Software Requirements and Design Document

for

FYP CONNECT

Prepared by

1. **22i-0773 Shayan**
2. **22i-0781 Ryan**
3. **22i-0789 Zubair**

Company name : Lucid Softwares

Date : 26 nov 2024

Table of Contents

1. Introduction

1.1 Purpose

This document outlines the software requirements for the Final Year Project (FYP) Management Portal, designed specifically for FAST University. The portal aims to simplify the management of undergraduate FYPs by automating administrative processes, improving communication, and offering tools to track project progress.

In this SRS document, we detail the portal's key features, catering to the needs of students, supervisors, and administrators. The goal is to create a centralized platform that manages every stage of the FYP journey, from submitting proposals to final evaluations.

1.2 Product Scope

The FYP Management Portal is a desktop application built with JavaFX, designed to tackle the inefficiencies in managing Final Year Projects (FYPs) at FAST University. With JavaFX, the portal offers an engaging and responsive interface that enhances the user experience while ensuring solid performance.

Purpose and Goals:

- **Simplify Administrative Work**: Automates tasks like scheduling evaluations, tracking deadlines, and managing project documentation in an easy-to-use desktop application.
- **Enable Effective Communication**: Creates a smooth communication flow between students, supervisors, and administrators in a user-friendly setting.
- **Centralized Progress Monitoring**: Provides tools to keep track of milestones, approvals, and deadlines, making sure projects stay on track.

Relevant Benefits:

- **Rich User Experience**: Utilizes JavaFX to offer a modern and dynamic interface with easy navigation.
- **Streamlined Processes**: Consolidates everything into one platform, removing the need for multiple communication channels or tools.
- **Desktop-Focused Solution**: A standalone application designed for offline use, perfectly suited to FAST University's academic environment.

By leveraging JavaFX, the FYP Management Portal strikes a balance between high performance, ease of use, and scalability, catering to the needs of students, supervisors, and administrators while supporting FAST University's goal of integrating technology into academic workflows.

1.3 Title

FYP Connect

Simplifying Final year project management!

1.4 Objectives

- **Simplify Project Management**: Provide a single platform to manage every stage of Final Year Projects, from initial proposals to final evaluations.
- **Improve Communication:** Make it easier for students, supervisors, and admins to stay connected and share updates.
- **Track Progress:** Help keep projects on track with tools to monitor milestones and deadlines.
- **Boost Accountability:** Keep clear records of progress and communication to ensure everyone stays accountable.
- **Increase Efficiency:** Reduce delays and eliminate manual work with streamlined, organized workflows.

1.5 Problem Statement

Managing Final Year Projects (FYPs) can be a complex and time-consuming task. Things like submitting proposals, scheduling evaluations, and tracking deadlines are often done manually through emails and documents, which can lead to confusion, missed deadlines, and added work for students, supervisors, and administrators.

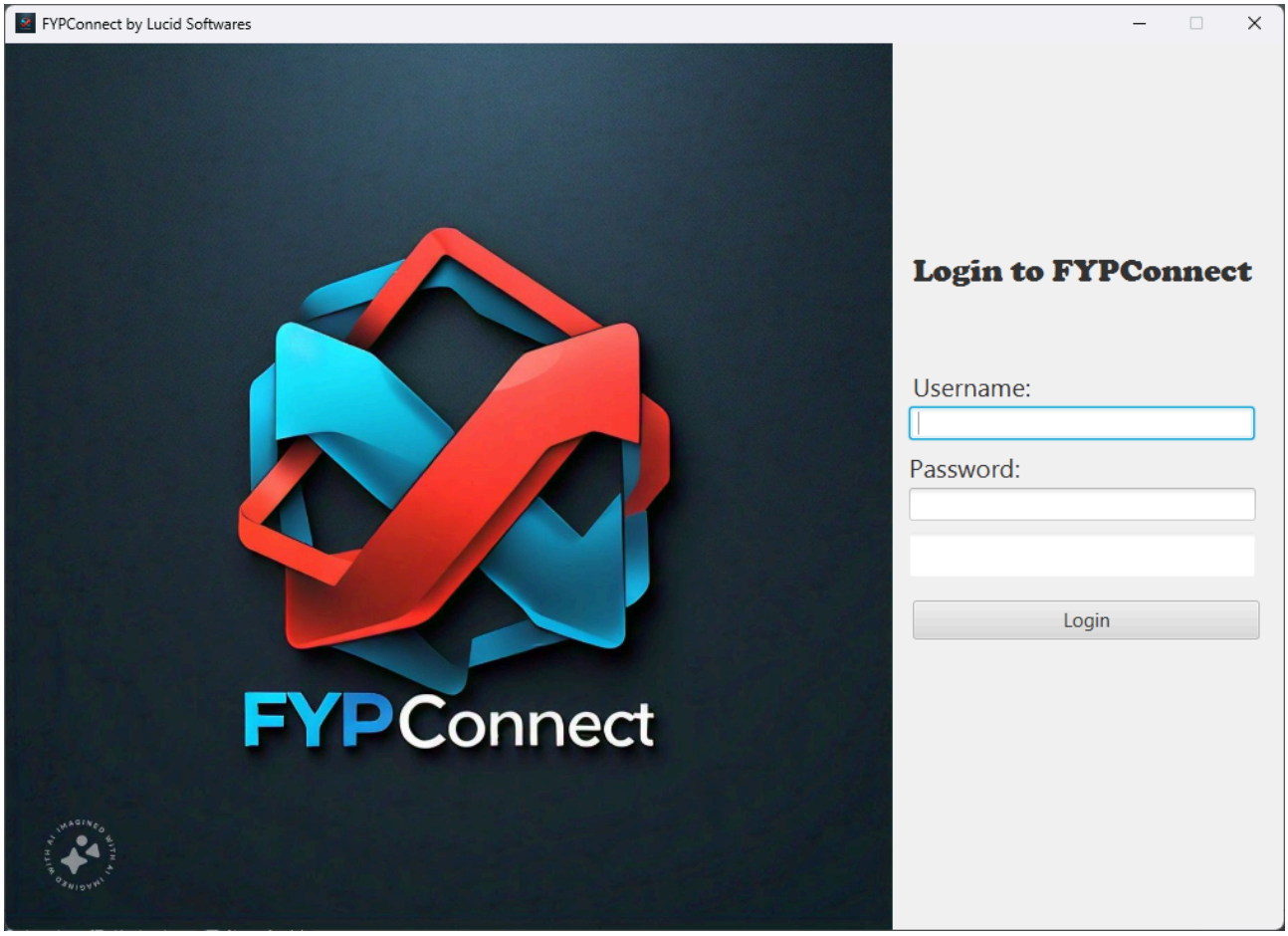
This project aims to address these challenges by developing a straightforward desktop application using JavaFX. The platform will bring all FYP management into one place, automate repetitive tasks, and make it easier to track progress and communicate. This will save time, minimize mistakes, and ensure a smoother process for everyone involved.

2. Overall Description

2.1 Product Perspective

The FYP Connect application is a fresh, standalone solution built to manage Final Year Projects at academic institutions such as FAST University. It is not an upgrade or replacement for any current system, but rather aims to fill the gap in project management by providing a centralized, automated, and easy-to-use platform.

While the application operates independently, it can integrate with existing systems like university email or databases to improve communication and data storage. Below is a high-level diagram that shows how the system's components interact with each other.



2.2 Product Functions

The main features of the FYP Connect application include:

- **Proposal Management:** Students can submit project proposals, and supervisors can review and approve them.
- **Progress Tracking:** Supervisors and admins can monitor milestones, deadlines, and project updates.
- **Document Handling:** A centralized space for uploading and accessing project documents.
- **Admin Controls:** Admins can manage users and system settings.
- **Communication Platform:** A seamless way for students, supervisors, and admins to stay in touch and share updates.

2.3 List of Use Cases

1. Create Groups
2. Submit Deliverables
3. Group Request
4. Join Groups
5. Search Functionality
6. Feedback Sharing
7. Mentorship Request
8. Resource Sharing
9. Supervise Projects
10. Set Timelines/Deadlines
11. Evaluate/Grade Students

2.4 Extended Use Cases

Use Case Name: Manage Users

(Shayan Memon – 22i-0773)

Scope

The system under design is the **FYP Connect** platform, used by administrators to manage user accounts (students and supervisors) for the Final Year Project process.

Level

User-goal level

Primary Actor

Admin

Stakeholders and Interests

- *Admin:* Wants to efficiently manage user accounts (students and supervisors) by creating, updating, and removing access as needed.
- *Students:* Expect active accounts to access FYP resources and collaborate with supervisors.
- *Supervisors:* Require active accounts to guide students and track progress.

Preconditions

- The admin is logged into the system.
- The admin has permission to manage users.

Postconditions

- The user account (student or supervisor) is successfully created, updated, or deleted.
- The correct role (e.g., student, supervisor, admin) is assigned to the user.
- All actions performed (like adding, updating, or deleting users) are saved in the system for record-keeping.
- Users are notified (by email or system message) if their account has been created or changed.

Main Success Scenario

Actor Actions	System Responsibility
1. Admin logs into the FYP Connect system and accesses "Manage Users."	2. System verifies login credentials and provides access.
3. Admin selects an option to add, update, or delete users.	4. System displays options for managing users (add, update, delete).
5. For adding, admin enters user details and submits.	6. System validates the user details.

7. System validates and creates the user account.	8. System creates the user account and assigns the correct role.
9. For updating, admin selects a user and modifies details.	10. System updates user information in the database.

11. System updates user information.	12. System logs the update for record-keeping.
13. For deleting, admin selects a user and confirms deletion.	14. System deletes the account, logs the action, and notifies the relevant user(s).

Extensions

- *3a. Invalid Input:* If the admin enters invalid or incomplete details during user creation, the system shows an error message and prompts for correction.
- *5a. User Not Found*: If the admin tries to update a user who does not exist, the system displays a "User not found" error.
- *7a. Deletion Blocked:* If the admin attempts to delete a user involved in active projects, the system blocks the deletion and provides a warning message.

Use Case Name: Supervise Projects

(Shayan Memon – 22i-0773)

Scope

The system under design is the **FYP Connect** platform, which enables supervisors to manage and supervise Final Year Projects (FYPs), from proposal submission to project completion.

Level

User-goal level

Primary Actor

Supervisor

Stakeholders and Interests

- *Supervisor:* Wants to efficiently upload project proposals, receive and manage student requests, approve students, and monitor project progress.
- *Students:* Want to view, request, and work on project proposals while receiving guidance from the supervisor.
- *Admin:* Ensures that supervisors manage student projects effectively and fairly.

Preconditions

- The supervisor is logged into the system.
- The supervisor has the necessary permission to upload and manage project proposals.
- Students are registered on the platform and can access available project proposals.

Postconditions

- The project proposal is uploaded and visible to students.
- Students are either accepted or rejected for the project, and they are notified.
- The supervisor is assigned to the project and begins overseeing it.
- The project has clear milestones, and progress is tracked.
- All actions and communications are saved for future reference.

Main Success Scenario

Actor Actions	System Responsibility
1. The supervisor logs into the FYP Connect platform.	2. System verifies login credentials and provides access.
3. The system shows an option to "Upload Project”	4. System displays the upload option for project
5. The supervisor uploads a project proposal, providing necessary details (title, description,requirements, etc.).	6. System validates the proposal details and checks for completeness.
7. The system validates and posts the proposal for students to view.	8. System publishes the proposal, making it available to students.
9. Students submit requests to the supervisor for project approval.	10. System collects student requests and notifies the supervisor.
11. The supervisor reviews the requests.	12. System presents the requests to the supervisor for review.

13. The supervisor accepts or rejects student requests.	14. System processes the acceptance or rejection and notifies the student.
15. Upon accepting a student, the supervisor is assigned to the project.	16. System links the supervisor to the project and assigns students.
17. The supervisor sets milestones, schedules meetings, and begins project supervision.	18. System tracks milestones, schedules, and updates meetings.
19. The supervisor monitors project progress through regular updates and meetings.	20. System logs project updates and facilitates communication.
21. All actions and communications related to the project are logged for future reference.	22. System saves all logs and records for future access.

Extensions

- *3a. Incomplete Proposal:* If the proposal information is incomplete or invalid, the system prompts the supervisor to correct and resubmit it.
- *7b. Reject Request:* If a student request is rejected, the student is notified, and the project remains open for other students to apply.
- *10a. Project Delays:* If the student fails to meet milestones or project deadlines, the supervisor can adjust deadlines or deduct student's marks.

Use Case Name: Set Timelines/Deadlines

(Shayan Memon – 22i-0773)

Scope

The system under design is the **FYP Connect** platform, used to set deadlines for student project proposals within the Final Year Project (FYP) management system.

Level

User-goal level

Primary Actor

Supervisor

Stakeholders and Interests

- *Supervisor:* Needs to set deadlines to ensure students submit project proposals on time.
- *Students*: Need clear deadlines to meet submission requirements.
- *Admin:* Ensures deadlines align with the overall project schedule.

Preconditions

- The supervisor is logged into the system.
- The supervisor has a project assigned or created that requires setting deadlines.
- Students are enrolled and assigned to the supervisor.

Postconditions

- Deadlines for project proposals are set successfully.
- Students are notified of the deadlines.
- The system tracks deadlines and sends alerts when submission dates are near.

Main Success Scenario

Actor Actions	System Responsibility
1. The supervisor logs into the FYP Connect platform.	2. System verifies login credentials and provides access.
3. The system shows the supervisor a list of their projects.	4. System displays the list of projects assigned to the supervisor.
5. The supervisor selects a project and chooses to set a deadline for proposal submission.	6. System prepares the deadline-setting interface for the selected project.
7. The system prompts the supervisor to choose a deadline date and time.	8. System provides date and time selection options.
9. The supervisor sets the deadline and submits it.	10. System validates the input and accepts the deadline.

11. The system checks that the deadline fits the overall project schedule.	12. System ensures that the deadline aligns with the FYP schedule.
13. A confirmation message is displayed, and students are notified of the new deadline.	14. System confirms the deadline, notifies students, and logs the action.
15. The system sends reminders to students as the deadline approaches.	16. System schedules and sends reminders to students.
17. The supervisor can view or edit the deadlines anytime.	18. System allows viewing and modification of deadlines.
19. The system logs the action for future reference.	20. System records the action for audit and tracking.

Extensions

- *7a. Notification Error:* If a student doesn't get notified, the supervisor can send a manual reminder or ask the admin to fix the issue.
- *Sa. Deadline Extension:* If extra time is needed, the supervisor can extend the deadline, and students will be informed.

Use Case Name: Evaluate/Grade Students

(Shayan Memon – 22i-0773)

Scope

The system under design is the **FYP Connect** platform, used to evaluate and grade students' project deliverables within the final year project management system.

Level

User-goal level

Primary Actor

Supervisor

Stakeholders and Interests

- *Supervisor:* Needs to evaluate students' project deliverables and assign grades based on the quality of work and progress made.
- *Students:* Expect fair and timely evaluation and feedback on their submitted deliverables.
- *Admin:* Monitors grading and ensures that evaluations are consistent across all projects.

Preconditions

- The supervisor is logged into the system.
- Students have submitted their project deliverables for evaluation.
- The supervisor has access to the grading criteria for the project deliverables.

Postconditions:

- Project deliverables are evaluated and graded.
- Feedback is provided to students.
- Grades are stored in the system for record-keeping, and both the students and admin are notified.

Main Success Scenario:

s

Actor Actions	System Responsibility
1. The supervisor accesses the "Evaluate/Grade Deliverables" section in the FYP Connect system.	2. System verifies the supervisor's credentials and grant access to the grading section.
3. The system displays a list of students who have submitted project deliverables.	4. System fetches and displays the list of submitted deliverables.
5. The supervisor selects a student's project deliverable for evaluation.	6. System loads the selected deliverable for review.
7. The system presents the deliverable along with the grading rubric/criteria.	8. System shows the grading rubric/criteria alongside the deliverable for easy reference.

9.The supervisor reviews the deliverable and assigns a grade based on the criteria.	10. System provides an interface for the supervisor to enter grades according to the rubric.
11. The supervisor provides feedback and comments on the deliverable.	12. System allows the supervisor to enter feedback and additional comments.
13. The grade and feedback are submitted, and the system records the evaluation.	14. System stores the evaluation, grades, and feedback, marking the process as complete.
15. The student is notified of their grade and feedback.	16. System sends a notification to the student about their evaluation and makes the feedback available to them.
17. The system logs the evaluation for future reference and reporting.	18. System records the evaluation data for audit and reporting purposes.

Extensions:

- *5a. Incomplete Deliverable:* If the student's project deliverable is incomplete, the supervisor can reject the submission, and the system notifies the student to resubmit with corrections.
- *7a. Grade Dispute:* If a student disputes their grade, the system allows them to submit a formal request for review, which the supervisor and admin can access for reconsideration.

Use Case Name: Submit Deliverables

22i-0789 M. Zubair Adnan

Scope:

The system under design is the FYP Connect platform, used by students to create project groups for the final year project.

Level:

User-goal level

Primary Actor:

Student

Stakeholder and Interests:

- *Students:* Need to submit project deliverables on time for evaluation and grading.
- *Supervisors:* Expect timely submissions for evaluation.
- *Admin:* Monitors submission deadlines and integrity.

Preconditions:

- Student is logged into the system.
- Student must be a part of a group.
- The submission window is open.
- The student has the completed deliverable.

Postconditions:

- Deliverable is successfully submitted.
- Deliverable is stored for future reference and grading.
- Supervisor is can now evaluate it.

Main Success Scenario:

Actor Actions	System Responsibility
1. Student logs into the FYP Connect system.	2. System verifies login credentials and provides access.
3. Student navigates to the "Submit Deliverables" section.	4. System displays the list of pending deliverables and corresponding deadlines.
5. Student selects the deliverable to submit.	6. System prepares the selected deliverable for upload.
7. Student uploads the project file and fills out required details (e.g., title, description).	8. System validates the file type and size and checks if all required fields are filled.

9. Student submits the deliverable.	10. System processes the submission, stores the file, and marks the submission as complete.
11. Student sees the confirmation of successful submission.	12. System tracks the submission for future reference and evaluation.
13. System displays a confirmation message and sends a notification to the supervisor.	

Extensions:

- *4a*. If the file does not meet the format or size requirements; the system displays an error message and prompts for correction.
- *ca.* If the student submits after the deadline, the system flags the submission as late and notifies both the student and the supervisor.
- *7a.* If there is a technical failure, the system notifies the student and provides troubleshooting options.

Use Case Name: Resource Sharing

22i-0789 M. Zubair Adnan

Scope:

The system under design is the FYP Connect platform, used by students to create project groups for the final year project.

Level:

User-goal level

Primary Actor:

Supervisor

Stakeholders and Interests:

- *Supervisors:* Need to share relevant templates and resources with students for their project work.
- *Students:* Need access to the provided templates to complete their project deliverables.
- *Admin*: Monitors template uploads to ensure compliance with university guidelines.

Preconditions:

- Supervisor is logged into the system.
- Templates are prepared and ready for upload.
- Students have access to the platform to download templates.

Postconditions:

- Templates are uploaded and accessible to students.
- Students are able to download the templates and use them for their project work.

Main Success Scenario:

Actor Action	System Responsibility
1. Supervisor logs into the FYP Connect system.	2. System verifies login credentials and provides access to supervisor dashboard.
3. Supervisor navigates to the "Resource Sharing" or "Upload Templates" section.	4. System displays the option to upload new templates.
5. Supervisor selects the file(s) to upload (templates).	6. System prompts for details like title, description, and relevant project sections.
7. Supervisor uploads the template(s).	8. System validates file type and size and stores the templates.
9. Supervisor sees confirmation of successful upload.	10. System displays a confirmation message and makes the uploaded template accessible to students.
11. Student logs into the system and navigates to the "Templates" section.	12. System lists the uploaded templates for students to download.

13. Student selects and downloads the required template.	14. System provides the download link and tracks the action.

Extensions:

- *4a.* If the template file doesn't meet format or size requirements, the system displays an error and prompts for a valid file.
- *5a.* If the upload fails due to technical issues, the system notifies the supervisor and provides troubleshooting options or support.
- *7a.* If the download fails, the system alerts the student and provides troubleshooting steps or a contact for support.

Use Case Name: Search Functionality

22i-0789 M. Zubair Adnan

Scope

Level: User-goal level

Primary Actor

Student

Stakeholder and Interests

- *Students:* Want to efficiently find groups, templates, and resources for their projects.
- *Supervisors:* Expect students to locate shared resources.
- *Admin:* Ensures the search functionality is responsive and accurate.

Preconditions

- Student is logged into the system.
- Relevant resources (groups, templates) exist in the system.
- The search engine is operational.

