StudiShare Software Model & Design

March 12th, 2025 – May 27th 2025

STUDISHARE

Marketplace

Members: Halil Bagosi (**Leader**), Ameraldo Bullari, Daniela Brahimi, Dea Hoxha, Dionis Beçi, Jagoda Andrea

Table of Contents

1. Executive Summary

1.1 Project Overview

The goal of StudiShare is to revolutionize the way students learn and collaborate by providing a comprehensive dynamic platform where they can freely exchange educational materials and resources. By facilitating the sharing of materials and textbooks, StudiShare aims to empower students to enhance their learning experience while promoting a culture of collaboration and knowledge sharing. Subject-matter-specialists can also find a new way to revitalize their knowledge on certain fields by helping out students in need. Additionally, by integrating accessibility features such as audio conversion for visually impaired users, StudiShare strives to ensure inclusivity and equal access to educational resources for all learners.

2. Product/Service Description

StudiShare's development is shaped by key factors that ensure it meets the needs of students, educators, and accessibility-focused users.

• **Educational Accessibility & Sharing** – Students need a centralized, user-friendly platform to access and share study materials, driving requirements for structured content organization and intuitive search.

• **Affordability** – High textbook costs necessitate a marketplace for affordable resources, influencing payment integration and pricing models.

• **Technology & Digital Learning** – Online education trends require AI-driven recommendations, cloud storage, and mobile accessibility.

• **Community & Engagement** – Peer-to-peer learning, mentorship, and rating systems encourage participation, shaping requirements for user interaction and feedback.

• **Accessibility & Inclusivity** – Features like text-to-speech enhance usability for visually impaired users, guiding UI/UX and assistive technology integration.

• **Security & Privacy** – Protecting user data and transactions necessitates encryption, secure authentication, and compliance with privacy regulations.

• **Scalability & Performance** – Growing user demand requires efficient cloud infrastructure, optimized search, and database management.

• **Market Differentiation** – StudiShare stands out by combining free and paid resource sharing with mentorship and accessibility, emphasizing usability and innovation.

2.1 Product Context

StudiShare is a self-contained educational platform that facilitates the sharing, buying, and selling of academic resources. However, it interfaces with external systems to enhance its functionality:

• **Payment Gateways** – Integrates with PayPal, credit/debit cards, and digital wallets (Apple Pay, Google Pay) for secure transactions.

• **Cloud Storage Services** – Utilizes cloud-based storage to manage user-uploaded content efficiently.

• **Video Conferencing Tools** – Links to platforms like Zoom, Google Meet, or Microsoft Teams for mentorship sessions.

• **Assistive Technologies** – Supports text-to-speech conversion for visually impaired users.

• **Authentication Systems** – Includes social logins via Google or Apple for a seamless user experience.

2.2 User Characteristics

1. **Students (Primary Users)**

• **Experience:** High school, undergraduate, and graduate students.

• **Technical Expertise:** Basic to moderate (comfortable with mobile apps, web navigation, and online payments).

• **Needs:** Access to study materials, mentorship, and affordable textbooks.

2. **Faculty & Mentors (Secondary Users)**

• **Experience:** Professors, tutors, and subject-matter experts.

• **Technical Expertise:** Moderate to advanced (familiar with digital learning tools and content creation).

• **Needs:** Uploading educational content, mentoring students, and managing subscriptions.

3. **Visually Impaired Users**

• **Experience:** Students with accessibility needs.

• **Technical Expertise:** Varies; relies on assistive technologies.

• **Needs:** Text-to-speech functionality and an accessible UI.

4. **Administrative Staff**

• **Experience:** University or platform administrators managing content and users.

• **Technical Expertise:** Moderate (familiar with user management and platform settings).

• **Needs:** Moderation, system maintenance, and policy enforcement.

2.3 Assumptions

Internet access – The platform requires an internet connection for content access and transactions.

• **Basic digital literacy** – Students and mentors are expected to navigate web and mobile applications.

• **Availability of third-party services** – Payment gateways, cloud storage, and video conferencing tools will remain operational.

• **Device compatibility** – Users are assumed to have access to modern smartphones, tablets, or computers.

2.4 Constraints and Dependencies

Constraints

• Security & Privacy Regulations – Compliance with data protection laws for secure transactions and content management.

• Accessibility Standards – Adheres to WCAG guidelines for visually impaired users.

• User Verification – Must ensure fair transactions while protecting against fraudulent activity.

• Payment Processing Limitations – Subject to third-party transaction fees and regional availability of payment services.

• Storage Limitations – May impose upload limits based on available cloud storage capacity.

Dependencies

• Third-Party Integrations – Requires external payment processors, cloud storage, and video conferencing tools.

• Mentorship Feature Activation – Depends on mentor participation and subscription models.

• Content Quality Control – Relies on a rating/review system to maintain high-quality educational materials.

• Scalability & Performance – Growth depends on cloud infrastructure and efficient database management.

3. Requirements

3.1 Functional Requirements

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
BR_01	Student shall access educational resources(search, filter, preview, download, purchase history)		1	21/03/25	M.Sc Edlira Cani
BR_02	The system shall facilitate study sessions by enabling mentors to manage lessons, availability, and session durations.(to organize live lessons, set availability, and manage session durations)		1	21/03/25	M.Sc Edlira Cani

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
BR_03	The system shall enable student work submission and mentor feedback process.		1	21/03/25	M.Sc Edlira Cani
BR_04	Student shall manage study materials.(upload, tag related subject)		2	21/03/25	M.Sc Edlira Cani
BR_05	The system shall have a text-to-speech option for materials and study materials.		2	21/03/25	M.Sc Edlira Cani
BR_06	The system shall allow users to modify their profile.	Users shall be able to change their profile picture, field of study, interest-subjects and username	2	21/03/25	M.Sc Edlira Cani
BR_07	The system shall rank mentors based on user rating and feedback.		3	21/03/25	M.Sc Edlira Cani
BR_08	The system shall allow users to customize which notifications they receive on their devices.		1	21/03/25	M.Sc Edlira Cani
BR_09	The system should support multiple languages for the user interface and content.		2	21/03/25	M.Sc Edlira Cani
BR_10	The system shall offer students the ability to download materials and lectures for offline access.		1	21/03/25	M.Sc Edlira Cani
BR_11	The system shall allow students to put a price on a material.	Students should be able to sell their materials to other students.	1	21/03/25	M.Sc Edlira Cani
BR_12	The system shall support sign language educational resources for hearing-impaired users.		2	21/03/25	M.Sc Edlira Cani
BR_13	The system shall allow users to view course syllabus for different subjects.		3	21/03/25	M.Sc Edlira Cani
BR_14	The platform shall allow mentors to conduct live classes.		3	21/03/25	M.Sc Edlira Cani
BR_15	The platform shall implement an algorithm for personalized recommendations based on user preferences and activity.		1	21/03/25	M.Sc Edlira Cani

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
BR_16	The system shall give users the ability to buy paid materials.	Students will be able to use bank transfer, credit card and online wallets (Apple Pay, Google Pay, Samsung Pay, PayPal)	2	21/03/25	M.Sc Edlira Cani
BR_17	The platform shall allow mentors to upload recorded lessons.	This enables mentors to provide on-demand learning content for students who miss live sessions or prefer self-paced study.	2	21/03/25	M.Sc Edlira Cani
BR_18	The system shall support audio-based study materials with accessibility features.	Option to have a transcript of audio materials.	2	21/03/25	M.Sc Edlira Cani
BR_19	The system shall allow students and mentors to list books for sale or rent.	Listing books with pricing and condition details, messaging between buyers and sellers, and secure transaction tracking.	2	21/03/25	M.Sc Edlira Cani
~~BR_20~~	~~The system shall offer a notification and alert system for key user activities.~~ • Users shall be able to delete their account if they decide they want to do so.	~~Alerts for mentorship session updates, study material availability, textbook marketplace activity, and important platform announcements.~~	~~1~~	~~21/03/25~~	~~M.Sc Edlira Cani~~
BR_21	Users shall be able to delete their account if they decide they want to do so.		1	21/03/25	M.Sc Edlira Cani
BR_22	Users shall be able to change/modify their credentials(email, password, birthdate, phone number).		1	21/03/25	M.Sc Edlira Cani
BR_23	The system shall allow users to request help on their schoolwork.	This will work in a forum-style where users can respond to the poster with relative solutions.	3	21/03/25	M.Sc Edlira Cani
BR_24	The system shall implement screenshot or screen-record blocking for mentor shared materials .		1	21/03/25	M.Sc Edlira Cani
BR_25	The system shall allow students to review, rate and report study materials.		3	21/03/25	M.Sc Edlira Cani
BR_26	The system shall allow students to create custom goals.	System reminds the student with notifications to stay on track	3	21/03/25	M.Sc Edlira Cani

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
BR_27	The system shall allow students to bookmark and categorize study materials.		3	21/03/25	M.Sc Edlira Cani
BR_28	The platform shall allow mentors to create quizzes or assignments for students to complete during or after a lesson.		3	21/03/25	M.Sc Edlira Cani
BR_29	The system shall include a feature for students to track their time spent on different study activities and sessions.		2	21/03/25	M.Sc Edlira Cani
BR_30	The system shall enable students to participate in monthly challenges or competitions related to their studies, with rewards for completion (e.g., badges, certificates).		3	21/03/25	M.Sc Edlira Cani
BR_31	The system shall allow students and mentors to communicate via direct messaging.		2	21/03/25	M.Sc Edlira Cani
BR_32	The system shall allow users to register and create an account		1	21/03/25	M.Sc Edlira Cani

3.2 Non-Functional Requirements

3.2.1 Product Requirements

- The system shall be able to maintain uptime in almost every moment because short scheduled maintenance ensures uninterrupted access for users.
- The system shall be compatible with Windows, macOS, and Linux operating systems and accessible via modern web browsers such as Chrome, Firefox, Safari and Edge without requiring additional software installation.
- The system should be available as a website and an application.
- Website load times should be less than 2 seconds.
- App load times should be less than 5 seconds.

3.2.2 Usability Requirements

- The platform should be optimized for both desktop and mobile devices.
- The system should have high contrast color options for the visually impaired users.
- The system shall give a tutorial to first time users on how to use the system.
- The user interface shall be intuitive and require no more than three steps to access study materials*(ensures ease of use).*
- The platform shall provide a dark mode option for user preference*(improves accessibility and comfort).*

- The system shall provide a consistent UI/UX across desktop and mobile versions.

3.2.3 Performance Requirements

- The platform shall be able to scale its processing power dynamically because a growing number of resources require efficient performance.
- The system shall support up to 10,000 simultaneous users and process at least 95% of search queries within 2 seconds under normal workload conditions.
- 99% of materials shall be available for download to buyers within 5 seconds of purchase.
- 90% of transactions shall be completed within 1 second.

3.2.4 Availability

- The system shall maintain 99.9% uptime with server redundancy to prevent service disruptions.
- The mean time between failure should not be less than 1 month.
- The system shall integrate with third parties such as: payment gateway or email service.
- The system shall automatically scale its resources to handle increased traffic during peak usage times.
- The system shall provide automatic failover to a backup server within 5 minutes in the event of a primary server failure*(this quantifies the failover time).*
- The system shall have a disaster recovery plan that includes regular backups of critical data*(this addresses major outages).*
- Automated alerts shall be generated for critical system events, such as server downtime, high CPU usage, or database errors*(this ensures timely response).*

3.2.5 Security

- Users shall have the option to enable Multi factor authentication(MFA) for added account security.
- The system shall have an end-to-end encryption for payments and private messages.
- The system shall perform *automated daily database backups* to ensure data integrity and prevent loss due to unexpected failures or cyberattacks.
- The system shall support high concurrent users without performance degradation or lag, ensuring a smooth experience even during peak usage times.

3.2.6 Organizational Requirements

- The system shall allow API integrations with tools such as Google Drive/Meet, OneDrive, and university portals because seamless connectivity improves user productivity.
- The system shall not exceed 500GB of storage for educational resources and user-generated content, with automatic archiving of materials older than five years to optimize space usage.
- The platform shall support integration with third-party payment providers for transactions*(ensures smooth financial transactions).*
- The system shall store all uploaded materials for a minimum of five years unless deleted by the user*(defines data retention policy).*

3.2.7 External Requirements

- The system shall comply with educational content regulations because preventing unauthorized sharing ensures legal and ethical usage of resources.
- The system shall ensure that all shared educational materials respect intellectual property rights by requiring users to acknowledge the source of their content and prohibiting unauthorized distribution of copyrighted materials.
- Users should be able to delete their account data.

4 User Scenarios/Use Cases

UC Name	***UC_01*** *- Access Educational Resources*
Summary	This use case allows students to search, filter, preview, and download educational resources.
Dependency	None
Actors	Student (Primary Actor)
Preconditions	The student must be logged into the system.
Description of the Main Sequence	1. Student navigates to the resource library. 2. Student searches for a resource by keyword, category, or author. 3. Student filters the search results based on preferences. 4. Student previews a resource before downloading. 5. Student downloads the selected resource.
Description of the Alternative Sequence	1. If no resources match the search criteria, the system displays a message and suggests similar resources. 2. If the student cancels the search, they are redirected to the main resource page.
Non-functional requirements	1. The search results should be displayed within 2 seconds. 2. The system must support at least 100 concurrent searches.
Postconditions	The student successfully downloads or previews an educational resource.

UC Name	*UC_02 - Manage Study Materials*
Summary	This use case allows students to upload and categorize study materials, with the system validating the selected categories to ensure their correctness.
Dependency	None
Actors	Student (Primary Actor)
Preconditions	The student must be logged in.
Description of the Main Sequence	**1.** Student selects "Upload New Material." 2.Student provides the required metadata (title, category, format). 3. Student uploads the material. 4.**Categorization of the material:** o **Step 4.1:** The system displays a list of predefined categories (e.g., "Mathematics," "Computer Science," "Literature"). o **Step 4.2:** If subcategories are available (e.g., "Algebra" under "Mathematics"), the system displays them as a dropdown list or hierarchy under the main categories. o **Step 4.3:** The student selects **one or more categories** that best represent the content of the material. The system allows for multi-category selection, which means a material can belong to multiple categories (e.g., "Computer Science" and "Java Programming"). o **Step 4.4:** If the student wants to create a new category, the system may allow them to do so (if they have the appropriate permissions). o **Step 4.5:** The system validates that the chosen category is valid (e.g., ensuring no errors such as selecting a non-existing category or an inappropriate format). o **Step 4.6:** Once the category is selected, the system links the material to the category (or categories) in the database. • **Step 5:** Student confirms the upload, and the system stores the material in the database along with its associated category(ies).

UC Name	***UC_02*** *- Manage Study Materials*
Description of the Alternative Sequence	**Step 1:** If the upload fails (e.g., due to file size or format issues), the system displays an error message. **Step 2:** If a duplicate material is detected (i.e., the same title and category already exist), the system notifies the user that the material already exists and offers options to overwrite, cancel, or change the title/category. **Step 3:** If an invalid category is selected (e.g., a category that has not been set up), the system prompts the student to select a valid category or notify them if the category is unavailable.
Non-functional requirements	The system should support file uploads of up to 500 MB. The system should categorize and store materials within 2 seconds of submission. The system should allow for flexible categorization with support for both **main categories and subcategories**. The system should provide validation to ensure that selected categories are appropriate and prevent selecting invalid or non-existent categories.
Postconditions	The material is uploaded successfully and categorized under the selected categories. The material is linked to its corresponding category or categories in the database for easy retrieval and sorting.

