1. Introduction

Domain description

Eduburd is an online tutoring platform designed to improve the educational experience for students while also simplifying tasks for tutors to teach and parents to view child progress. The platform addresses the challenges faced by our client, who provides tutoring services to foreign students across various grade levels through local tutors.

Although our client currently serves over 100 students, many tasks—like booking classes, handling payments, coordinating between tutors-students and tutor-parents, and managing administrative duties—are still done manually through WhatsApp and Telegram. This traditional approach consumes a lot of time and results in less efficiency.

Our solution is designed to make these tasks easier and more efficient by automating them. Instead of relying on manual processes, we'll help simplify and speed things up, making everything run more smoothly for both the admin and the users. This will save time, reduce mistakes, and ensure things get done more reliably.

Current system and limitations

Currently, the client relies on WhatsApp and Telegram for all the tasks.

- Manual transactions: All transactions, class bookings, class scheduling, parent-tutor interactions, and tutor-student interactions are conducted manually via WhatsApp and Telegram.The client manually manages all interactions.

- Manual payment process: Students pay class fees to the admin, who then pays the tutors' salaries. There is no system to streamline these payments.

- Lack of class tracking system: The client has no system to track the classes conducted by each tutor every month. Tutors report their classes at the end of the month, which is both time-consuming and prone to errors.

- Loyalty and trust based tutors’ salary payments: The current salary payment system is based on loyalty and trust between tutors and the client, without any tracking or validation of the classes conducted.

- Large user base: With over 100 students on the platform, the lack of an automated system leads to the operational inefficiencies.

- Manual tutor selection: Tutors are selected based on their resumes, qualifications, experience, and performance in a trial class. This process is currently done manually.

- Lack of child progress tracking: no dedicated feature for parents to view updates on their child's performance, class attendance, or feedback from tutors.

Goal & objectives

Goal: To develop an automated online tutoring platform that enhances efficiency, improves user experience, and streamlines administrative tasks for tutors, students, and parents.Automate class bookings, payments, and communications to eliminate manual processes.Provide parents with an easy way to track their child’s progress.Create a reliable class tracking system for admins and tutors.Improve tutor selection with a standardized process.Enhance the overall operational efficiency of the platform.

Objectives:

● Develop a user-friendly interface that makes it easy for students, tutors, and parents to navigate and use the platform effectively.

● The system shall be scalable to handle increasing numbers of users and sessions.

● Implement an efficient scheduling system for tutoring sessions

● Facilitate parental involvement in their children's education by providing updates on academic progress.

● Ensure that the platform is accessible through all devices.

● Ensure that all user data is protected and that the platform complies with relevant data privacy regulations.

● Create a secure online environment where tutors-students and tutors-parents can interact without concerns about data breaches or misuse.

Assumptions

- All users will have access to the internet and compatible devices that support the platform.
- All users will have a basic knowledge of navigating online platforms for booking classes and managing payments.
- Users will provide accurate and up-to-date information for class bookings and payment processing.
- All users will have access to an email address.

2. Feasibility Study

Technical feasibility

Technologies used so far:

- HTML, CSS and Javascript used as the front end developing technologies.
- PHP is used as the back end developing technology.
- MySQL used as the DBMS technology

Applications and the tools used:

- Visual Studio Code - an IDE to edit and maintain the source code.
- GitHub - used as the storage and collaborative platform to collect all the source codes.
- Figma - used to design the UI/UX.
- Draw.io - used to draw all the diagrams needed (Activity diagrams, use case, class diagram, ER diagram)
- Google Drive - a google folder which maintains our documentations and other resources.
- JIRA - project management tool used to distribute the workload among the group members.
- Google DOCS, Canva - used to do the documentation requirements.
- Whatsapp and Google meet - used to do frequent communication and have meetings.

These technologies and applications are free, readily available and the team members are familiar. This platform is developed using the above mentioned technologies and tools making it evident that the proposed system is technically feasible.

Financial feasibility

- No additional costs as all technologies and tools used are free and open-source.
- We use our own computers and hardware.
- Meetings are conducted via free platforms like Google Meet or WhatsApp, with only minimal data or travel expenses.

Given no extra costs; our proposed system is financially feasible.

Operational feasibility

- The platform is built to support the client's mission of improving the efficiency of online tutoring for both students and tutors. It tackles the main issues by automating manual tasks like booking, payments, and communications.
- The platform is well-suited for foreign students working with local tutors, offering an interactive and easy-to-navigate learning environment.
- Students, parents, and tutors will quickly adopt the platform, as it makes tasks more manageable and fosters better communication and learning.
- By automating tasks like scheduling and payments, the platform reduces the admin's workload and streamlines everyday operations, saving time for everyone involved.

Schedule feasibility

As of now, the project is progressing well within the original 9-month timeline. We are using the modified Agile methodology, with ongoing requirement gathering as new features are added. The team has successfully completed UI/UX design, frontend development, and part of the backend development. The workload distribution ensures the project will meet its completion date.

Total Estimated Man Hours:

- Total months: 9
- Total weeks: 36
- Work hours per week : 10 [Weekdays - 5 hours | Weekends - 5 hours]
- Team members: 4
- Total man hours : 10*4*36 = 1440 hours

Legal and ethical feasibility

- All development code is written from scratch by the team, ensuring no copyright conflicts.
- All the user passwords that are created are hashed.
- Any user data that is stored in the databases will not have any access nor will be shared with any external parties to the system.
- Documents such as payment proof will be accessible only by the admin.

- Payments are securely processed by the admin, ensuring tutors receive payment at the end of each month.
- Tutor accounts will undergo thorough verification to prevent fake accounts.

Therefore it can be declared that our system is legally feasible.

3. Requirements

Stakeholders / Actors

The main actors of this system are the admin, tutor, parent and students. The users of the website will be tutors, students and parents.

1. **Admin:** Manages users, approves tutors based on their CVs, handles payments, monitors tasks and manages assignments.
2. **Tutors:** Deliver lessons, upload assignments, track student progress, and grade assignments.
3. **Students:** Book classes, attend sessions, complete assignments, and receive grades.
4. **Parents:** Monitor their child's progress, communicate with tutors, and track assignment grades.

Functional and Non-Requirements

Functional requirements

Tutor

- Profile management
- Class management
- Manage teaching and learning materials
- Upload assignments
- Create , check submissions and give grades
- View parent comments
- View marks and grades of students
- Accept/reject student requests
- Manage announcements
- Start and end class confirmation
- Upload, view announcements
- Access resources

Student

- Profile management
- Filter tutors
- View tutor profiles
- Request class sessions
- Submit assignments

- View marks and grades of assignments
- Access resources
- View due dates
- Start and end class confirmations
- Submit payment receipt
- Approve / Reject parent request
- Decline the given approval
- Chat with tutor
- View announcements

Parent

- Profile management
- Send request to add student
- Edit/delete request if status pending
- Delete a child from the list
- View student marks and grades
- View student progress graph
- Read FAQs
- Chat with tutor
- Access resources
- Add, edit ,delete and view resources
- View students' enrolled tutors and their profiles
- View progress graph of child
- View class history and upcoming classes
- View site announcements
- View assignment status and submission due dates

Admin

- Profile management
- User management
- Content management
- Class and schedule management
- Progress and performance monitoring
- Support and help desk
- Payment approvals
- Manage announcements
- Manage resources
- Manage feedbacks/ chats and comments
- Prepare MIS reports
- Manage FAQs

Non-functional requirements

1. Performance
    - The system is built upon the MVC architecture so that performance improvement of the system is ensured.
    - The system responds to user interactions within 2 seconds to ensure a smooth user experience.

2. Scalability
    - The system is scalable to handle increasing numbers of users and sessions.

3. Security
    - Session and cookie handling is implemented for user authentication.
    - User authentication and authorization mechanisms ensure that only authorized users can access specific functionalities.
    - Data encryption techniques are employed to secure sensitive information.

4. Compatibility
    - The system provides a user-friendly interface accessible on smartphones and tablets.

5. Availability
    - The system shall have an uptime of 99.9% to ensure it is available to users at all times.

6. Reliability
    - Databases will be updated frequently to ensure data integrity and availability.

7. Reusability
    - The system will be developed using reusable components to promote efficiency and reduce development time.
    - Rigorous error checking and testing will be conducted during the integration of components to ensure seamless functionality.

In-scope and out-scope

In-scope

1. User management

- Student, tutor, parent, and admin account registration and management.
- Login management.

2. Profiles creation

- Detailed profile creation including qualifications and experience for tutors.
- Separate profile creation for students by including their grade.
- Separate profile creation for parents.

3. Search and booking System

- Students have a search function with filters (subject, ratings, grade).
- Booking, rescheduling, and canceling sessions.
- Sending and receiving requests.

4. Interactive tutoring sessions

- Live online sessions with conference platforms.
- Option for one-to-one sessions.
-

5. Learning resources

- Resource library with study materials, practice tests, worksheets and past papers.
- All three users tutors, students, parents have the access to add, view, download the resources.

6. Progress tracking

- Performance reports and feedback.
- Achievement tracking.
- View marks and grades for each subject of the student.

7. Parental Involvement

- Monitoring their child's progress.
- Communicate with relevant tutors whom their child or children has/have enrolled.

8. Assignments and due dates management

- This system allows tutors to create assignments, set due dates, and ensure visibility for both students and parents.

Out-scope

- Support for multiple languages
- Built-in tools for marketing, advertising, or promotional campaigns
- SMS notification
- Development of custom content creation tools for tutors.

Constraints and limitations

- Need for a stable internet connection for uninterrupted access.

4. Proposed System's Architecture

Our proposed system architecture is MVC architecture. MVC (Model-View-Controller) is a design pattern widely used in software development to organize application components. It separates concerns into three interconnected parts:

Components and their functionalities

Model: Manages the system's data logic.Handles CRUD operations for classes, students, tutors,admin,parents and assignments.Implements database interactions for storing user details, class schedules, and grades.

View: Handles the user interface, presenting information to users in an interactive and visually accessible way.Provides the user interface for Admin, Tutors, Students, and Parents.

Displays dashboards, forms, progress reports, and assignment grades.Ensures a responsive design for access across devices.

Controller: Acts as the mediator between model and view. It processes user input, updates the model, and refreshes the view accordingly.Manages workflows such as tutor approval,parent approval, assignment grading, and class scheduling.

Component Interactions

Admin workflow:Admin interacts with the view to approve tutors and monitor system activity.Controller updates the model to save tutor details and system changes.

Tutor Workflow:Tutors upload assignments and grade them via the view.Controller validates inputs and updates assignment details in the model.

Student Workflow:Students book classes and complete assignments through the view.Controller checks availability and records assignment submissions in the model.

