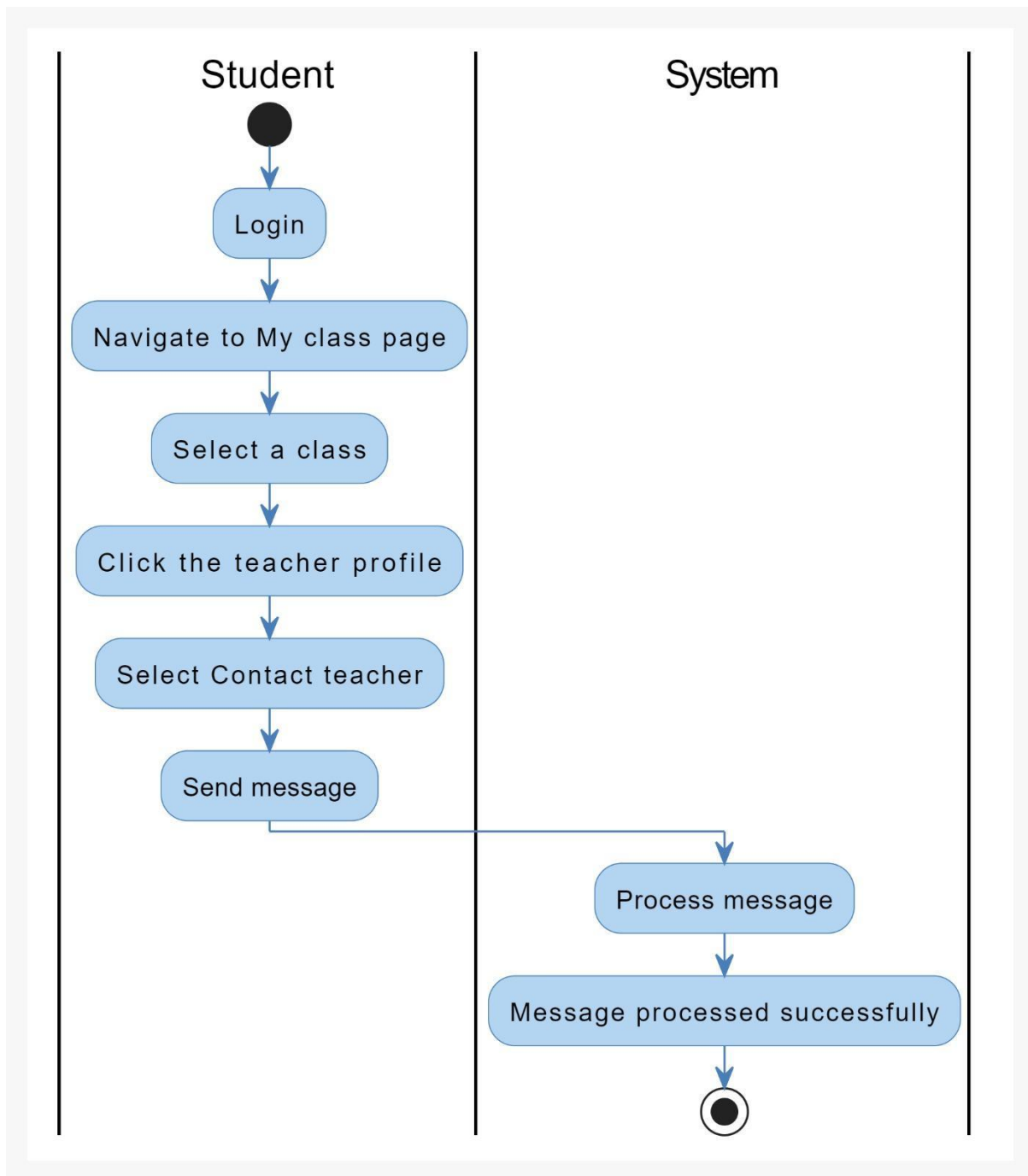
Contact support



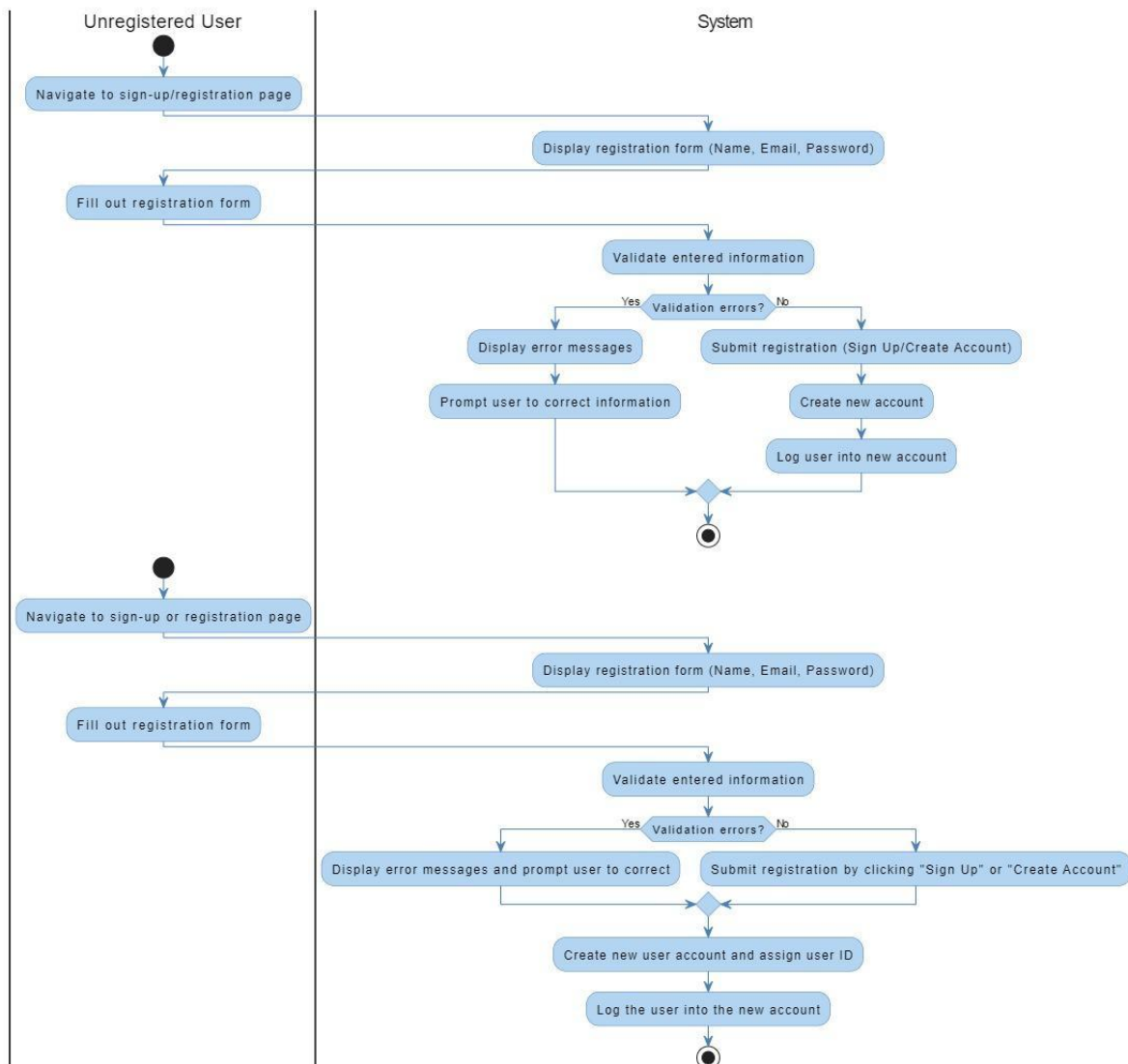
Search advertisements



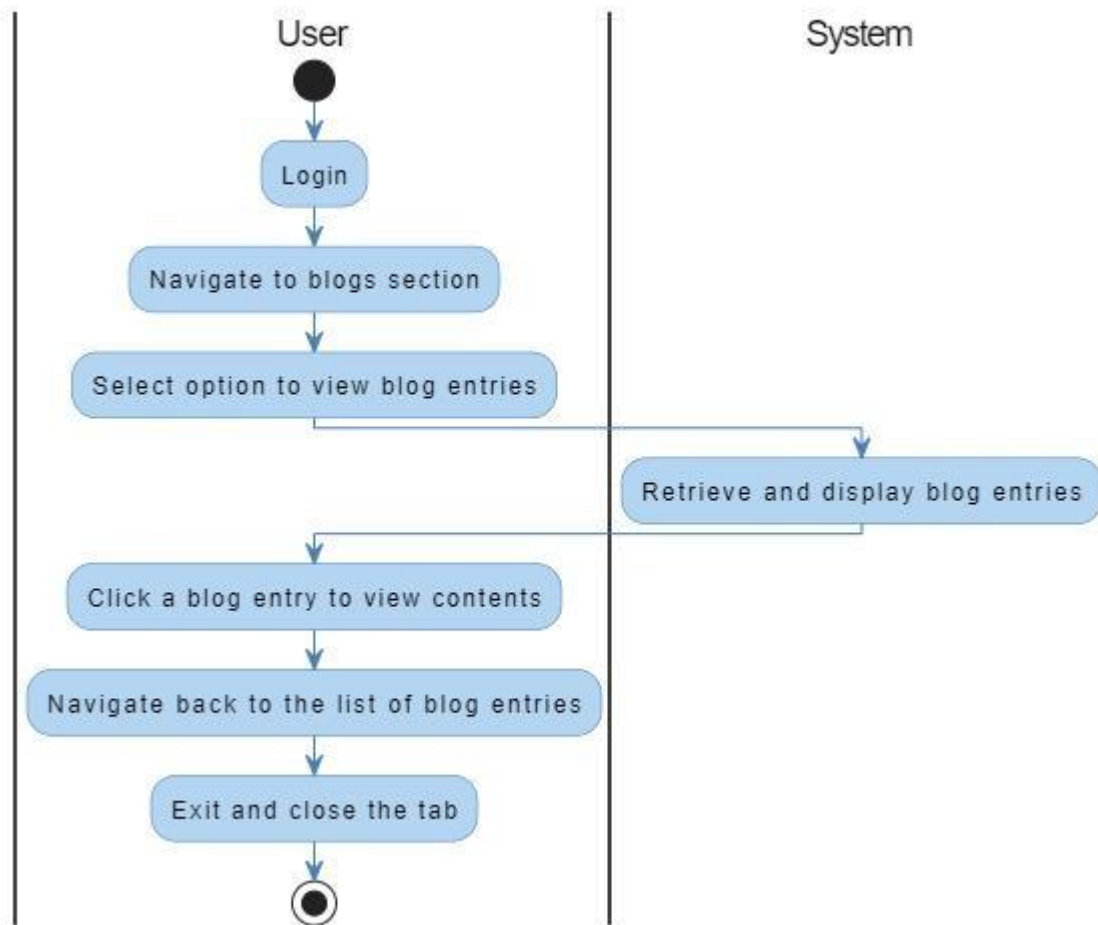
Contact teacher



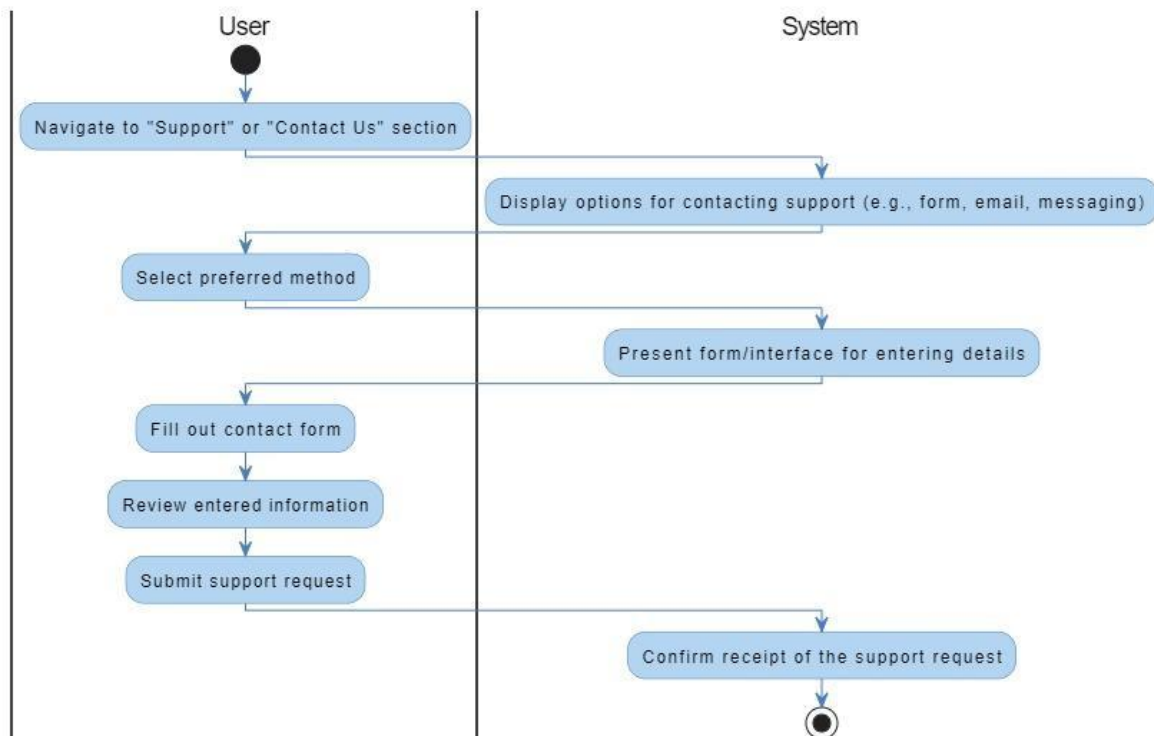
Sign up and login



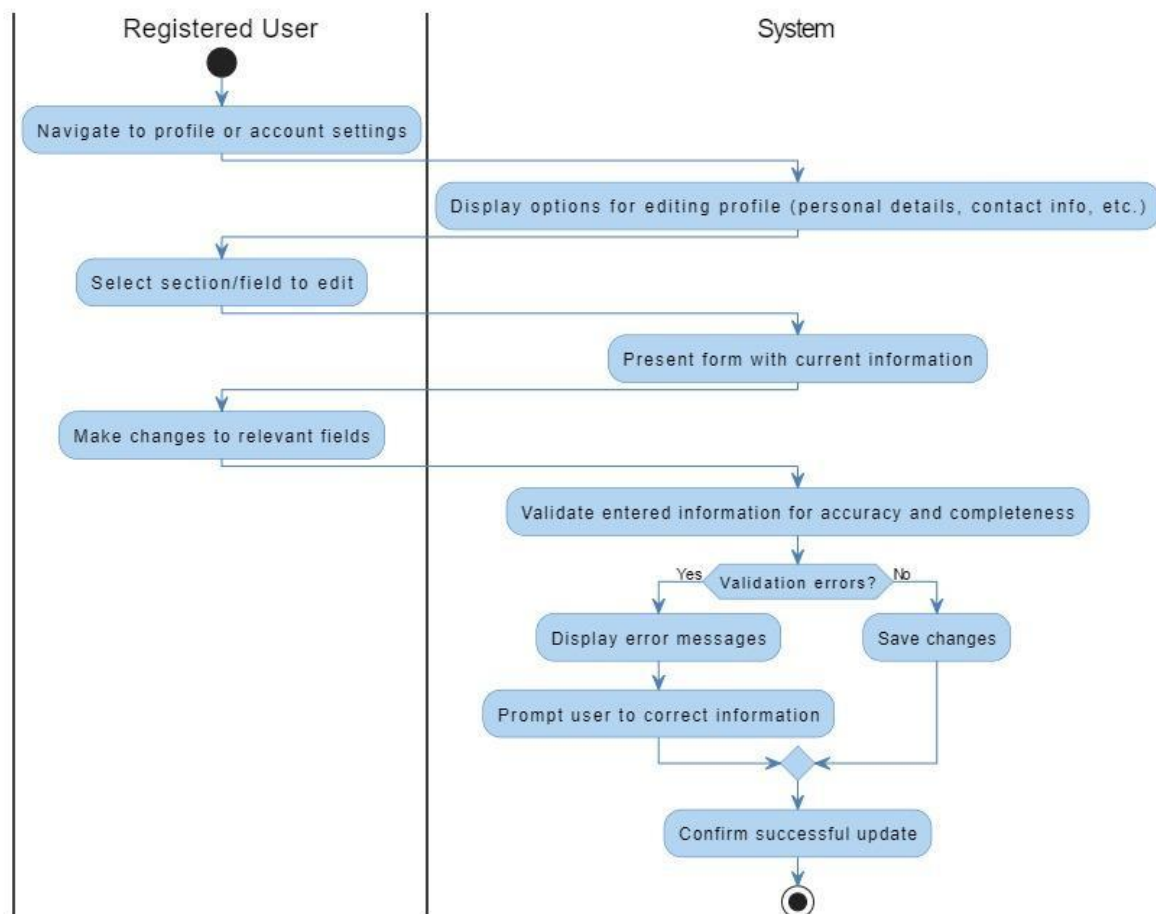
View blogs



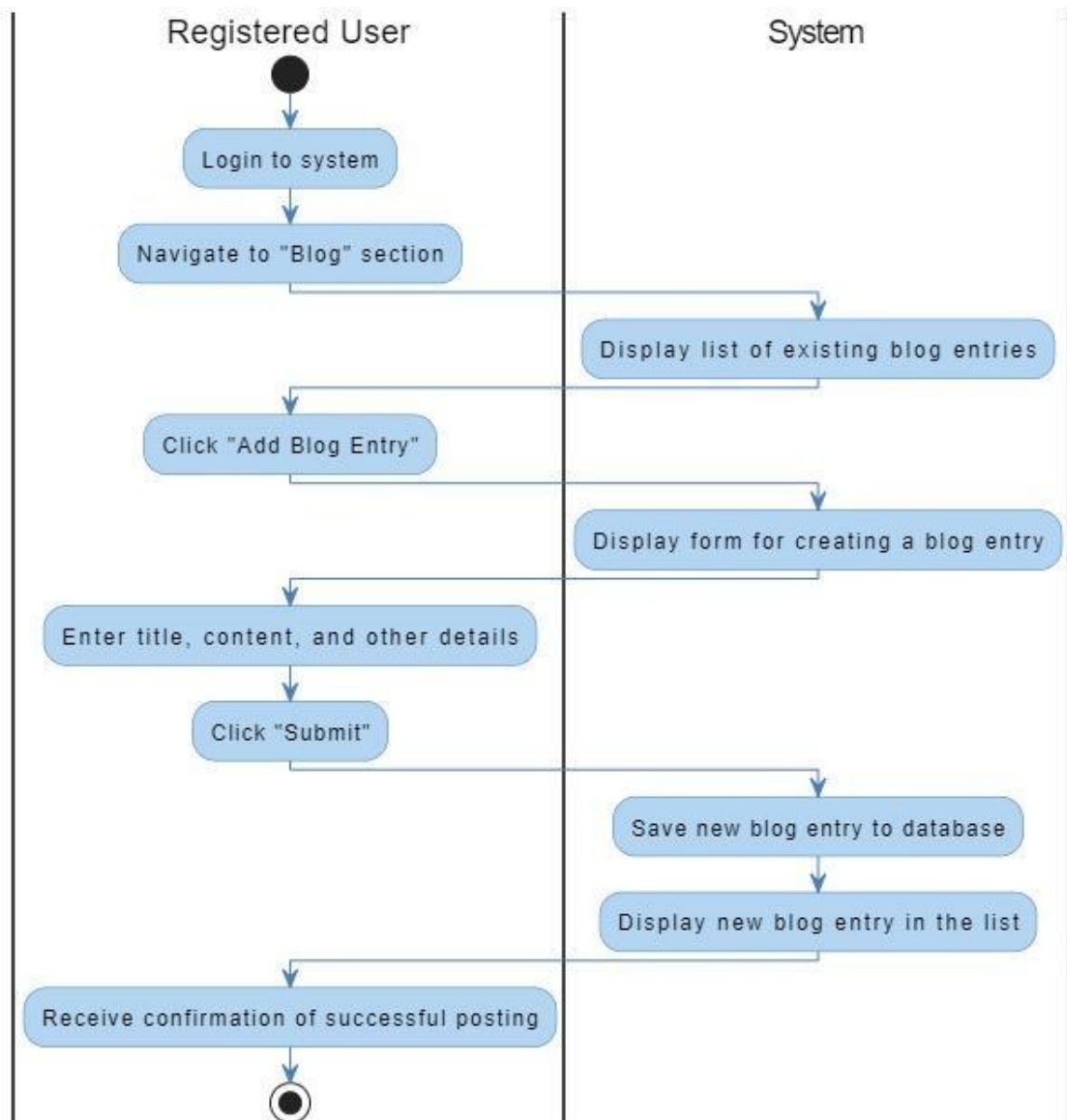
Contact information



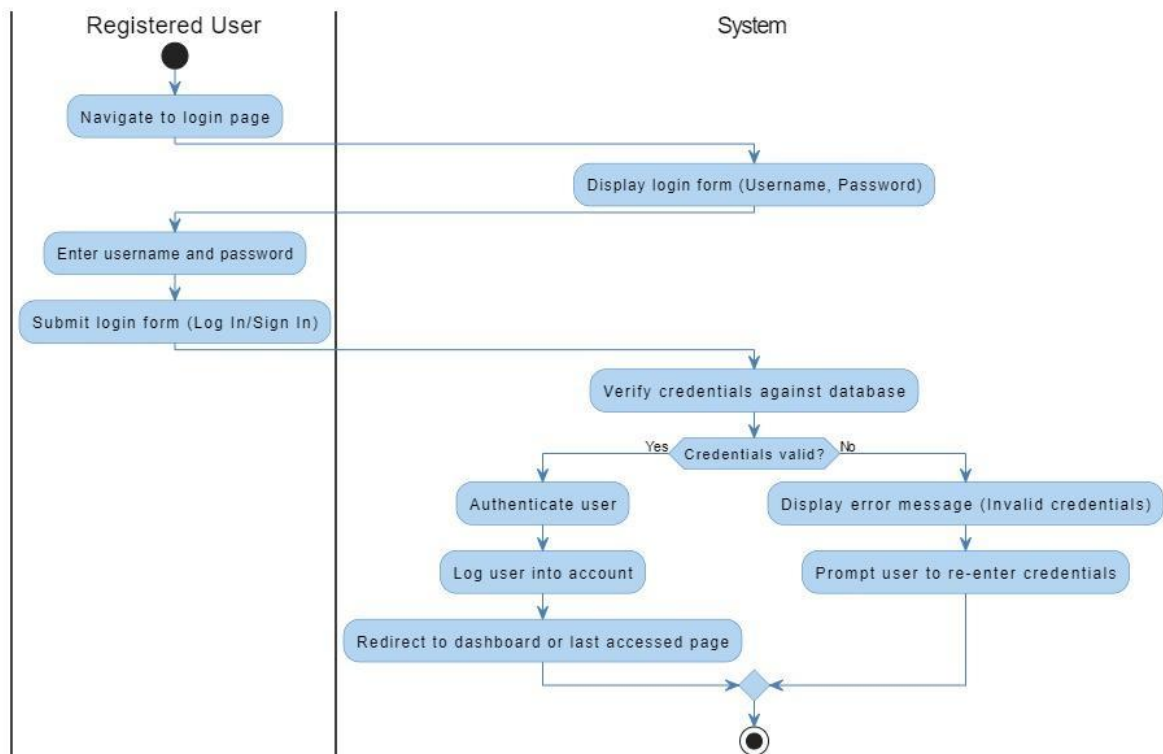
Edit profile



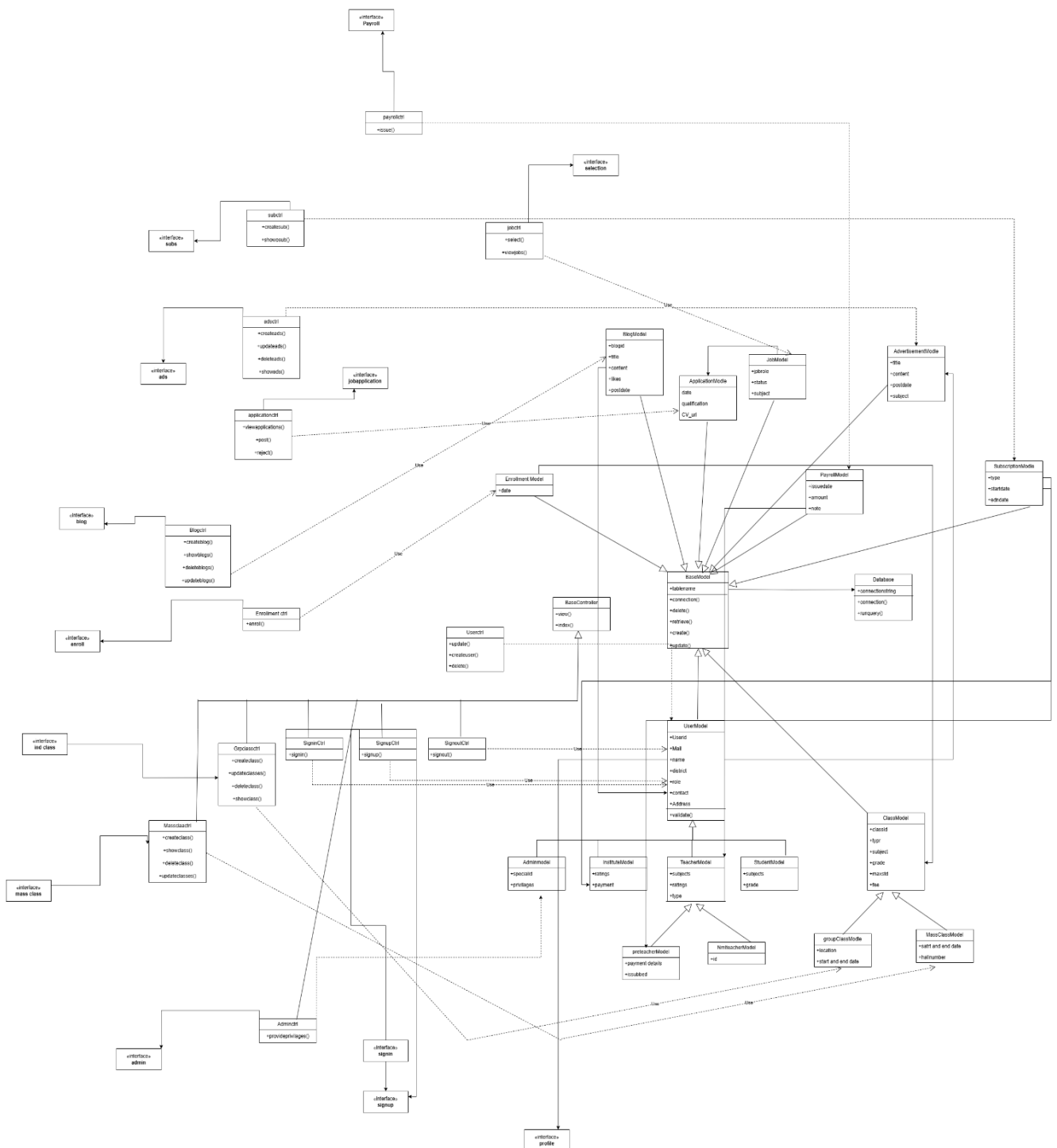
Add blog entries



USER LOGIN



Class Diagram



Technologies Used in the Project

1. **Front-End Development:**
 - **HTML:** Structure the content and elements of web pages.
 - **CSS:** Style and format the visual presentation of web pages.
 - **JavaScript (Vanilla JS):** Enhance interactivity and functionality on the clientside.