UC Name	*UC_03 - Student Work Submission & Mentor Feedback*
Summary	This use case allows students to submit assignments and mentors to provide feedback.
Dependency	None
Actors	Student (Primary Actor), Mentor (Secondary Actor)
Preconditions	The student must be enrolled in a course.
Description of the Main Sequence	1. Student uploads an assignment. 2. System notifies the assigned mentor. 3. Mentor reviews the submission. 4. Mentor provides feedback. 5. Student views the feedback.
Description of the Alternative Sequence	1. If submission deadline has passed, system rejects submission. 2. If mentor is unavailable, system assigns a different mentor.
Non-functional requirements	1. Feedback should be stored securely. 2. Students should be notified within 10 minutes of mentor feedback.
Postconditions	Student receives mentor feedback successfully.

UC Name	***UC_04*** *- Facilitate Study Sessions*
Summary	This use case allows mentors to organize and manage live lessons, set their availability, and manage session durations.
Dependency	None
Actors	Mentor (Primary Actor), Student (Secondary Actor)
Preconditions	Mentor must be registered in the system.
Description of the Main Sequence	1.Mentor schedules a live lesson. 2.Mentor sets the availability hours for the lesson. 3.Mentor manages the session duration. 4.Students join the session as per the scheduled time
Description of the Alternative Sequence	1.If the mentor cancels the session, students are notified. **2.** If a student cannot join, they are notified that they missed the session.
Non-functional requirements	System should support live streaming without buffering.
Postconditions	The live study session is successfully conducted or scheduled, and any associated materials (if applicable) will be subject to expiration after a set period.

UC Name:	***UC_5*** *- Text-to-Speech for Study Materials*
Summary	The system shall provide a text-to-speech (TTS) feature that enables students to listen to study materials aloud with customizable voice settings.
Dependency:	The system depends on the availability of a Text-to-Speech (TTS) engine (Azure TTS) and internet connectivity.
Actor	Primary Actor: Student
Preconditions:	● The student is logged into the system. ● The student has study materials available in text format. ● The system has access to a text-to-speech engine.
Description of the Main Sequence:	1.The student selects the text-to-speech option for a study material. 2.The system uses Azure Text-to-Speech (for Windows.MacOS and Linux) 3.The system displays voice customization options (male/female voice, speed). 4.The student selects desired voice settings. 5.The system processes the text using the predefined TTS algorithm. 6.The system begins reading the text aloud. 7.The student can pause, resume, or stop the playback at any time.
Description of the Alternative Sequence:	If the selected study material is not in a compatible text format, the system displays an error message. If the TTS engine fails to process the text, the system notifies the student and suggests checking the internet connection.

Non-functional requirements:	● The system shall use **Azure Text-to-Speech** for all operating systems (Windows, macOS, Linux). ● The response time for text-to-speech conversion shall not exceed 2 seconds. ● The system shall support two voice types (male and female) and allow the student to adjust speech speed.
Postconditions:	● The text material is successfully converted to speech. ● The student can control playback using pause, resume, and stop options.

UC Name	***UC_06*** *– Modify user profile*
Summary	The system shall allow students and mentors to modify their profile information, including profile picture, username, interest subjects, and field of study.
Dependency	The system depends on the availability of student profile data and an operational database.
Actors	Primar Actors: Student, Mentor
Preconditions	1. The student/mentor is logged into the system. 2. The system has access to the student/mentor's profile data.
Description of the Main Sequence	1. The student/mentor navigates to the profile settings page. 2. The system displays the current profile information (profile picture, username, interest subjects, field of study). 3. The student/mentor selects an option to modify one or more profile details. 4. The system allows the student to update the chosen fields. 5. The student saves the changes. 6. The system updates the profile information in the database and confirms the changes.
Description of the Alternative Sequence	1. If the student/mentor uploads an unsupported file format for the profile picture, the system displays an error message. 2. If the username is already taken, the system prompts the student to enter a different username.

UC Name	*UC_06 – Modify user profile*
	3. If the system fails to update the profile due to a connection issue, it notifies the student and prompts them to retry later.
Non functional requirements	1. The system shall ensure that updates are reflected within 2 seconds of saving changes. 2. The system shall validate username uniqueness before saving. 3. The system shall support standard image formats (PNG, JPG) for profile pictures.
Postconditions	1. The student/mentor's updated profile information is stored successfully. 2. The system confirms the update and reflects the changes in the student's profile view.

UC Name	*UC_07 - Rank Mentors Based on User Rating and Feedback*
Summary	The system ranks mentors based on user ratings and feedback to ensure quality mentorship.
Dependency	None
Actors	**Primary Actor:** System **Secondary Actor:** Students
Preconditions	• The student must have completed a mentorship session. • The system must have stored past feedback and ratings.
Description of the Main Sequence	• **Step 1:** The student accesses the mentor feedback section. • **Step 2:** The student submits a rating and feedback for the mentor. • **Step 3:** The system updates the mentor's ranking based on the new feedback. • **Step 4:** The ranking list of mentors is refreshed.
Description of the Alternative Sequence	• **Step 1:** If no rating is provided, the system does not update the ranking. • **Step 2:** If an invalid rating is submitted, the system prompts the student to enter a valid rating.
Non functional requirements	• The ranking system should update in real-time.

	• The system should ensure data privacy for feedback submissions.
Postconditions	The mentor's ranking is updated based on feedback received

UC Name	***UC_08*** **-** *Customize Notification Preferences*
Summary	The system shall allow users to customize which notifications they receive
Dependency	The system shall offer a notification and alert system for key user activities
Actors	• **Primary Actor:** Student • **Secondary Actor:** System
Preconditions	The student must be logged into the system.
Description of the Main Sequence	• **Step 1:** The student navigates to the notification settings. • **Step 2:** The student selects which types of notifications to enable or disable. • **Step 3:** The system saves the new notification preferences.
Description of the Alternative Sequence	• **Step 1:** If no changes are made, the system retains the previous settings.(default settings) • **Step 2:** If an error occurs, the system notifies the student to retry later.
Non functional requirements	• The system should apply changes instantly. • The system must ensure persistent storage of preferences.
Postconditions	The student receives only the selected notifications.

UC Name	***UC_09*** *- Support Multiple Languages*
Summary	The system should support multiple languages for the UI and content.
Dependency	None
Actors	● **Primary Actor:** Student ● **Secondary Actor:** System
Preconditions	● The student must be logged into the system. ● The system must have multiple language options available.
Description of the Main Sequence	● **Step 1:** The student navigates to language settings. ● **Step 2:** The student selects a preferred language from the available options. ● **Step 3:** The system updates the UI and content language accordingly.
Description of the Alternative Sequence	● **Step 1:** If a language is not available, the system notifies the student and defaults back to English ● **Step 2:** If the system encounters an error, it retains the previous language settings.
Non functional requirements	● The system should store language preferences for future sessions. ● The UI should be responsive to language changes without requiring a restart.
Postconditions	The system displays the UI and content in the selected language.

UC Name	***UC_10 -*** *Download Notes and Lectures*
Summary	The system shall offer students the ability to download materials and lectures for offline access.
Dependency	None
Actors	• **Primary Actor:** Student • **Secondary Actor:** System
Preconditions	• The student must be logged in. • The student must have access to the materials or lectures.
Description of the Main Sequence	• **Step 1:** The student navigates to a material or lecture. • **Step 2:** The student selects the download option. • **Step 3:** The system processes and downloads the file to the student's device.
Description of the Alternative Sequence	• **Step 1:** If there is insufficient storage, the system notifies the student. • **Step 2:** If an error occurs, the system prompts the student to retry.
Non functional requirements	• The system should ensure secure downloads. • The system should provide a progress indicator for large files.
Postconditions	The selected material or lecture is successfully downloaded for offline access.

UC Name	***UC_11*** *- Sell Notes*
Summary	The system shall allow students to put a price on their materials and sell them to other students.
Dependency	The system shall give users the ability to buy paid materials
Actors	• **Primary Actor:** Student • **Secondary Actor:** System
Preconditions	• The student must be logged in. • The student must upload and categorize the material before selling.
Description of the Main Sequence	• **Step 1:** The student navigates to the material-selling section. • **Step 2:** User uploads a material to the platform. • **Step 3:** User sets a price for the material. • **Step 4:** The system verifies the material and lists it for sale.
Description of the Alternative Sequence	• **Step 1:** If the price is not set, the system prompts the student to enter a value. • **Step 2:** If an error occurs, the system notifies the student and does not list the material for sale.
Non functional requirements	• The system should ensure secure transactions. • The system should prevent unauthorized sharing of paid content.
Postconditions	• The material is listed for sale on the platform.

UC Name	***UC_12 -*** *Access Sign Language Educational Resources*
Summary	This use case describes how a hearing-impaired user accesses sign language educational resources available on the platform.
Dependency	The platform must have sign language resources uploaded and tagged appropriately.
Actors	• **Primary Actor:** Hearing-impaired student • **Secondary Actor:** System
Preconditions	The user is registered and logged in. Sign language resources are available in the system.
Description of the Main Sequence	• **Step 1:** The user logs into the platform. • **Step 2:** The user navigates to the educational resources section. • **Step 3:** The user selects the filter for sign language educational resources. • **Step 4:** The system displays a list of sign language resources. • **Step 5:** The user selects a resource to view or download. • **Step 6:** The system opens the resource for the user.
Description of the Alternative Sequence	• **No Available Resources:** If no sign language resources exist, the system displays a message indicating that no resources are currently available and suggests checking back later.
Non functional requirements	• The resource retrieval and display process should complete within a few seconds. • The UI must comply with accessibility standards.
Postconditions	• The hearing-impaired user is able to view or download the selected sign language resource.

UC Name	***UC_13*** *- View Course Syllabus*
Summary	This use case outlines how a user views the course syllabus for different subjects on the platform.
Dependency	Course syllabi must be available and associated with their respective subjects.
Actors	• **Primary Actor:** Student • **Secondary Actor:** System
Preconditions	• The user is registered and logged in. • The course syllabi are uploaded and properly linked to the subjects.
Description of the Main Sequence	• **Step 1:** The user navigates to the courses section**.** • **Step 2:** The user selects a subject of interest. • **Step 3:** The system displays the available courses and their associated syllabi. • **Step 4:** The user clicks on the course syllabus link. • **Step 5:** The system opens and displays the detailed course syllabus.
Description of the Alternative Sequence	• **Step 1:** If the syllabus is missing, the system shows a message stating that the course syllabus is not currently available.
Non functional requirements	• The syllabus must load within 5 seconds. • The display should be clear and formatted for easy reading.
Postconditions	• The user successfully views the complete course syllabus for the selected subject.

UC Name	***UC_14*** *- Conduct Live Classes*
Summary	This use case details how a mentor conducts live classes using the platform.
Dependency	The mentor must have the necessary permissions and a stable internet connection.
Actors	• **Primary Actor:** Mentor • **Secondary Actor:** System • **Tertiary Actor:** Students
Preconditions	The mentor is registered, logged in, and has access to the live class module. The live class functionality is enabled and properly configured.
Description of the Main Sequence	• **Step 1:** The mentor logs into the platform and navigates to the live classes section. • **Step 2:** The mentor schedules or initiates a live class session. • **Step 3:** The system verifies the mentor's credentials and connectivity. • **Step 4:** The mentor starts the live class, and the system broadcasts the session. • **Step 5:** Students join the live session and interact via chat or audio/video as supported by the platform. • **Step 6:** The system records the session if that feature is enabled.
Description of the Alternative Sequence	• **Connection Issues:** If the mentor experiences connectivity issues, the system displays an error message and offers a retry option or switches to a backup connection.
Non functional requirements	• The live class session must start within a few seconds of initiation. • The system should support real-time interaction without significant latency.
Postconditions	• The live class is successfully conducted, and students are able to participate in real time.

UC Name	***UC_15*** *- Generate Personalized Recommendations*
Summary	This use case describes how the platform generates personalized recommendations for users based on their preferences and activity.
Dependency	The system must have an algorithm in place that collects and processes user data. We think **TF-IDF + Cosine Similarity Algorithm** will be best for our case.
Actors	• **Primary Actor:** System • **Secondary Actor:** Student
Preconditions	The user is registered and logged in. The user has set preferences and has an activity history on the platform.
Description of the Main Sequence	• **Step 1:** The system continuously collects data on user preferences and activity. • **Step 2:** The system runs the recommendation algorithm at regular intervals or upon significant user activity. • **Step 3:** The system generates a list of personalized study materials and courses. • **Step 4:** The system displays the personalized recommendations on the user's dashboard or a dedicated recommendations page. • **Step 5:** The user reviews and selects a recommendation for further interaction.
Description of the Alternative Sequence	• **Insufficient Data:** If the user has limited activity or preferences, the system displays a default set of popular or trending resources instead of personalized recommendations.
Non functional requirements	• Recommendations must be generated and displayed within a few seconds. • The algorithm must securely handle and process user data.
Postconditions	• The user sees a list of personalized recommendations, enhancing their overall experience on the platform.

UC Name	***UC_16*** *- Purchase Paid Materials via Multiple Payment Methods*
Summary	This use case describes how a user purchases paid materials using various payment methods (bank transfer, credit card, and online wallets including Apple Pay, Google Pay, Samsung Pay, and PayPal).
Dependency	None
Actors	• **Primary Actor:** Student • **Secondary Actor:** Payment Gateway
Preconditions	User must be registered and logged in. The user has selected a paid material for purchase. A valid payment method is available.
Description of the Main Sequence	• **Step 1:** The user selects a paid material and clicks the purchase option. • **Step 2:** The system displays available payment methods: bank transfer, credit card, and online wallets. • **Step 3:** The user chooses a payment method. • **Step 4:** The system processes the payment via the selected method. • **Step 5:** The system confirms the successful transaction. • **Step 6:** The system grants the user access to the purchased material.
Description of the Alternative Sequence	• **Payment Failure:** If the transaction fails, the system notifies the user and offers to retry or select another payment method. • **Invalid Payment Details**: If entered details are invalid, the system prompts the user to re-enter correct payment information.
Non functional requirements	• Payment processing must complete within 10 seconds. • All payment data must be securely encrypted.
Postconditions	• The user gains access to the paid material. • The transaction is recorded in the system.