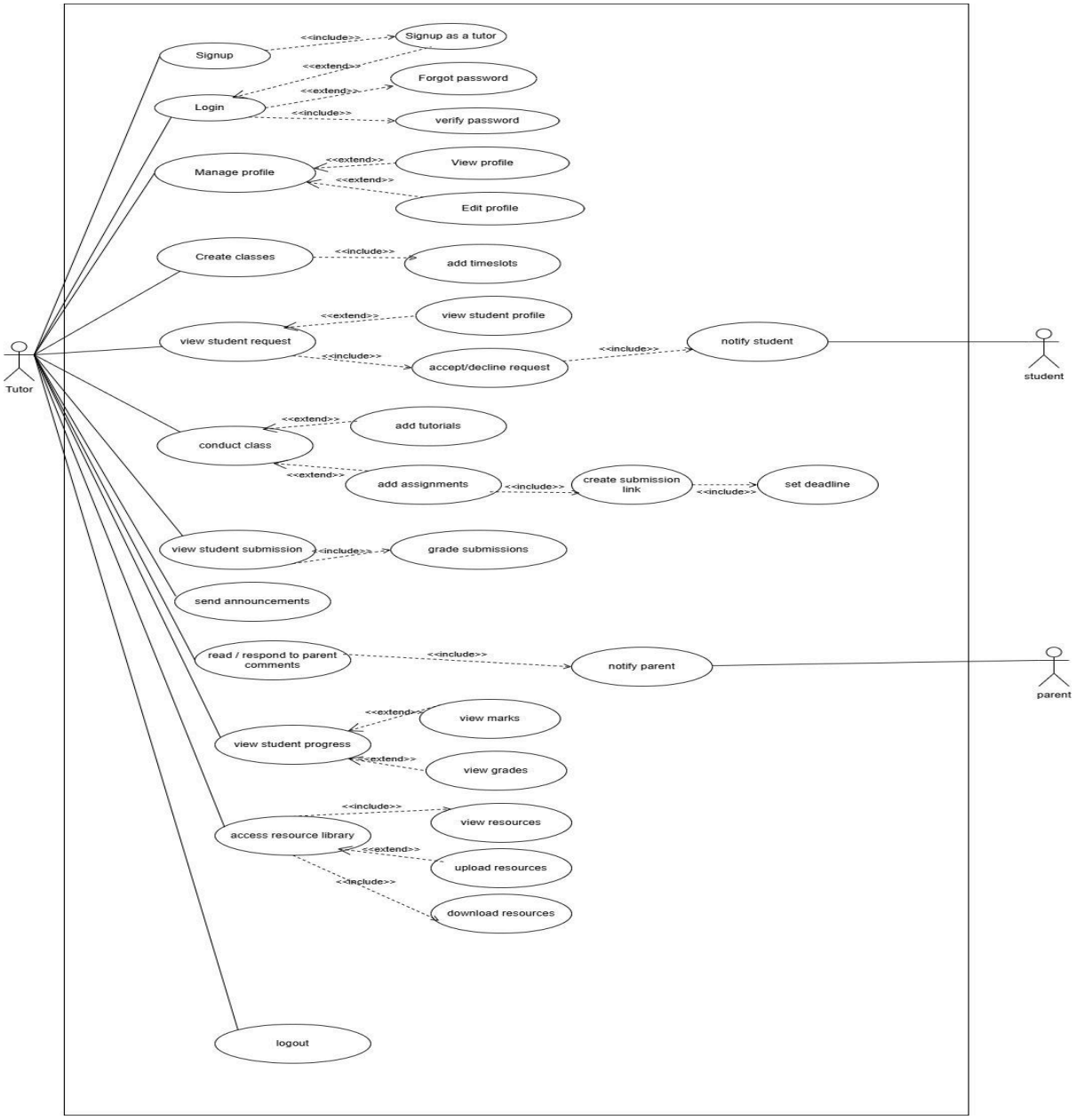
Parent Workflow:Parents login and send a request to add their child by entering the child’s details.After approval, the child appears in the parent’s "My Child" list.

Parents access the view to monitor their child's academic progress.Controller fetches data from the model and displays detailed progress reports.

5. System Design Diagrams

Use case diagrams

Tutor:



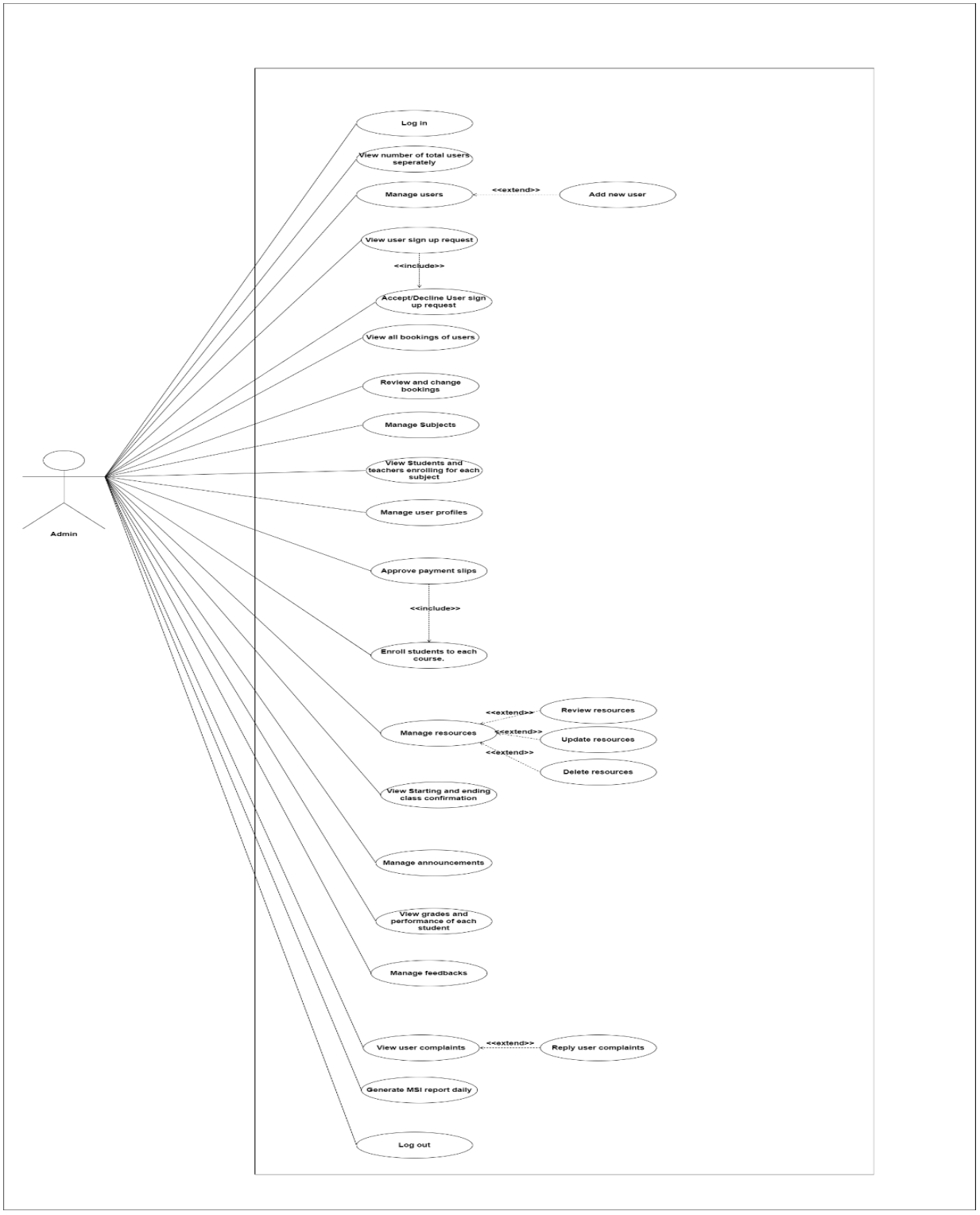
Student:


Signup
signup as a student
Login
forgot password
verify password
Manage profile
view profile
edit profile
Search tutors
View tutor profile
Cancel request
Send request to tutor
Notify tutor
Tutor
Send feedback / comments to tutor
View parent profile
Approve request
Decline the approval
View parent request
Reject request
Notify parent
Parent
Book classes
cancel bookings
Make payment to class
Get enrolled to the class after payment approval
Join class
View announcement
View marks
View progress
View grades
Admin
View resources
Access resource library
Download resources
Upload resources
View assignments
Edit submission
Add submission
Give reviews and ratings to tutor
Logout
Student
<<include>>
<<extend>>

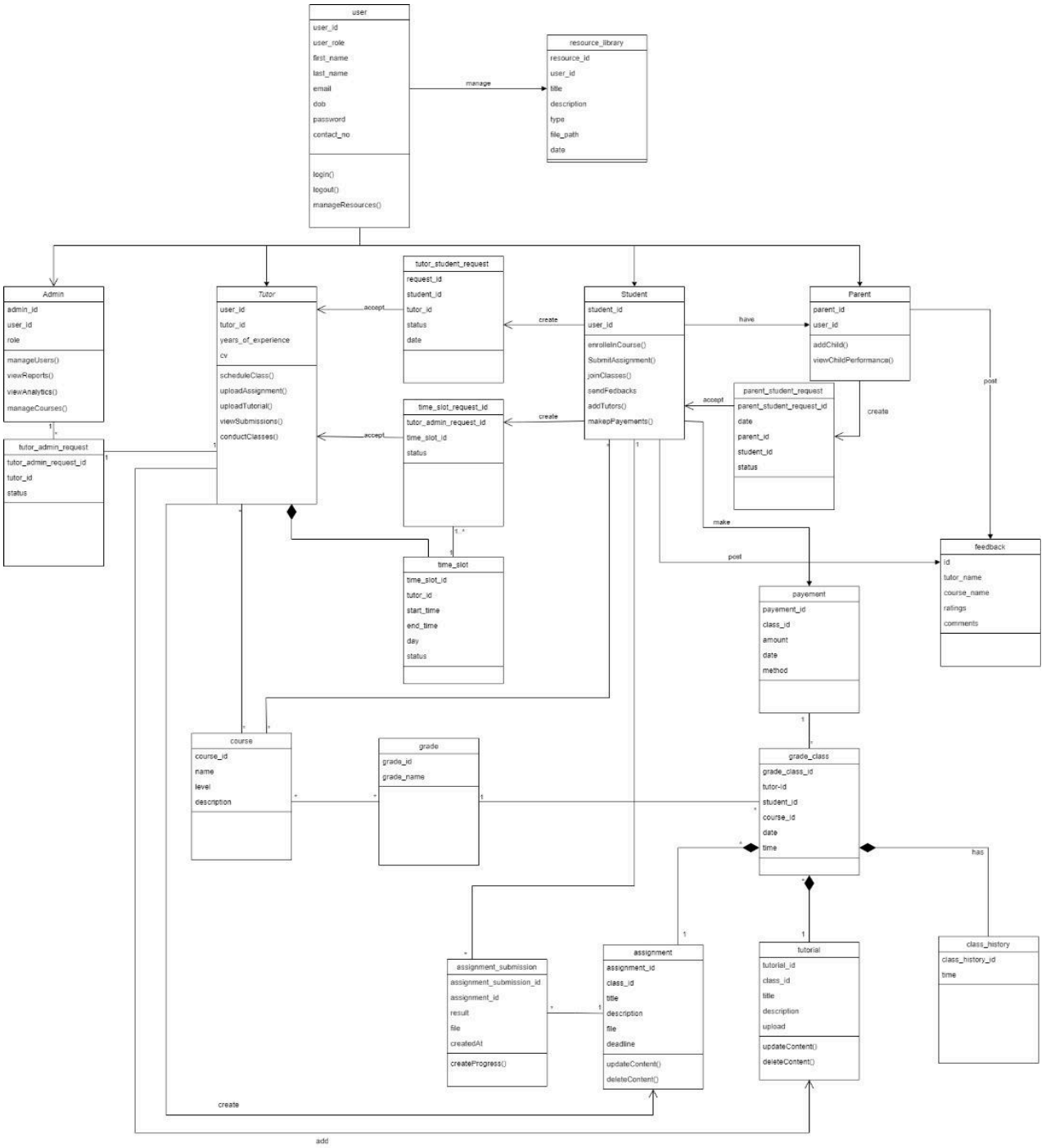
Parents:


signup
<<include>>
Signup as parent
<<extend>>
Login
<<extend>>
forgot password
<<include>>
verify password
<<extend>>
view profile
Manage profile
<<extend>>
edit profile
Send request to student to add as child
<<include>>
notify student
Student
View child in child list
View child profile
Parent
View enrolled subjects of child
View enrolled tutor's profile
Add comments to enrolled tutor
<<include>>
Notify tutor
Tutor
View class history
View upcoming classes of child
View marks
<<include>>
View child progress
<<include>>
View grades
<<include>>
View progress graph with filters
View announcements
Read FAQs
View assignment due deadlines
View submission status
View resources
<<include>>
Access resource library
<<extend>>
Download resources
<<extend>>
Upload resources
Logout