Postconditions:

- The student successfully finds the resource.
- The student can access or download the resource directly from the search results.

Main Success Scenario

Actor Actions	System Responsibilities
1. Student logs into the FYP Connect system.	2. System verifies login and provides access.
3. Student navigates to the "Search" bar.	4. System displays the search bar and prompts for input.
5. Student enters a search query (e.g., group name, template) and submits it.	6. System processes the search query.
7. –	8. System displays a list of relevant search results (e.g., groups, templates, resources).
9. Student selects the desired result from the list.	10. System directs the student to the chosen resource (e.g., group page, template download).
11. Student accesses or downloads the resource.	12. System provides the resource or download link and tracks the access.

Extensions

- *3a.* If no results are found, the system displays "No results found" and suggests related queries or categories.

Use Case Name: Create Groups

22i-0789 M. Zubair Adnan

Scope:

The system under design is the FYP Connect platform, used by students to create project groups for the final year project.

Level:

User-goal Level

Primary Actor:

Student

Stakeholder and Interests:

- *Students:* Want to create and manage groups for their final year project to collaborate with peers and submit their project proposals.
- *Supervisors:* Need to oversee the groups and ensure that each group is properly formed and assigned to a supervisor.

Preconditions:

- The students are logged into the system.
- The students are registered for the final year project course.
- The student must not be a part of any group before.

Postconditions:

- A project group is successfully created.
- The project group is visible to other students.
- Other students can request to become a part of the group.
- The group members can collaborate and submit project proposals.

Main Success Scenario

Actor Actions	System Responsibilities
1. Student logs into the FYP Connect system.	2. System verifies login and provides access to the student's dashboard.
3. Student enters the group name and other required details.	4. System validates the input for any errors (e.g., duplicate group name).
5. (Optional) Student invites other students to join the group.	6. System sends invitations to the selected students.
7. (Optional) Invited students accept the invitations.	8. System updates the group with accepted members and notifies the group creator.

9. Group is successfully created, and the students can collaborate within the group.	10. System creates the group, marks it as active, and displays it in the group listings.

Extensions

- *2a.* Duplicate Group Name. In case a group already exists with the name input by the student, display an error.
- *3a.* If a student attempts to invite more members than the allowed limit, the system displays an error message and prevents additional invites.

Use Case name: Feedback sharing

(Ahmad Ryan – 22i-0781)

Scope

The system under design is the ***"FYP Connect"*** platform, that will be used by supervisors to track and provide feedback on students' work, and by students to track deliverables, view and respond to feedback regarding their final year project.

Level

User-goal level

Primary Actor

Supervisor (can provide feedback)

Stakeholders and Interests

- *Students:* Want to receive feedback from supervisors to improve their FY project work.
- *Supervisors:* Need to provide feedback on students' work and track their progress to ensure project quality and guide students effectively.
- *Admin / University Management:* Interested in ensuring that proper feedback is provided to the students on their work.

Preconditions

- The supervisor is logged into the system.
- The student has submitted a project deliverable or has a project in progress.
- The feedback option is available within the system for the project.

Postconditions

- Feedback is successfully shared with the student.
- The student can view and respond to the feedback.

- The feedback is logged and tracked for future reference.

Main Success Scenario

Actor Action	System Response
1. The supervisor logs into the FYP Connect system and accesses "Feedback".	2. The system displays a list of students' deliverables or project progress.
3. The supervisor selects a student's project or deliverable to review.	4. The system presents the details of the selected deliverable or project.
5. The supervisor provides feedback by adding comments and suggestions.	6. The system validates the input and allows submission if all required fields are filled.
7. The supervisor submits the feedback.	8. The system stores the feedback, updates the student's project, and sends a notification to the student.
9. The student views the feedback and responds with questions or comments.	10. The system logs the feedback exchange and makes it available for both the supervisor and the student to track in the future.

Extensions

- *5a. Incomplete Feedback:* If the supervisor attempts to submit incomplete or minimal feedback, the system prompts them to provide more detailed comments.

Use case name: Join groups

(Ahmad Ryan – 22i-0781)

Scope

The system under design is the ***"FYP Connect"*** platform, that will be used by supervisors to track and provide feedback on students' work, and by students to track deliverables, view and respond to feedback regarding their final year project.

Level

User-goal level

Primary Actor

Student

Stakeholders and Interests

- *Students:* Want to join existing project groups to collaborate with other students and participate in group for their final year project.
- *Supervisors:* Need to ensure that all students are grouped effectively to promote collaboration and project success.
- *Admin:* Monitors group memberships and ensures compliance with institutional policies regarding group sizes and formations.

Preconditions

- The student is logged into the system.
- The student is registered in the final year project course.
- There are available groups that the student can join.

Postconditions

- The student successfully joins the selected project group.
- The student is added to the group member list and can participate in group activities.
- The group is notified of the new member joining.

Main Success Scenario

Actor Action	**System Response**
1. The student logs into the FYP Connect system and navigates to the "Groups" section.	2. The system displays a list of available groups with details (group name, project topic, and current members) and any invitations to join a group.
3. The student selects a group to join.	4. The system prompts the student to confirm the intention to join the selected group.
5. The student confirms the request to join the group.	6. The system validates the request, checking if the group has not reached its member limit.

7. The already grouped members are notified to confirm the new member's request.	8. Once confirmed, the system adds the student to the group, sends a notification to other group members, and grants access to group resources and communication tools.

Extensions

- *ca. Group Full:* If the group has reached its maximum member limit, the system displays an error message and suggests other available groups.
- *5a. Request Pending*: If the group requires approval to join, the system informs the student that their request is pending and will notify them once approved.

Use case name: Mentorship request

(Ahmad Ryan – 22i-0781)

Scope

The system under design is the ***"FYP Connect"*** platform, that will be used by supervisors to track and provide feedback on students' work, and by students to track deliverables, view and respond to feedback regarding their final year project.

Level

User-goal level

Primary Actor

Student

Stakeholders and Interests

- *Students:* Want to seek mentorship from supervisors to gain guidance and support for their final year projects.
- *Supervisors:* Need to review and respond to mentorship requests from students to provide appropriate support.

Preconditions

- The student is logged into the system.
- The student has identified a supervisor they wish to request mentorship from.
- The supervisor is available for mentorship.

Postconditions

- A mentorship request is successfully sent to the supervisor.
- The supervisor is notified of the request and can accept or decline it.
- The system tracks the status of the mentorship request.

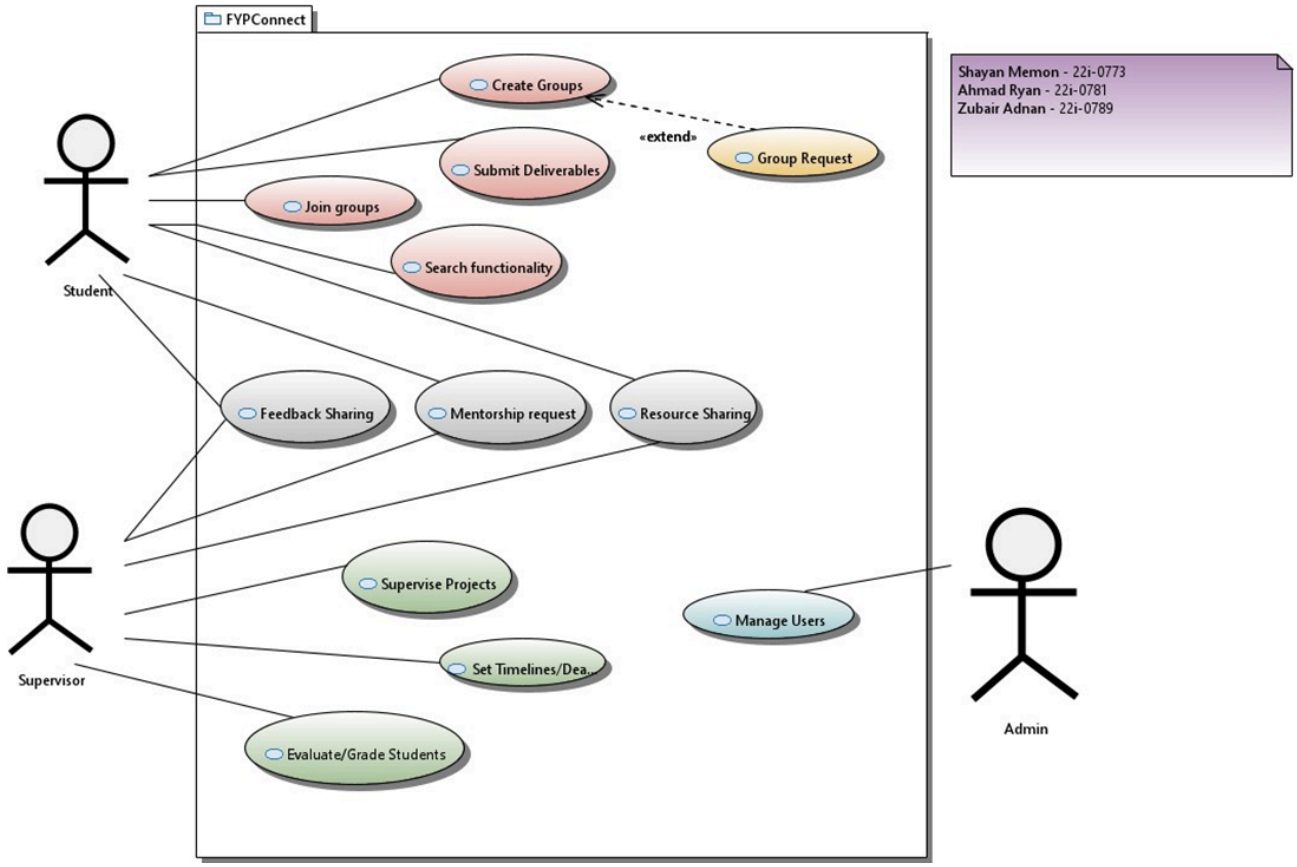
Main Success Scenario

Actor Action	System Response
1. The student logs into the FYP Connect system and navigates to the "Mentorship Request" section.	2. The system displays a list of supervisors available for mentorship.
3. The student selects a supervisor to request mentorship from.	4. The system prompts the student to provide a message outlining their request and reasons for seeking mentorship.
5. The student submits the mentorship request.	6. The system validates the request and sends it to the selected supervisor.
7. The supervisor receives a notification about the mentorship request.	8. The supervisor reviews the request and decides whether to accept or decline it.
9. The system notifies the student of the supervisor's decision.	10. If the request is accepted, the system allows the student to schedule meetings with the supervisor for mentorship sessions.

Extensions

- *4a. Request Information:* If the student fails to provide adequate information in the request, the system prompts them to include more details before submission.
- *8a. Request Expiration:* If the supervisor does not respond to the mentorship request within a certain timeframe, the system automatically notifies the student and allows them to withdraw the request or send a reminder.
- *Sa. Decline Notification:* If the supervisor declines the request, the system informs the student of the decision and provides the option to request mentorship from another supervisor.

2.5 Use Case Diagram



3. Other Nonfunctional Requirements

3.1 Performance Requirements

Response Time: The application should load and respond quickly, completing most actions like submitting deliverables or loading feedback within 3 seconds for the majority of users. This ensures a smooth and hassle-free experience.

System Resource Usage: The app should run efficiently without overloading the CPU or memory, even on standard computers, so it works seamlessly without slowing down the system.

Offline Functionality: Users should be able to perform key tasks like submitting projects and managing feedback offline. The app will sync data automatically when an internet connection is restored.

Real-Time Updates: Any changes (e.g., project status or feedback updates) should be updated locally within **5 seconds** to ensure users are working with the latest data.

3.2 Safety Requirements

Data Integrity: The app should automatically save progress and have recovery options in case of unexpected shutdowns. This will ensure no data is lost and users can pick up right where they left off.

Input Validation: The app should check that all user inputs, like project details or deadlines, are in the correct format. If there's an error, the user should get a clear and helpful message to fix it.

3.3 Security Requirements

- ***User Authentication****: Every user (student, supervisor, or admin) must log in securely with a unique username and password to access the system.*
- ***Role-Based Access****: Features should be limited based on user roles. For instance, students can only manage their own projects, supervisors can view and grade their assigned projects, and admins have full access to user data and system settings.*
- ***Privacy****: The system should protect personal information, like student details and project grades, from unauthorized access. For example, only the assigned supervisor should be able to view and grade a student's project.*

3.4 Software Quality Attributes

Usability: The app should be simple to use for all roles (students, supervisors, admins). Tasks like submitting projects, sharing feedback, or grading should take no more than 3 steps, with clear guidance and error messages when needed.

Maintainability: The code should be modular and well-documented, making it easy to update or fix. Minor changes should take up to 2 weeks, and major updates up to 4 weeks.

Performance: The app should stay fast, with data loading in under 3 seconds, even as the number of users or projects grows.

Portability: The app should run smoothly on Windows, macOS, and Linux with minimal adjustments.

Interoperability: The app should work with other university systems, like email clients or local databases, for seamless communication and data management.

Scalability: It should handle increasing users and projects easily, without requiring significant infrastructure changes.

3.5 Business Rules

Student Role: Students can submit project proposals, request mentorship, join groups, upload deliverables, and track their project's progress. They cannot access or change other students' projects or feedback.

Supervisor Role: Supervisors can review, approve, or reject proposals, oversee project work, provide feedback, assign grades, and set deadlines. They only have access to projects assigned to them.

Admin Role: Admins can manage all users (students and supervisors), set global deadlines, adjust system settings, and view all projects and feedback. They have full access to make system-wide changes, like adding users or updating configurations.

Access Control: Students can only manage their own projects. Supervisors can access and grade their assigned projects. Admins have full access to all projects and user data.

Feedback and Evaluation: Supervisors must give feedback for each student submission and finalize grades before a project is marked as complete. Admins can review feedback and grades but cannot change them.

3.6 Operating Environment

Hardware Platform:

- **Minimum**: Intel i3 or equivalent, 4GB RAM, 500MB free disk space
- **Recommended**: Intel i5 or higher, 8GB RAM, 1GB free disk space

Operating System:

- Windows 10 or later
- macOS 10.12 or later
- Ubuntu 18.04 or later

Software Components:

- Java Development Kit (JDK 8 or higher)
- JavaFX for the user interface
- Local database (e.g., SQLite or H2) for storage
- Git for version control and team collaboration

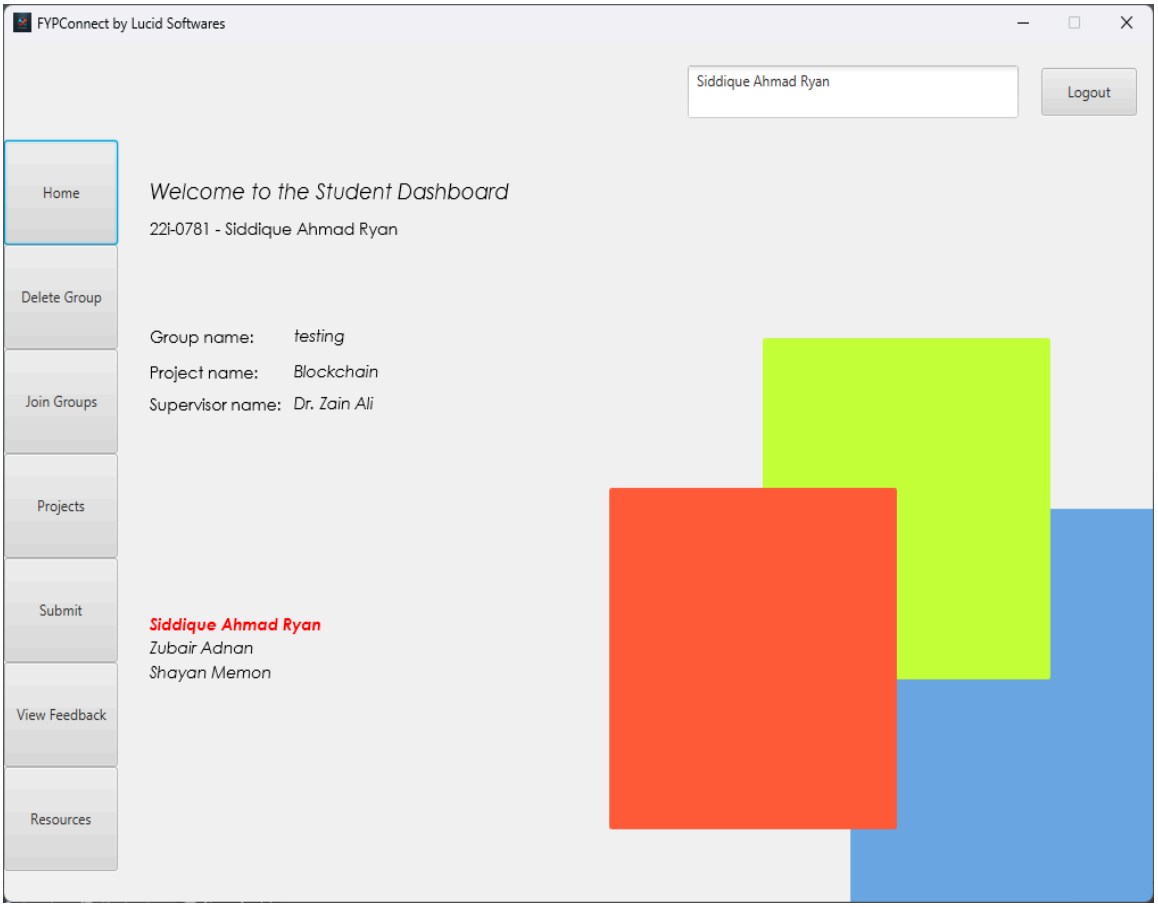
Network Requirements:

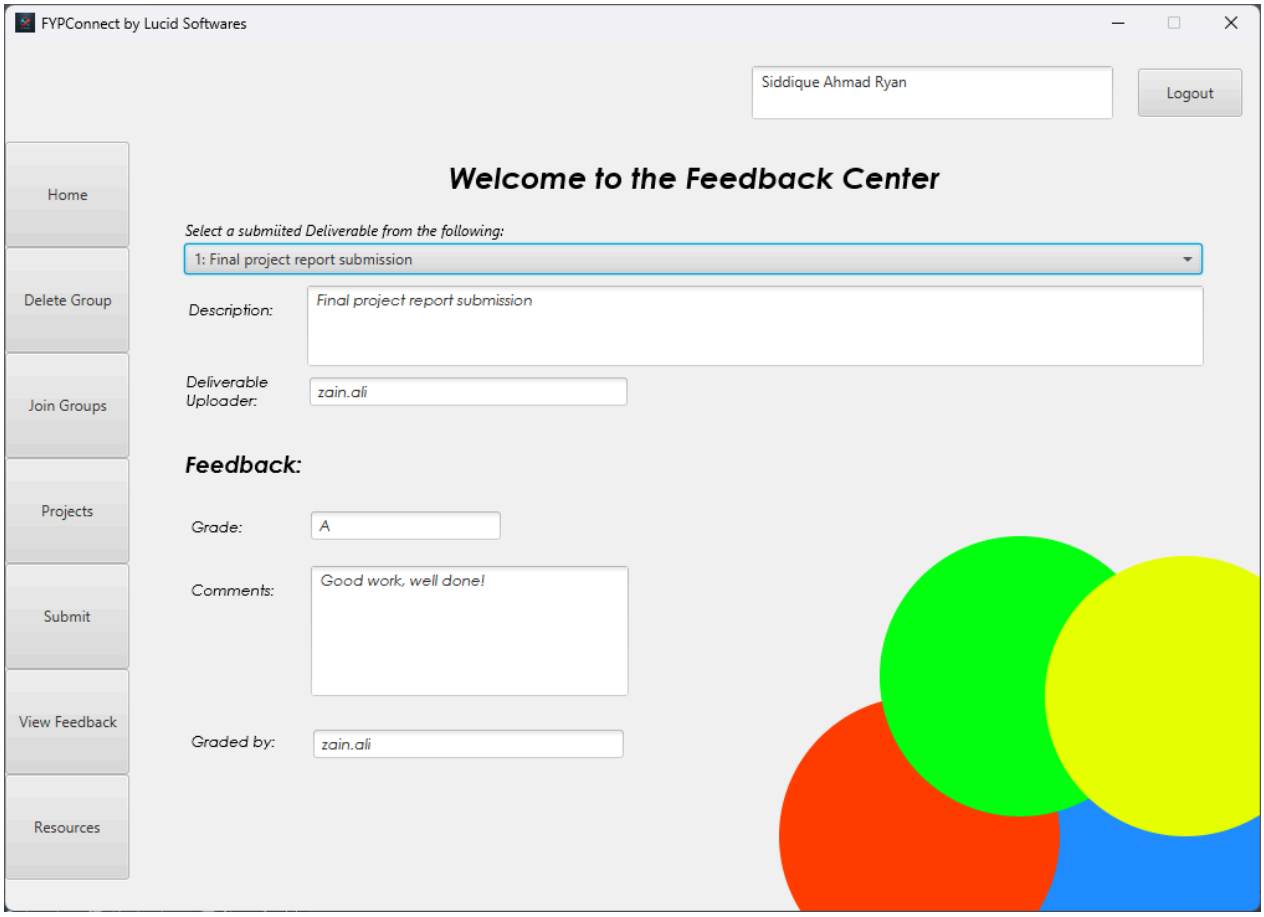
- Local Area Network (LAN) for on-campus deployment
- Internet connection for syncing backups and sending notifications