UC Name	***UC_17*** *- Upload Recorded Lessons*
Summary	This use case describes how a mentor uploads a recorded lesson to the platform for students to access later.
Dependency	None
Actors	• **Primary Actor:** Mentor • **Secondary Actor:** System
Preconditions	Mentor must be registered and logged in. The mentor has a recorded lesson file available in a supported format.
Description of the Main Sequence	• **Step 1:** The mentor navigates to the "Upload Recorded Lesson" section. • **Step 2:** The mentor selects the recorded lesson file from their device. • **Step 3:** The mentor provides any required metadata (e.g., lesson title, description, subject). • **Step 4:** The mentor submits the upload request. • **Step 5:** The system validates the file format and metadata. • **Step 6:** The system uploads the recorded lesson and stores it in the cloud. • **Step 7:** The system confirms the successful upload and makes the lesson available for students.
Description of the Alternative Sequence	• **File Validation Failure:** If the file format is unsupported or metadata is missing, the system displays an error message and prompts the mentor to correct the issue and retry the upload.
Non functional requirements	• The upload process should complete within a reasonable time (e.g., under 10 seconds for typical file sizes). • The file must be securely stored with appropriate access controls.
Postconditions	The recorded lesson is successfully uploaded and accessible on the platform for student viewing.

UC Name	***UC_18** – Uploadig audio-type materials*
Summary	The system allows the students and mentors to upload voice materials and audio files to explain topics.
Dependency	None
Actors	**Primary Actor:** User **Secondary Actor:** System
Preconditions	• The user must be registered and logged into the platform.
Description of the Main Sequence	1. The user uploads an audio-based study material. 2. The system generates an automatic transcript. 3. Users can access the audio file and its transcript. 4. The system provides text-to-speech options for written materials.
Description of the Alternative Sequence	• If transcription fails, the system notifies the uploader and offers manual input. • If text-to-speech is not available for a specific language, the system informs the user.
Non-functional requirements	• The system should support multiple languages. • Audio and text data should be accessible through screen readers.
Postconditions	• Audio-based materials are successfully uploaded and accessible with enhanced accessibility features.

UC Name	***UC_19 –*** *Selling/Lending used books*
Summary	The system allows book selling and lending from students to other students.
Dependency	None
Actors	**Primary Actor:** Student **Secondary Actor:** Student
Preconditions	• The seller must be registered and logged into the platform. • The seller must provide book details (e.g., price, condition).
Description of the Main Sequence	1. A seller lists a used textbook with relevant details. 2. A buyer searches for a textbook and views the listing. 3. The buyer initiates a conversation with the seller through messaging. 4. The buyer and seller agree on the transaction. 5. The system facilitates secure payment and confirms the purchase.
Description of the Alternative Sequence	• If the buyer backs out before purchase, the listing remains active. • If payment fails, the transaction is not completed, and the system notifies the buyer.
Non-functional requirements	• Secure payment handling with encryption. • Review and reporting mechanism for sellers and buyers.
Postconditions	• A successful textbook sale or purchase transaction is completed

UC Name	***UC_20 – Account suspension***
Summary	The system allows the users to permanently delete their account.
Dependency	None
Actors	**Primary Actor:** User **Secondary Actor:** System
Preconditions	• The user must be logged into their account. • The system must verify the user's identity before account deletion or suspension.
Description of the Main Sequence	1. The user navigates to the account settings page. 2. The user selects either the "Suspend Account" option. 3. The system prompts the user to confirm their action. 4. If the user confirms: 1. The system deactivates the account, allowing reactivation later. 5. The system logs the action and notifies the user via email.
Description of the Alternative Sequence	• If the user decides not to proceed, they can cancel the action before confirmation.
Non-functional requirements	• Suspended accounts should allow reactivation within a predefined period.
Postconditions	• The user's account is suspended temporarily.

UC Name	***UC_21 –*** *Account deletion*
Summary	The system allows the users to permanently delete their account.
Dependency	None
Actors	**Primary Actor:** User **Secondary Actor:** System
Preconditions	• The user must be logged into their account. • The system must verify the user's identity before account deletion.
Description of the Main Sequence	1. The user navigates to the account settings page. 2. The user selects either the "Delete Account" option. 3. The system prompts the user to confirm their action. 4. If the user confirms: o The system permanently removes the user's account and associated data. 5. The system logs the action and notifies the user via email.
Description of the Alternative Sequence	• If the user attempts to delete their account but has active transactions or pending payments, the system prevents deletion and notifies the user. • If the user decides not to proceed, they can cancel the action before confirmation.
Non-functional requirements	
Postconditions	• The user's account is deleted permanently.

UC Name	***UC_22 –*** *Changing credentials*
Summary	The system allows the user to change their email, password, phone number or birthdate using one of these credentials for verification.
Dependency	None
Actors	**Primary Actor:** User **Secondary Actor:** System
Preconditions	• The user must be logged into their account. • The system must verify the user's identity before making changes.
Description of the Main Sequence	1. The user accesses the account settings page. 2. The user selects the credential they wish to modify (password, birthdate, phone number). 3. The system prompts the user to verify their identity (e.g., entering the current password or a verification code). 4. The user enters the new credentials. 5. The system validates the new credentials and updates the user's account. 6. The system confirms the update and notifies the user via email or SMS.
Description of the Alternative Sequence	• If the user enters incorrect verification details, the system prevents changes and prompts re-entry. • If the user attempts to set an invalid birthdate (e.g., below the minimum age requirement), the system rejects the change. • If the password does not meet security criteria, the system prompts the user to enter a stronger one.
Non-functional requirements	• Password changes must enforce security best practices (e.g., minimum length, special characters). • Sensitive changes (e.g., phone number modification) should require two-factor authentication (2FA).
Postconditions	• The user's credentials are successfully updated in the system.

UC Name	***UC_23** – Students requesting help*
Summary	Allows students to request help from mentors or other students on their schoolwork.
Dependency	None
Actors	**Primary Actor:** Student **Secondary Actor:**Mentor
Preconditions	1. The student must be logged into the system. 2. There must be available mentors or other students willing to provide help.
Description of the Main Sequence	1. Student logs into the system. 2. Student navigates to the "Help Request" section. 3. Student selects the type of help needed (e.g., assignment, concept clarification). 4. Student submits the help request. 5. System matches the request with an available mentor or peer. 6. Mentor or peer accepts the request. 7. System notifies the student that help has been assigned.
Description of the Alternative Sequence	1. Student logs into the system. 2. Student navigates to the "Help Request" section. 3. Student submits the help request. 4. No mentors or peers are available. 5. System notifies the student that no one is available and suggests trying again later.
Non-functional requirements	The system should notify the student within 5 seconds of request submission whether help is available.
Postconditions	Help request is successfully assigned, or the student is notified of the unavailability of help.

UC Name	***UC_24 –*** *Screenshot/Screenrecording blocking*
Summary	Prevents students from taking screenshots or screen recordings of mentor-shared materials to protect intellectual property.
Dependency	None
Actors	**Primary Actor:** Mentor **Secondary Actor:**Student, System
Preconditions	1. The mentor must have uploaded or shared the material. 2. The student must have access to the shared material.
Description of the Main Sequence	1. Mentor uploads and shares study material. 2. Student opens the shared material. 3. System detects any attempt to take a screenshot or screen recording. 4. System blocks the screenshot or recording and displays a warning message. 5. System logs the attempt for security monitoring.
Description of the Alternative Sequence	1. Mentor uploads and shares study material. 2. Student opens the shared material. 3. System fails to detect a screenshot attempt due to device limitations. 4. No warning is displayed, but the session is logged for review.
Non-functional requirements	The system should detect and block screenshot and recording attempts within 1 second of detection and log the attempt for future reference.
Postconditions	Screenshot or recording attempt is blocked, and the attempt is logged in the system.

UC Name	***UC_25 –*** *Rate, report, review materials*
Summary	Allows students to review, rate and report study materials.
Dependency	None
Actors	**Primary Actor**: Student
Preconditions	1. The student must be logged into the system. 2. The student must have accessed or interacted with the study material.
Description of the Main Sequence	1. Student navigates to a study material page. 2. Student selects the option to leave a review and rate the material. 3. Student enters a review and selects a rating. 4. System saves the review and rating. 5. The study material's overall rating is updated based on the new input.
Description of the Alternative Sequence	0. Student identifies inappropriate study material. 0. Student selects the option to report the material. 0. System prompts the student to provide a reason for reporting. 0. Student submits the report. 0. System logs the report and notifies administrators for review.
Non-functional requirements	• The rating and review submission should be intuitive and quick. • The system should prevent duplicate reviews from the same user. • The reporting system should immediately flag content for review by administrators.
Postconditions	• The review and rating are successfully saved and displayed for other users. • Reported materials are logged and sent for administrative review.

UC Name	***UC_26 –*** *Custom goals creation*
Summary	Allows students to create custom study goals and receive reminders to stay on track.
Dependency	None
Actors	**Primary Actor**: Student
Preconditions	The student must be logged into the system.
Description of the Main Sequence	1. Student navigates to the goal-setting feature. 2. Student creates a study goal and sets a target deadline. 3. Student enables reminder notifications. 4. System saves the goal and schedules reminders. 5. System sends notifications to remind the student of their goal.
Description of the Alternative Sequence	1. Student navigates to the goal-setting feature. 2. Student creates a study goal but does not set a deadline. 3. System prompts the student to enter a deadline before saving.
Non-functional requirements	• The reminder system should be customizable (e.g., frequency, notification type). • Notifications should be sent without significant system delay.
Postconditions	• The study goal is successfully created and saved in the system. • Reminder notifications are sent according to the student's preferences.

UC Name	***UC_27*** **-** *Bookmarking study materials*
Summary	Allows students to bookmark study materials for quick access.
Dependency	None
Actors	**Primary Actor**: Student
Preconditions	1. The student must be logged into the system. 2. The study material must be available.
Description of the Main Sequence	1. Student navigates to a study material. 2. Student selects the option to bookmark the material. 3. System saves the bookmarked material in the student's profile. 4. Student can access bookmarked materials from the profile.
Description of the Alternative Sequence	1. Student attempts to bookmark a study material without logging in. 2. System prompts the student to log in before proceeding.
Non-functional requirements	• The bookmarking feature should be easily accessible from the study material page. • Bookmarked materials should load quickly when accessed.
Postconditions	• The study material is successfully bookmarked and stored in the student's profile for quick access.

UC Name	***UC_28*** **-** *Creating Quizzes*
Summary	The platform shall allow mentors to create quizzes for students to complete during or after a lesson.
Dependency	None
Actors	Actors: Mentor (p), System (s)
Preconditions	-The mentor must be logged into the platform.

Description of the Main Sequence	-The mentor selects the "Create Quiz" option. -The mentor fills in the necessary details (e.g., questions, options, deadlines). -The system saves the quiz details. -The system assigns the quiz to selected students.
Description of the Alternative Sequence	-If the mentor does not provide all required details, the system prompts them to complete the missing fields. -If there is an error while creating the quiz, the system logs the issue and notifies the mentor.
Non functional requirements	-The system shall ensure that quizzes load within 2 seconds to provide a smooth user experience.
Postconditions	-The quiz is available to the students.

UC Name	UC_29 - Tracking Time Spent on Study Activities
Summary	The system shall include a feature for students to track their time spent on different study activities and sessions.
Dependency	None
Actors	Student (p), System (s)
Preconditions	-The student must be logged into the platform. -The student must enable the time-tracking feature in their preferences.
Description of the Main Sequence	-The student starts a study activity and selects "Start Session." -The system begins recording the time for the activity. -The student completes the study activity and selects "End Session." -The system logs the total time spent on the activity and updates the student’s dashboard.
Description of the Alternative Sequence	-If the student forgets to end the session, the system automatically stops the timer after a predefined idle period. -If there is a system error during time tracking, the system logs the error and notifies the student.
Non functional requirements	-The system must record time in real-time without significant delays.
Postconditions	-The student’s study time is recorded and visible in their dashboard.

UC Name	***UC_30*** *- Monthly Challenges or Competitions*
Summary	The system shall enable students to participate in monthly challenges or competitions related to their studies, with rewards for completion (e.g., badges, certificates).
Dependency	None
Actors	Student (p), System (s)
Preconditions	-The student must be logged into the platform. -The student must have access to the monthly challenge.
Description of the Main Sequence	-The student views available monthly challenges or competitions. -The student selects a challenge to participate in. -The student completes the required activities to finish the challenge. -Upon completion, the system rewards the student with badges or certificates. -The rewards are added to the student's profile and displayed on their dashboard.
Description of the Alternative Sequence	-If the student fails to complete the challenge within the set timeframe, the system sends a reminder and allows them to reattempt. -If the challenge cannot be accessed due to a system error, the system logs the error and informs the student.
Non functional requirements	-The system shall handle high traffic volumes during competitions without performance degradation, ensuring all users can participate seamlessly
Postconditions	-The student is rewarded with badges/certificates and their participation is recorded.

UC Name	UC_31 - Messaging Between Students and Mentors
Summary	The system allows students and mentors to communicate through a built-in messaging feature.
Dependency	None
Actors	Student (p), Mentor (p), System (s)
Preconditions	The user (student or mentor) must be logged into the platform. Messaging must be enabled for both parties.
Description of the Main Sequence	-The student or mentor selects the "Messages" feature from the platform. -The user searches for a mentor/student or selects a contact from the list. -The system opens a chat window and allows the user to type a message. -The user sends the message. -The system delivers the message to the recipient in real-time. -The recipient receives a notification and can respond.
Description of the Alternative Sequence	-If the recipient is offline, the system stores the message and delivers it when they are online. -If messaging is disabled for a user, the system notifies the sender that communication is restricted. -If there is a system error, the message is queued for retry or the user is notified of the failure.