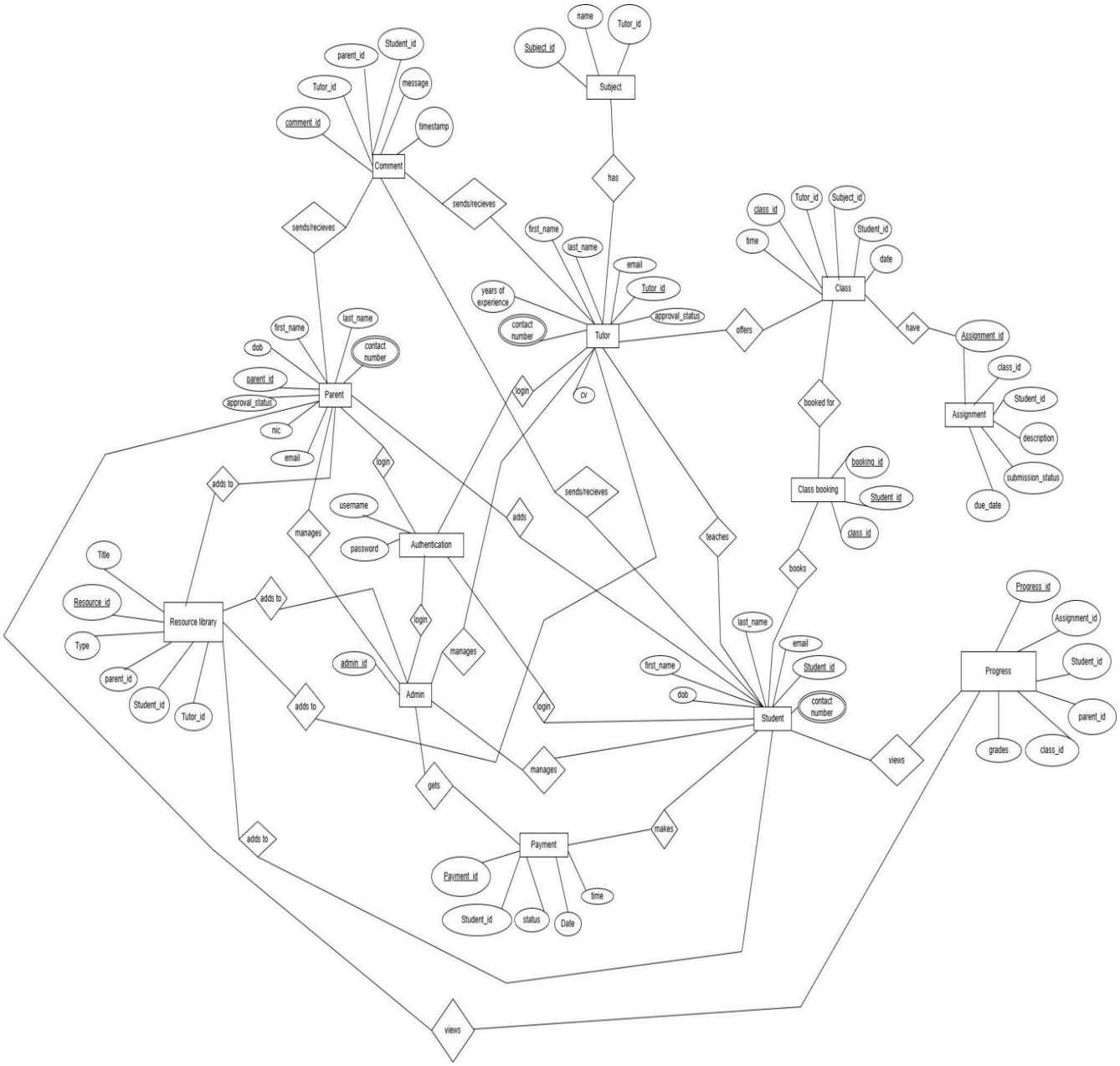
Admin:


Log in
View number of total users seperately
Manage users
<<extend>>
Add new user
View user sign up request
<<include>>
Accept/Decline User sign up request
View all bookings of users
Review and change bookings
Manage Subjects
View Students and teachers enrolling for each subject
Manage user profiles
Approve payment slips
<<include>>
Enroll students to each course.
Manage resources
<<extend>>
Review resources
<<extend>>
Update resources
<<extend>>
Delete resources
View Starting and ending class confirmation
Manage announcements
View grades and performance of each student
Manage feedbacks
View user complaints
<<extend>>
Reply user complaints
Generate MSI report daily
Log out
Admin

Class diagram

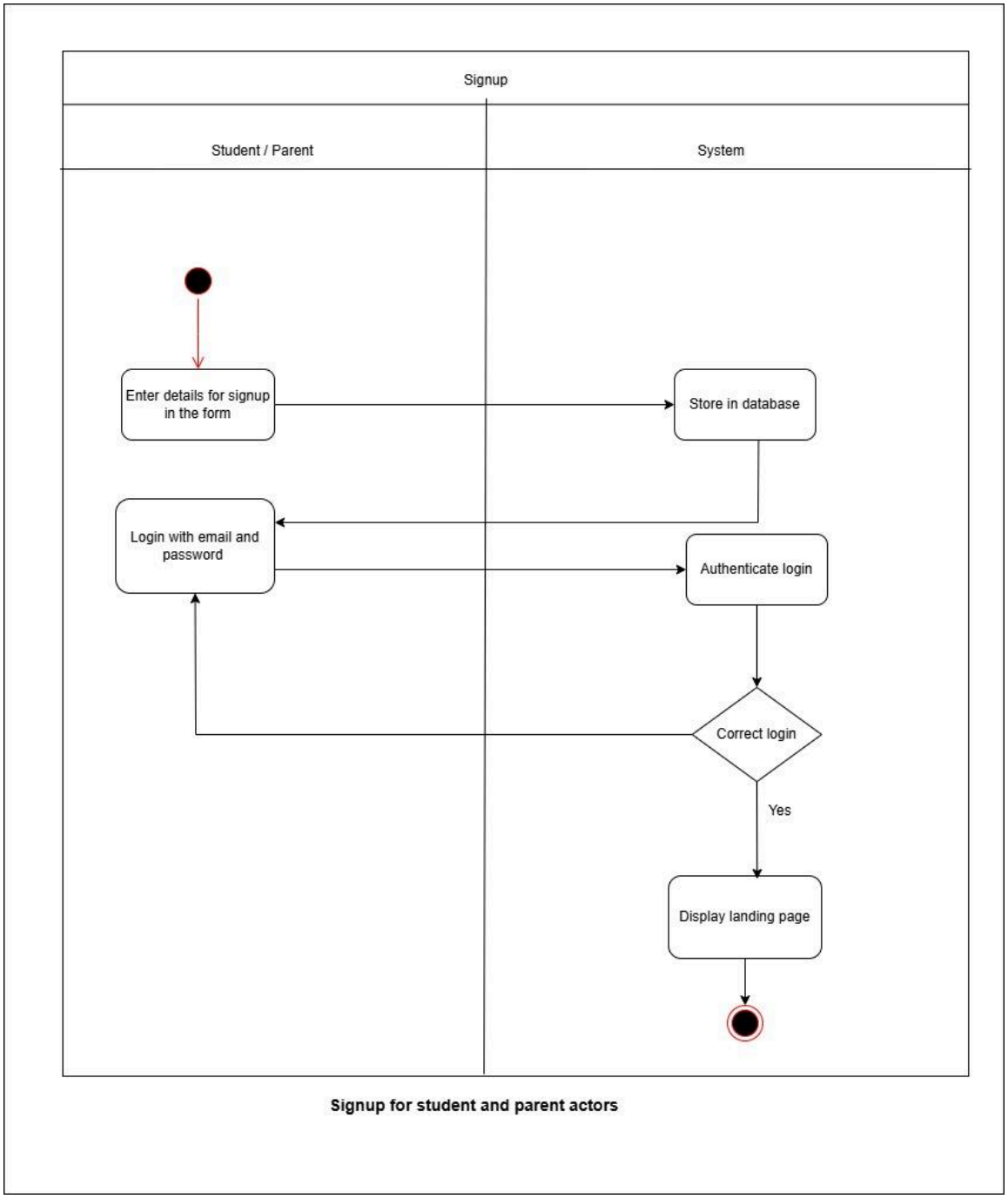


ER Diagram

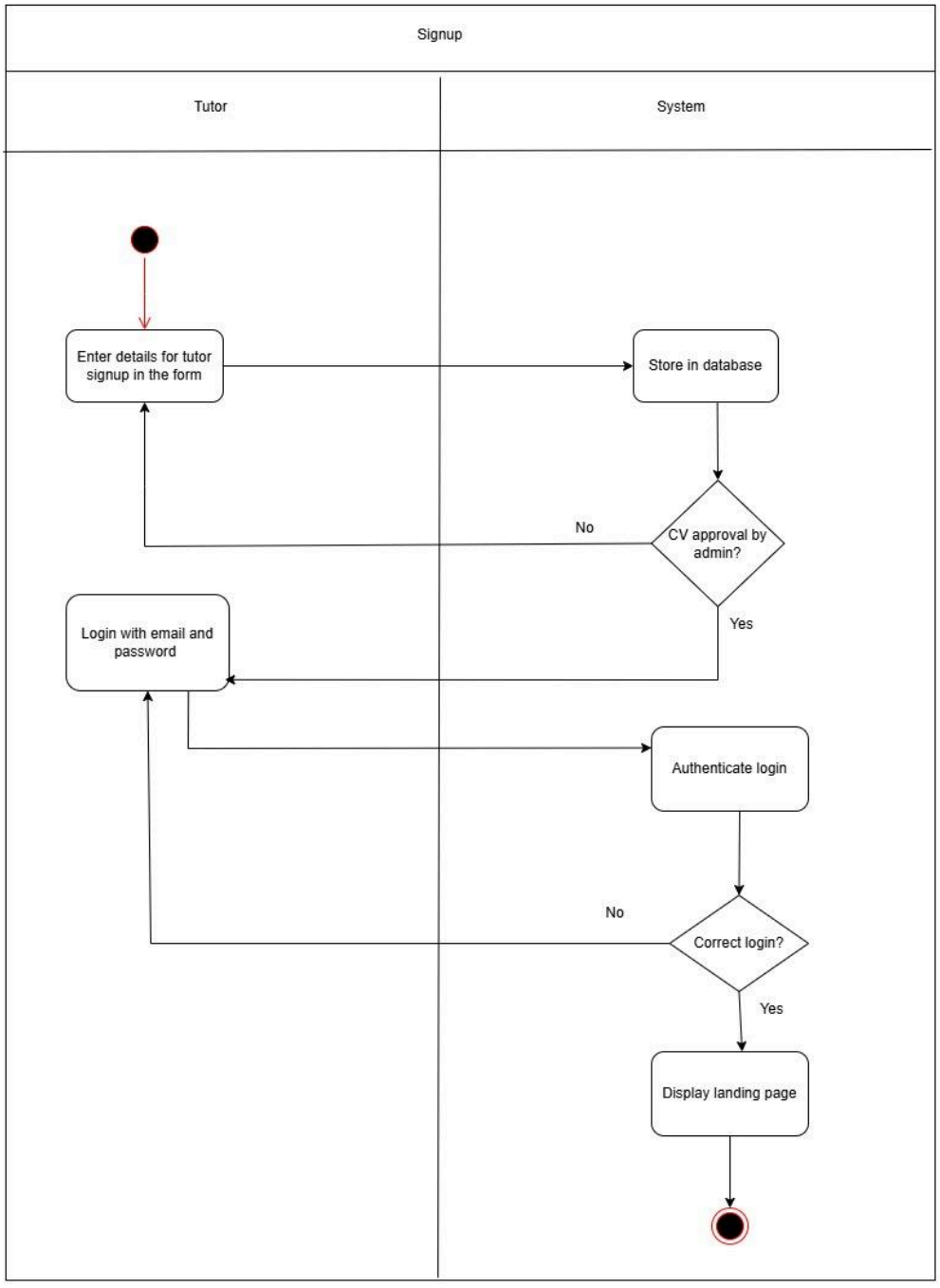

Comment
comment_id
Student_id
parent_id
Tutor_id
message
timestamp
sends/recieves
sends/recieves
Subject
Subject_id
name
Tutor_id
has
Parent
parent_id
first_name
last_name
dob
contact number
approval_status
nic
email
Tutor
first_name
last_name
email
Tutor_id
years of experience
contact number
approval_status
cv
login
offers
Class
class_id
Tutor_id
Subject_id
Student_id
time
date
have
Assignment_id
Assignment
class_id
Student_id
description
submission_status
due_date
booked for
Class booking
booking_id
Student_id
class_id
books
teaches
Authentication
username
password
login
adds to
manages
adds
sends/recieves
Resource library
Resource_id
Title
Type
parent_id
Student_id
Tutor_id
adds to
Admin
admin_id
login
manages
adds to
Student
first_name
last_name
email
Student_id
dob
contact number
login
manages
views
Progress
Progress_id
Assignment_id
Student_id
parent_id
grades
class_id
gets
makes
Payment
Payment_id
Student_id
status
Date
time
adds to
views