3.7 User Interfaces

The **FYP Connect** system includes the following user interfaces, designed to ensure a user-friendly experience for students, supervisors, and admins. Below are the key characteristics of each interface:

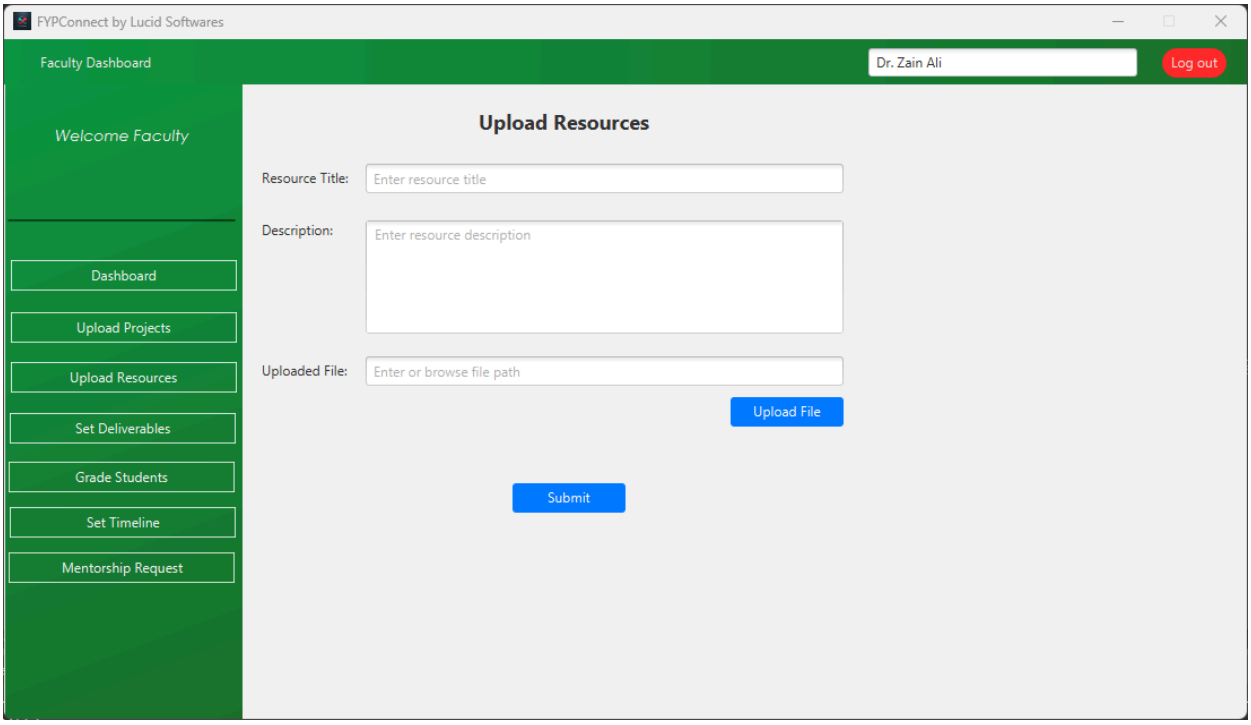
1. Student Interface

- **Purpose**: Allows students to manage group activities, submit deliverables, and communicate with supervisors.
- **Features**:
  - Dashboard displaying project milestones, deadlines, and supervisor feedback.
  - Forms for group creation, deliverable submission, and mentorship requests.
  - Search functionality to browse projects or resources.
  - Buttons for **Submit**, **Cancel**, and **Request Help** visible on key screens.
  - Navigation bar for easy access to core modules (e.g., Groups, Deliverables, Messages).
  - Error messages for invalid file uploads or failed submissions, displayed as pop-ups or banners.
  - 



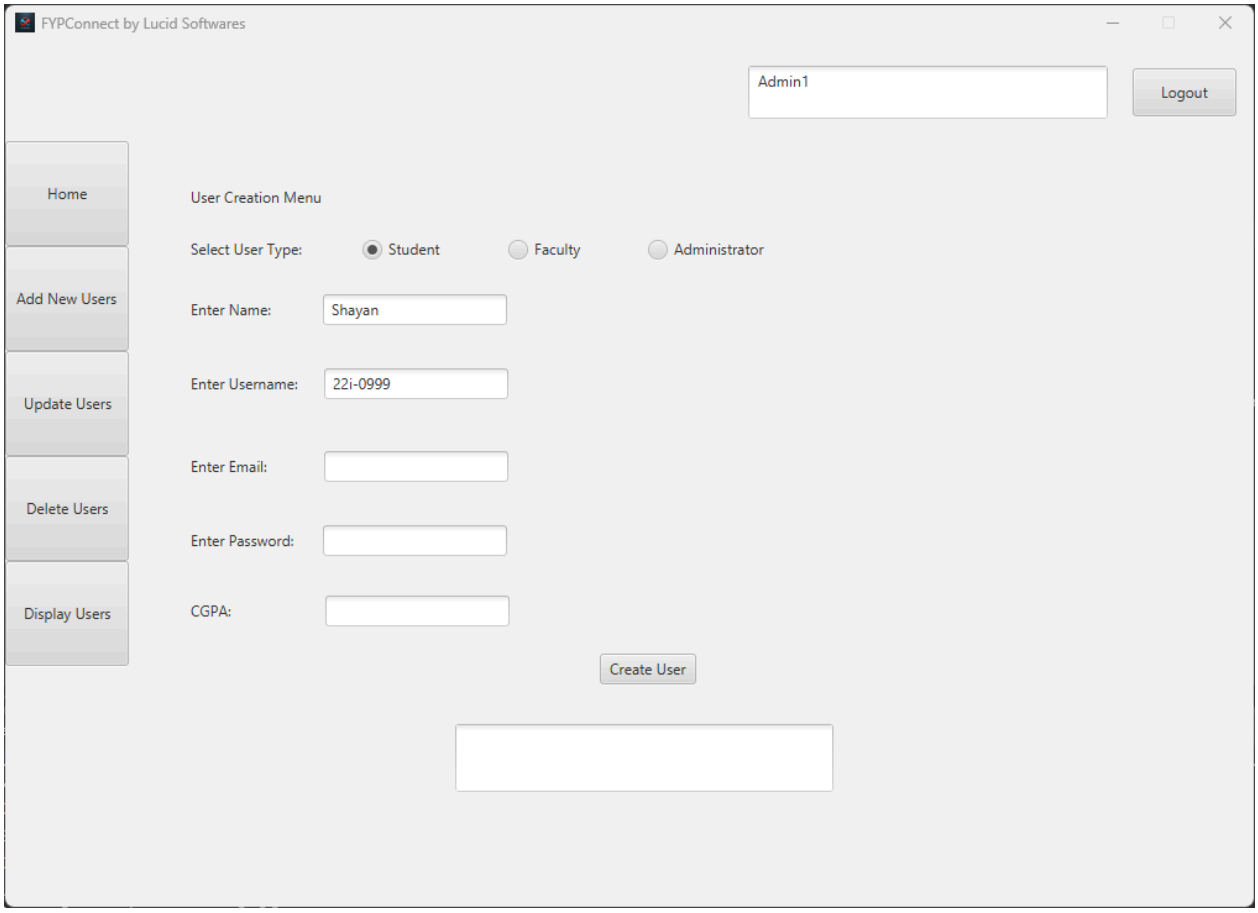
2. Supervisor Interface

- **Purpose**: Enables supervisors to review student progress, provide feedback, and manage timelines.
- **Features**:
    - Dashboard displaying supervised projects, pending approvals, and deadlines.
    - Project review screen with options to evaluate, approve, or return deliverables.
    - A timeline management module with tools to set or update project deadlines.
    - Feedback entry form with pre-defined templates and manual input.
    - Navigation bar and action buttons such as **Approve**, **Reject**, and **Edit Timeline**.

- 

3. Admin Interface

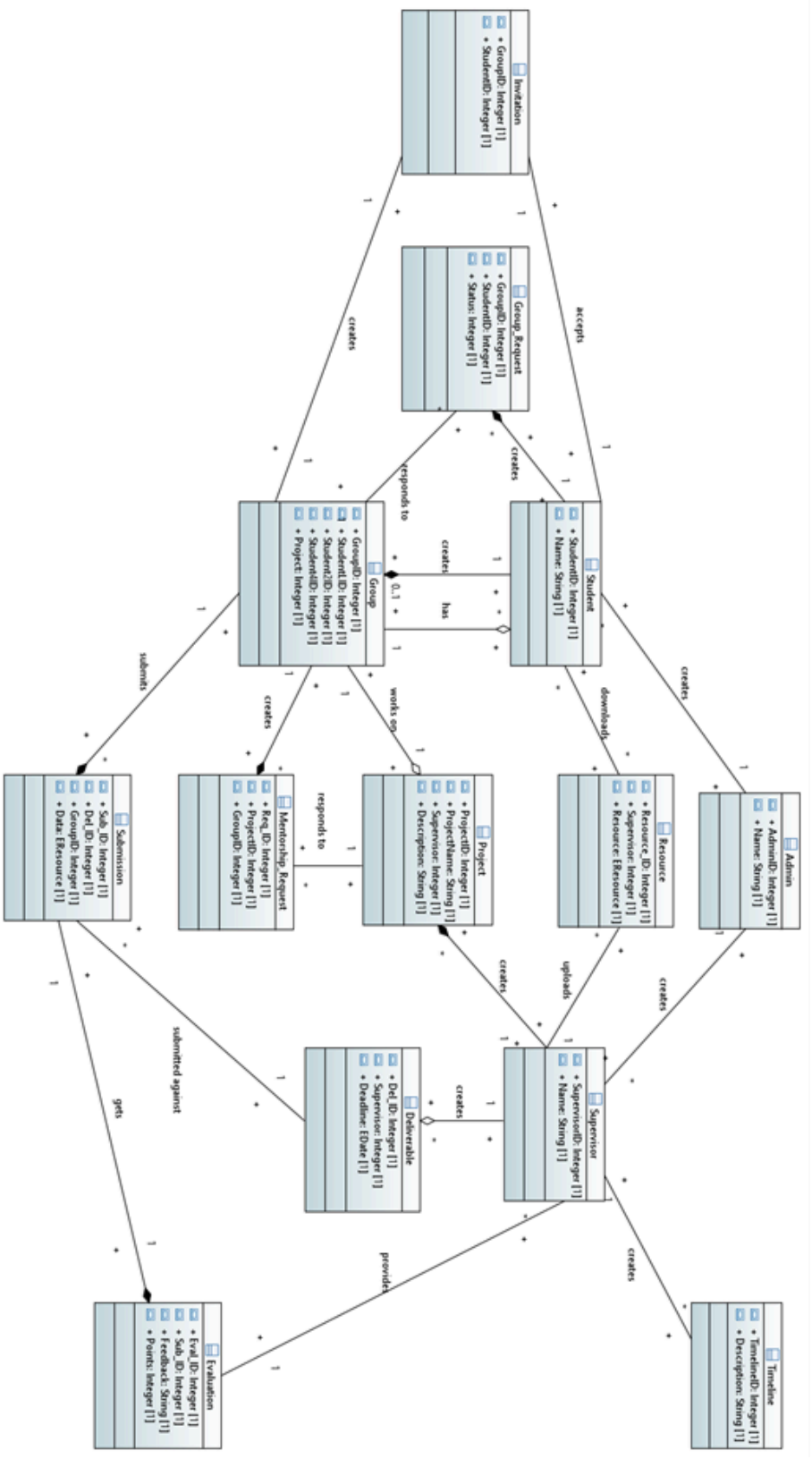
- **Purpose**: Provides admins with control over user management, system settings, and overall project tracking.
- **Features**:
  - User management panel for adding, removing, or updating user accounts.
  - Buttons for **update , delete , add** users.
  - Error-handling messages for invalid inputs or system conflicts.



General Interface Characteristics

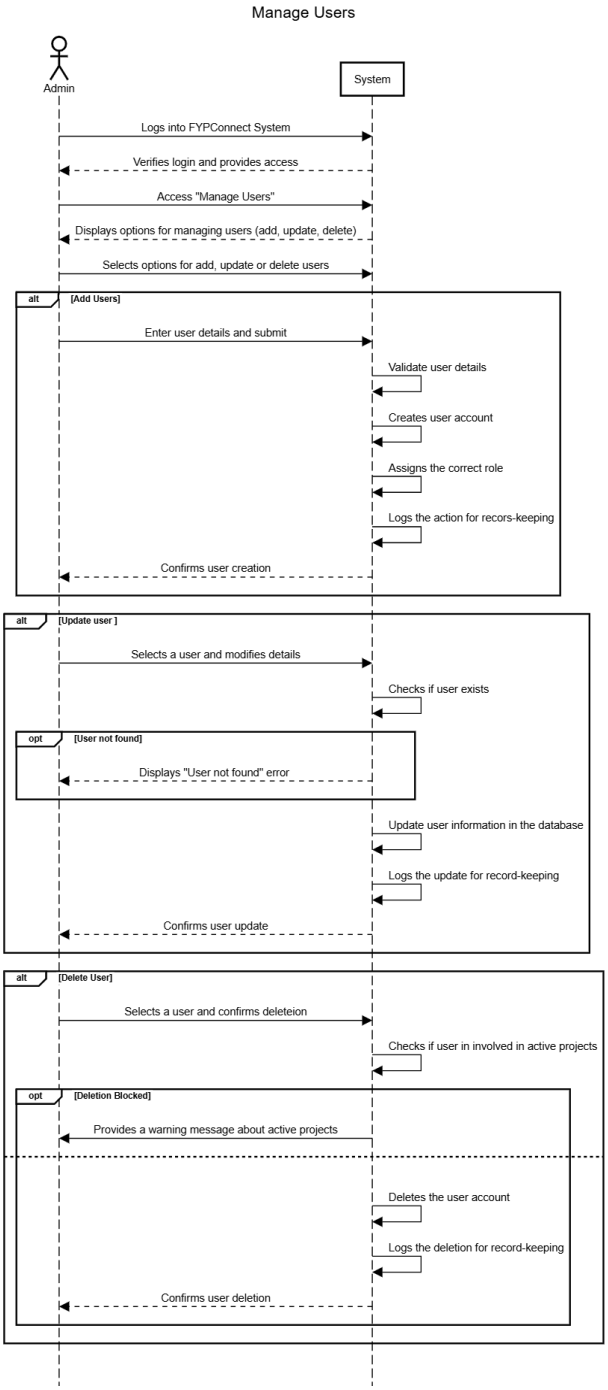
- **Consistent Layout**: All interfaces will follow a consistent design with clear navigation, easy-to-read fonts, and intuitive layouts.
- **Standard Elements**:
    - **Navigation Bar**: Present on every screen for quick access to major modules.
    - **Error Messages**: Clear and concise messages displayed in modals or banners, with instructions to resolve issues.