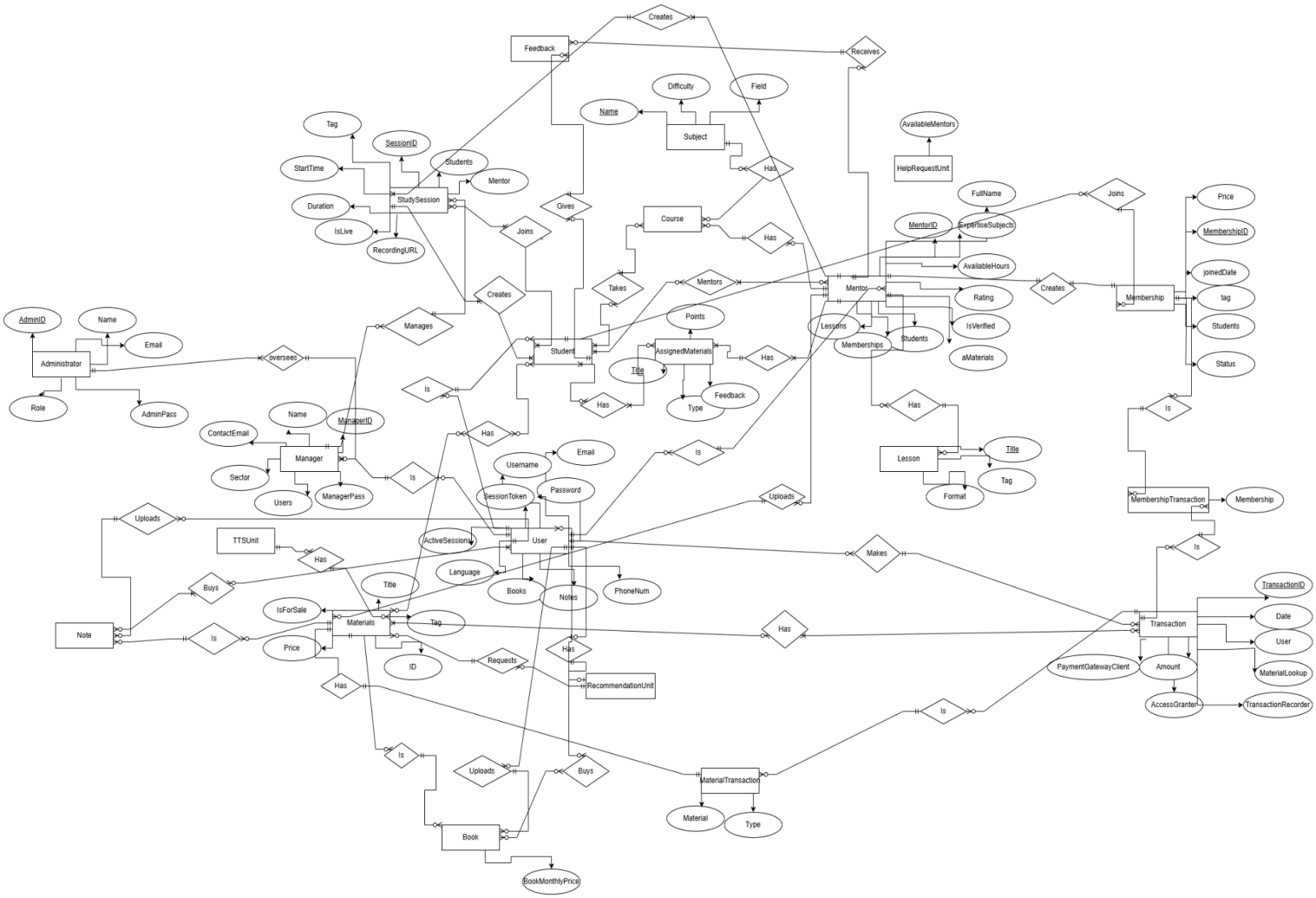
Non functional requirements	-The system shall ensure end-to-end encryption for all messages exchanged between users to maintain privacy and data security
Postconditions	-The message is successfully sent and received.

UC Name	UC_32 - Creating Accounts
Summary	The system shall allow users to create personal profiles.The system allows students to select and modify their interest-subjects when first registering. Mentors shall present their degree to be registered as such.
Dependency	None
Actors	Student (p), Mentor(p), System (s)
Preconditions	-The user (student or mentor) must be registering for an account. -Mentors must provide proof of their degree for verification.
Description of the Main Sequence	-The user selects the "Create Profile" option during registration or from the settings. The system prompts the user to enter personal details (e.g., name, email, profile picture). **For students**: -The system provides a list of subjects for the student to select their interests. -The student selects or modifies their interest-subjects. **For mentors**:

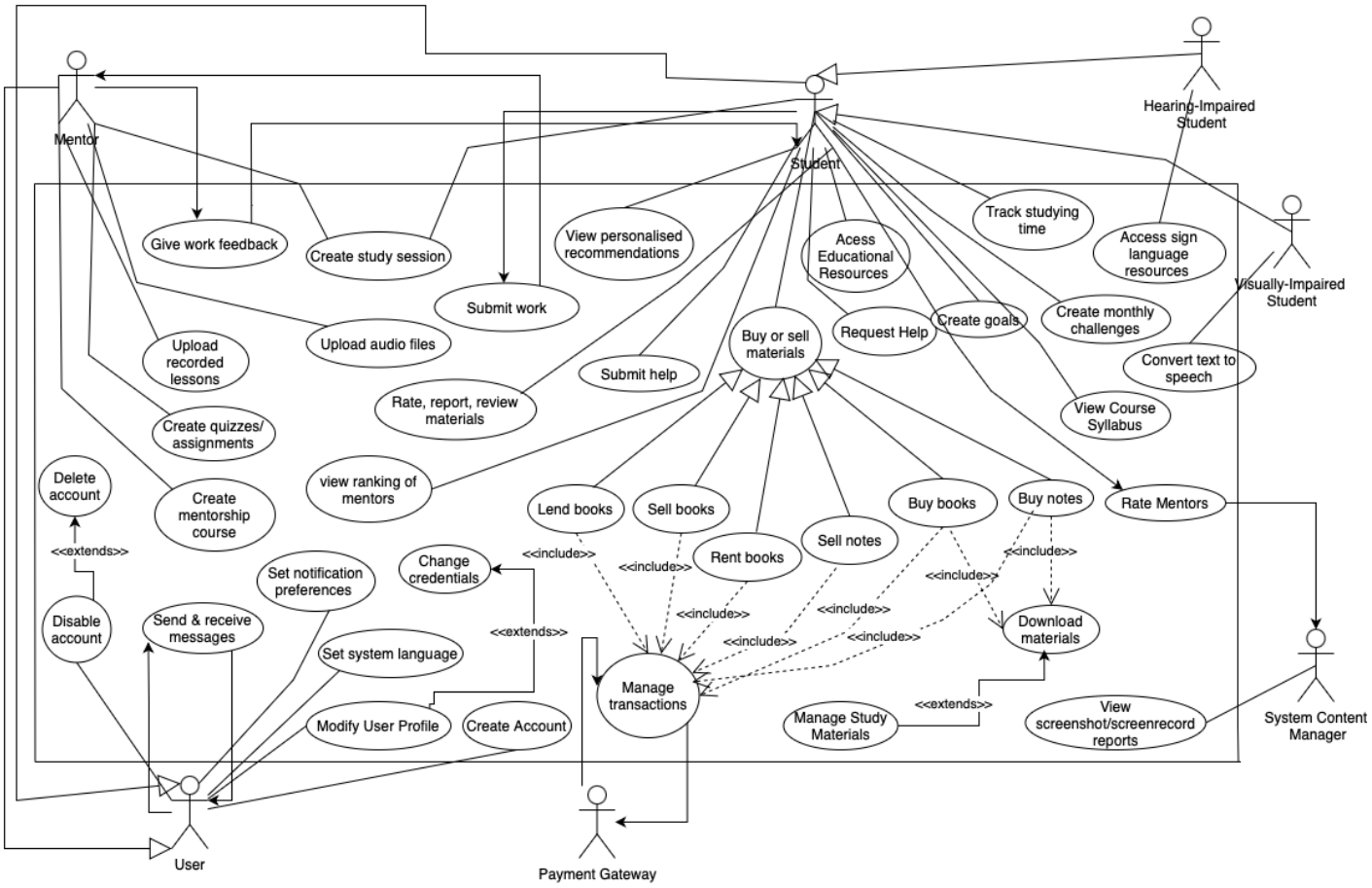
	-The mentor uploads proof of their degree. -The system verifies the document before approving mentor status. -The user submits the profile information. -The system saves the profile and displays it on the platform.
Description of the Alternative Sequence	-If the user does not complete all required fields, the system prompts them to enter missing details. -If the mentor's degree verification fails, the system notifies them and requests resubmission. -If there is a system error while saving the profile, the system logs the issue and prompts the user to retry later
Non functional requirements	-The system must securely store and protect user data. -Mentor verification should be handled promptly to avoid delays in registration.
Postconditions	-The user's profile is created and viewable to others.