Activity diagrams

Signup

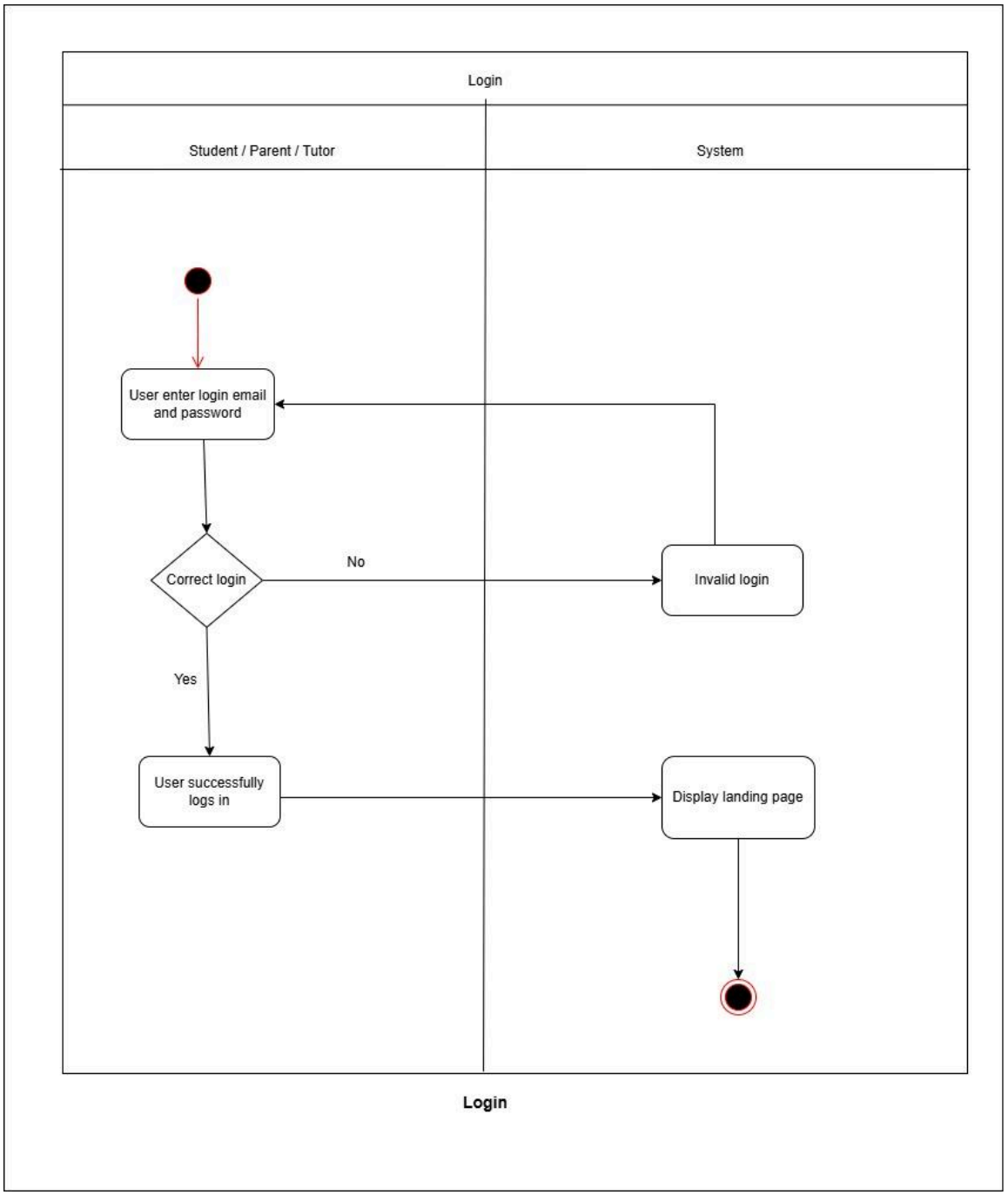


Signup for student and parent actors



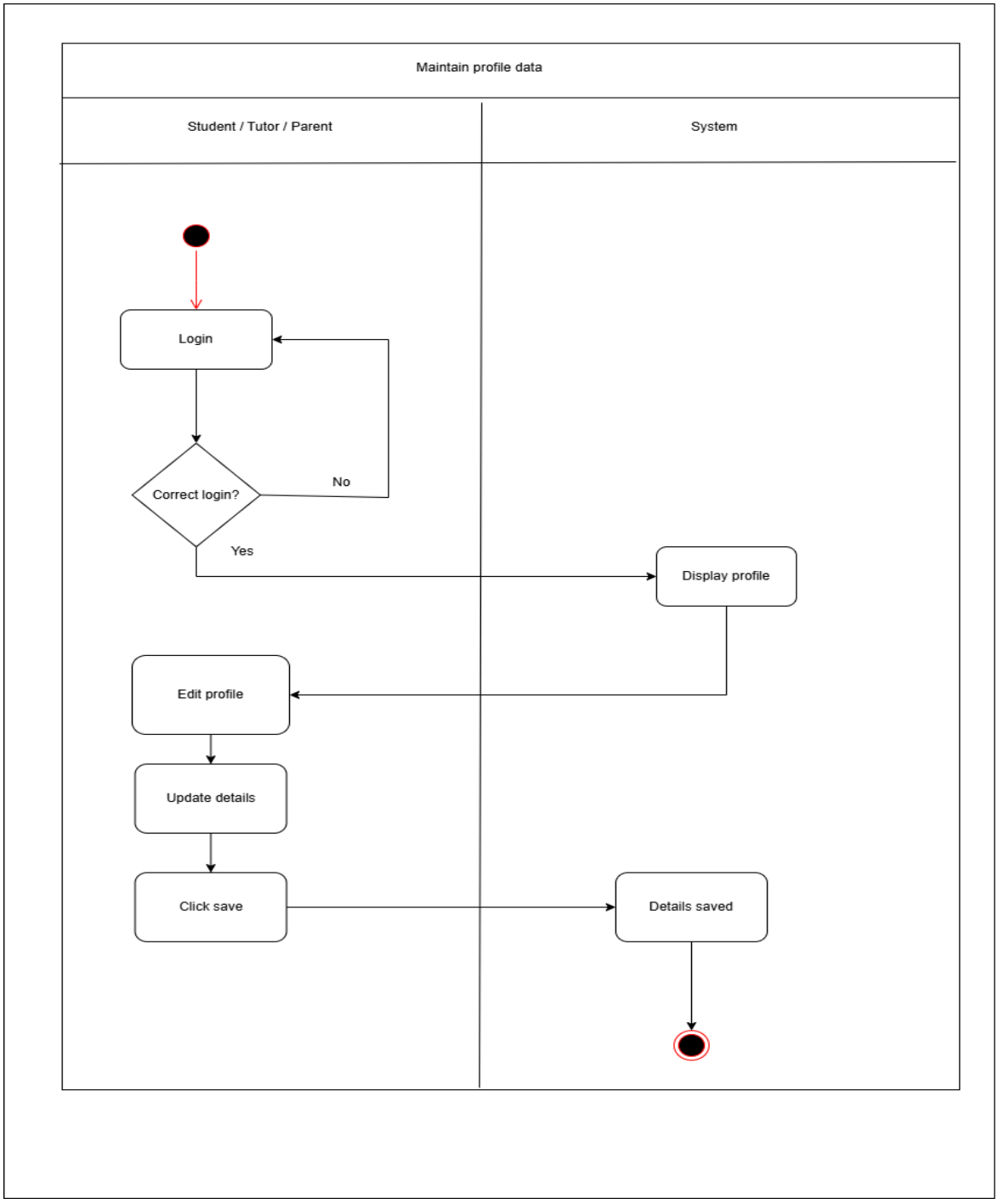
Signup for tutor actor

Login

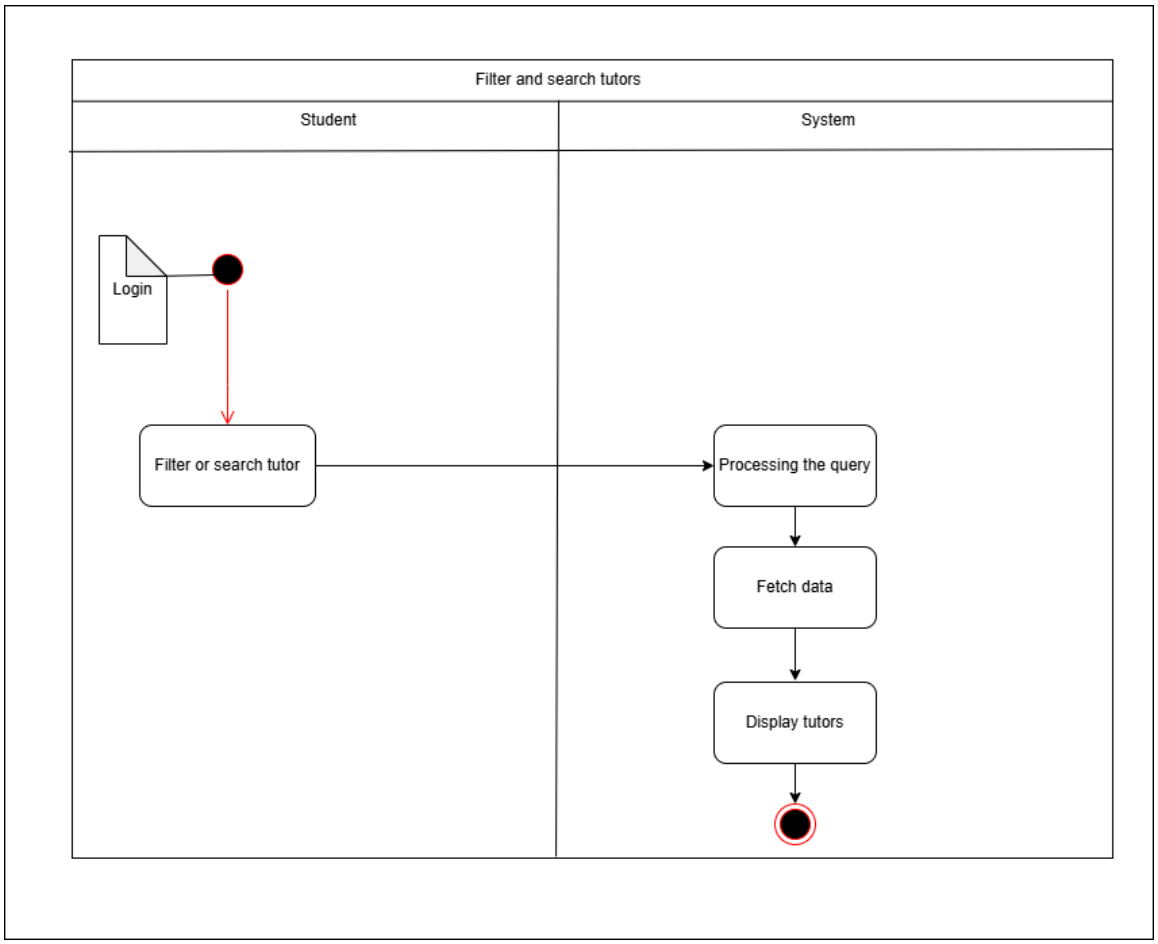

Login
Student / Parent / Tutor
System
User enter login email and password
Correct login
No
Invalid login
Yes
User successfully logs in
Display landing page