4. Domain Model



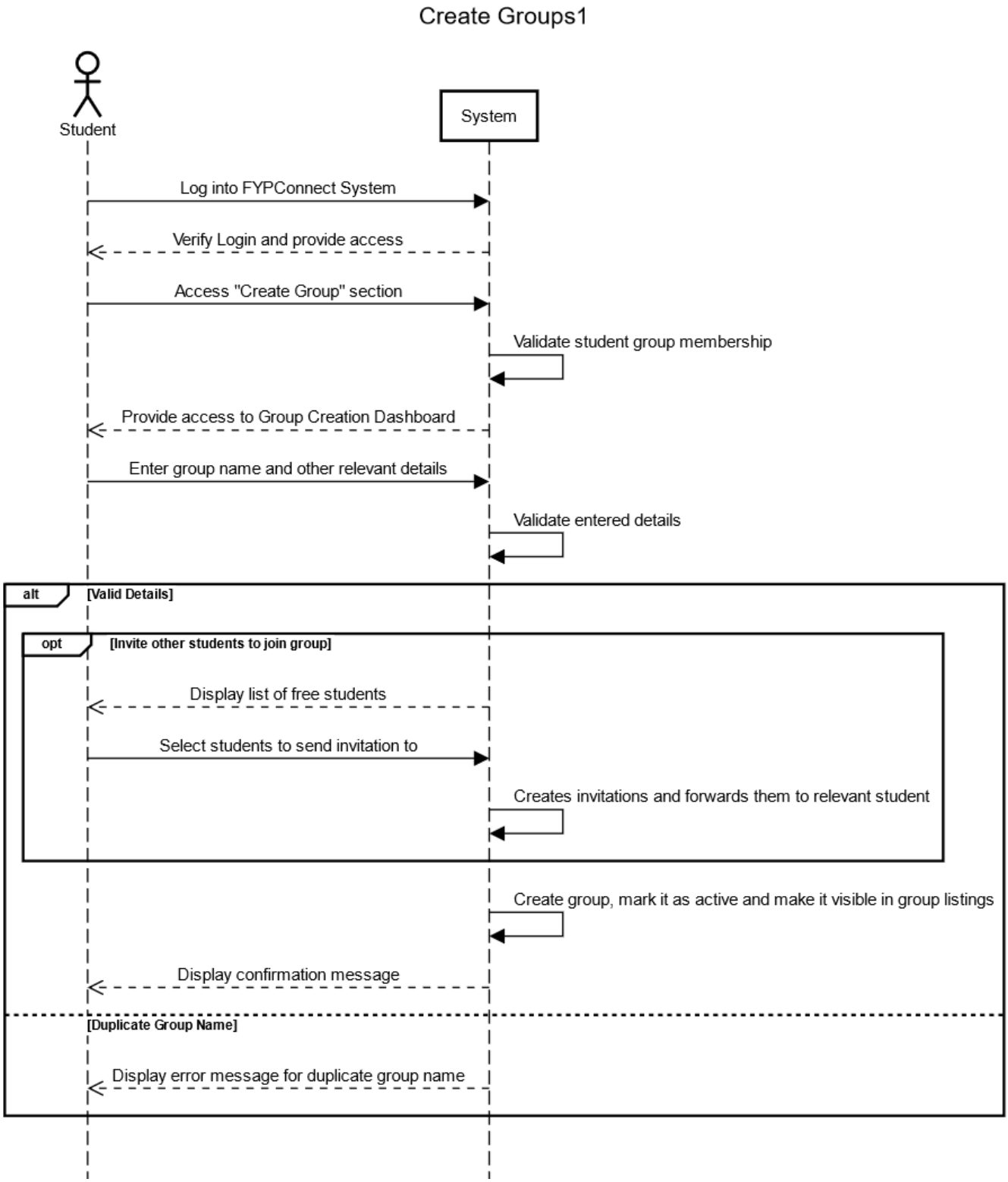
5. System Sequence Diagram

1. Manager Users

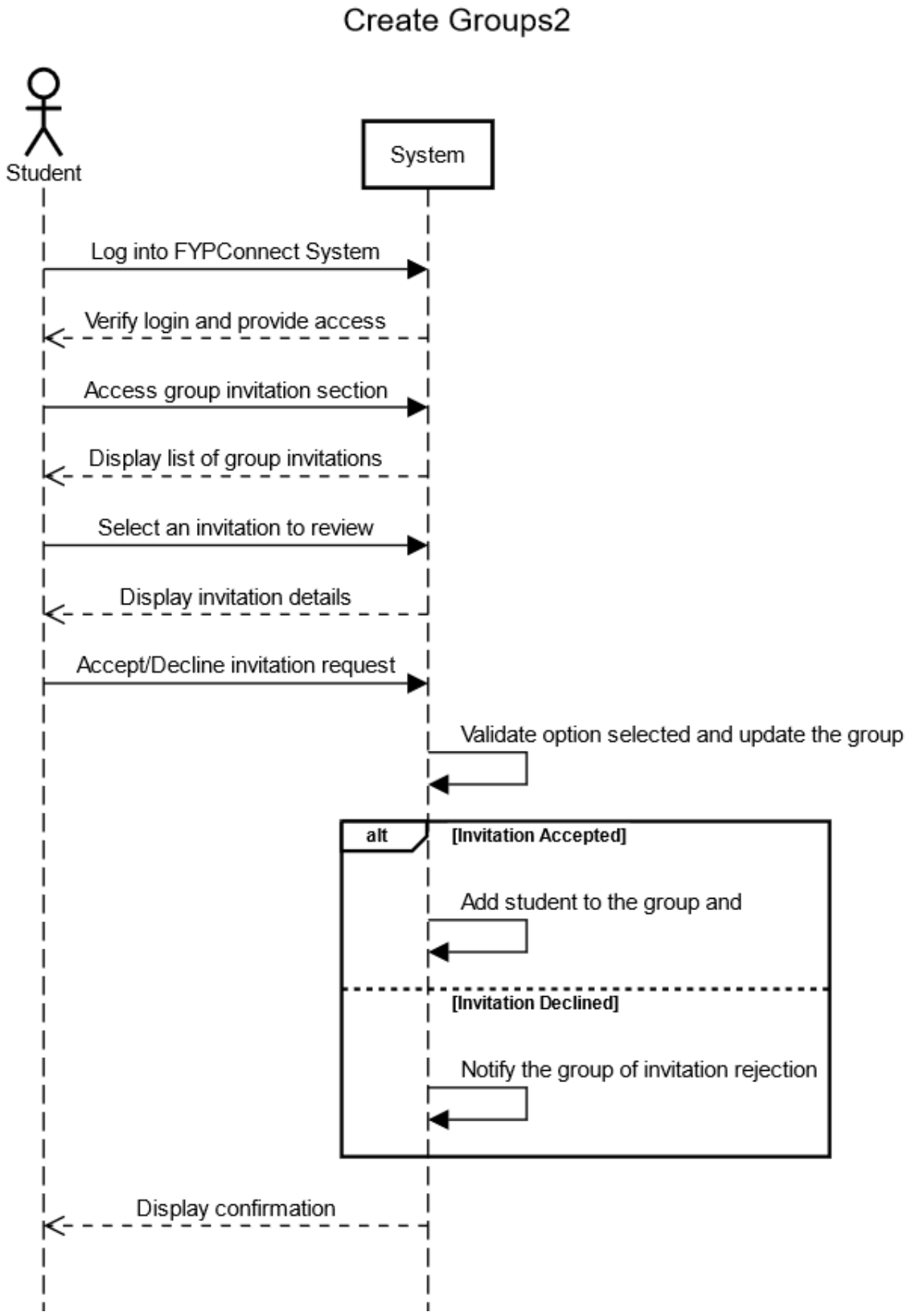


2. Create Groups

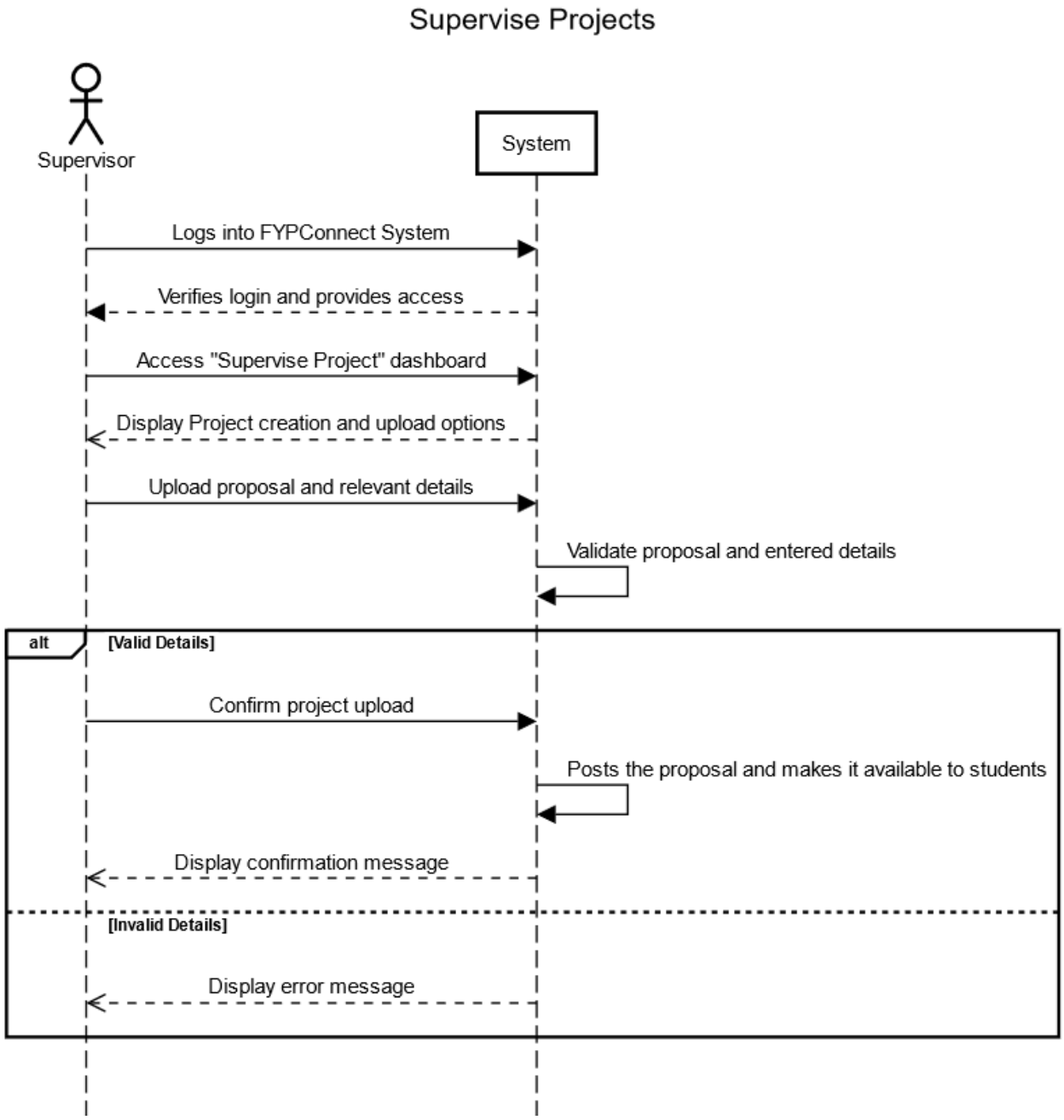
a. Create Groups 1



b. Create Groups 2

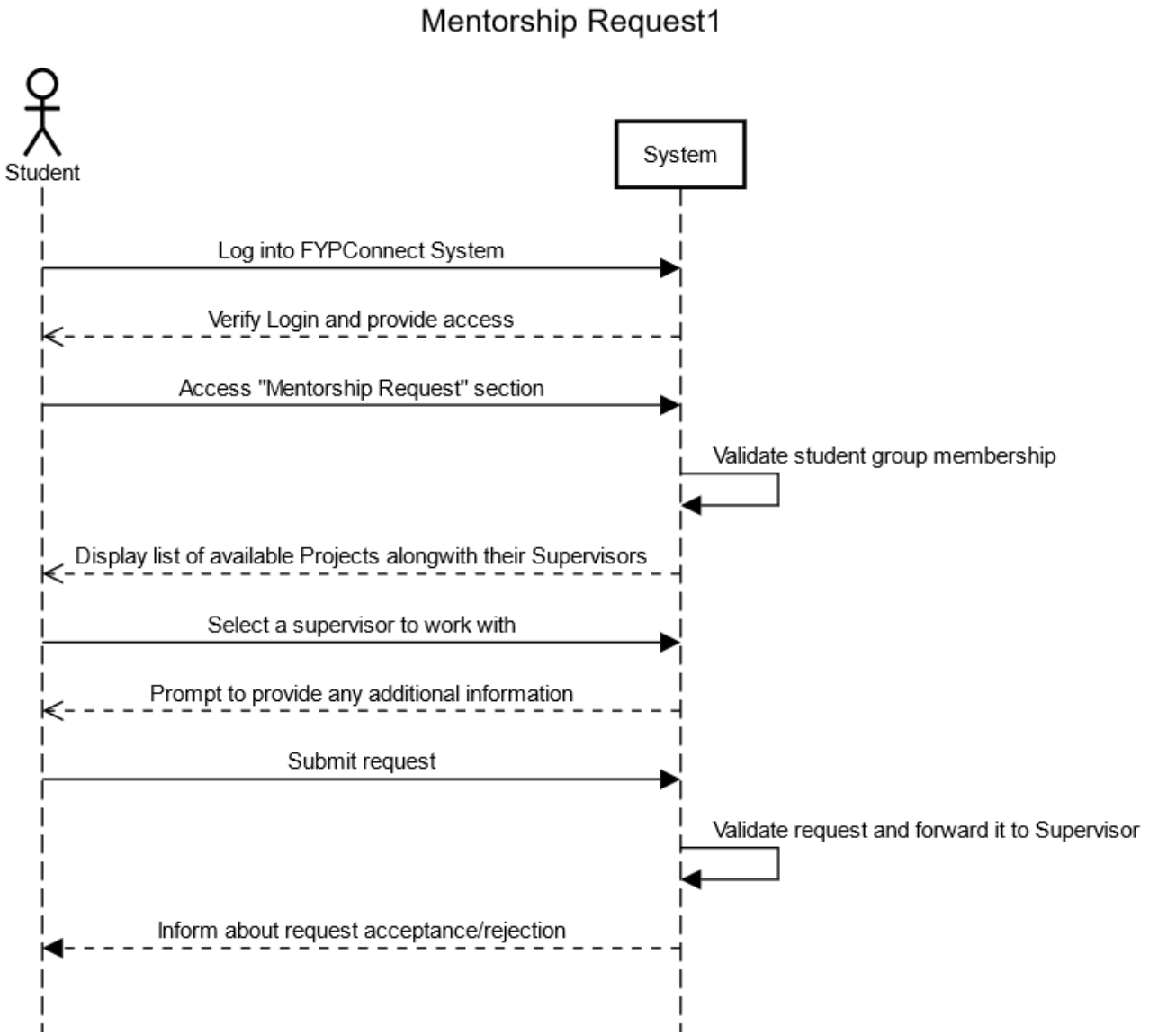


3. Supervise Projects

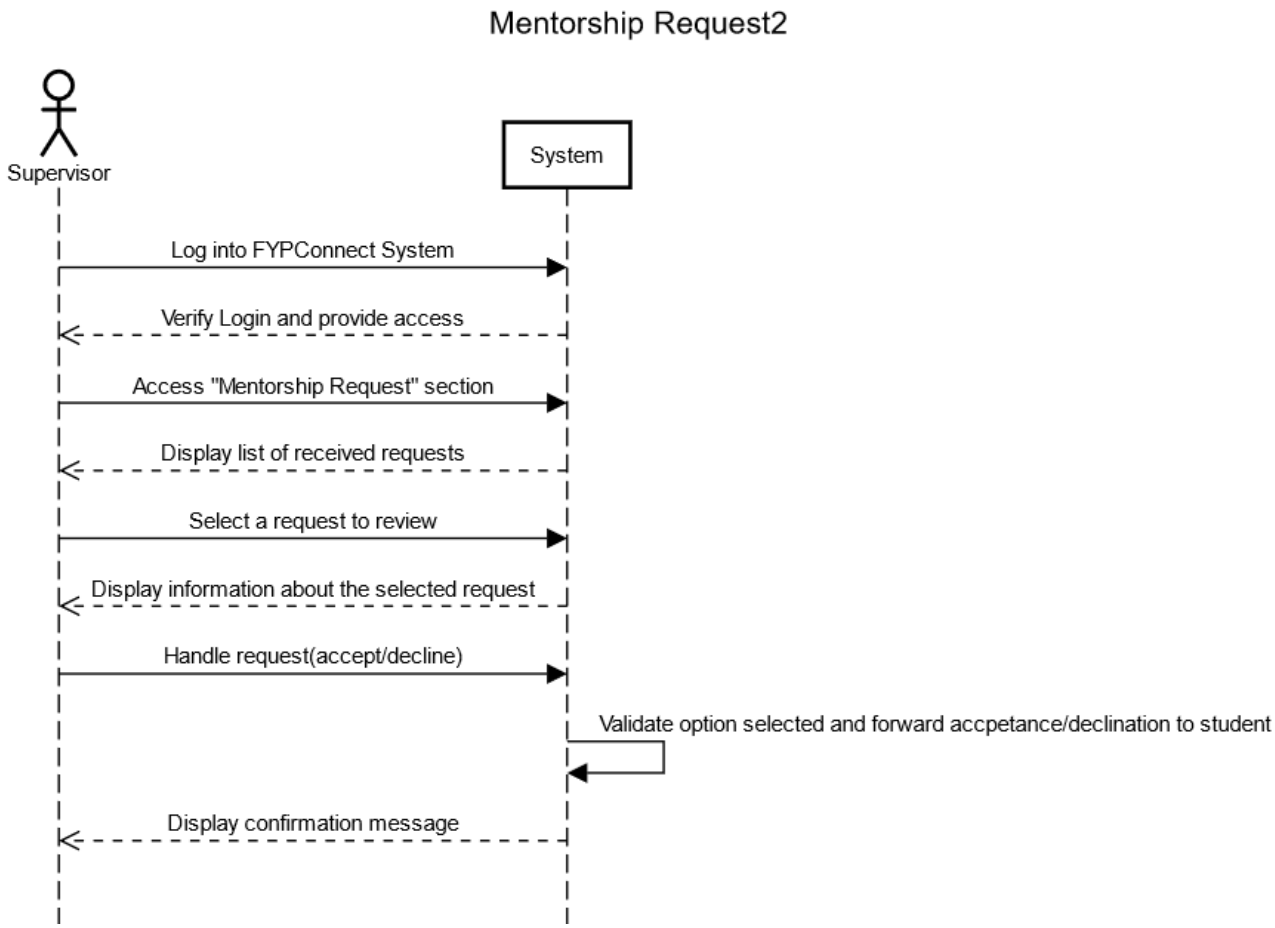


4. Mentorship Requests

a. Mentorship Request 1

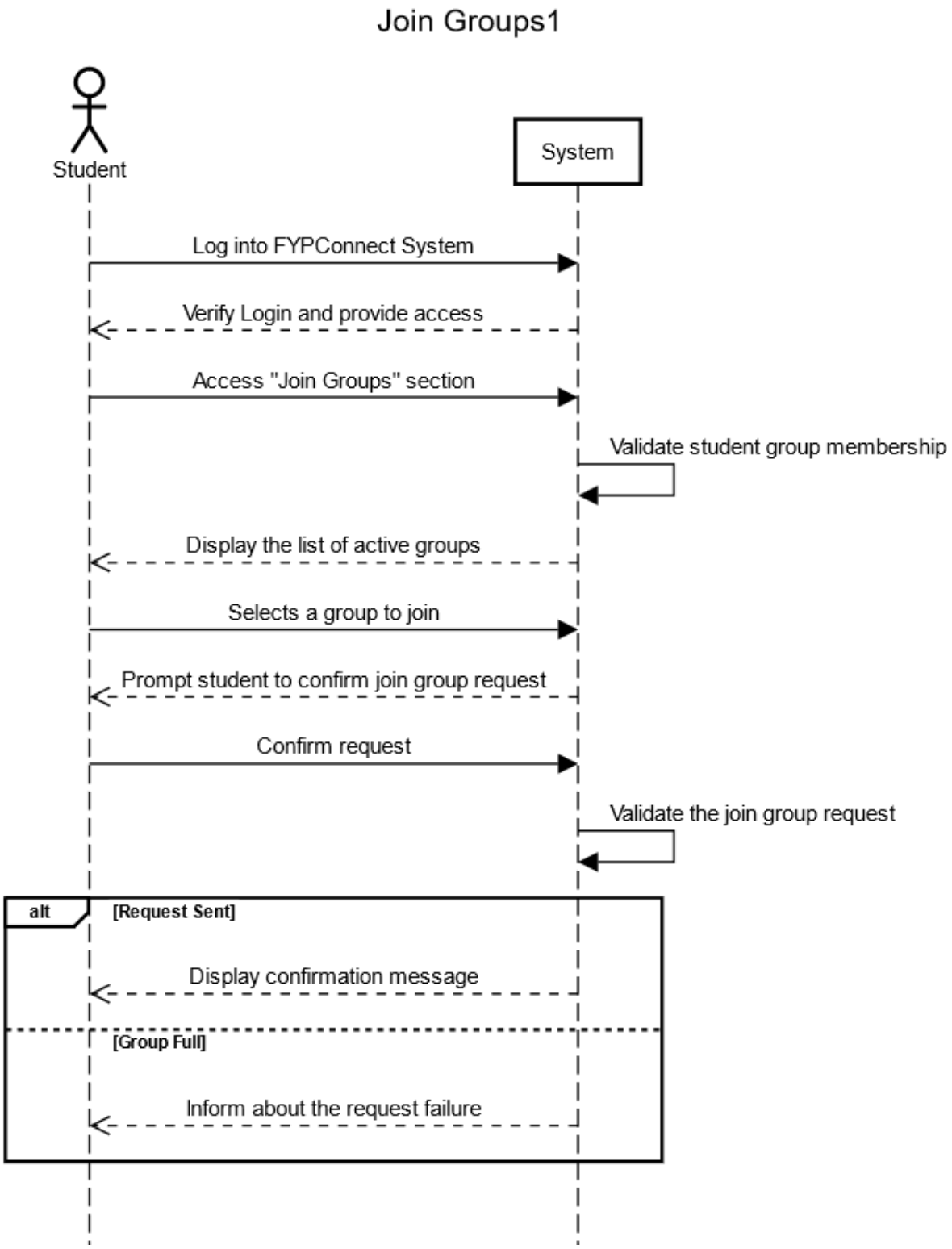


b. Mentorship Request 2

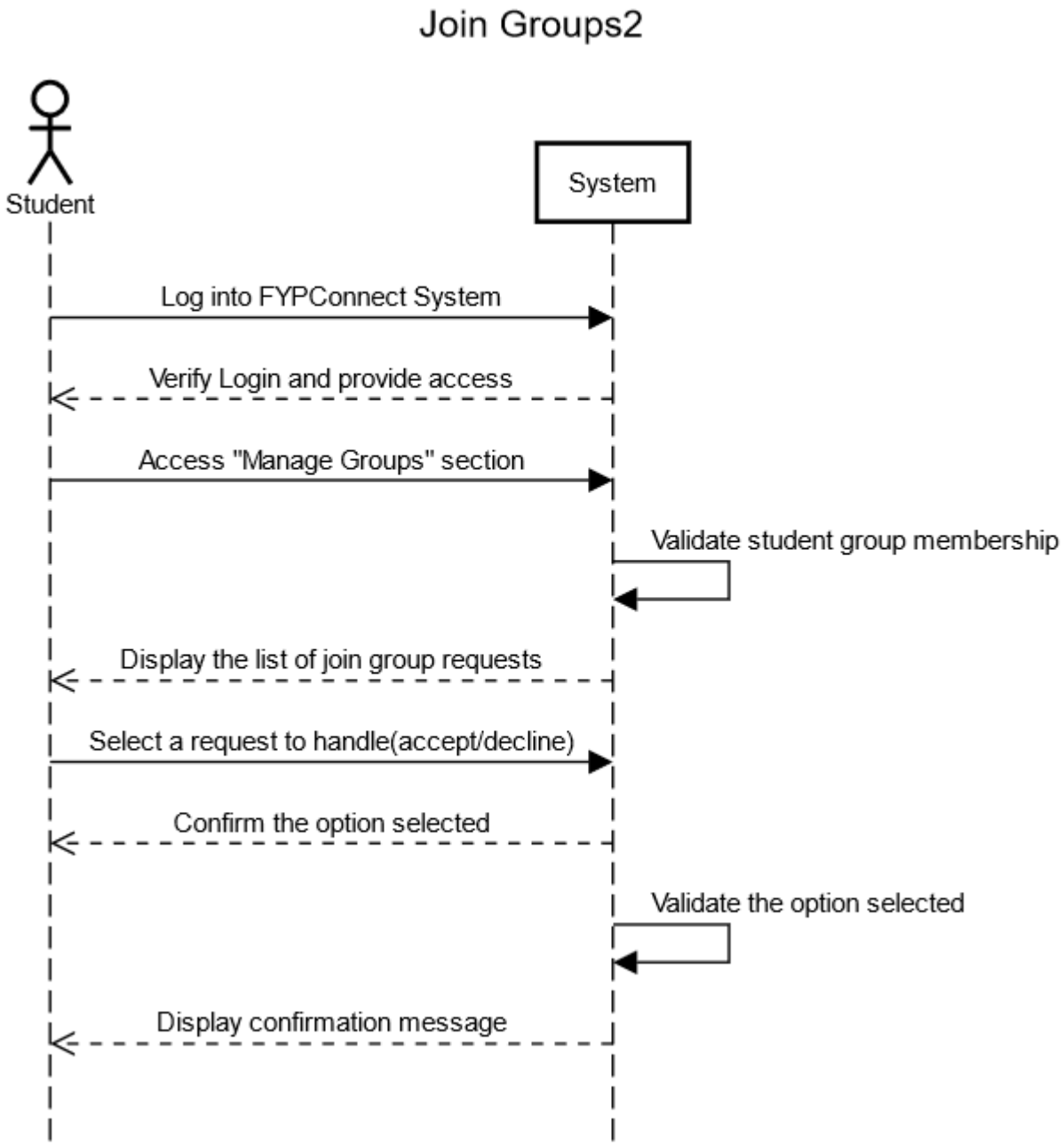


5. Join Groups

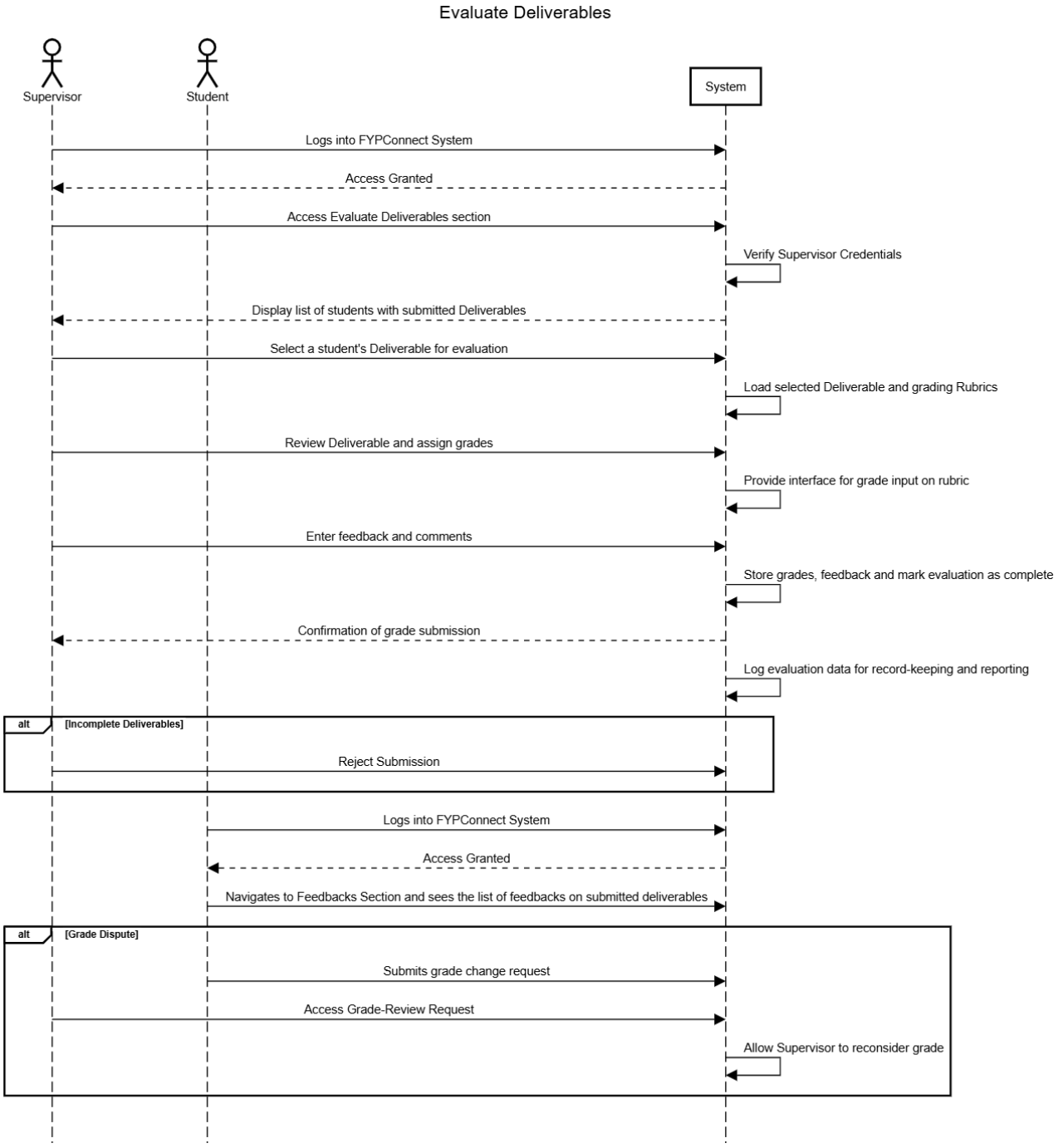
a. Join Group 1



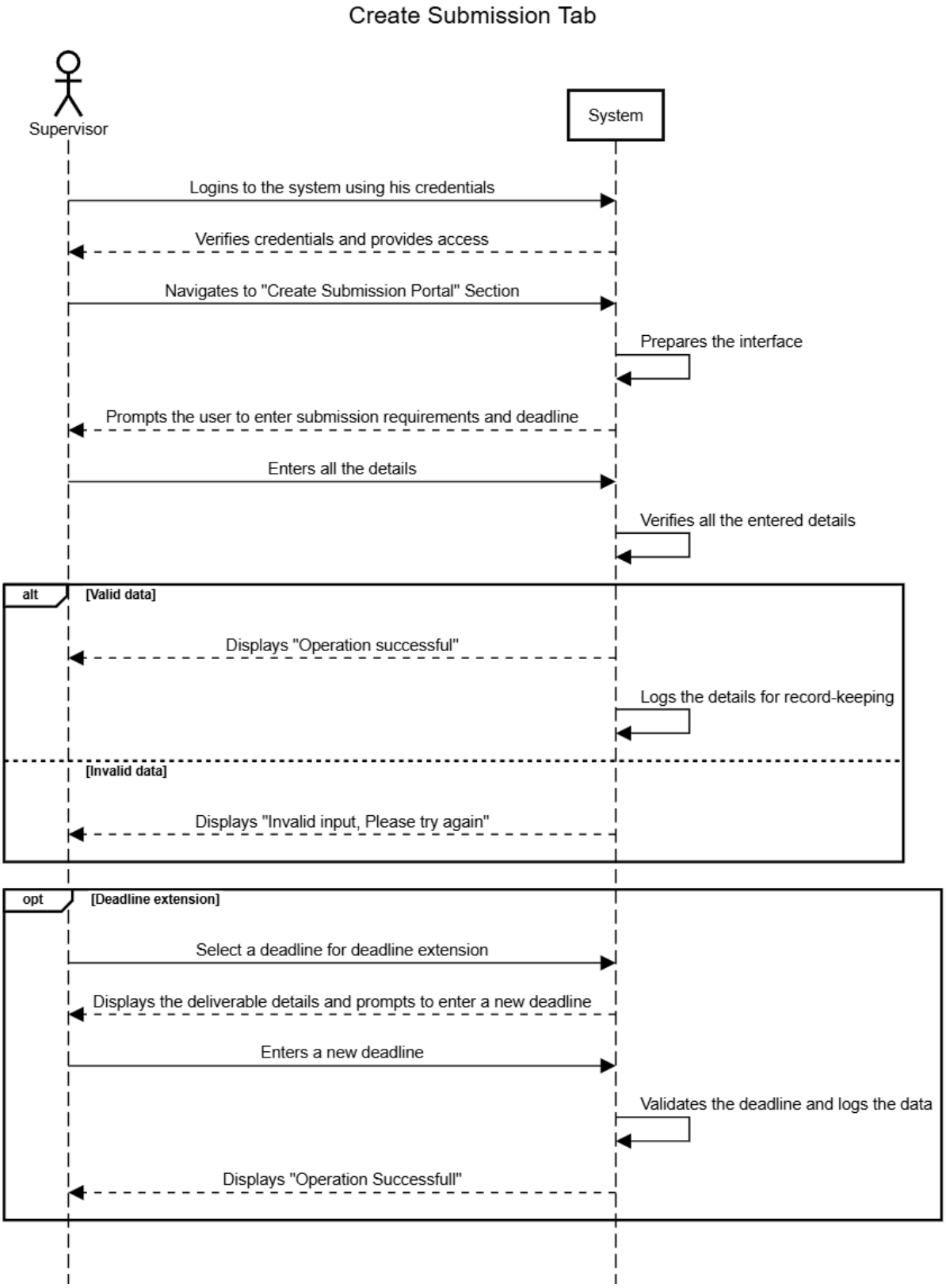
b. Join Group 2