5. Diagrams

5.1 ER diagram

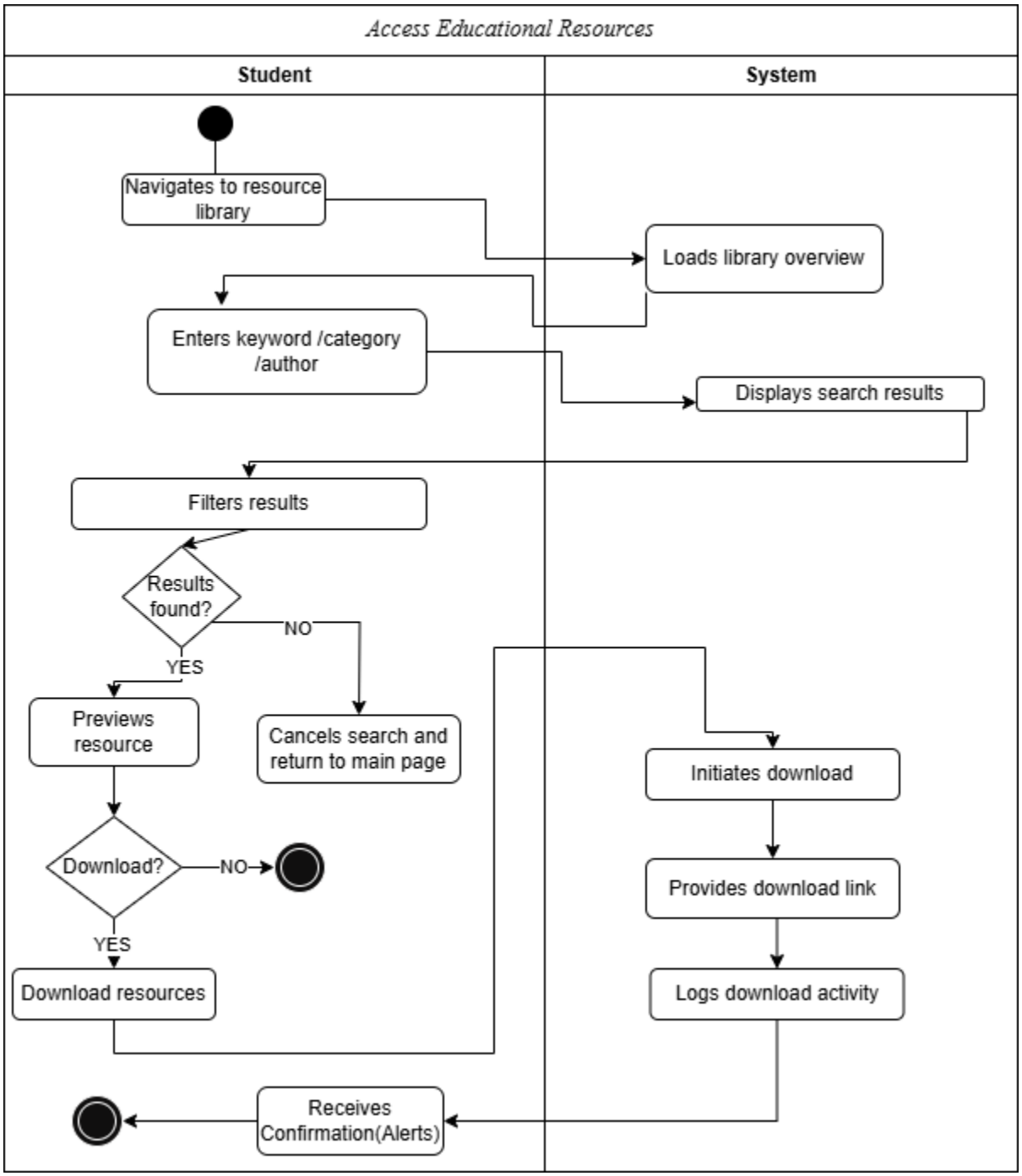


5.2 Use Case Diagram

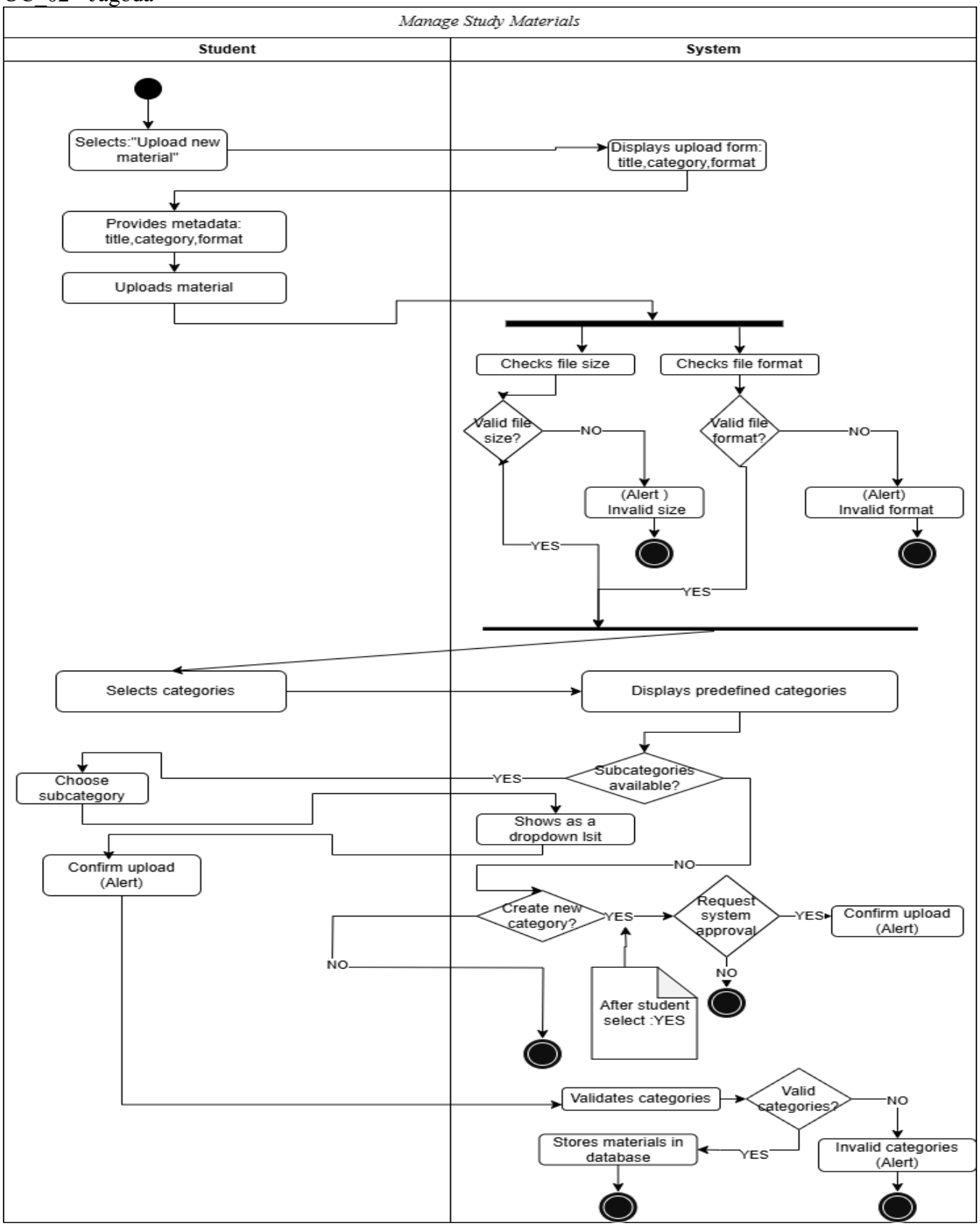


5.3 Activity Diagrams

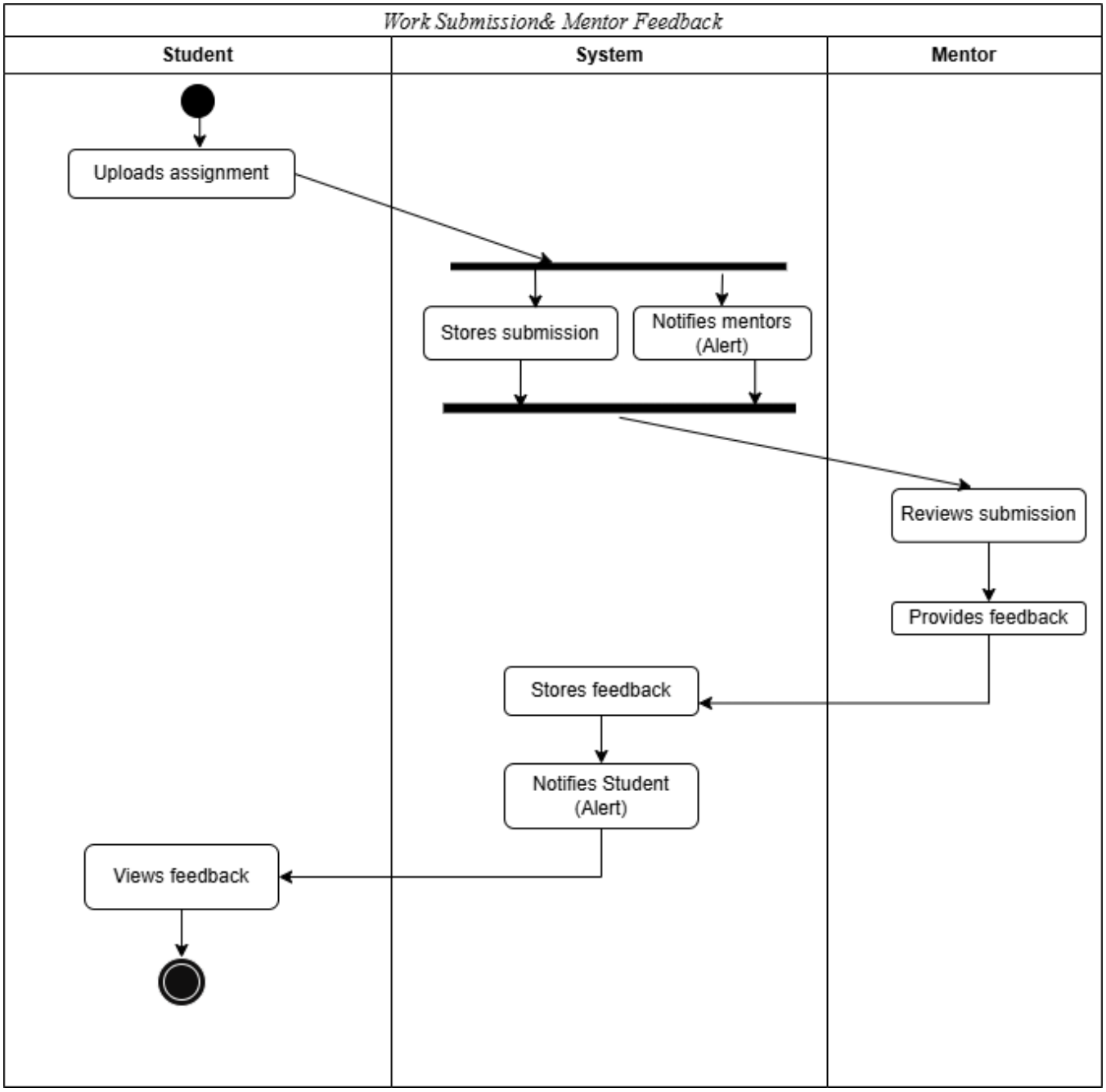
UC_01- Jagoda



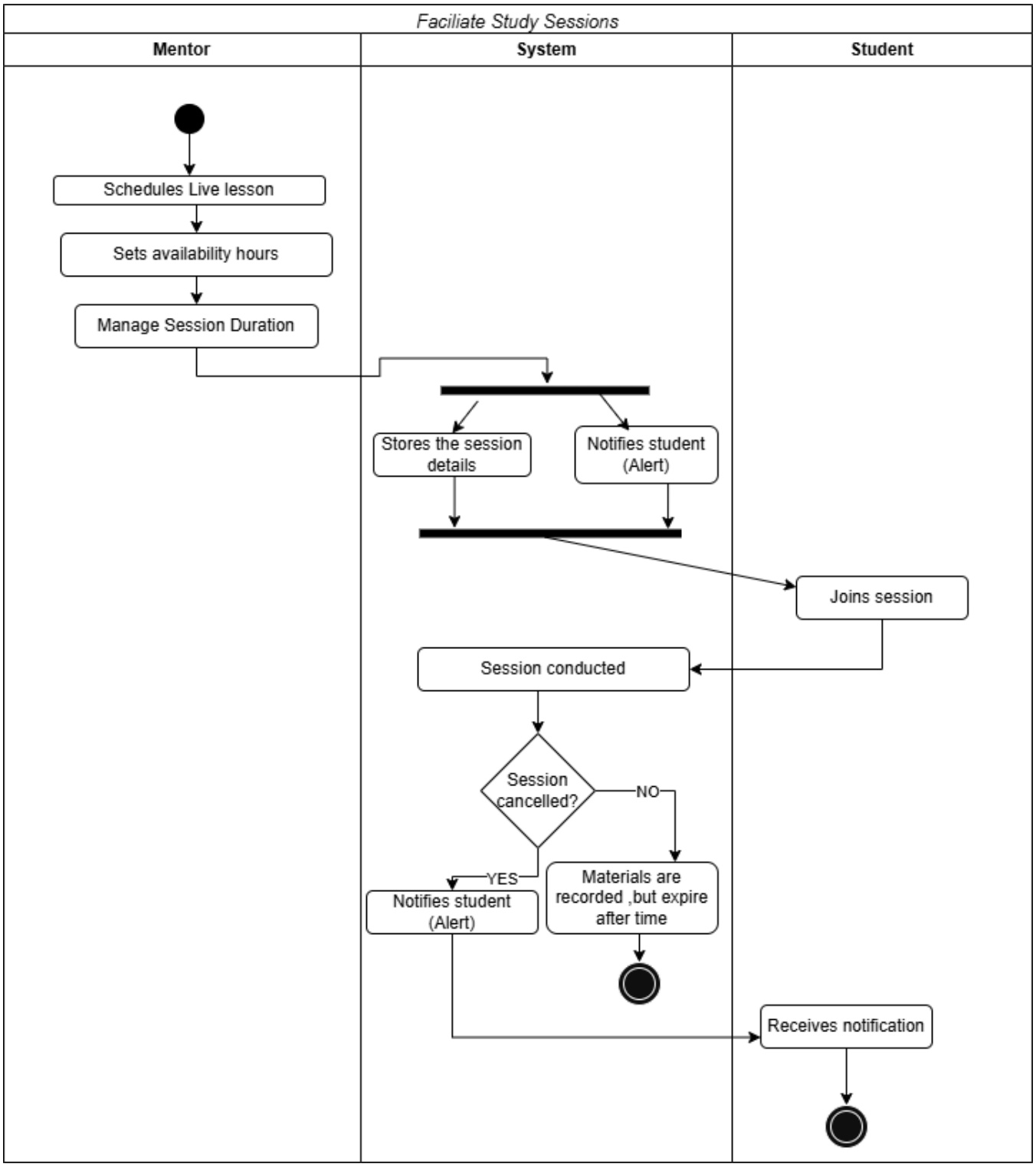
UC_02 - Jagoda


Manage Study Materials
Student
System
Selects:"Upload new material"
Displays upload form: title,category,format
Provides metadata: title,category,format
Uploads material
Checks file size
Checks file format
Valid file size?
NO
Valid file format?
NO
(Alert) Invalid size
(Alert) Invalid format
YES
YES
Selects categories
Displays predefined categories
Choose subcategory
YES
Subcategories available?
Shows as a dropdown lsit
Confirm upload (Alert)
NO
Create new category?
YES
Request system approval
YES
Confirm upload (Alert)
NO
NO
After student select :YES
Validates categories
Valid categories?
NO
Stores materials in database
YES
Invalid categories (Alert)

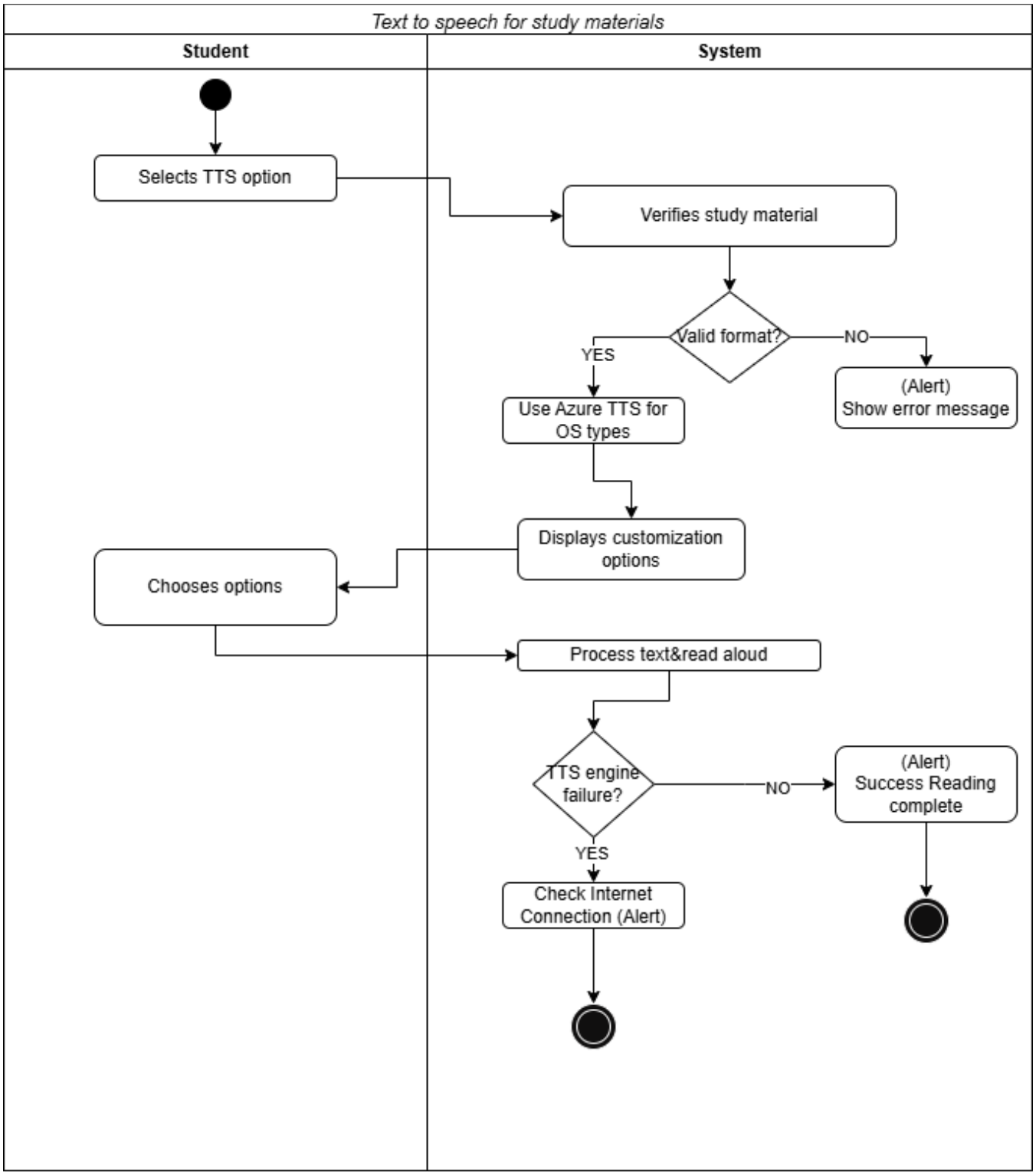
UC_03 - Jagoda


Work Submission& Mentor Feedback
Student
System
Mentor
Uploads assignment
Stores submission
Notifies mentors (Alert)
Reviews submission
Provides feedback
Stores feedback
Notifies Student (Alert)
Views feedback