Login

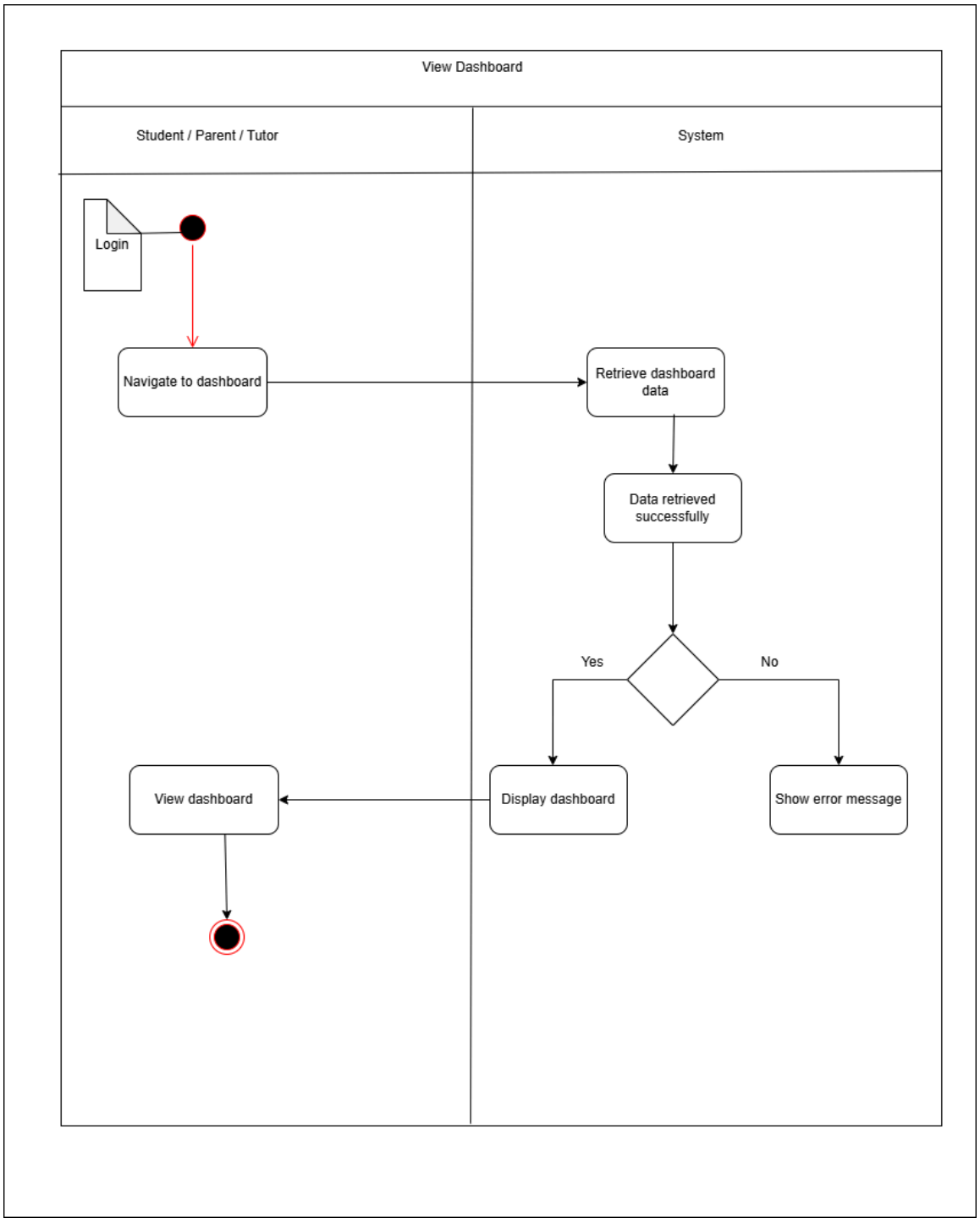
Profile management



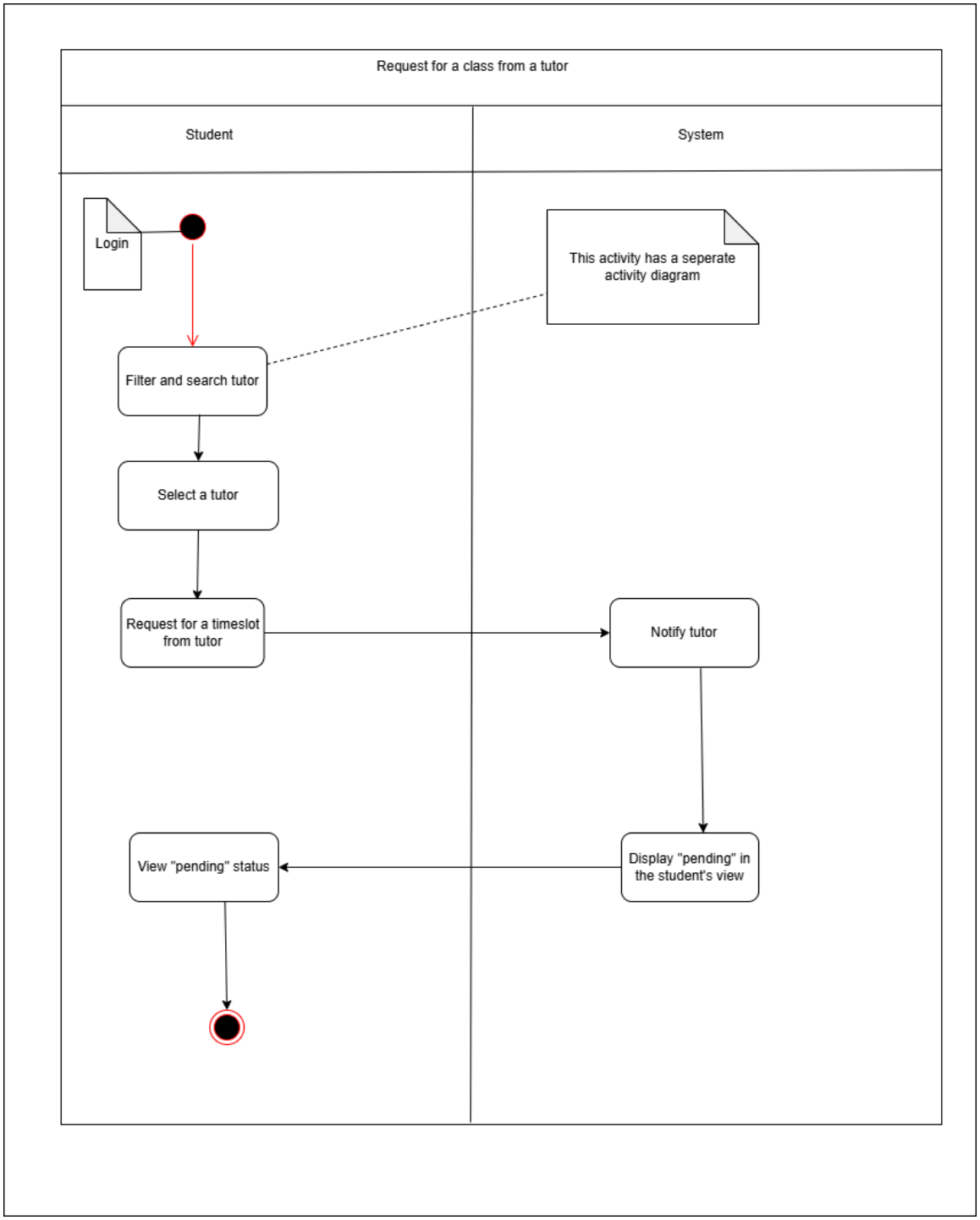
Filter and search tutors



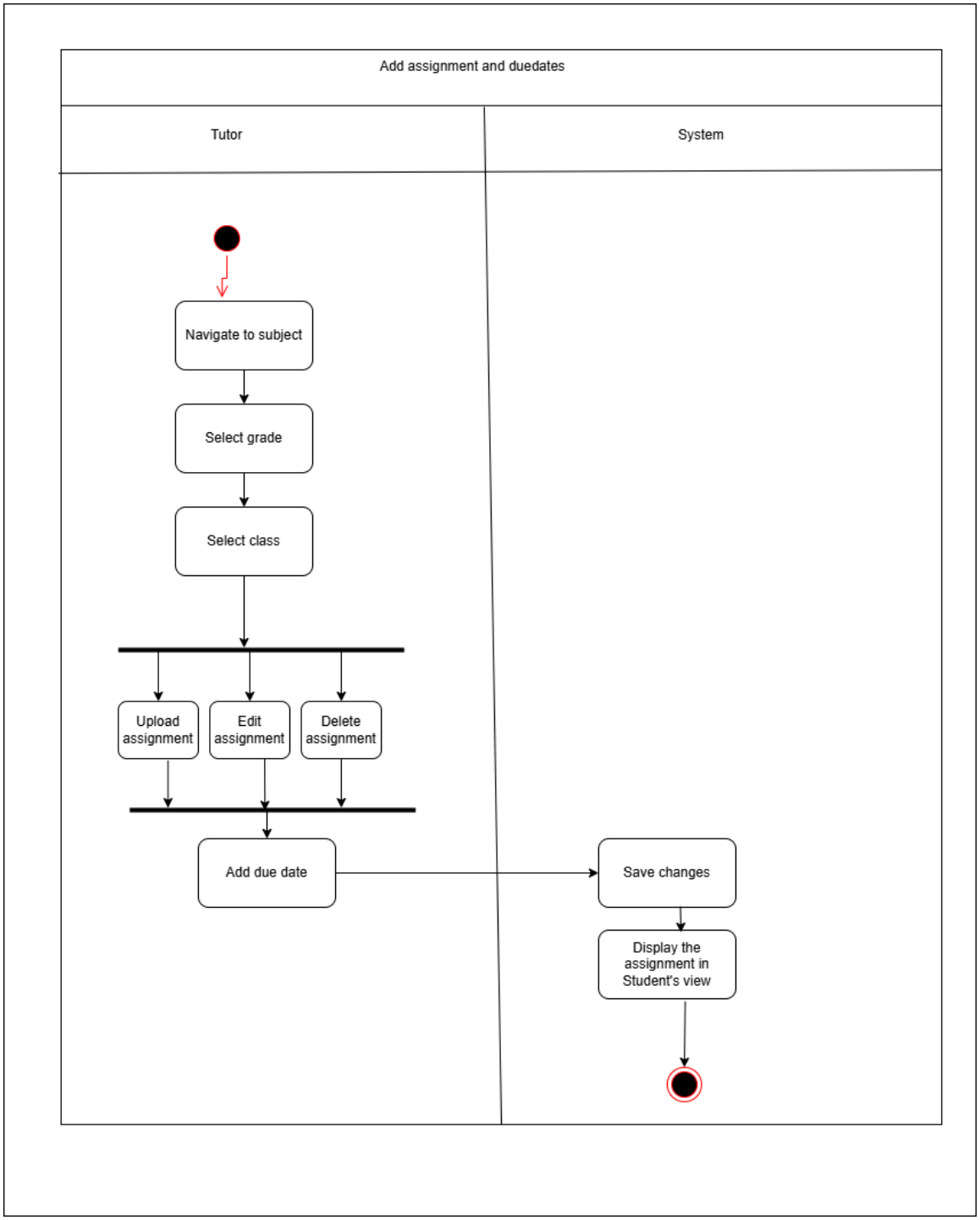
View dashboard

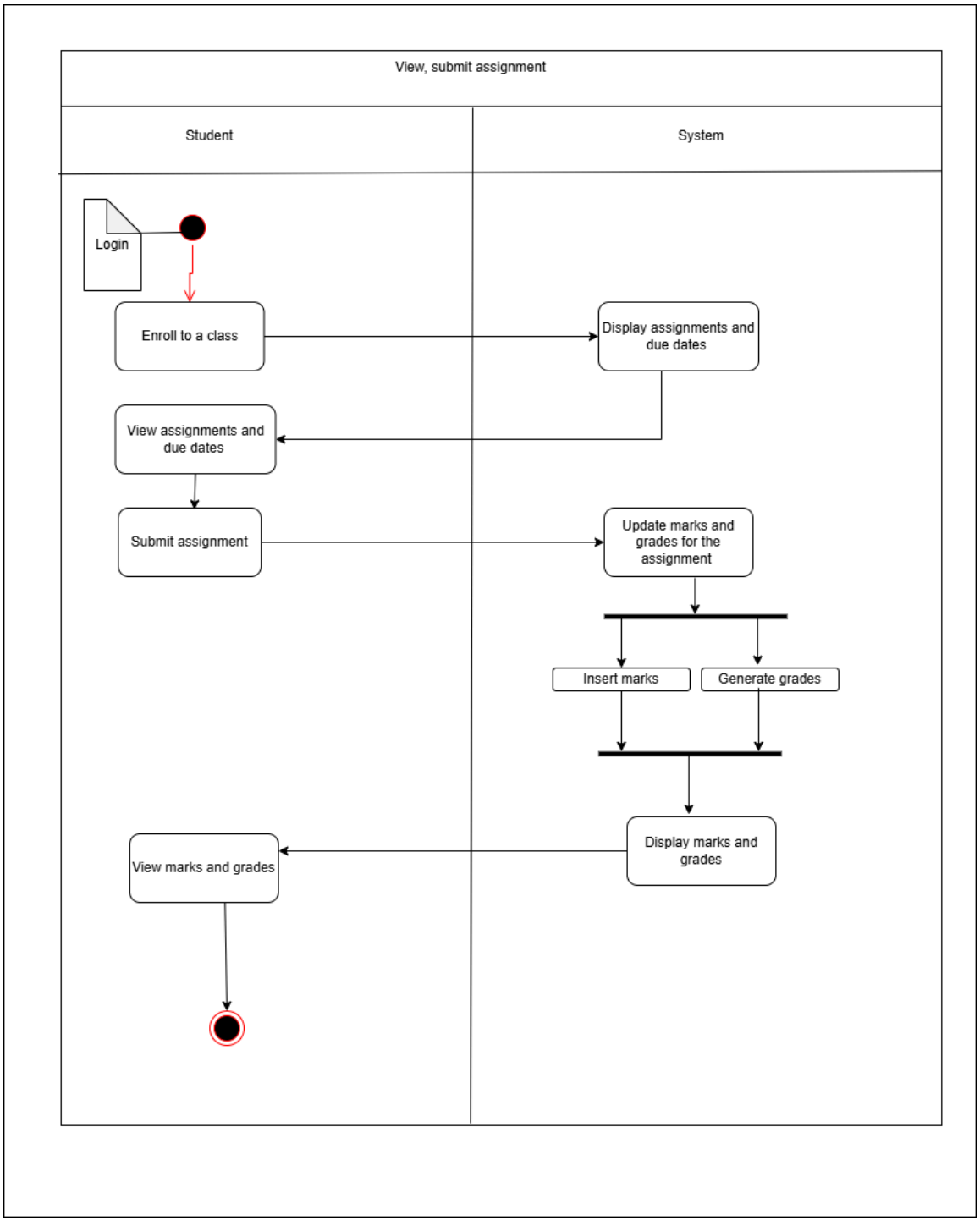


Request class by student

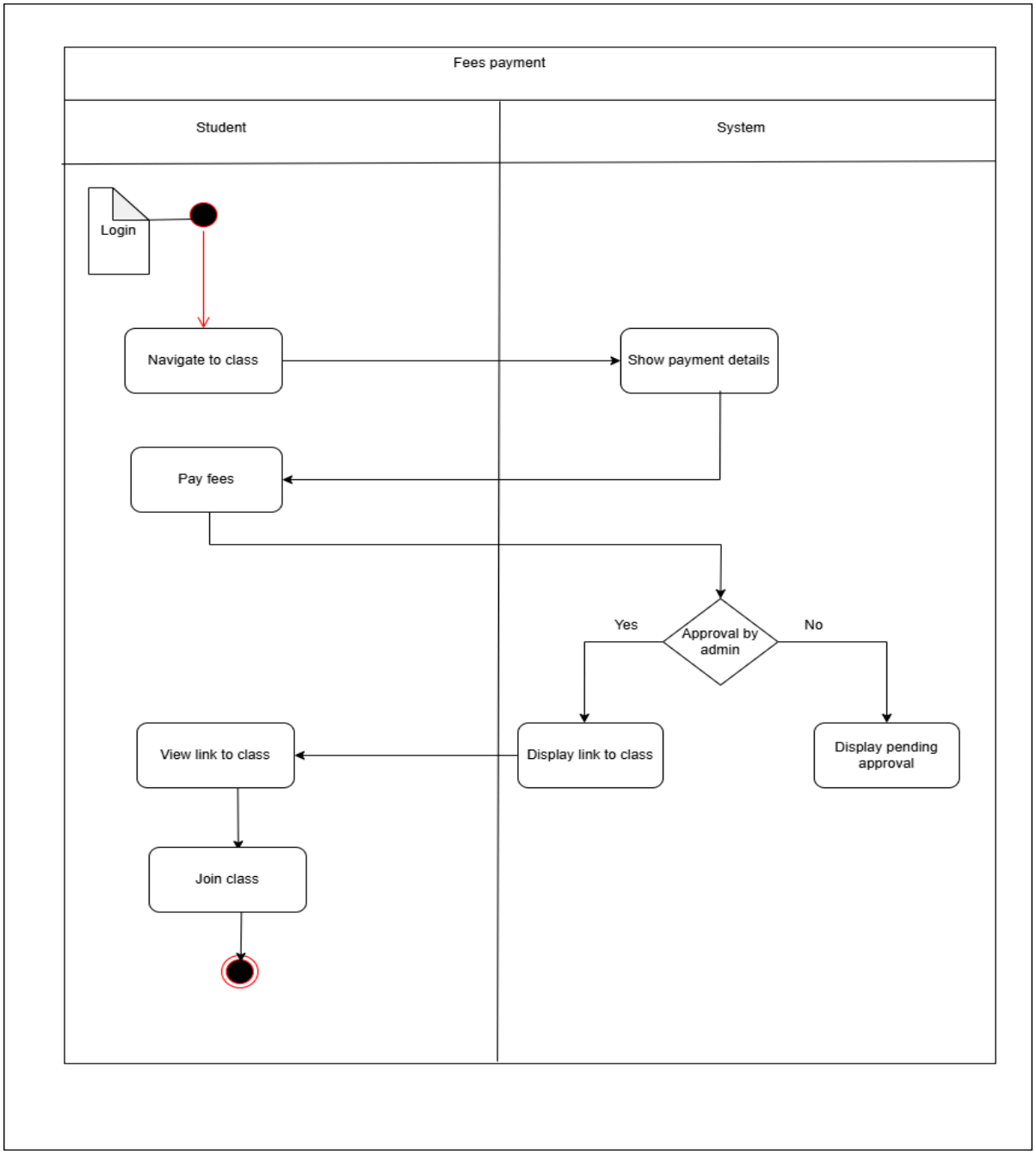


Assignments management