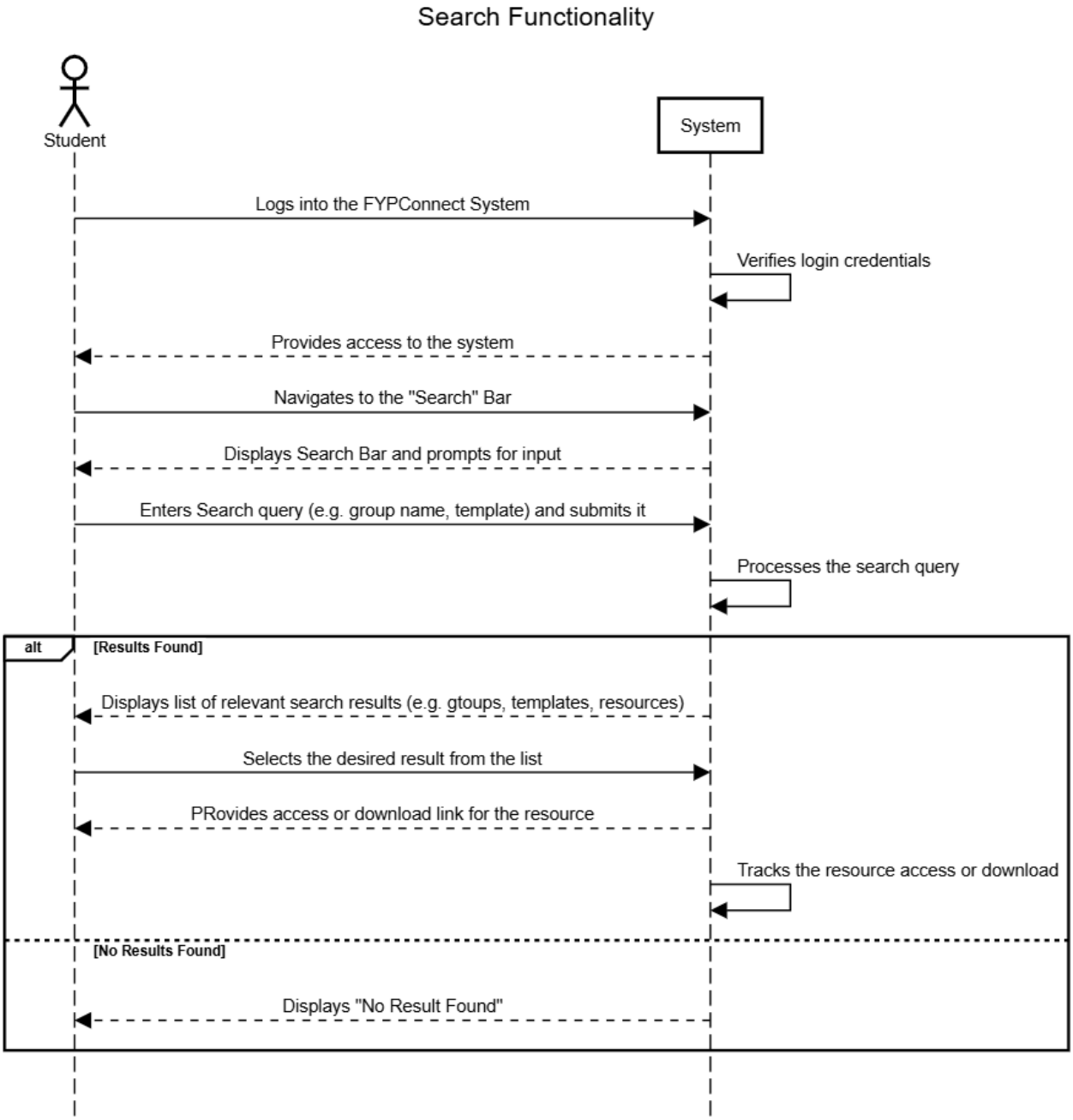
6. Evaluate Deliverables



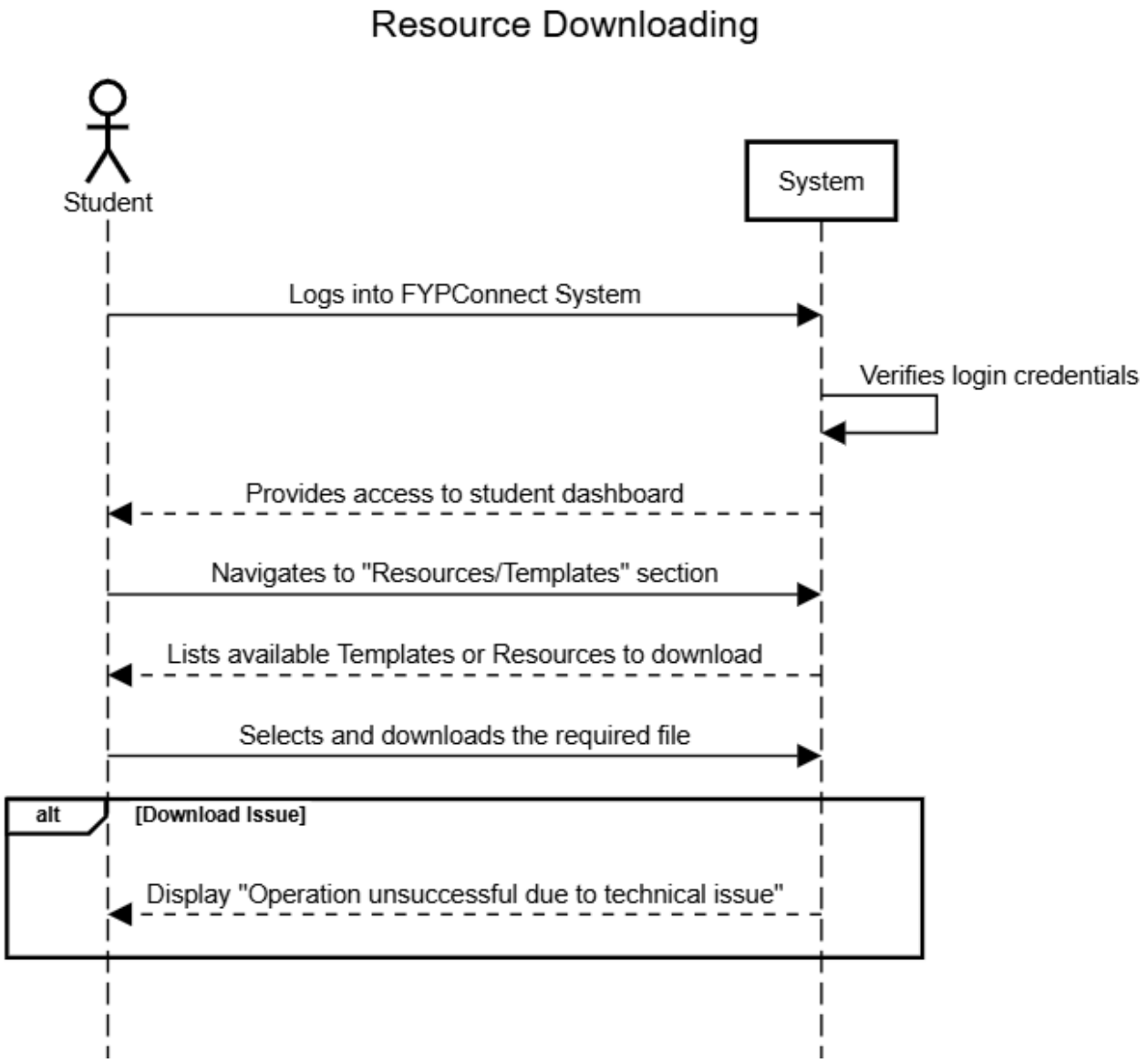
7. Submission Tab



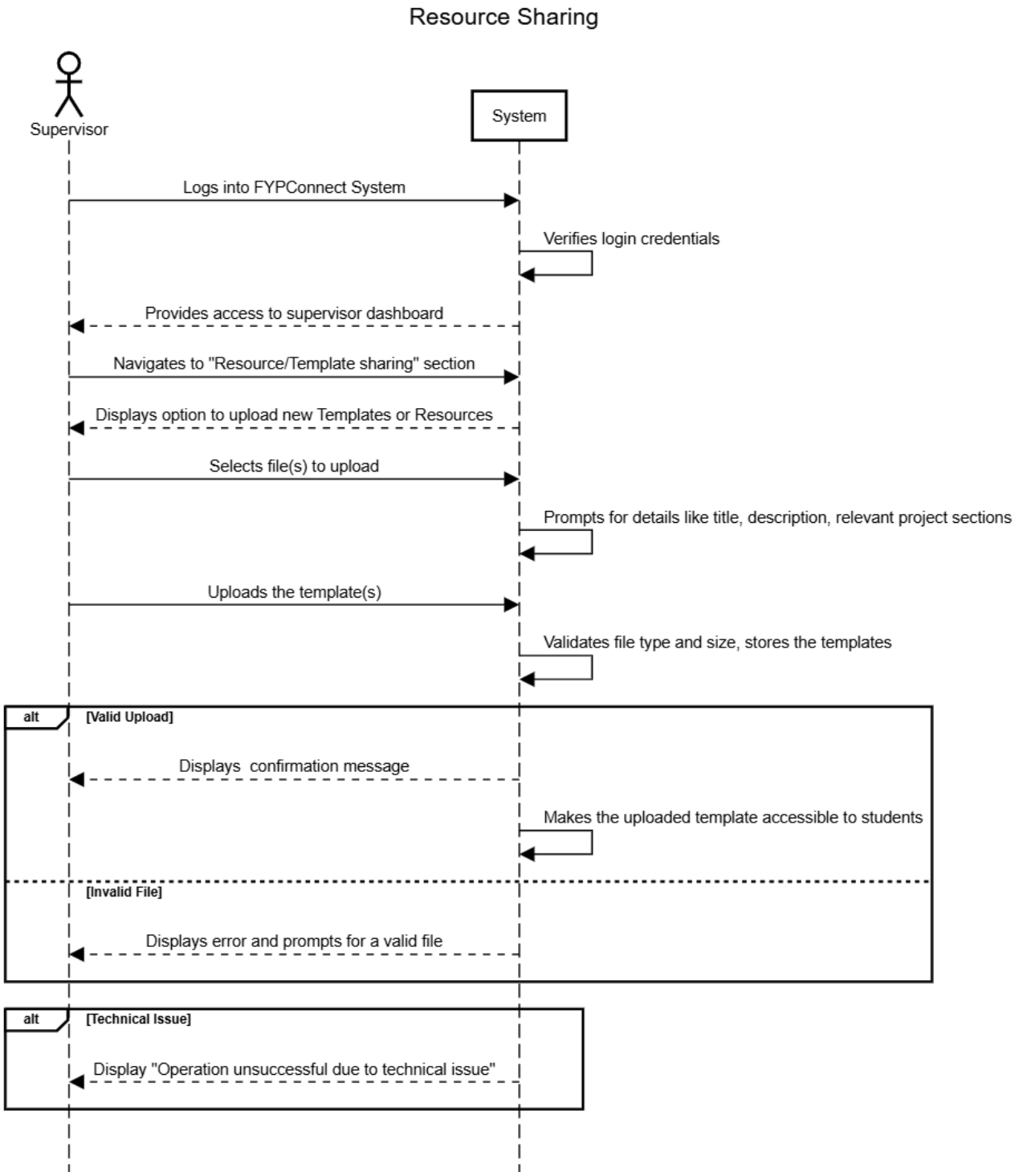
8. Search Functionality



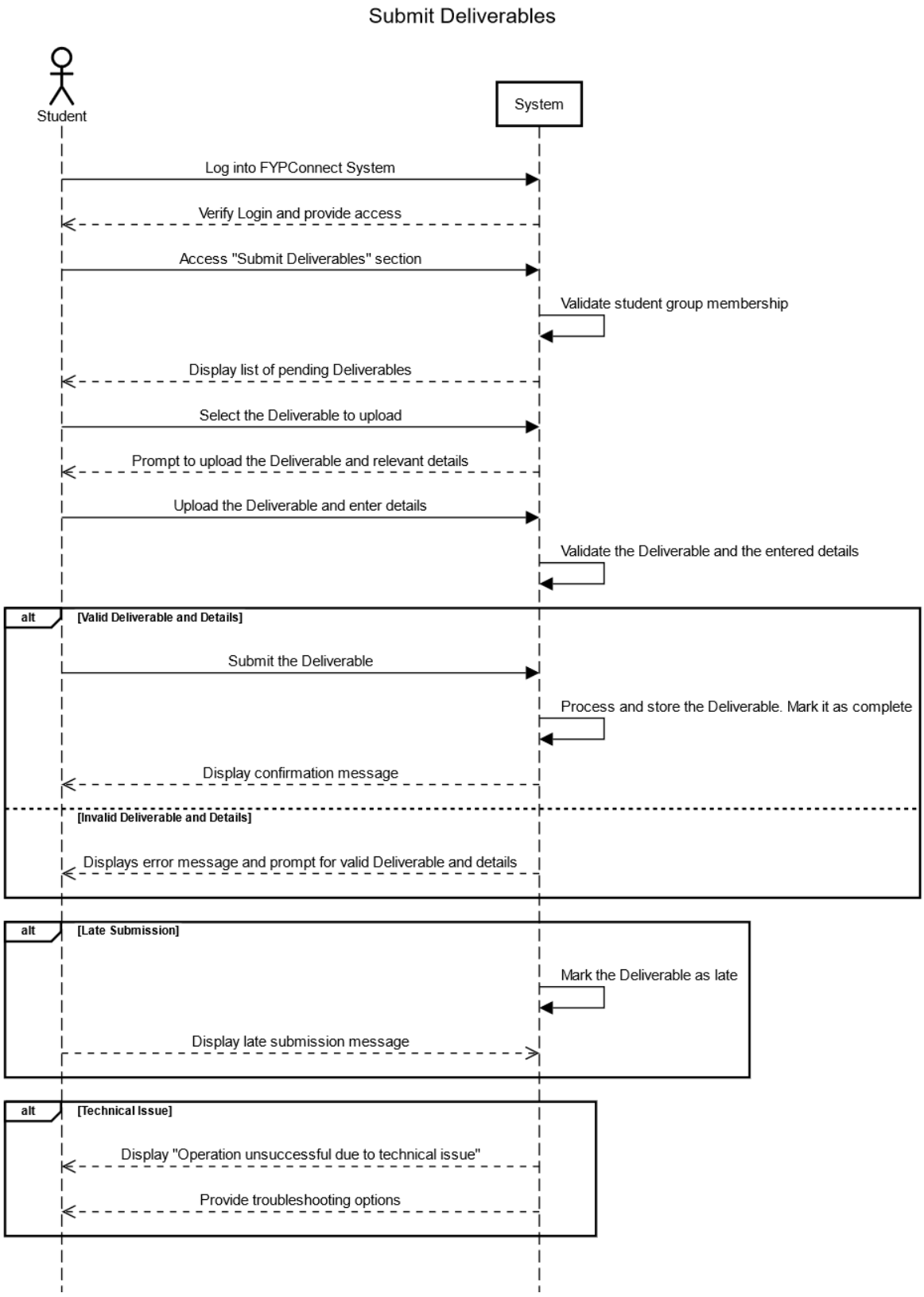
9. Resource Uploading



10. Resource Downloading

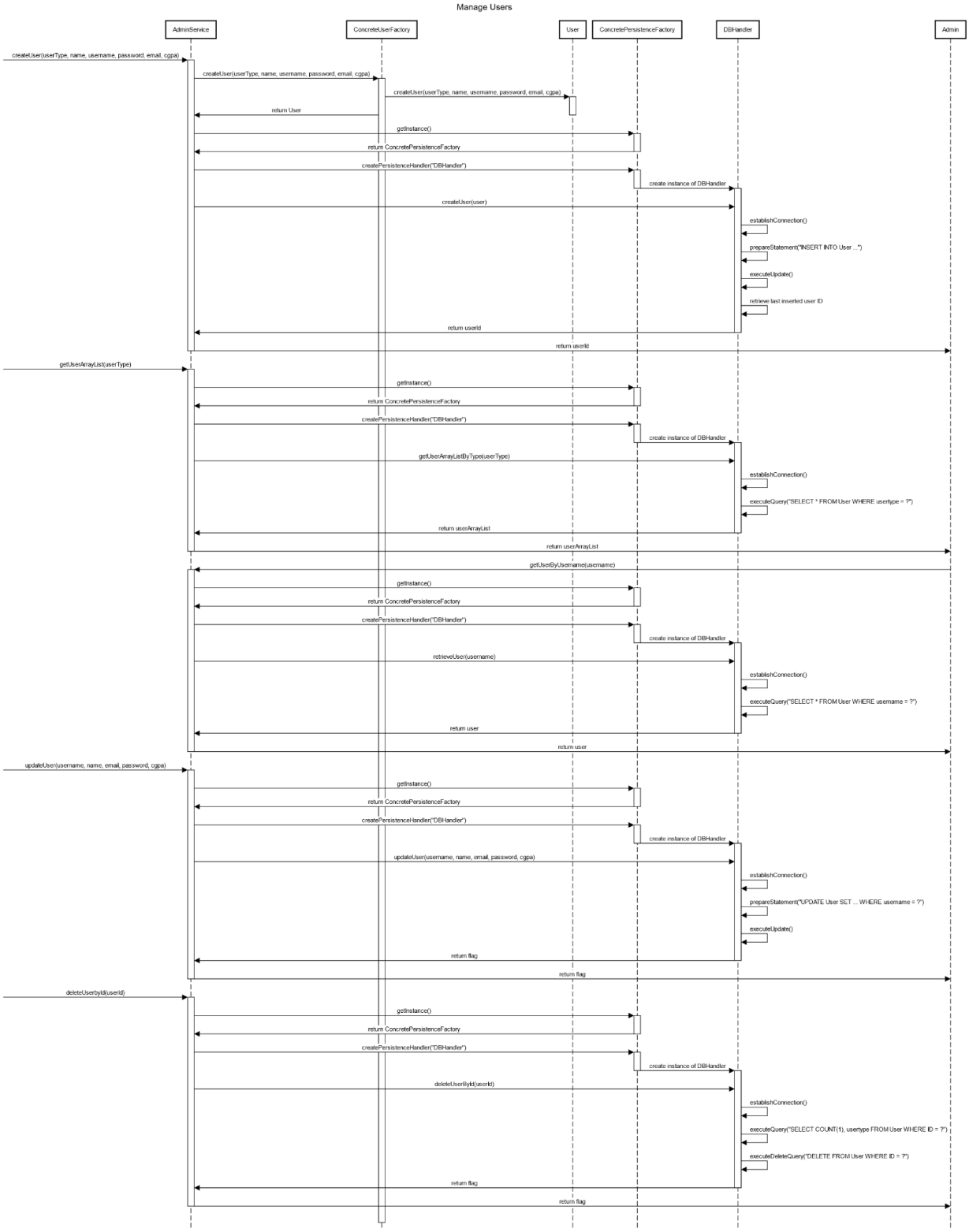


11. Submit Deliverables

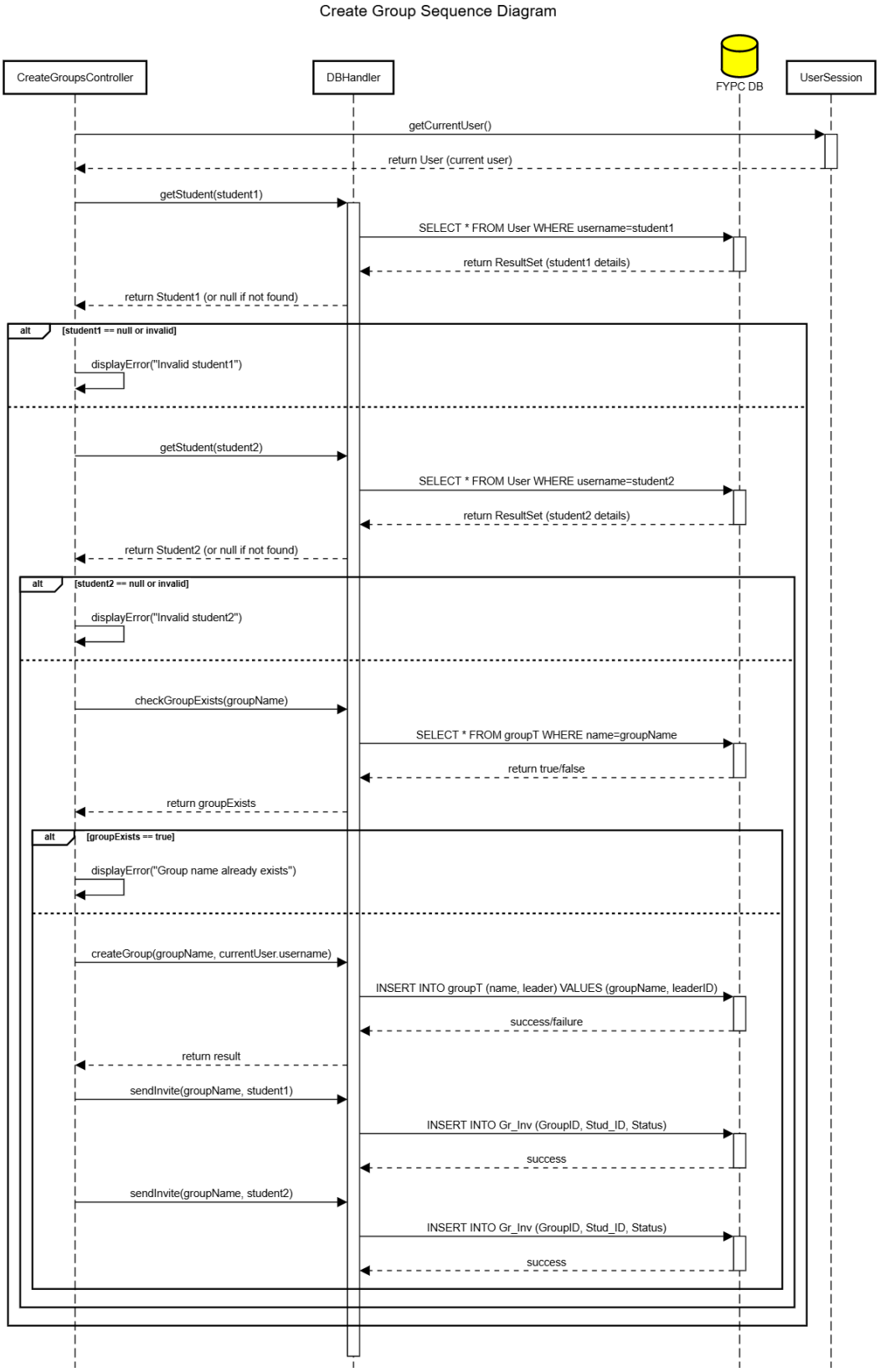


6. Sequence Diagram

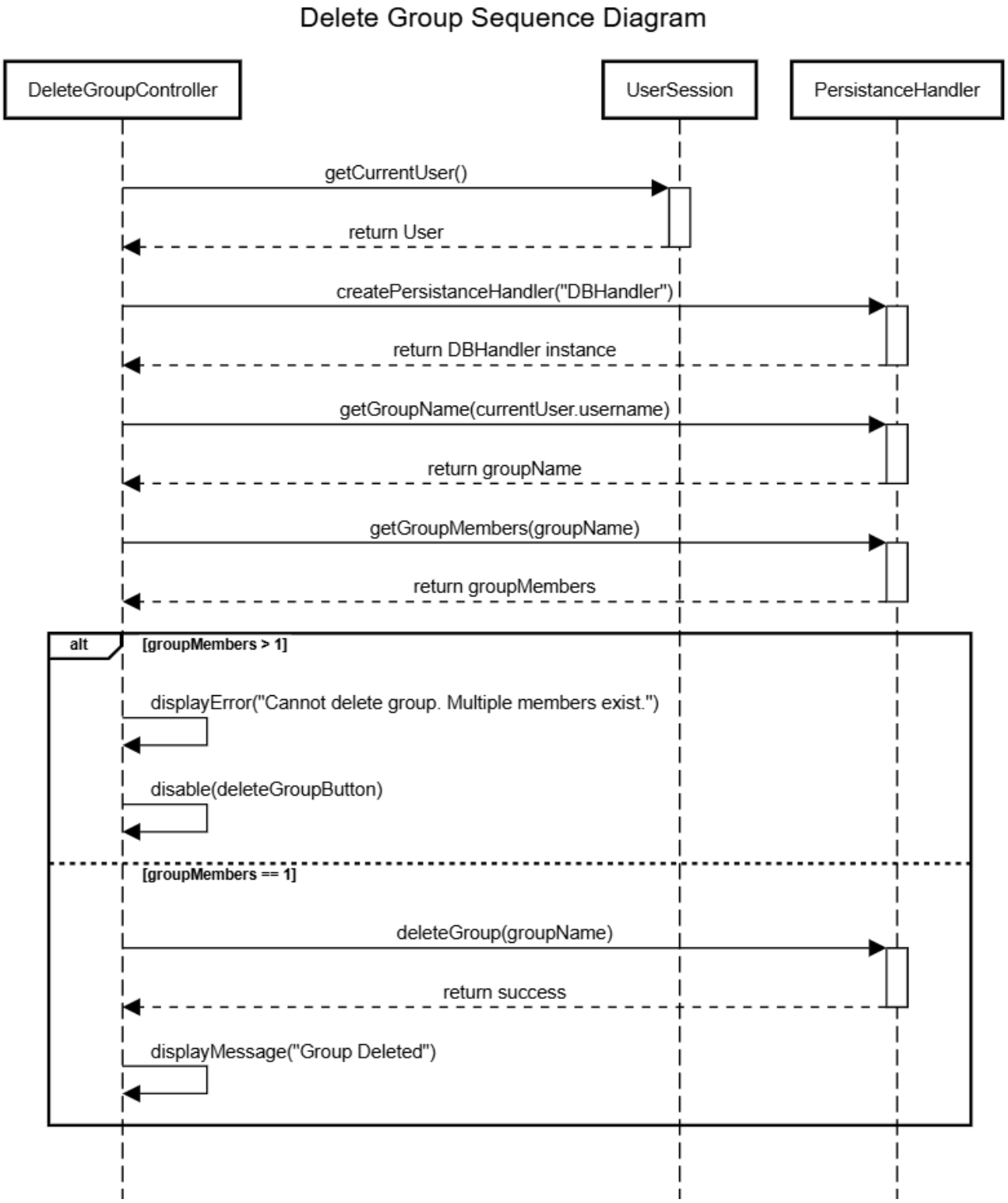
1. Manage Users



2. Create Groups



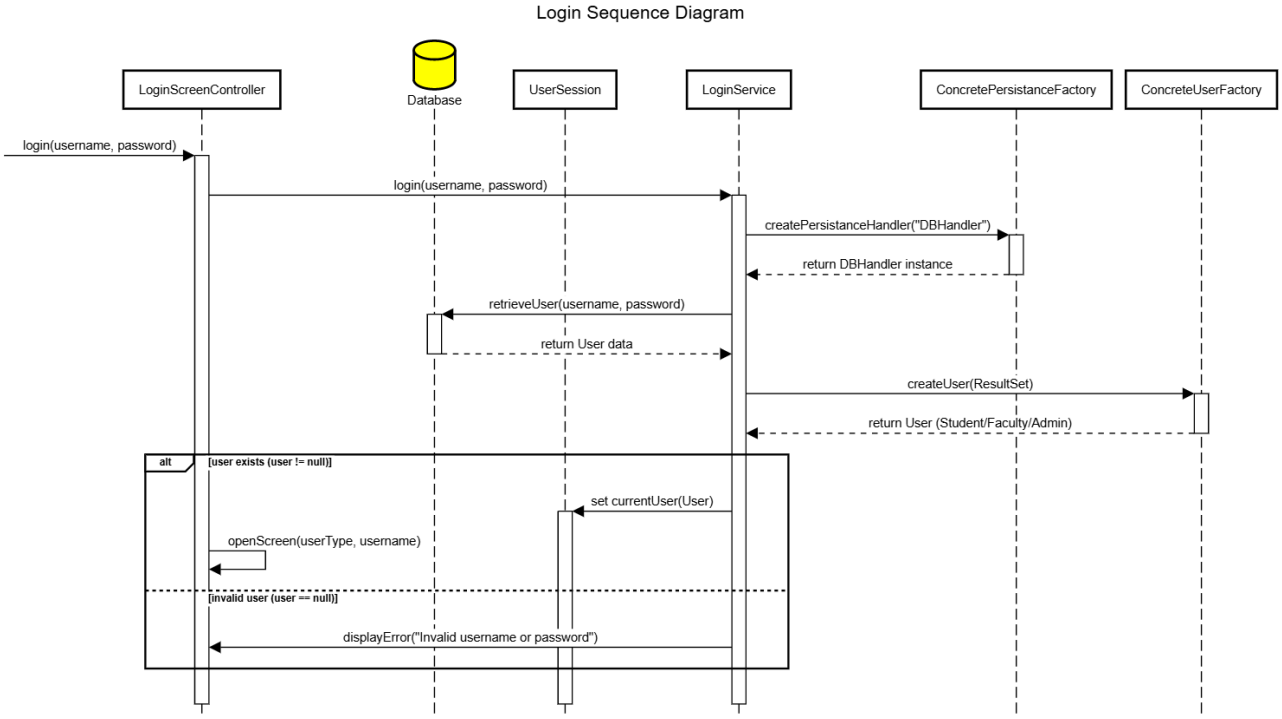
3. Delete Groups



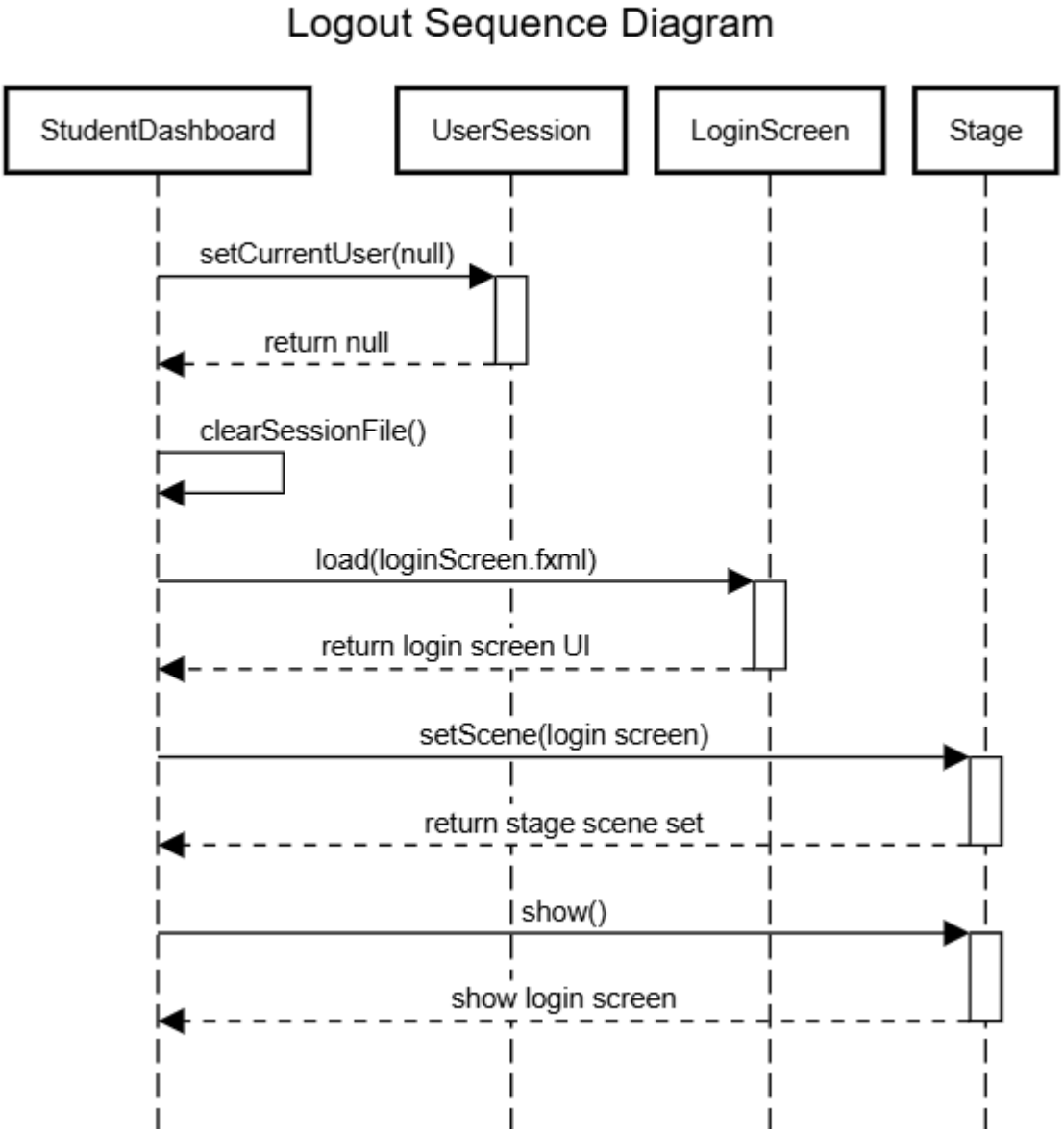
4. Join Group


Join Groups Controller Flow
UserSession
joinGroupsController
PersistanceHandler
getCurrentUser()
User (Current User)
isGroupLeader(username)
boolean (isLeader)
alt
[User is a leader]
getGroupRequests(username)
String[] (requests)
getStudent(requests[0])
User (requester)
acceptRequest(groupName, selectedperson)
boolean (isAccepted)
[User is not a leader]
getGroupInvitations(username)
String[] (invitations)
getGroupLeader(invitation)
String (leader)
acceptInvite(groupName, username)
boolean (result)
getAvailableGroups()
String[] (groups)
sendRequest(groupName, username)
boolean (result)
declineInvite(groupName, username)
boolean (result)
declineRequest(groupName, selectedperson)
boolean (isDeclined)
getGroupName(username)
String (groupName)