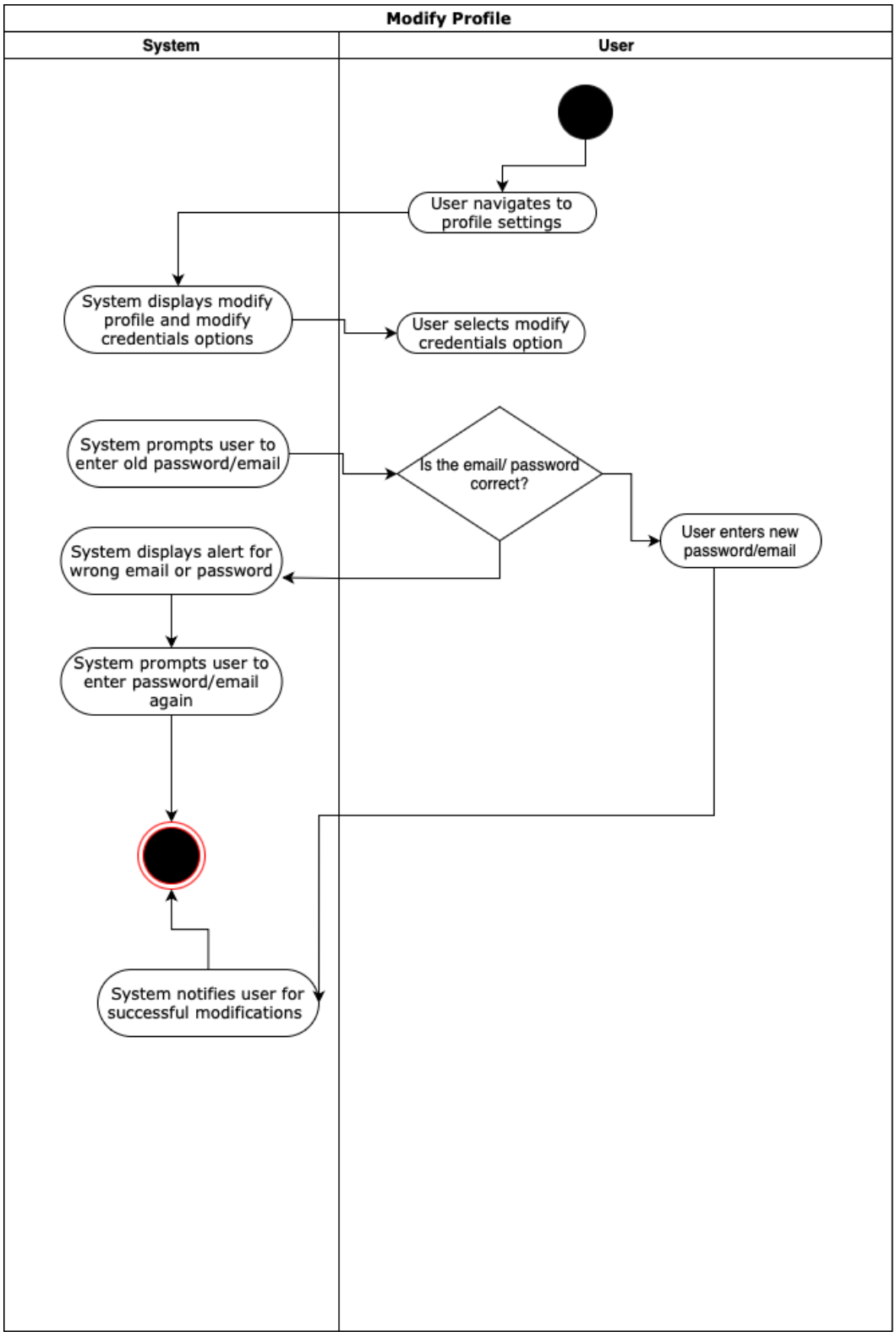
UC_04 - Jagoda



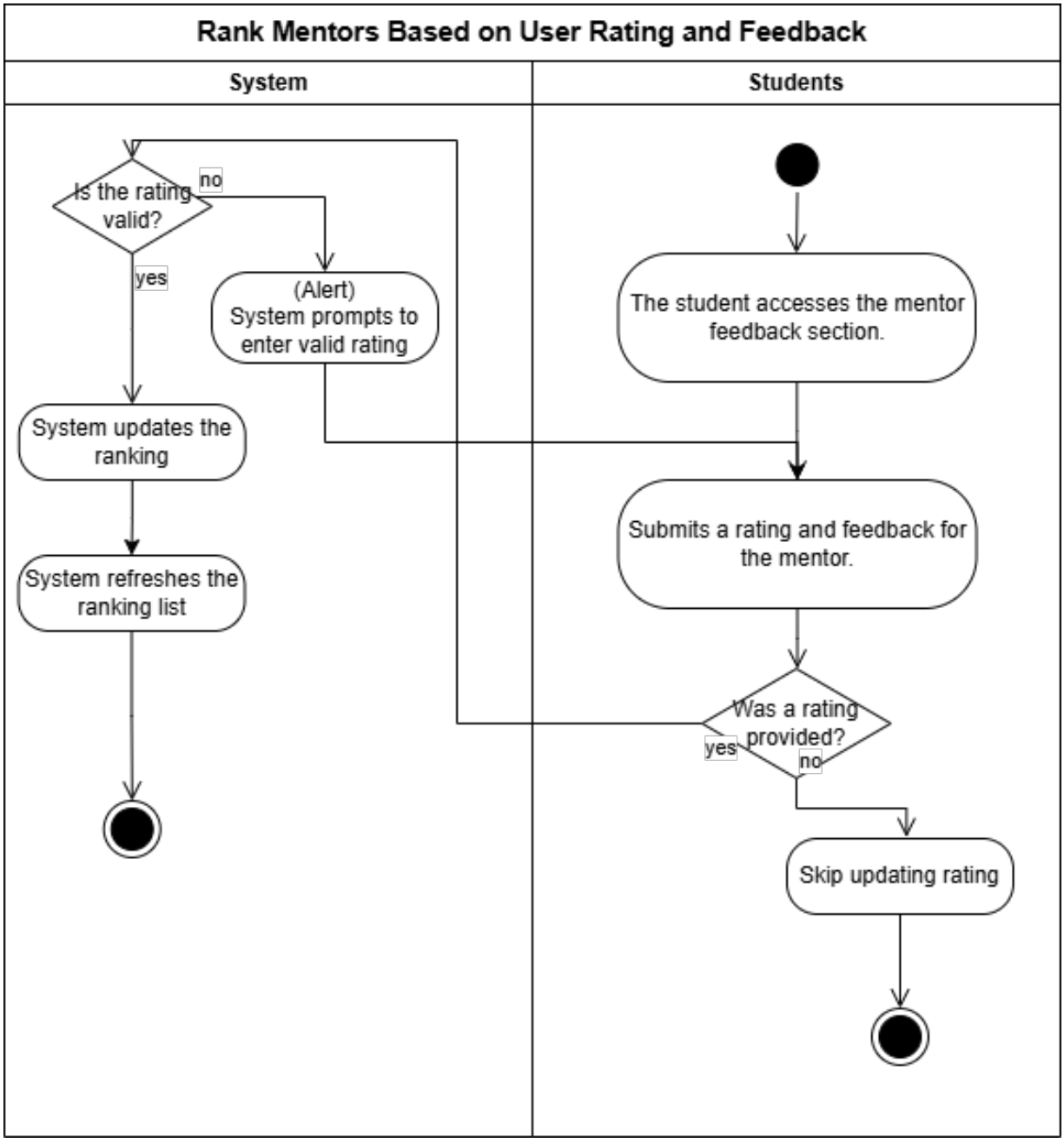
UC_05 - Jagoda

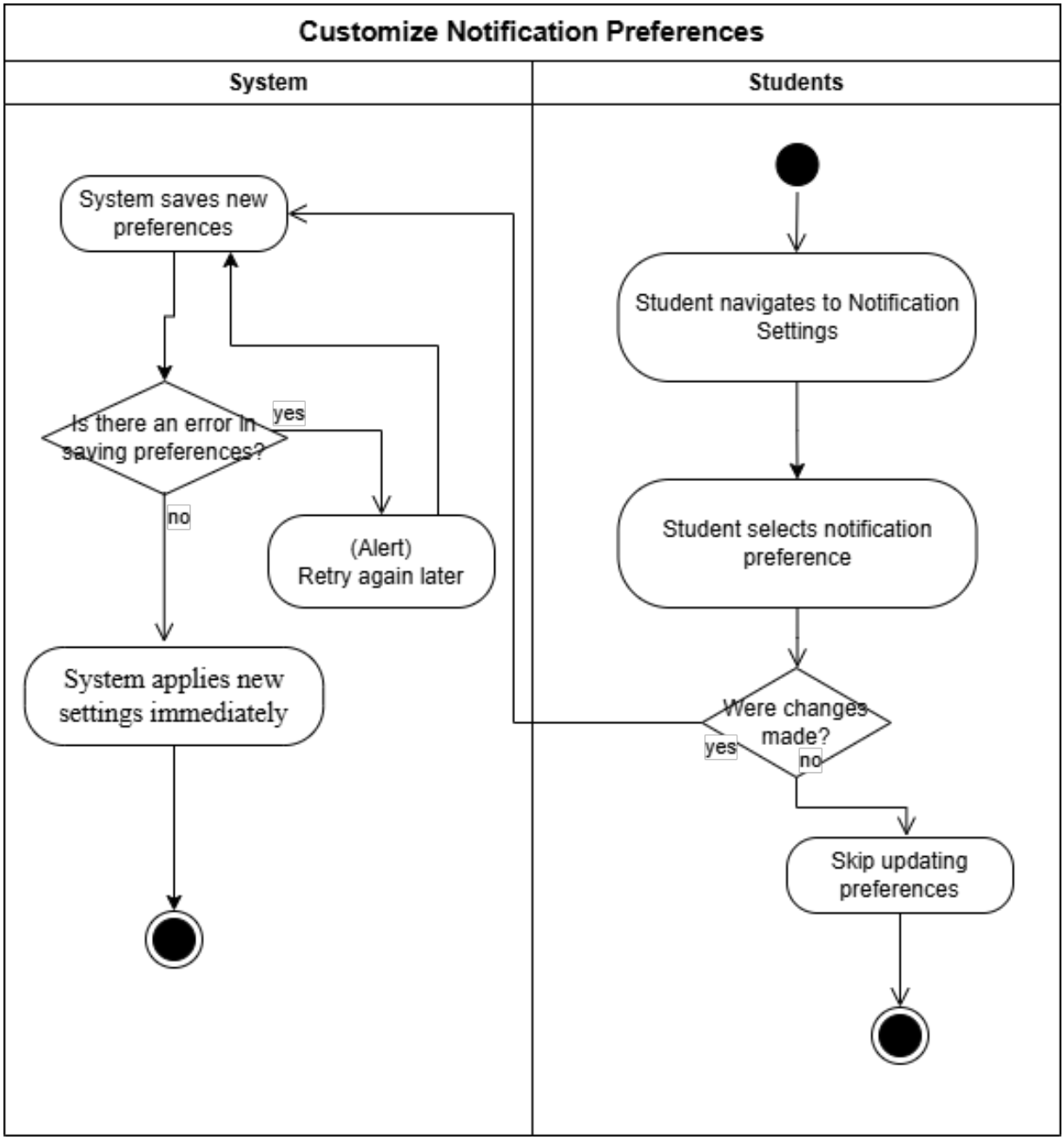

Text to speech for study materials
Student
System
Selects TTS option
Verifies study material
Valid format?
YES
NO
(Alert)
Show error message
Use Azure TTS for
OS types
Displays customization
options
Chooses options
Process text&read aloud
TTS engine
failure?
NO
(Alert)
Success Reading
complete
YES
Check Internet
Connection (Alert)

UC_06 - Halil

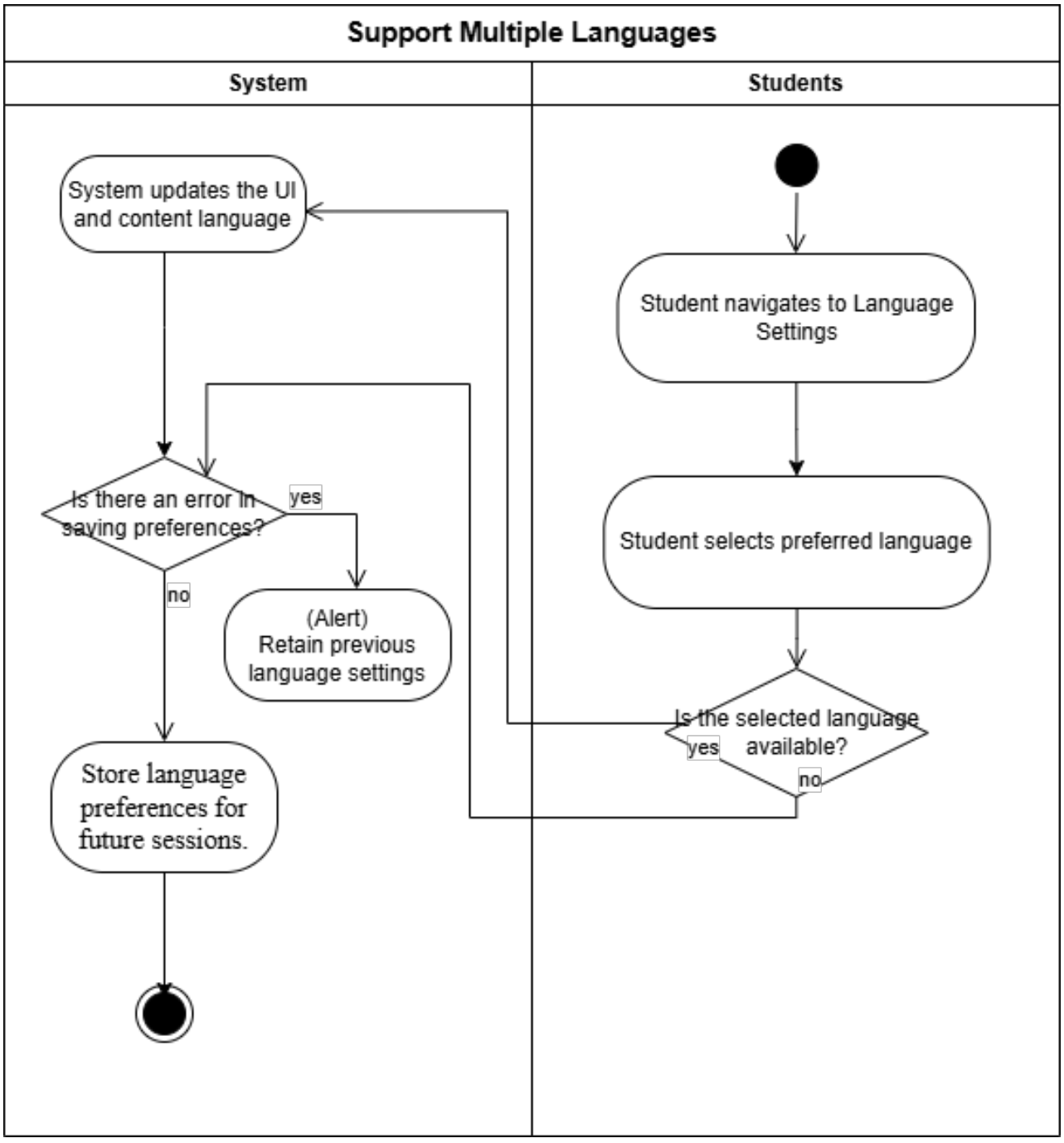

Modify Profile
System
User
User navigates to profile settings
System displays modify profile and modify credentials options
User selects modify credentials option
System prompts user to enter old password/email
Is the email/ password correct?
User enters new password/email
System displays alert for wrong email or password
System prompts user to enter password/email again
System notifies user for successful modifications

UC_7 - Dionis

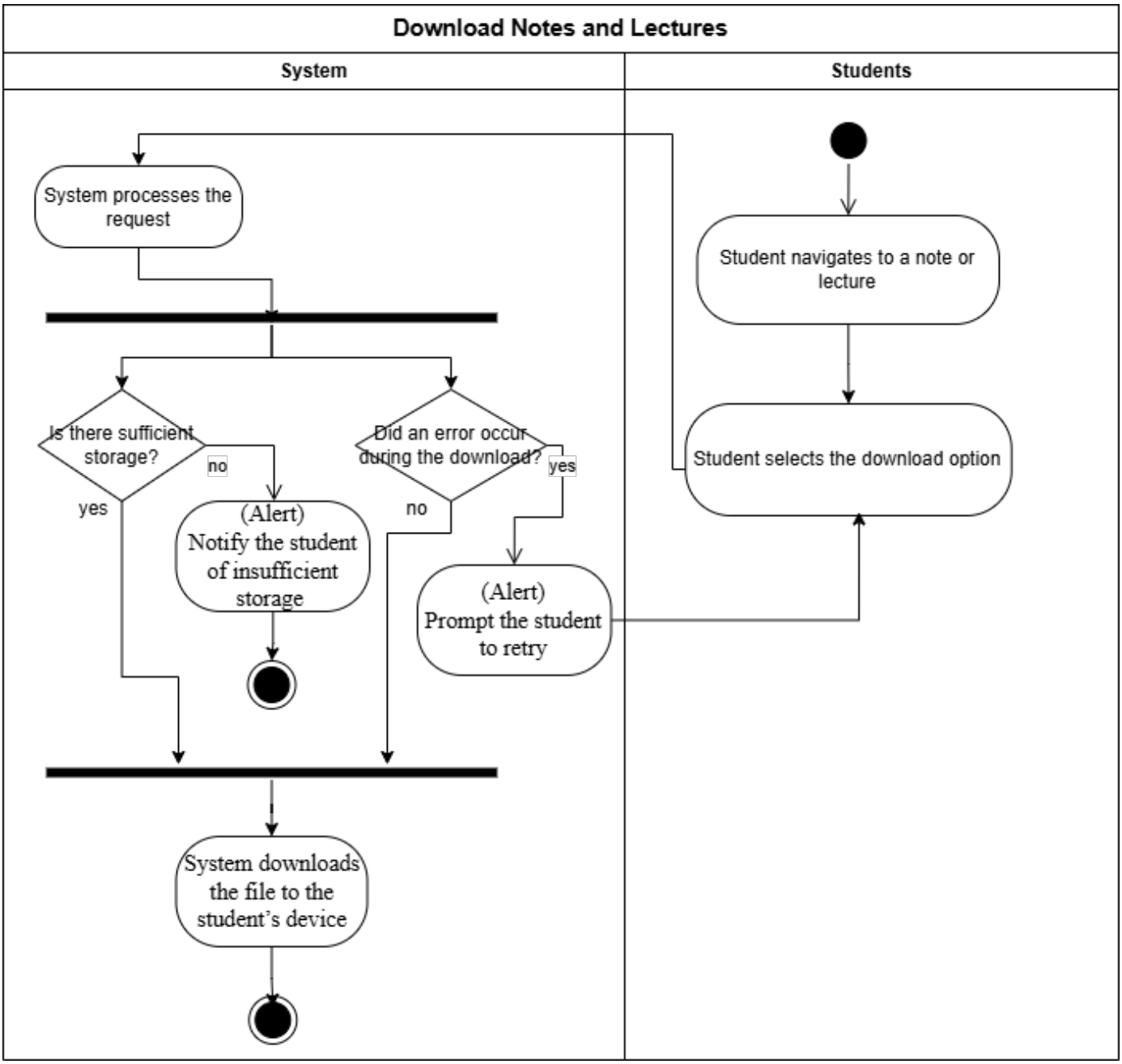

Rank Mentors Based on User Rating and Feedback
System
Students
Is the rating valid?
no
yes
(Alert)
System prompts to enter valid rating
System updates the ranking
System refreshes the ranking list
The student accesses the mentor feedback section.
Submits a rating and feedback for the mentor.
Was a rating provided?
yes
no
Skip updating rating