2. **Back-End Development:**
 - **PHP:** Server-side scripting language for processing logic and interacting with the MySQL database.
 - **MySQL:** Relational database management system for storing and managing structured data.
3. **Payment Gateway:** ○ **PayHere:** Secure platform for handling online payment transactions.
4. **Version Control:**
 - **Git:** Version control system for managing and tracking changes to your project's source code.
5. **Integrated Development Environment (IDE):**
 - **Visual Studio Code (VS Code):** Lightweight and powerful IDE for coding across various languages.
6. **Communication and Collaboration:**
 - **WhatsApp:** Instant messaging for quick team communication.
 - **Zoom:** Video conferencing for remote meetings and collaboration sessions.
7. **Diagrams and Design:**
 - **Draw.io:** Web-based tool for creating diagrams such as flowcharts, process diagrams, and wireframes.
 - **Canva:** Design platform for creating graphics, posters, social media posts, and other visual content.
 - **Adobe Photoshop:** Image editing software for designing and editing graphics, logos, and other visual elements.
8. **Documentation:**
 - **Microsoft Office Package:**
 - **Microsoft Word:** For creating and formatting textual content, such as project documentation, user manuals, and reports.
 - **Microsoft Excel:** For data analysis, calculation, and organization of project-related information.
 - **Microsoft PowerPoint:** For creating presentations to communicate project progress, outcomes, and findings.