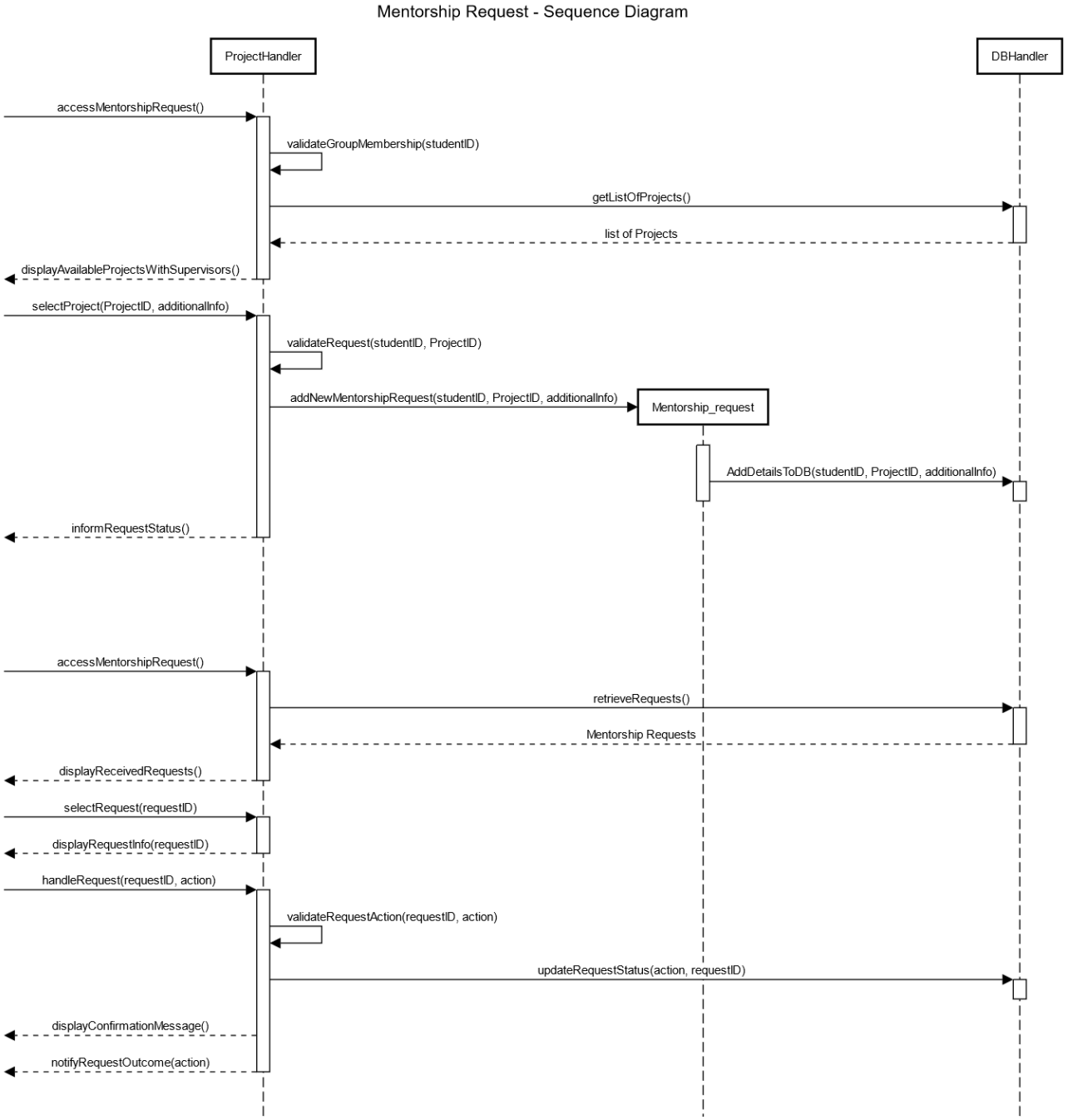
5. Login


Login Sequence Diagram
LoginScreenController
Database
UserSession
LoginService
ConcretePersistanceFactory
ConcreteUserFactory
login(username, password)
login(username, password)
createPersistanceHandler("DBHandler")
return DBHandler instance
retrieveUser(username, password)
return User data
createUser(ResultSet)
return User (Student/Faculty/Admin)
alt
[user exists (user != null)]
set currentUser(User)
openScreen(userType, username)
[invalid user (user == null)]
displayError("Invalid username or password")

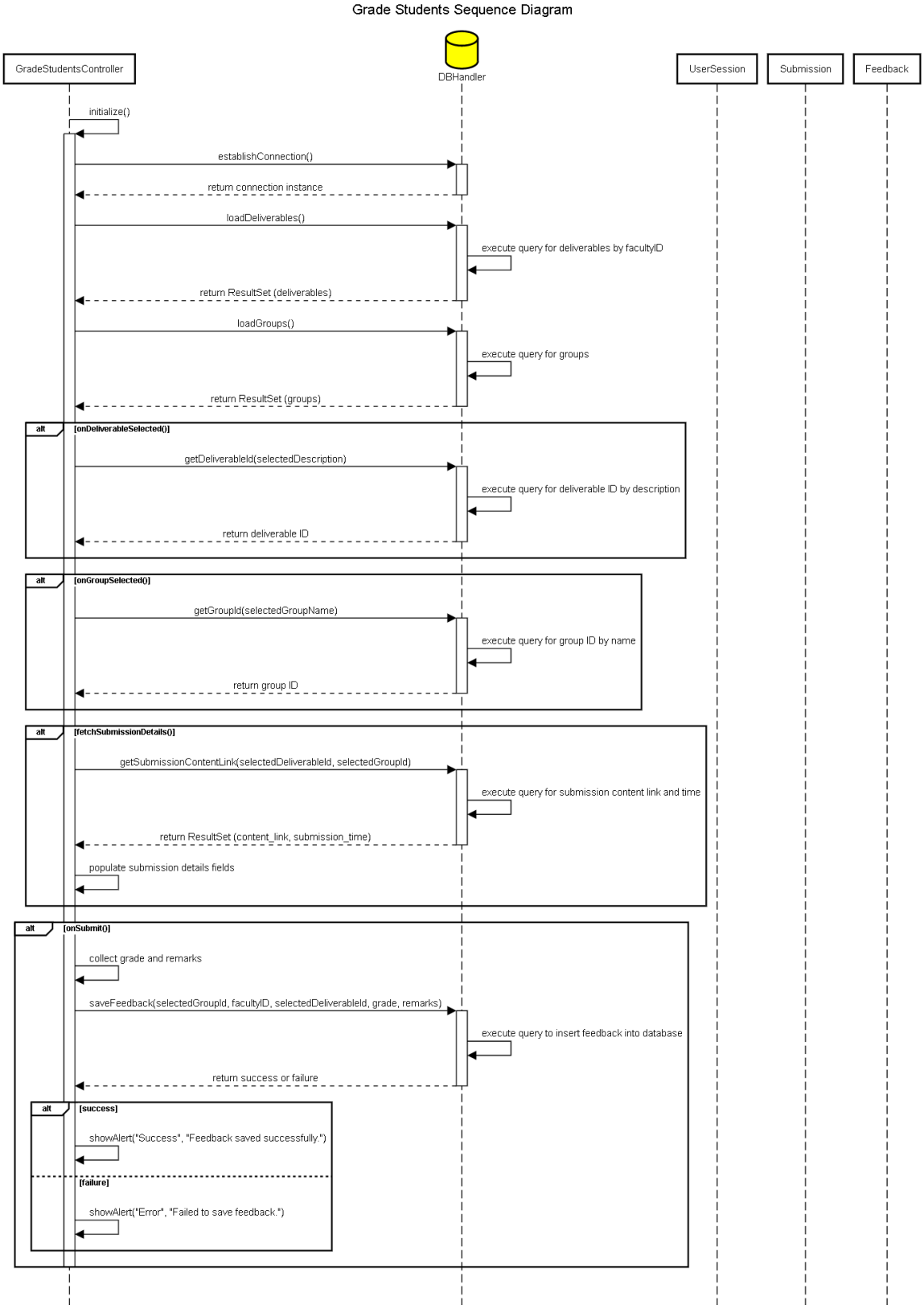
6. Logout


Logout Sequence Diagram
StudentDashboard
UserSession
LoginScreen
Stage
setCurrentUser(null)
return null
clearSessionFile()
load(loginScreen.fxml)
return login screen UI
setScene(login screen)
return stage scene set
show()
show login screen