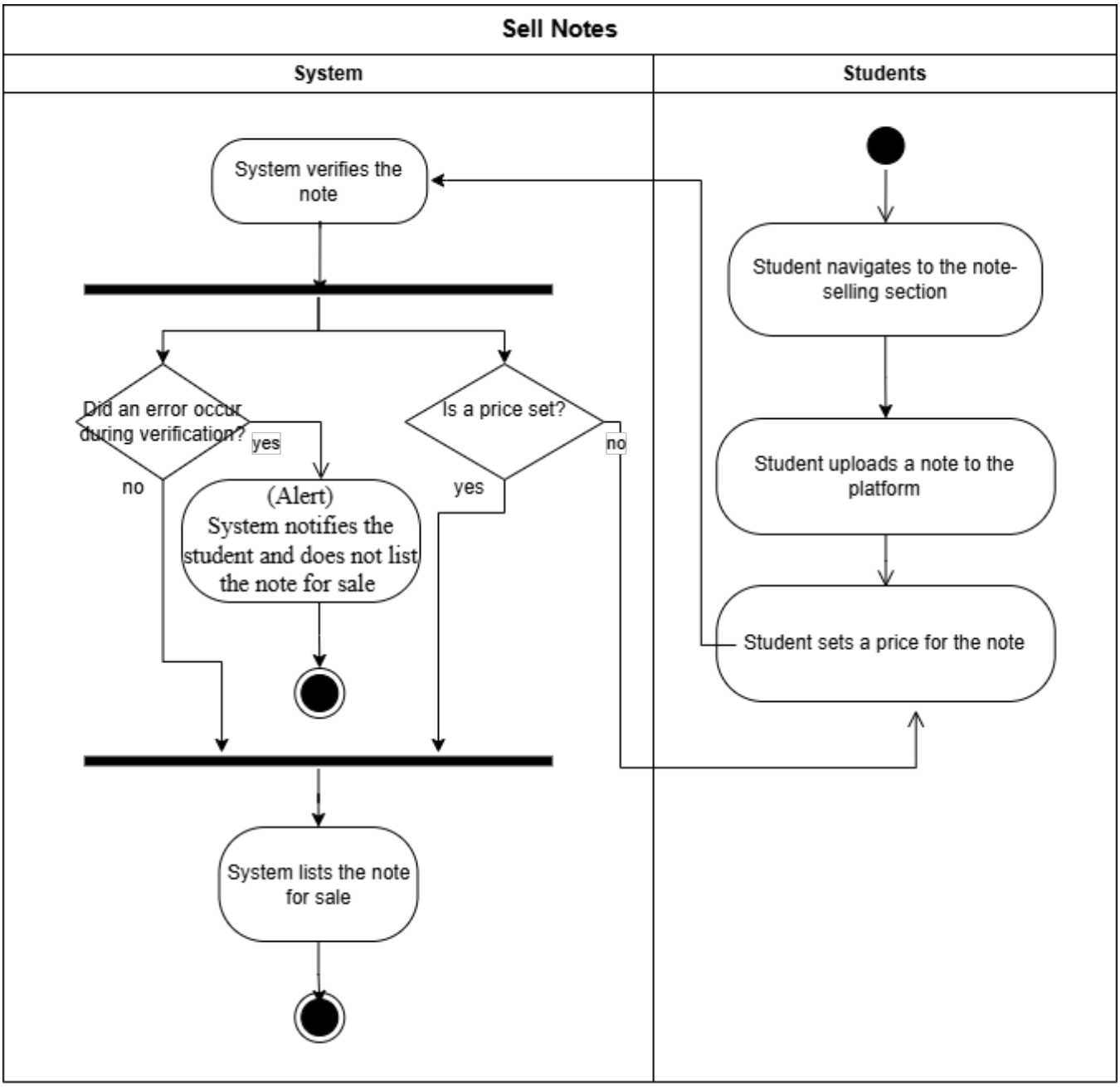

Customize Notification Preferences
System
Students
System saves new preferences
Is there an error in saving preferences?
yes
no
(Alert) Retry again later
System applies new settings immediately
Student navigates to Notification Settings
Student selects notification preference
Were changes made?
yes
no
Skip updating preferences