Progress So Far

Our team has made significant strides in developing a comprehensive web-based platform that caters to the needs of various user roles. At the core of the system, we have implemented a secure login and signup system, ensuring the integrity and confidentiality of user data.

On the user-facing side, we have created customized home pages for all user roles, providing a personalized experience tailored to their specific needs and permissions. This includes:

- System administrators can now manage the entire system through a robust CRUD (Create, Read, Update, Delete) interface, giving them full control over the platform.
- General users can access the integrated blog features, allowing them to stay informed and engaged with the latest content.
- A notification system has been implemented, keeping all users updated on relevant events and updates within the platform.

For student-centric functionalities, we have empowered students to access their enrolled classes, search for and enroll in new classes, and manage their academic schedules. On the instructor side, we have enabled teachers to schedule classes and manage their course materials, fostering a seamless learning environment.

Furthermore, we have introduced capabilities for institutes to schedule their own classes and coordinate with students and faculty, promoting a collaborative and organized approach to education.

Looking ahead, our development team is currently focused on integrating additional features to enhance the platform's capabilities. These include:

- Implementing an advertisement CRUD system to enable the management of promotional content
- Designing a subscription model to offer premium services and generate revenue
- Integrating a messaging system that allows users to communicate via SMTP protocol
- Incorporating a payment gateway to facilitate secure financial transactions
- Creating a job platform to connect employers with potential candidates

The user interface (UI) designs for these upcoming features have already been completed, and our team is now dedicated to the backend implementation and integration of these critical functionalities.

We are confident that the successful completion of these remaining tasks will solidify our web platform as a comprehensive and versatile solution, catering to the diverse needs of our users.

Next Step

Moving forward, the team has identified the next steps to advance the project:

The primary focus will be on integrating messaging protocols and payment gateways. The team will research and implement the necessary solutions to enable user-to-user communication and secure financial transactions within the platform.

Concurrently, the team members will take on specialized roles to drive progress in specific areas of the system:

- Saranga will work on enhancing the student-facing CRUD functionalities, ensuring students have a seamless experience managing their class enrollments, schedules, and related data.
- Randila will focus on developing the teacher-specific CRUD capabilities, enabling instructors to effectively manage their course materials, class schedules, and student interactions.
- Dulshan will be responsible for implementing the institute-level CRUD features, allowing administrators to oversee their course offerings, faculty, and institutional operations within the platform.
- Raheem will handle the system-wide CRUD functionalities, including the management of user accounts, general content, and other cross-cutting administrative tasks.

The team aims to complete these tasks and deliver the project in its entirety by the end of February. Regular progress reviews and coordination will be crucial to ensure the successful integration of these new features and the overall project's timely completion.