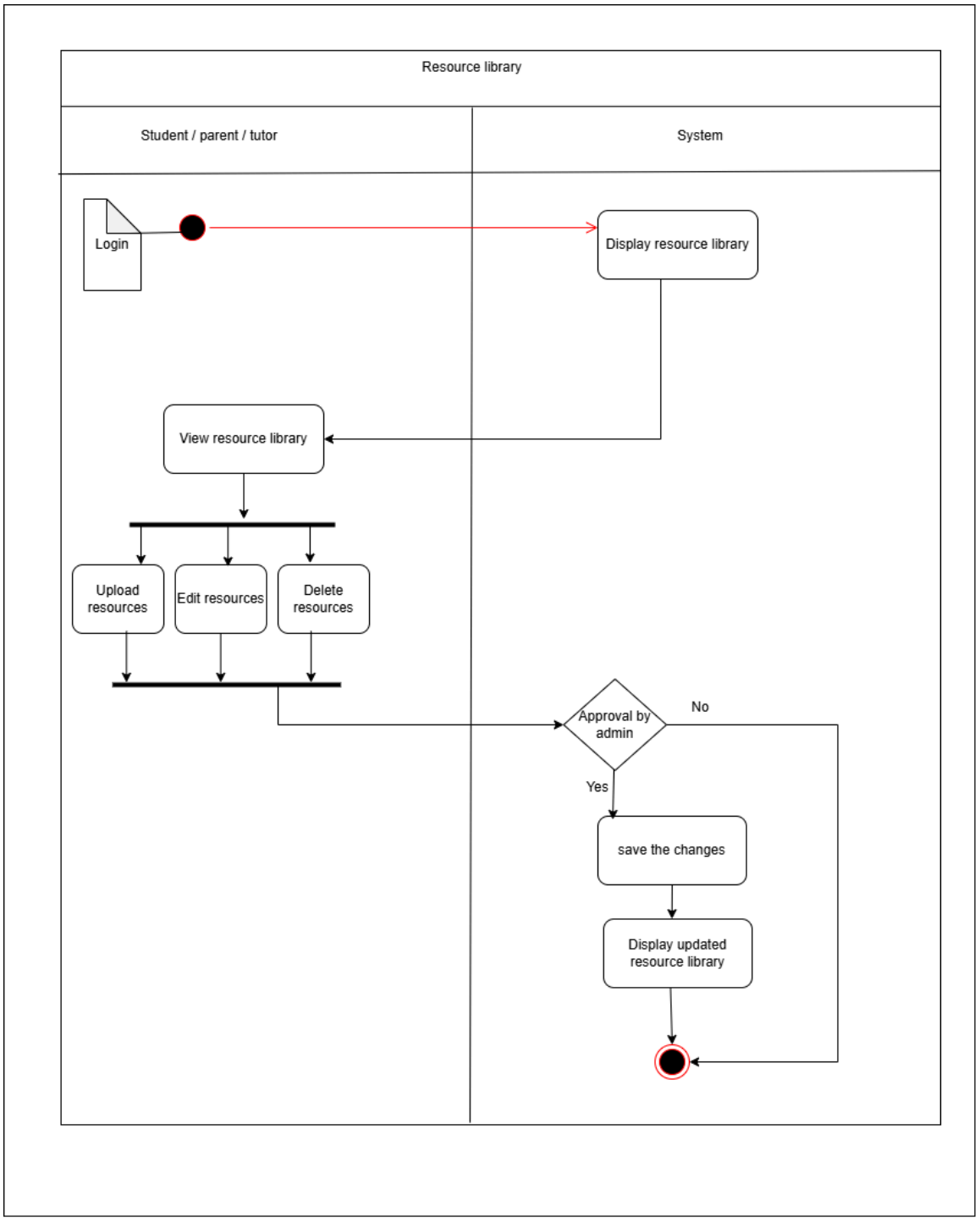



View, submit assignment
Student
System
Login
Enroll to a class
Display assignments and due dates
View assignments and due dates
Submit assignment
Update marks and grades for the assignment
Insert marks
Generate grades
Display marks and grades
View marks and grades