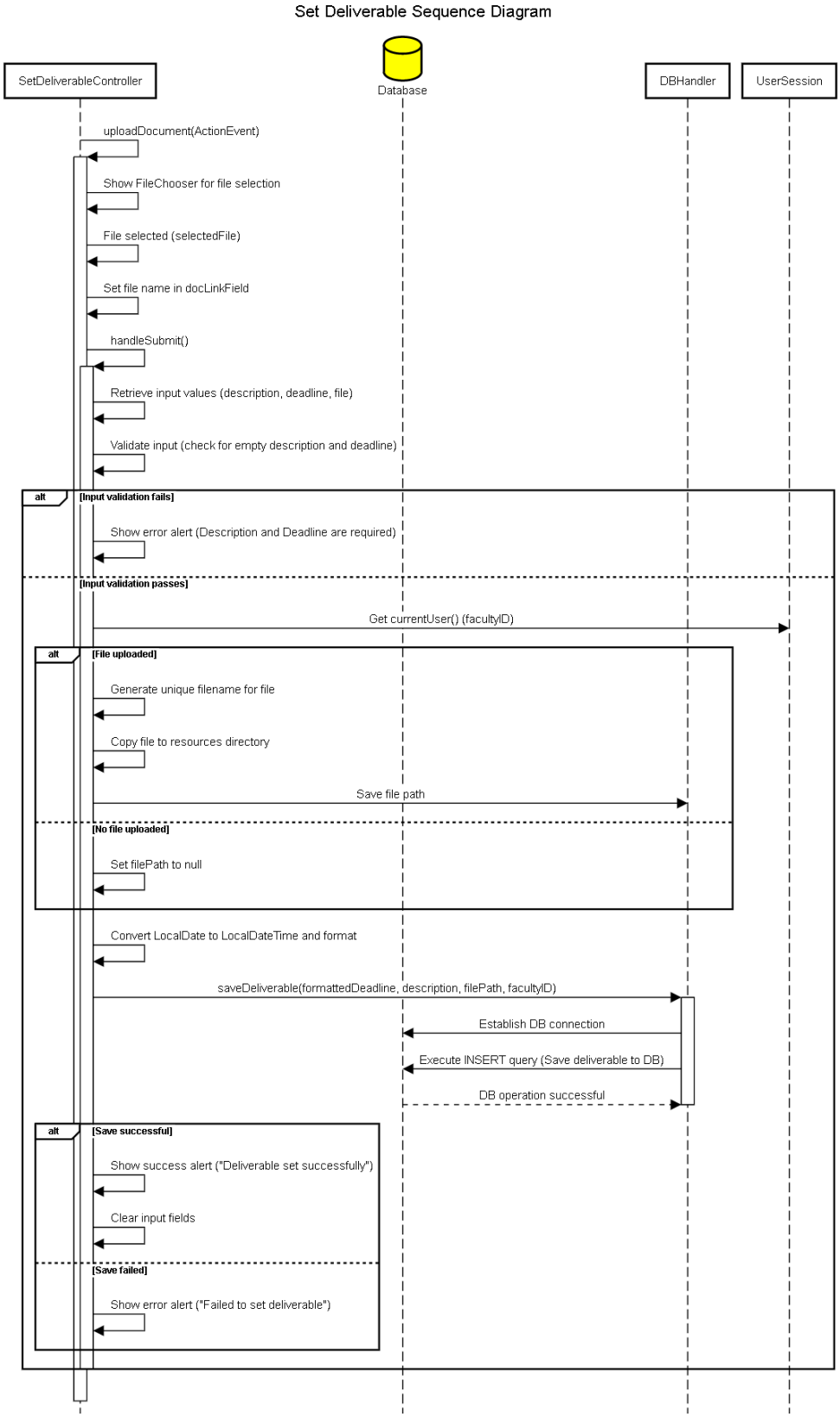
7. Mentorship Request



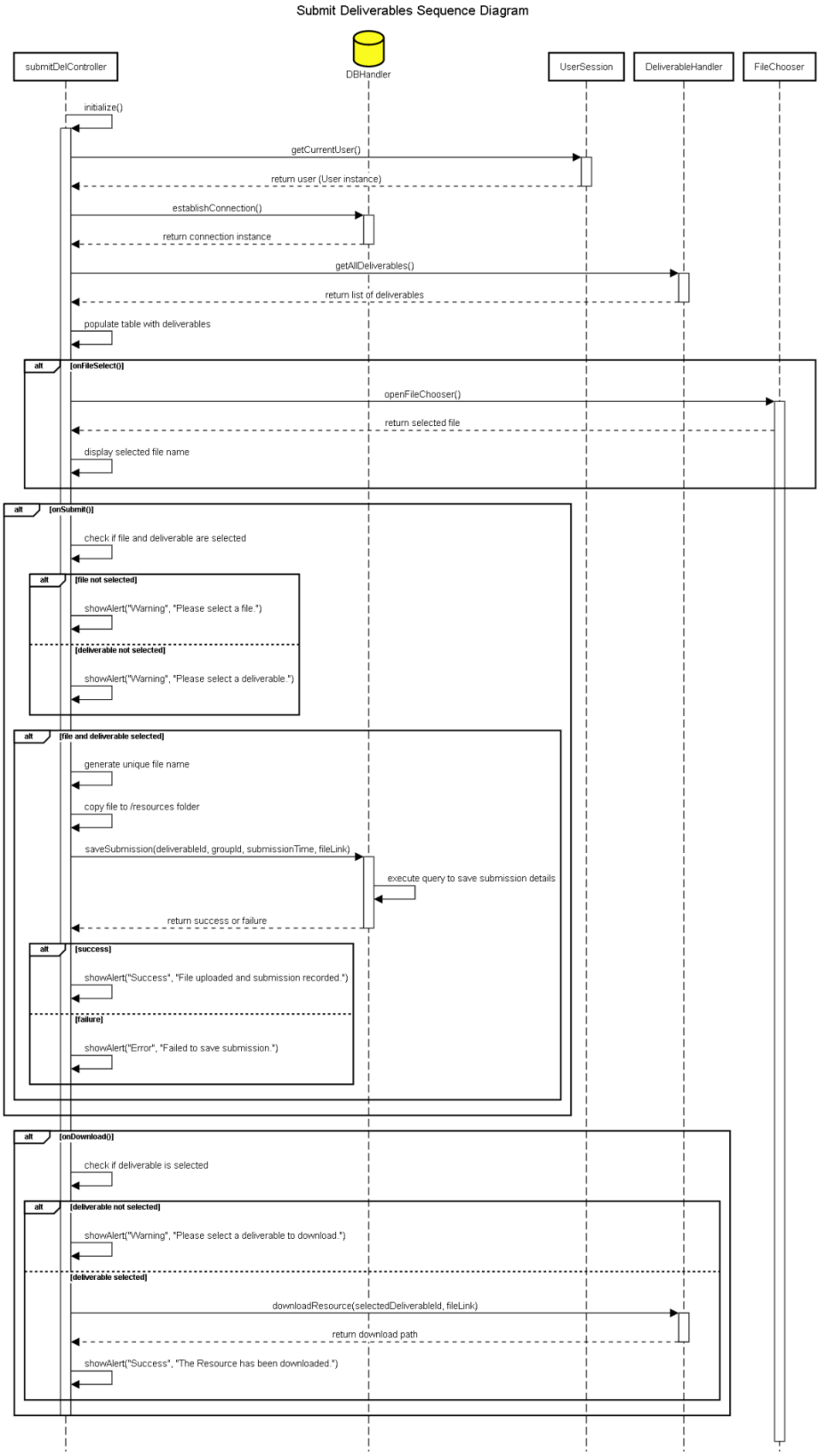
8. Evaluate Deliverables



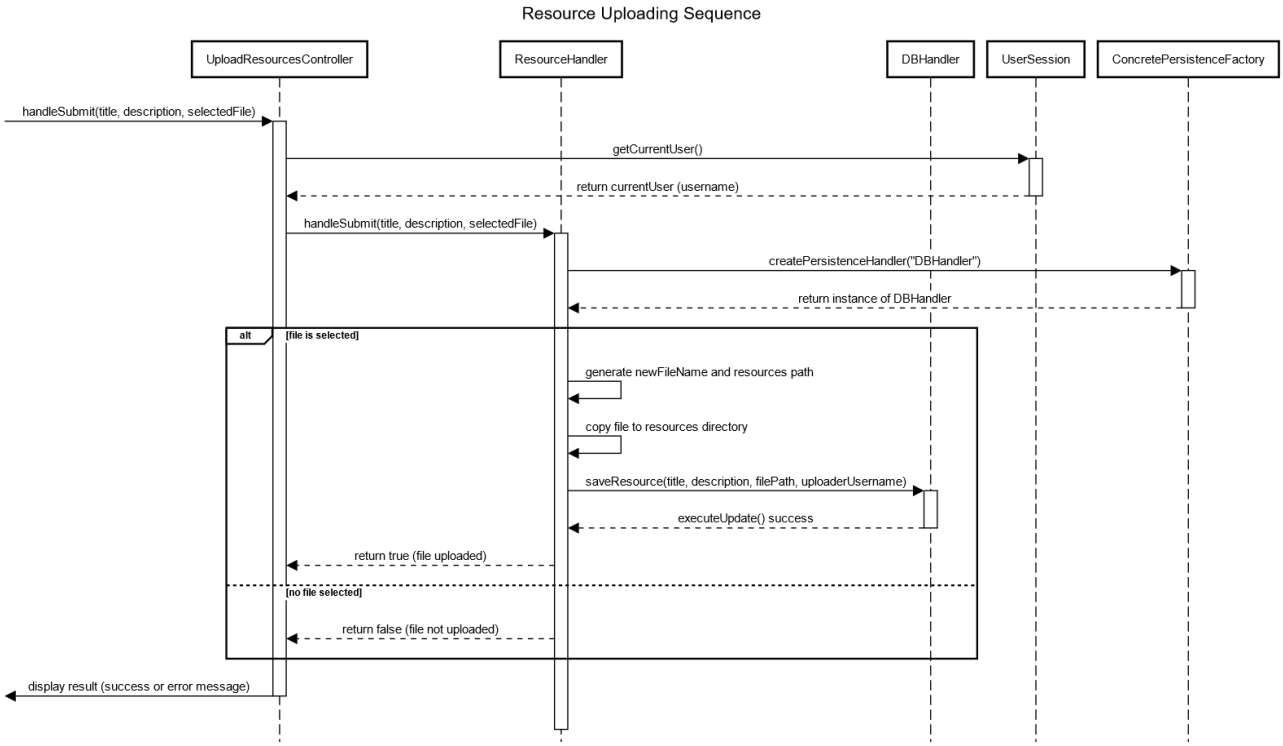
9. Create Deliverable



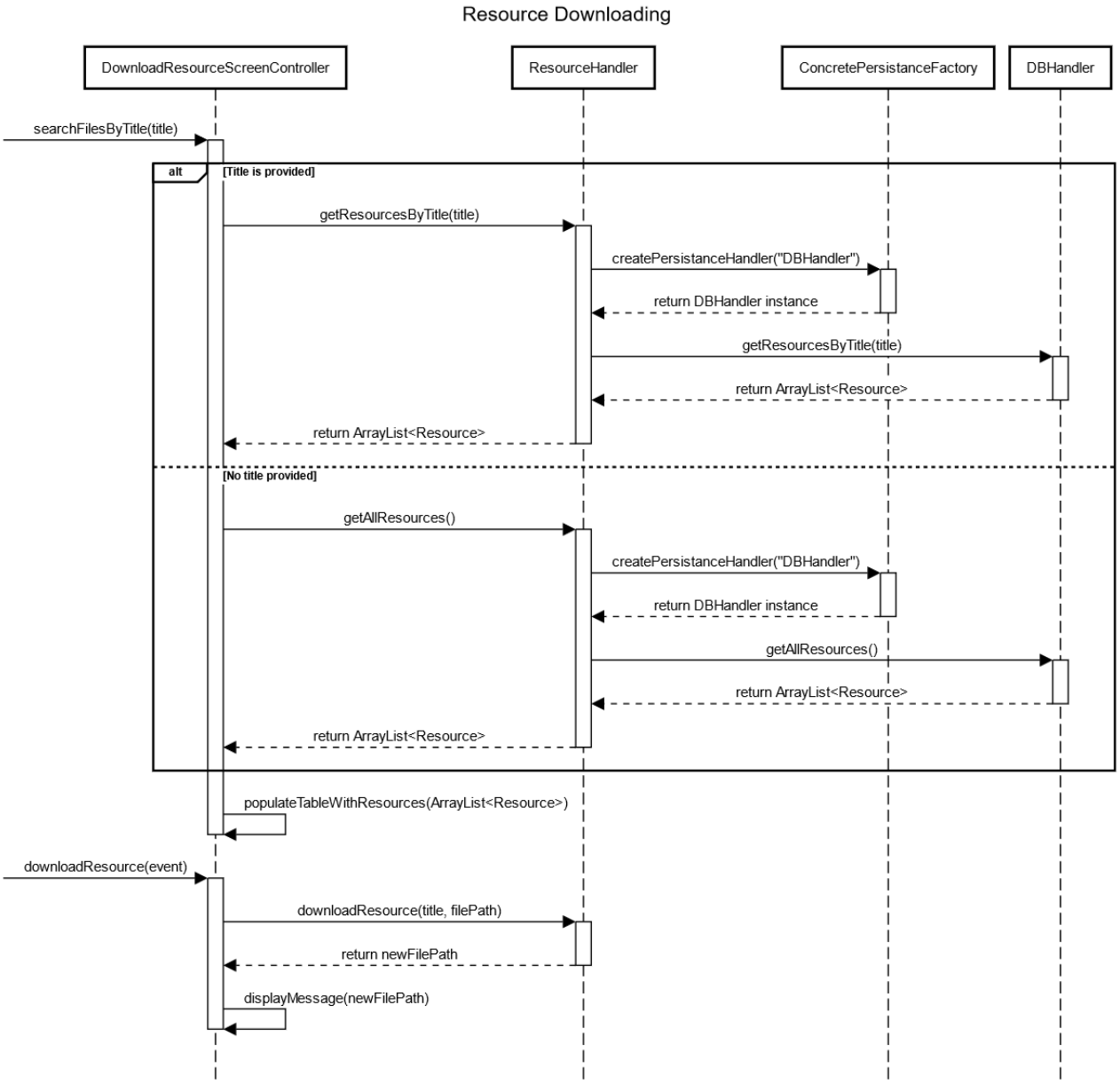
10. Submit Deliverable



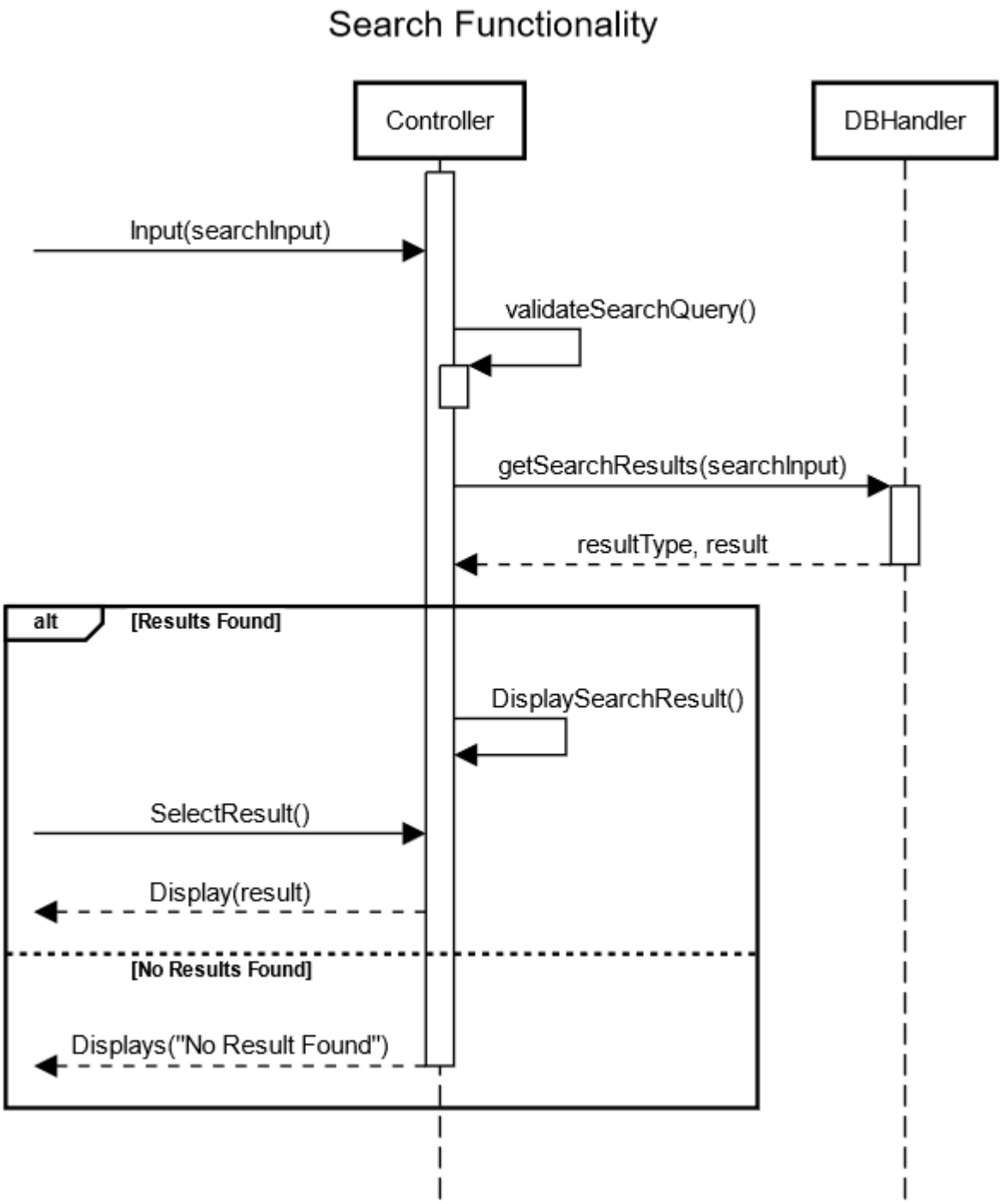
11. Resource Uploading



12. Resource Downloading

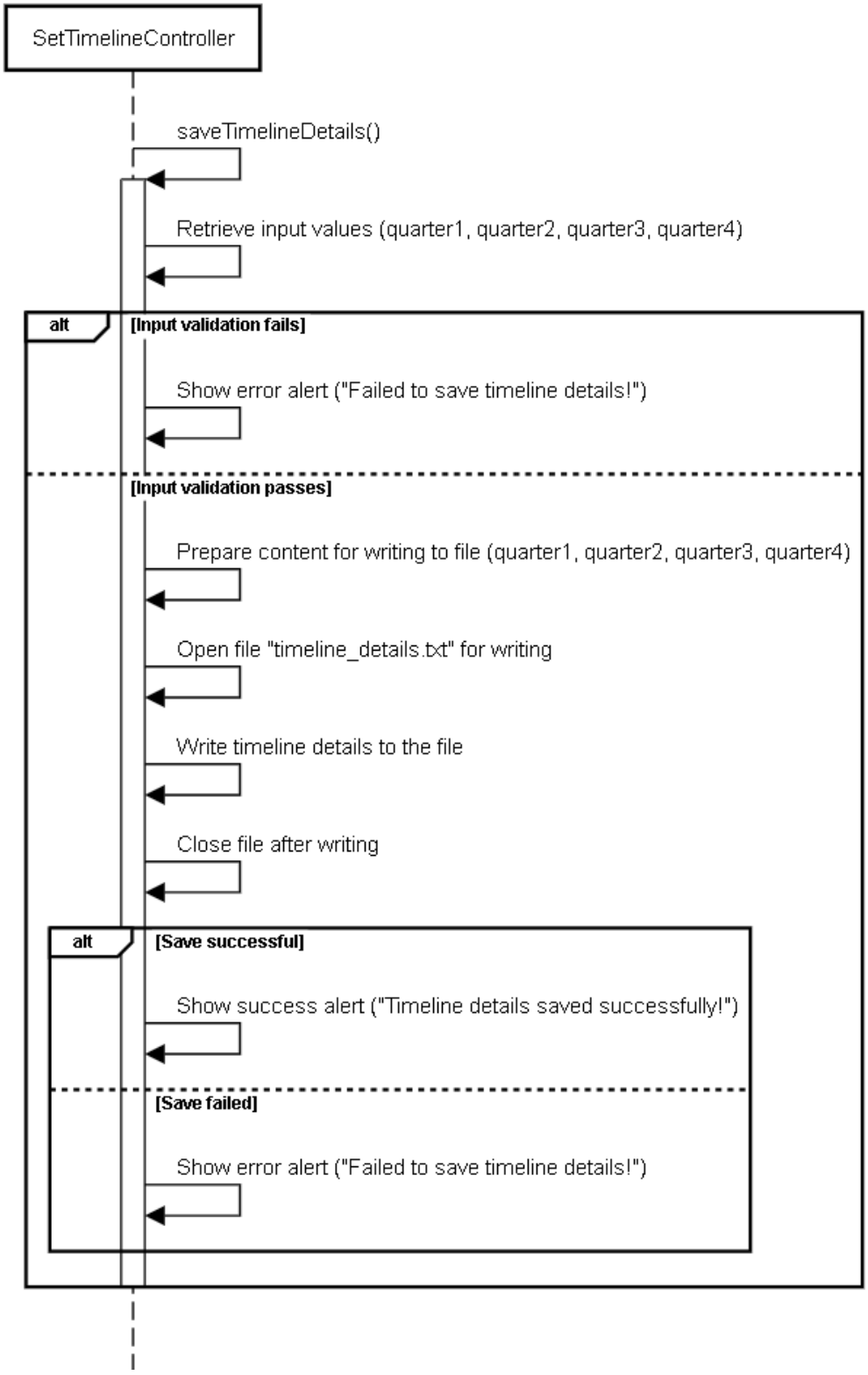


13. Search Functionality

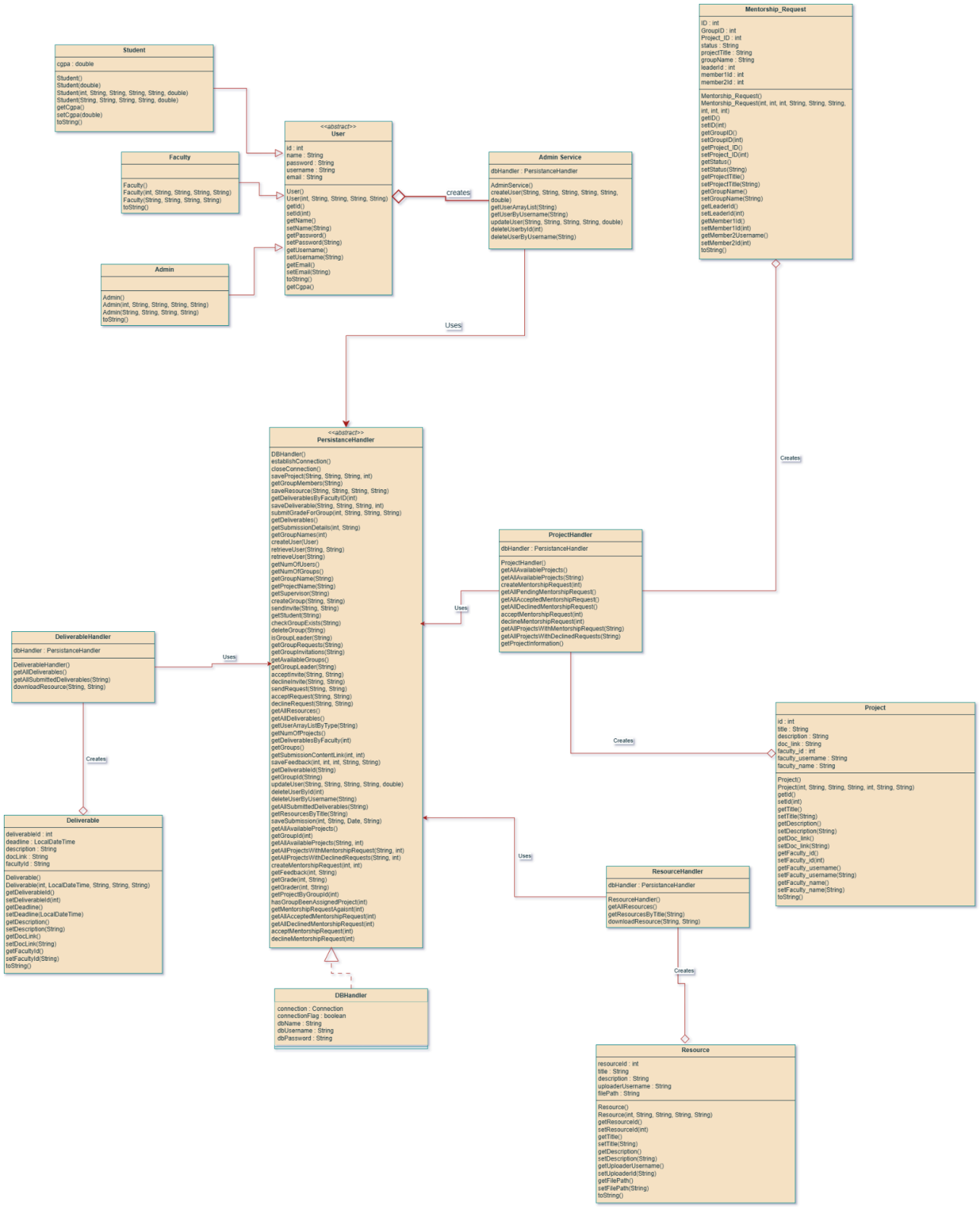


14. Set Timeline

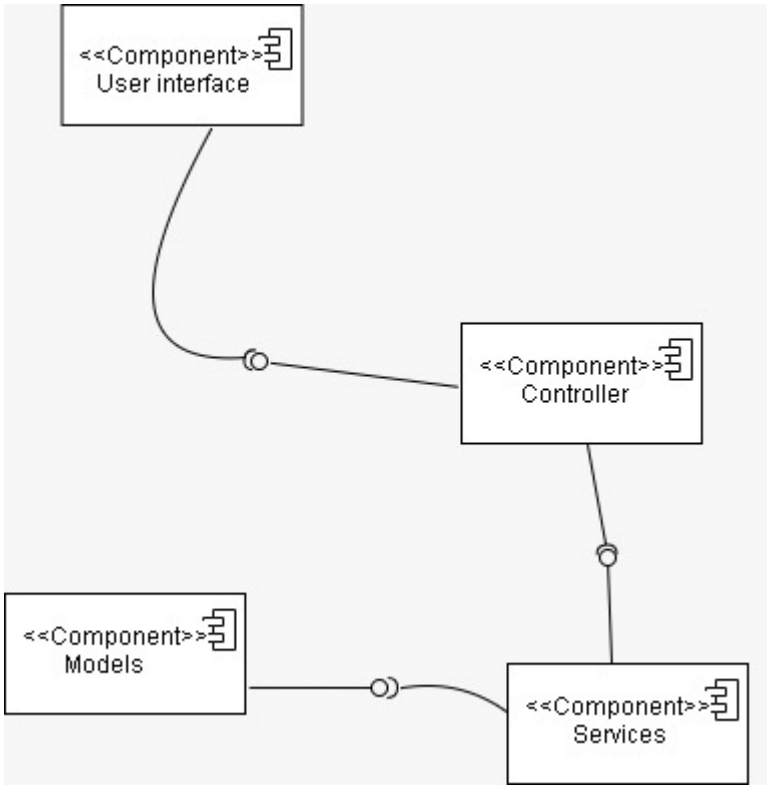
Set Timeline Sequence Diagram



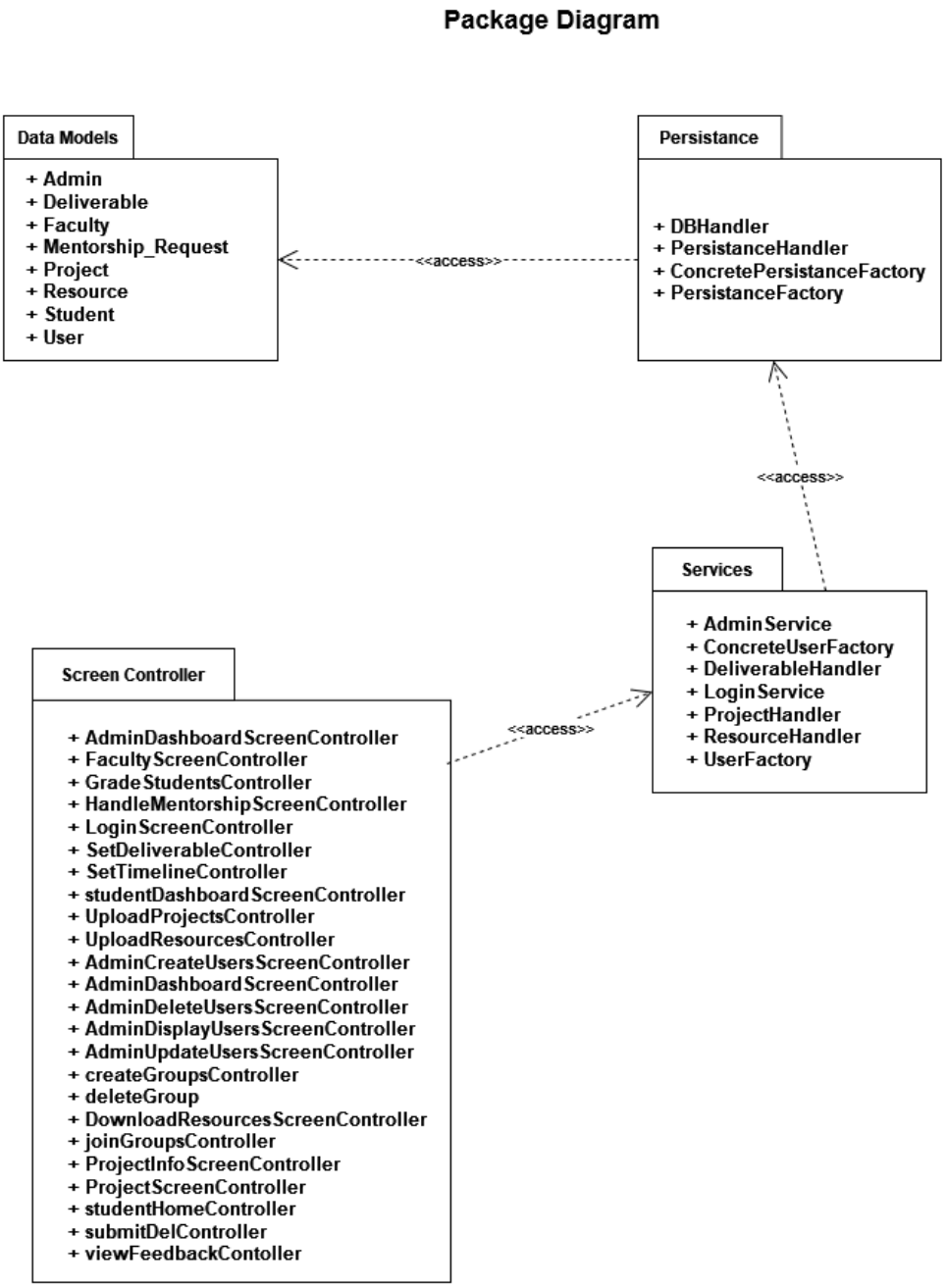
7. Class Diagram



8. Component Diagram



9. Package Diagram



10. Deployment Diagram