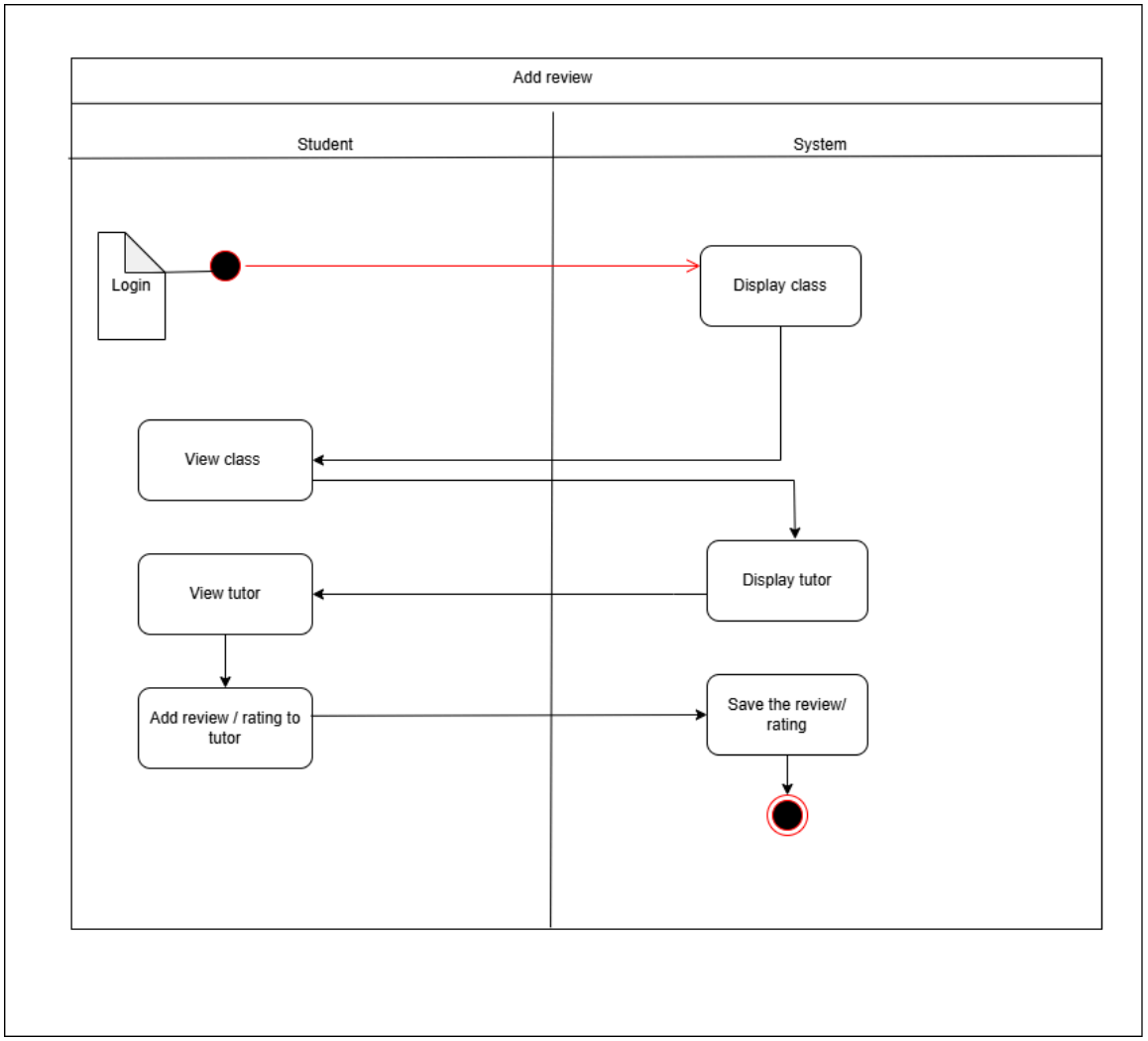
Fees management

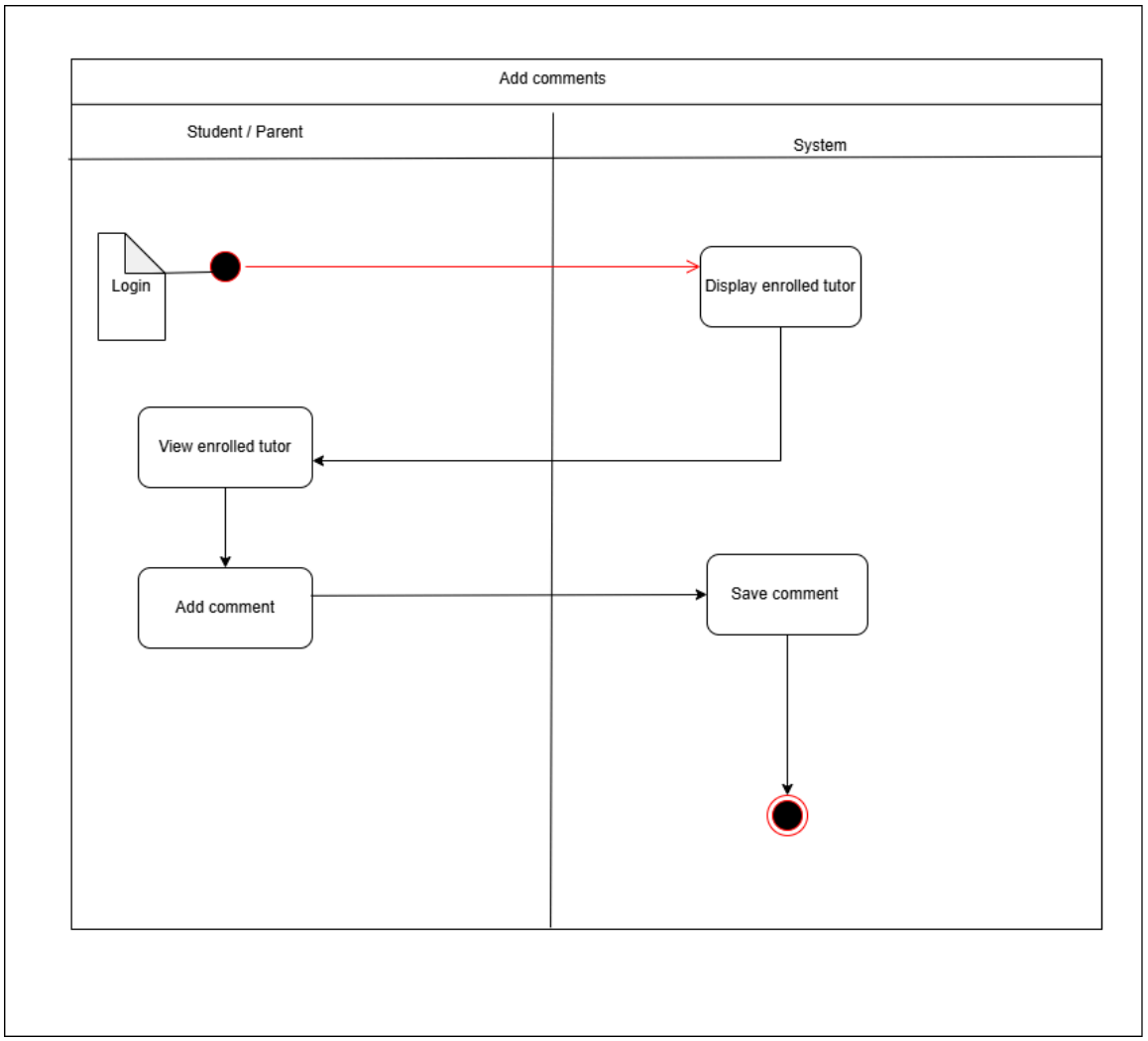


Resource library

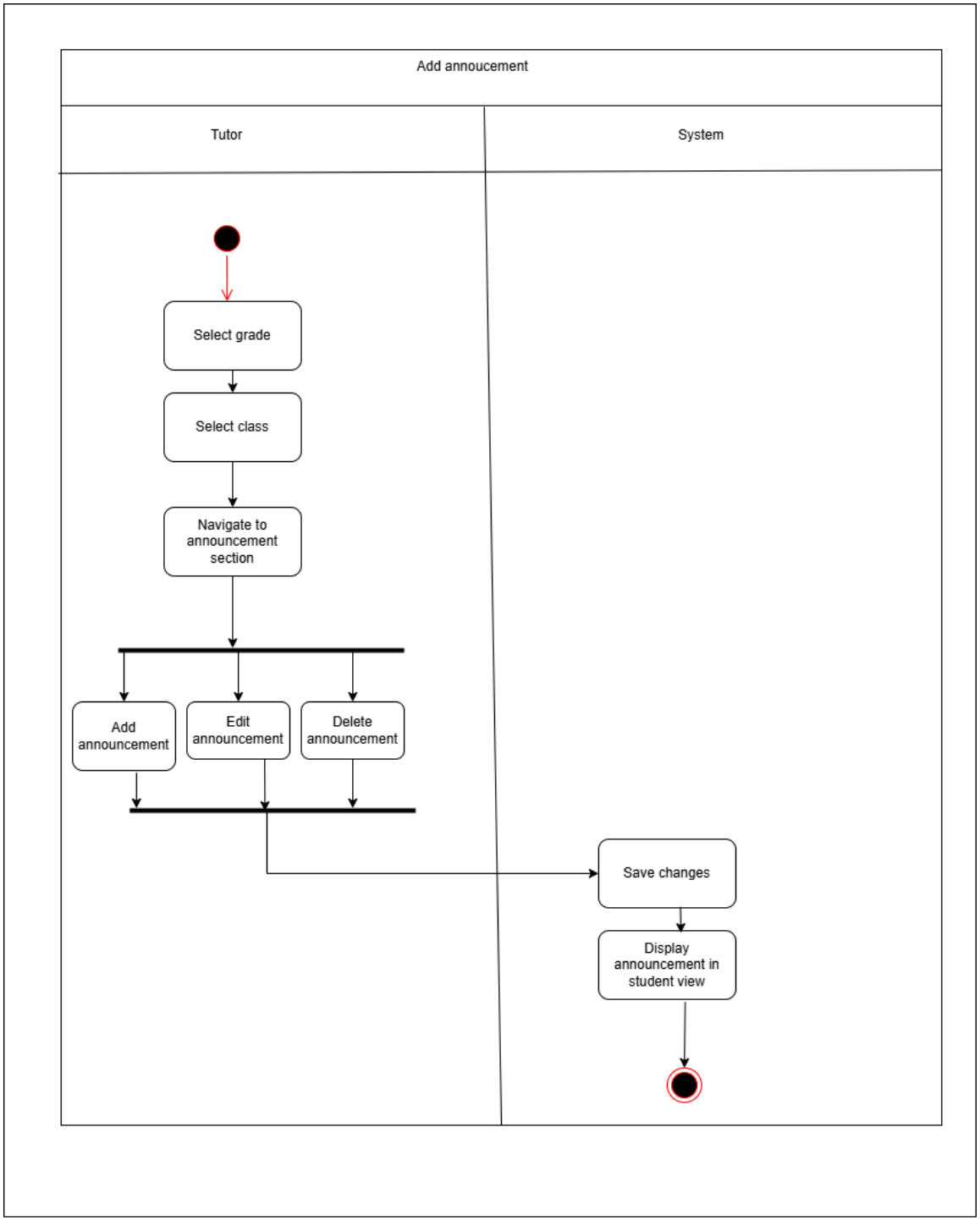


Add reviews, ratings and comments

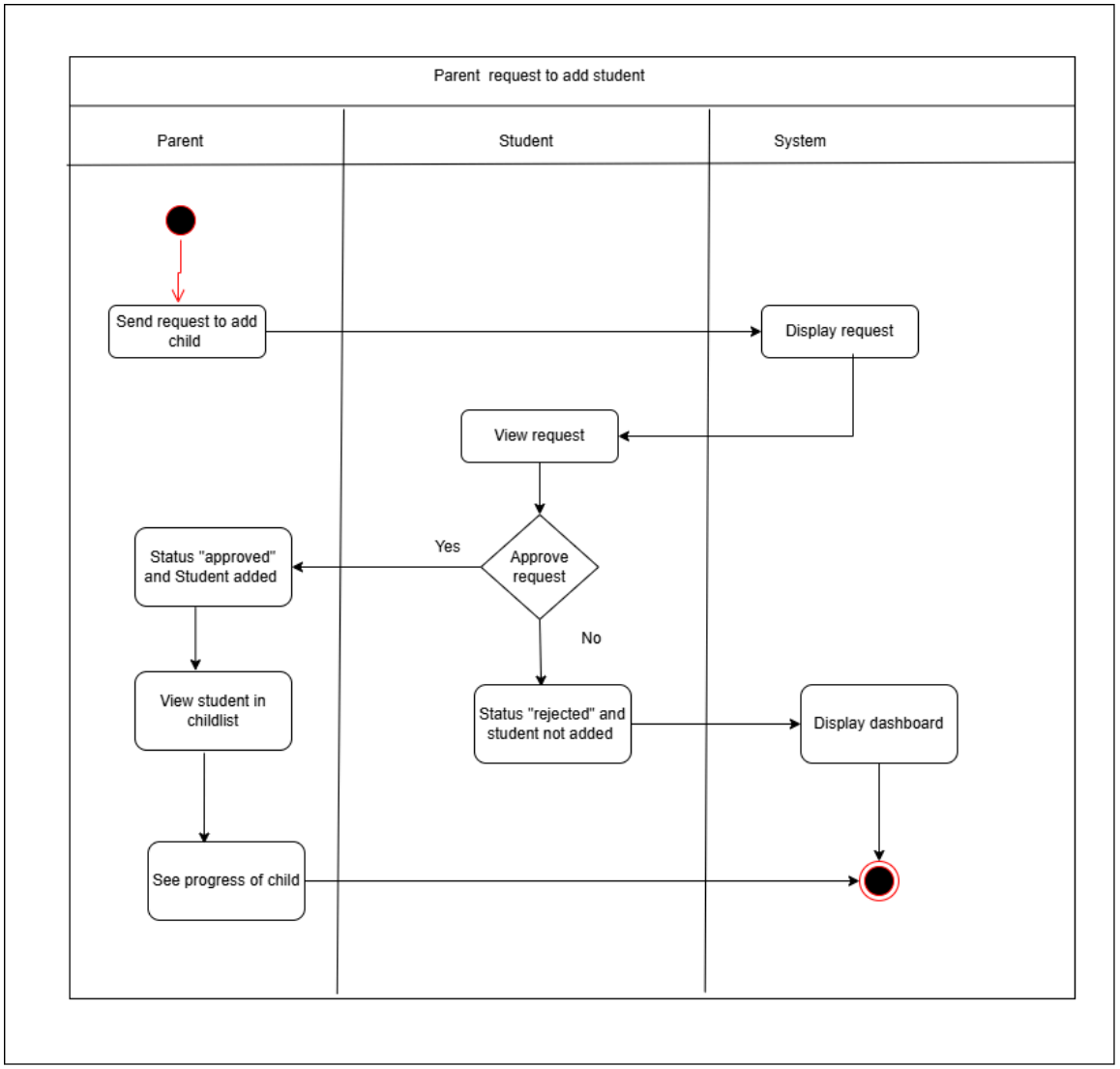



Add comments
Student / Parent
System
Login
Display enrolled tutor
View enrolled tutor
Add comment
Save comment

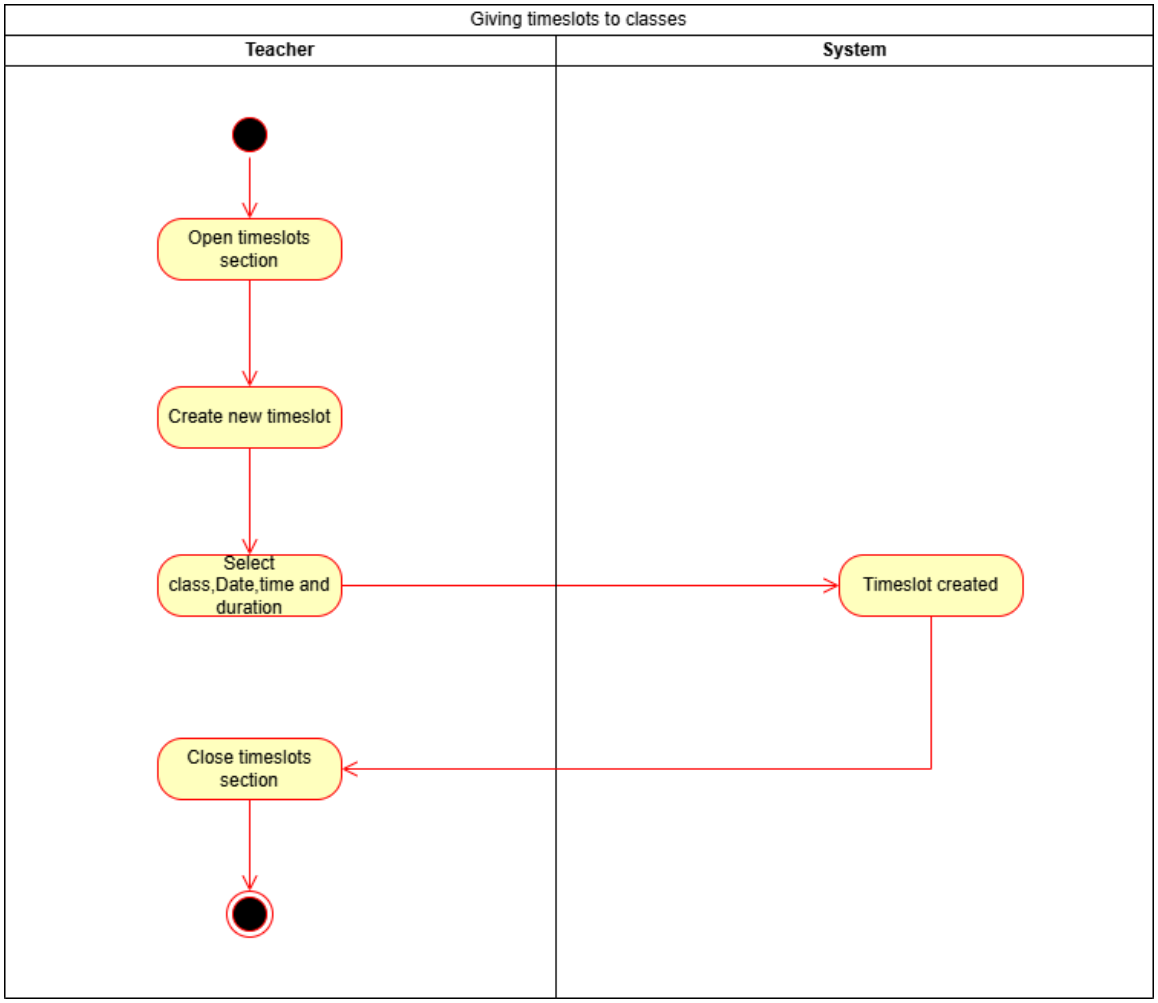
Announcement



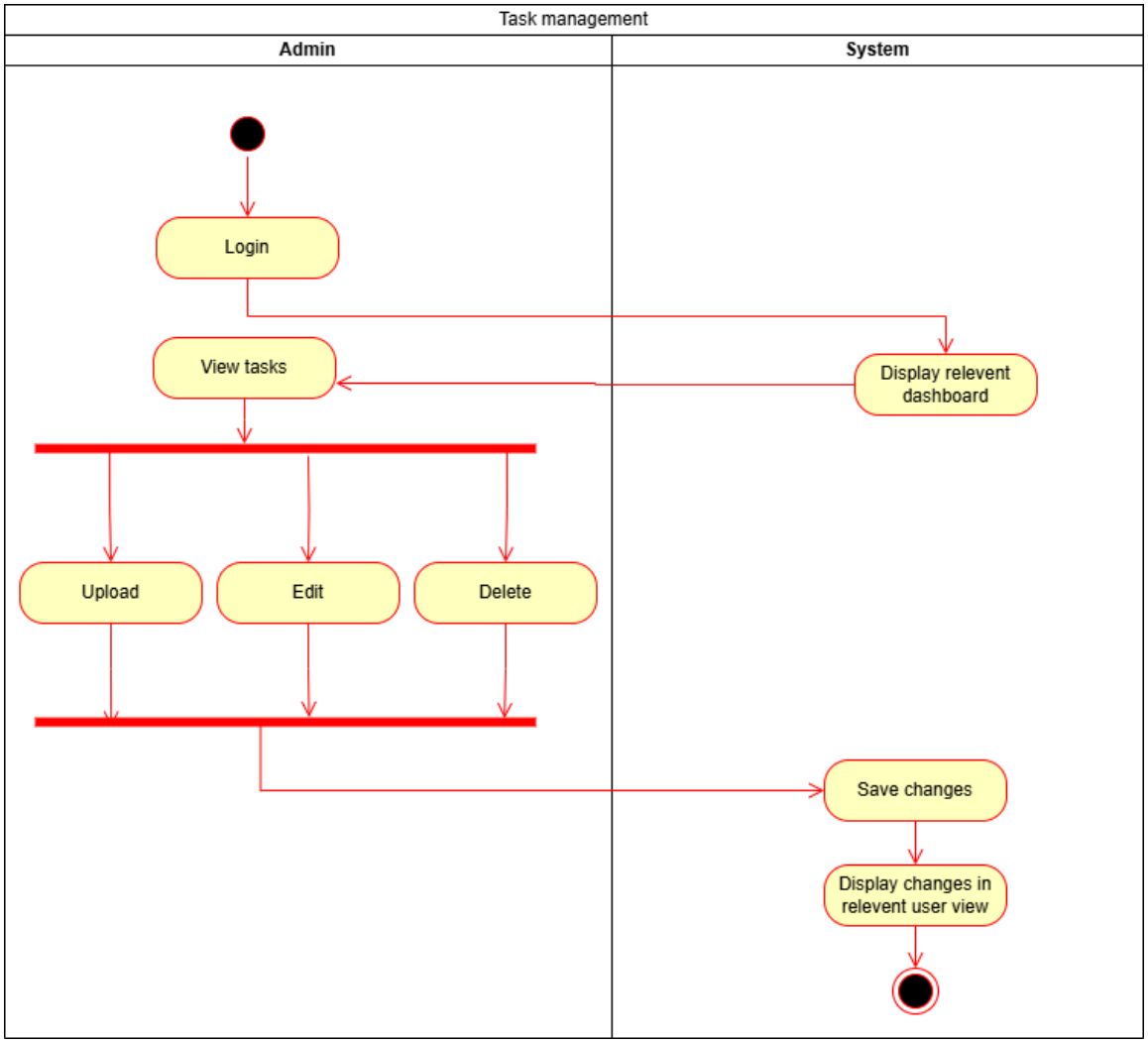
Add child request



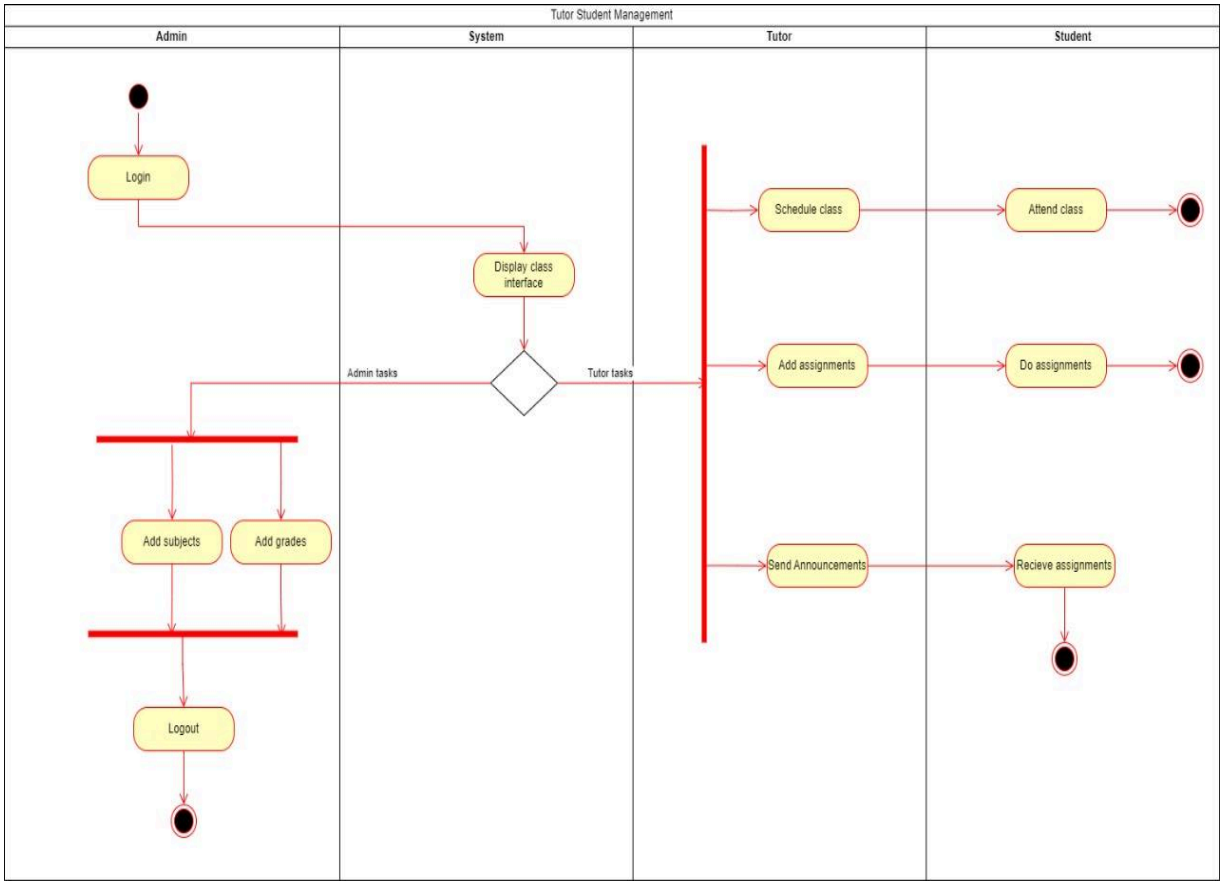
Time slot management



Task management



Tutor - student management



6. Current Progress

System development progress concerning the system requirements

We’re making progress on the system, with most of the frontend and UI design work nearly complete. While there’s still work to do on the database, backend and authentication.

Percentage of the system completed now

Frontend development - 98%
UI designing - 100%
Database - 70%
Authentication - 90%
Backend development - 40%

Remaining tasks / work?

Frontend development - 2%
Database expansion - 30%
Backend development - 60%
Testing and deployment - 95%
Update changes and maintenance - 97%

Each member’s contribution

Team member	Contribution
Farshad M.F.M 22020233	UI design and frontend development for the **admin** actor, database development, backend development for the **admin** CRUD functionality,MVC architecture development
Sajidha M.S.F 22020853	UI design and frontend development for the **parent** actor, database development, backend development for the **parent** CRUD functionality, MVC architecture development.
Rashmika A.D.C 22020812	UI design and frontend development for the **student** actor, database development, backend development for the **student** CRUD functionality, MVC architecture development.
Kuruppu Arachchi K.A.D.A.A 22020519	UI design and frontend development for the **tutor** actor, database development, backend development for the **tutor** CRUD functionality, MVC architecture development