UC_9 - Dionis

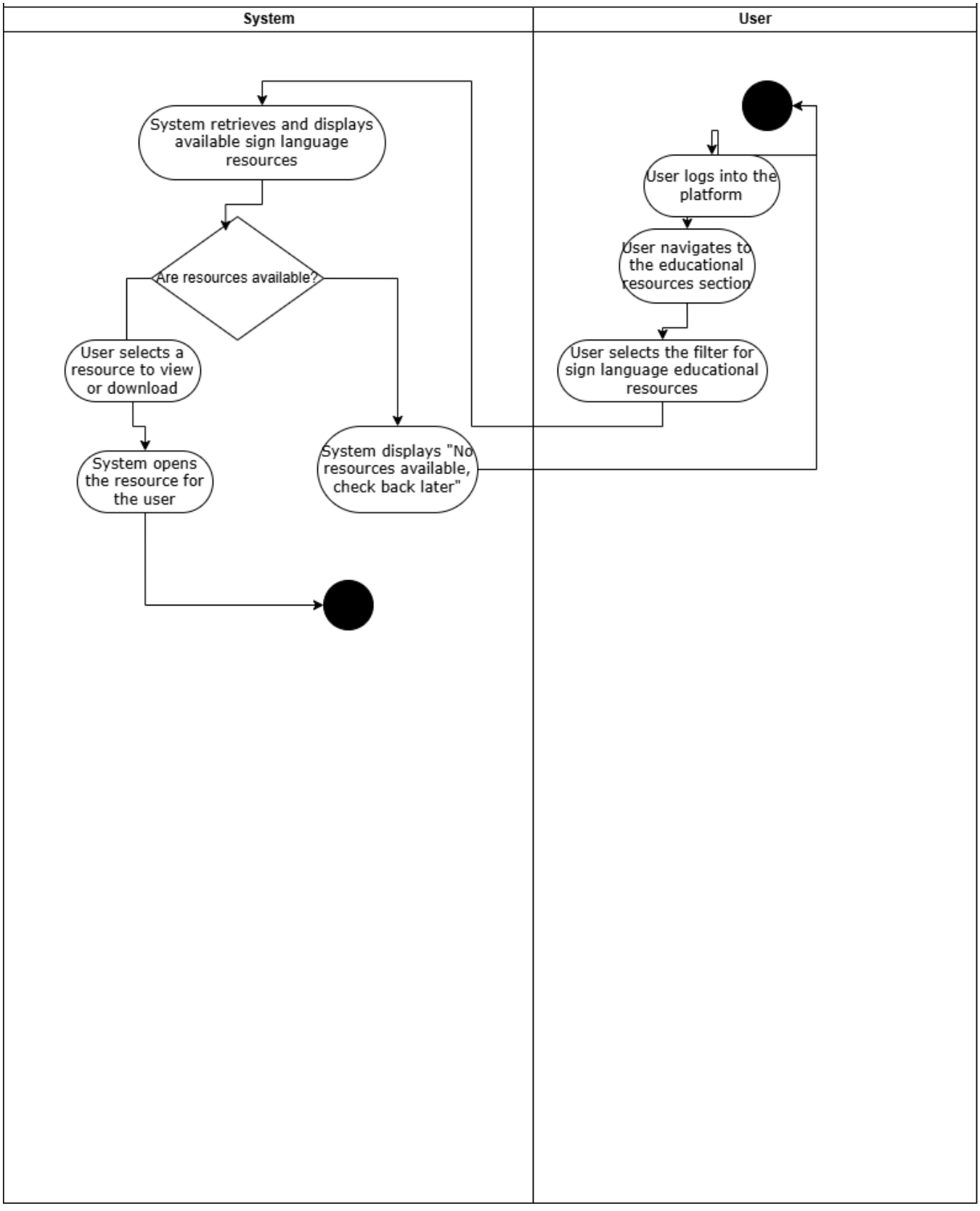


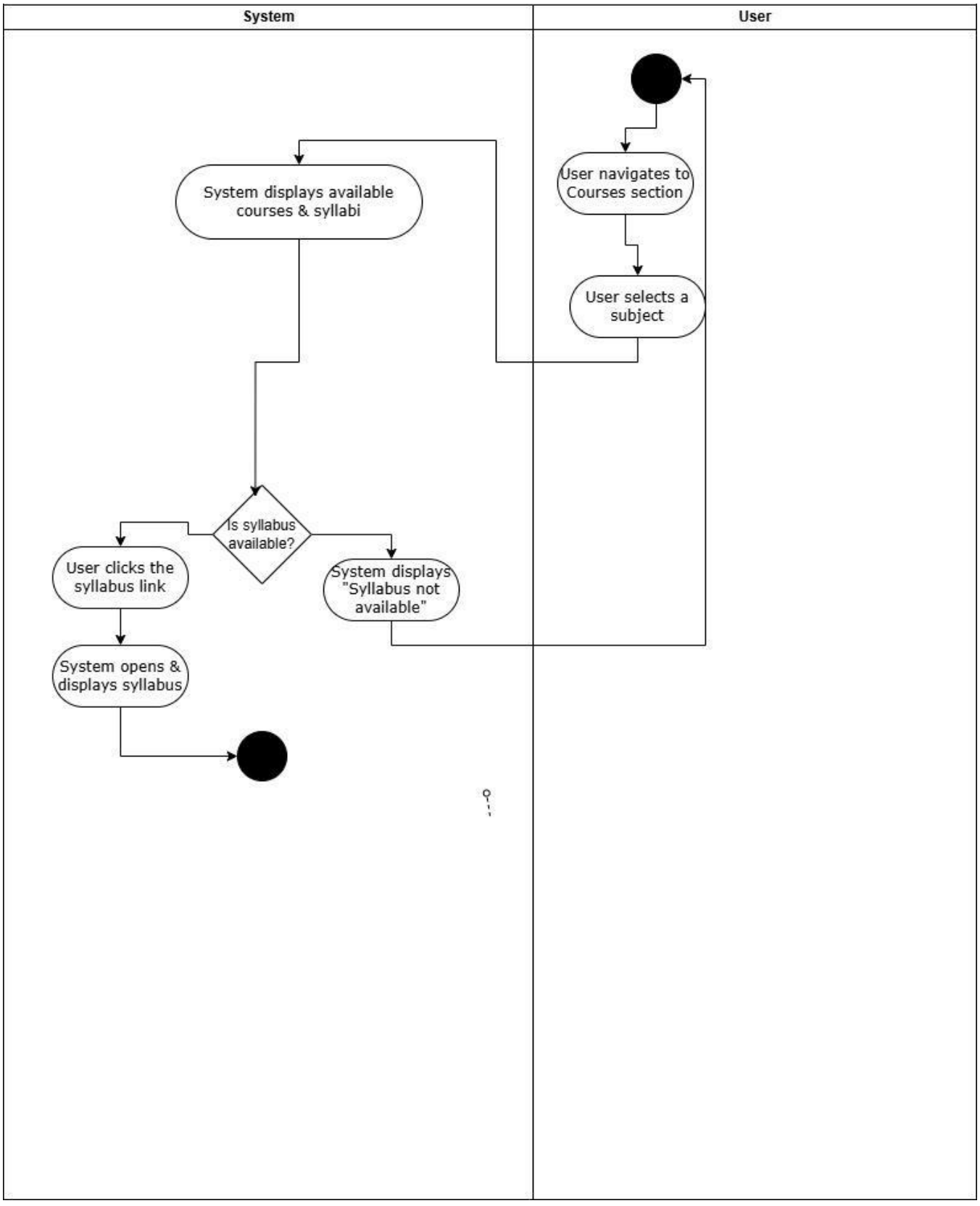
UC_10 - Dionis

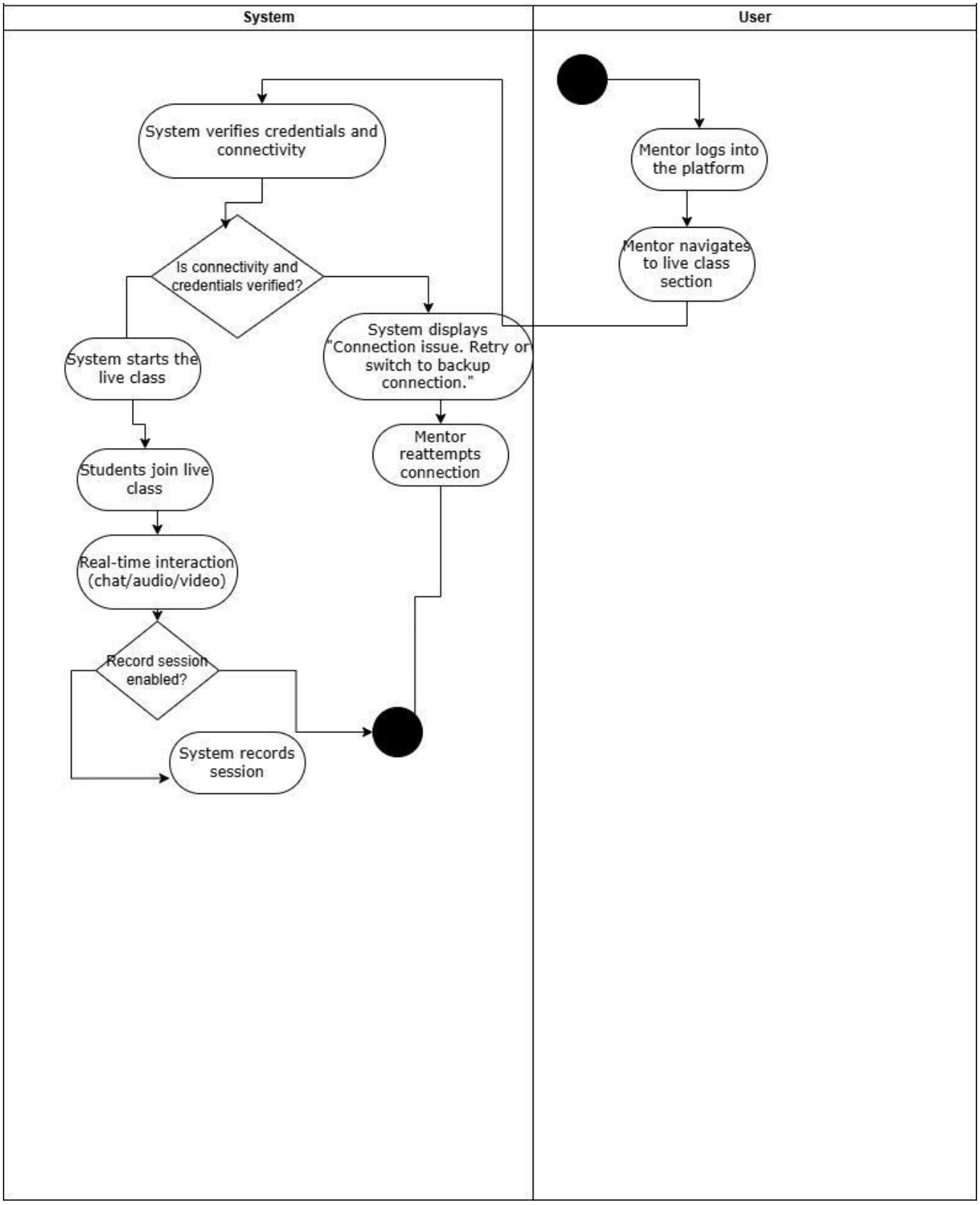

Download Notes and Lectures
System
Students
System processes the request
Student navigates to a note or lecture
Is there sufficient storage?
no
yes
Did an error occur during the download?
yes
no
Student selects the download option
(Alert) Notify the student of insufficient storage
(Alert) Prompt the student to retry
System downloads the file to the student's device


Sell Notes
System
Students
System verifies the note
Student navigates to the note-selling section
Did an error occur during verification?
yes
no
(Alert) System notifies the student and does not list the note for sale
Is a price set?
no
yes
Student uploads a note to the platform
Student sets a price for the note
System lists the note for sale

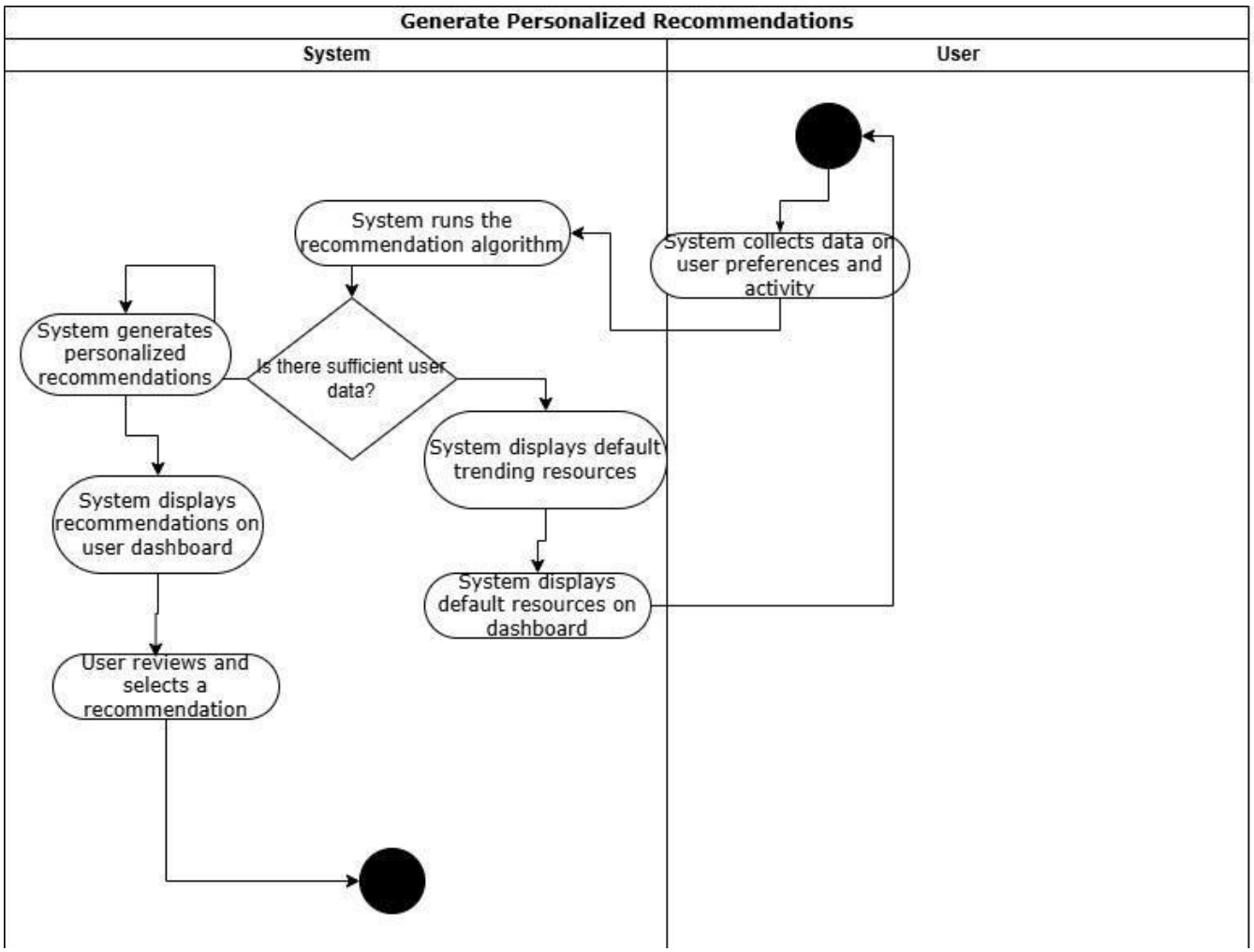
UC_12 - Marvi


System
User
System retrieves and displays available sign language resources
Are resources available?
User selects a resource to view or download
System opens the resource for the user
System displays "No resources available, check back later"
User logs into the platform
User navigates to the educational resources section
User selects the filter for sign language educational resources

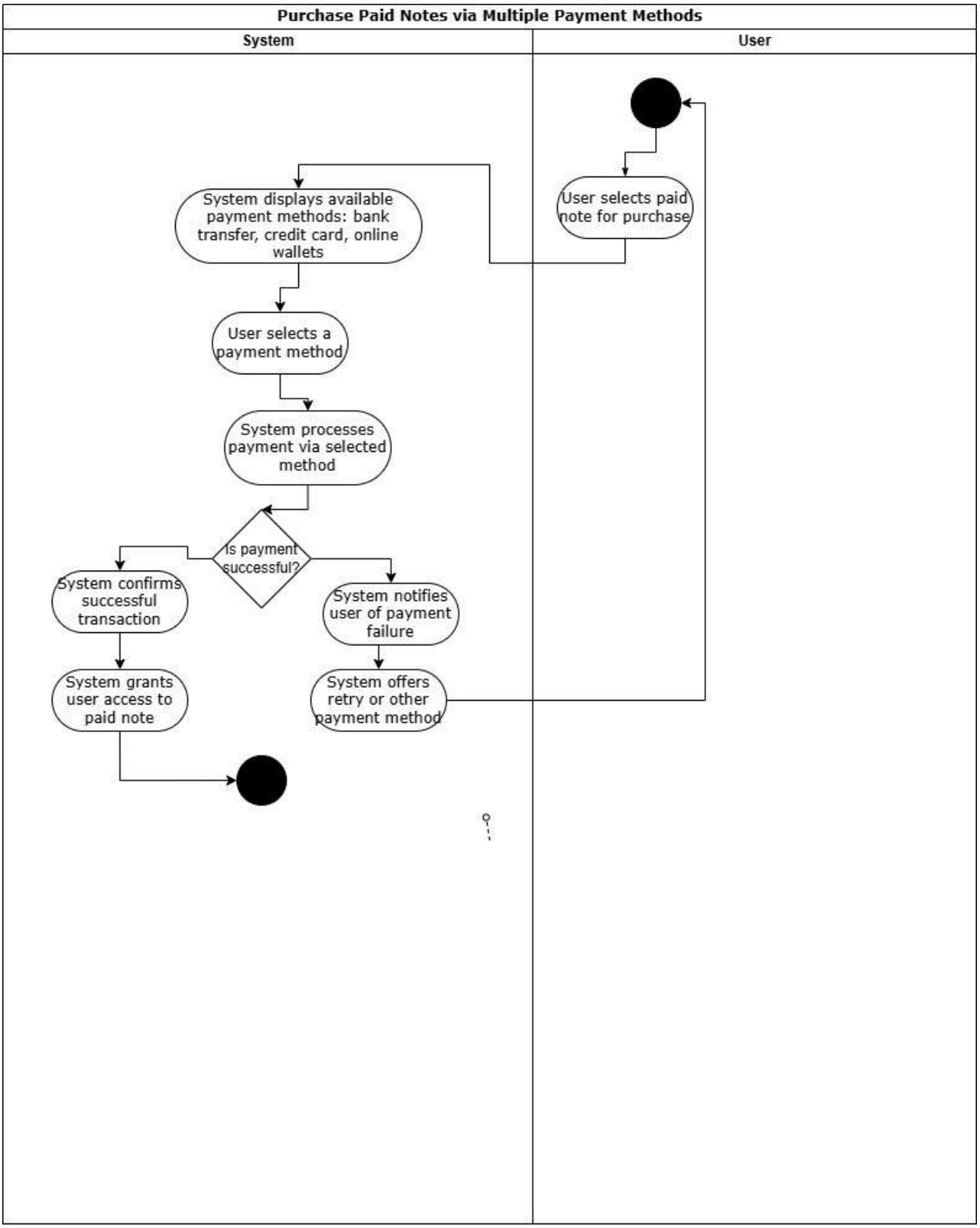

System
User
User navigates to Courses section
User selects a subject
System displays available courses & syllabi
Is syllabus available?
User clicks the syllabus link
System displays "Syllabus not available"
System opens & displays syllabus


System
User
System verifies credentials and connectivity
Is connectivity and credentials verified?
System starts the live class
Students join live class
Real-time interaction (chat/audio/video)
Record session enabled?
System records session
System displays "Connection issue. Retry or switch to backup connection."
Mentor reattempts connection
Mentor logs into the platform
Mentor navigates to live class section

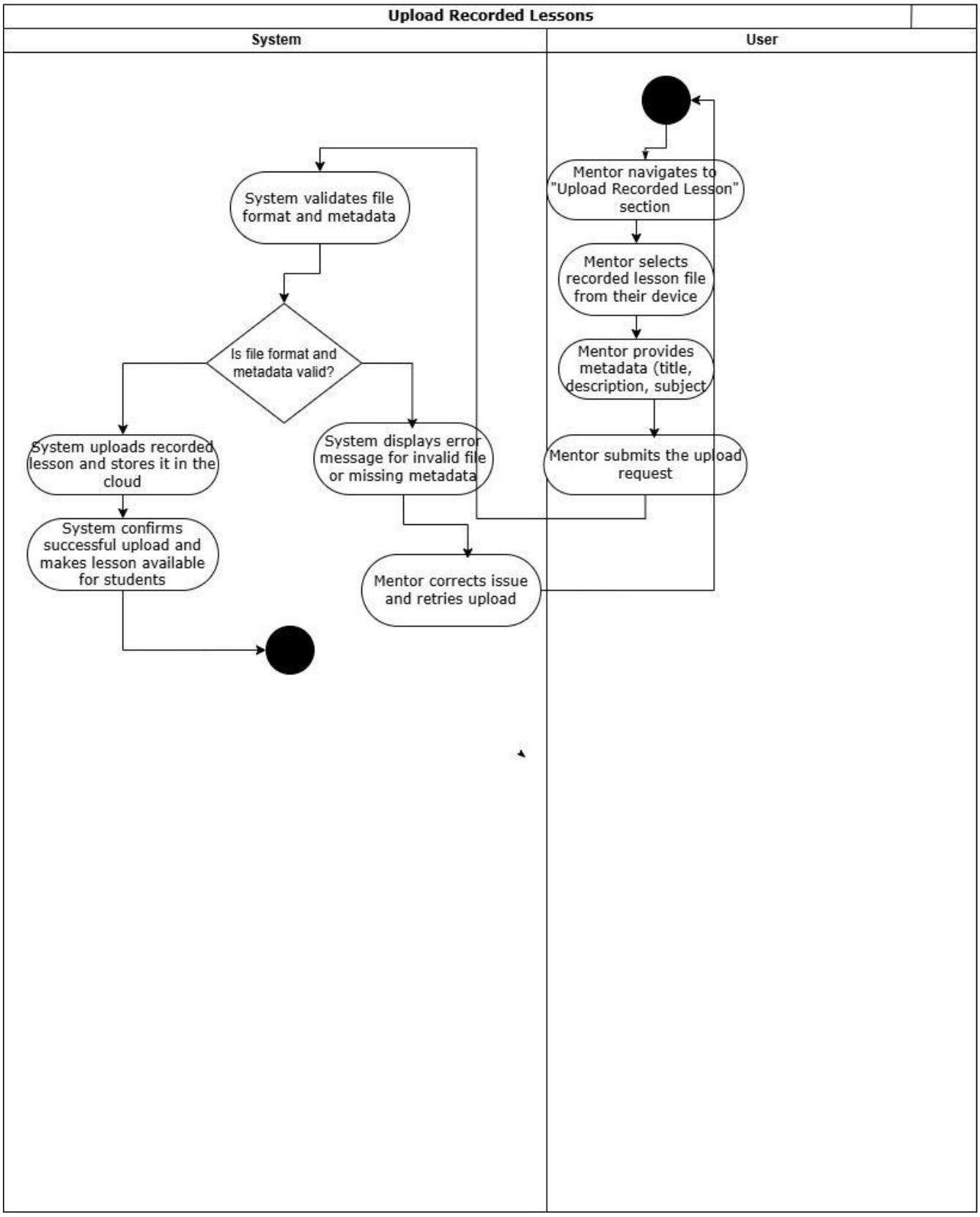
UC_15 - Halil


Generate Personalized Recommendations
System
User
System runs the recommendation algorithm
System collects data on user preferences and activity
System generates personalized recommendations
Is there sufficient user data?
System displays default trending resources
System displays recommendations on user dashboard
System displays default resources on dashboard
User reviews and selects a recommendation

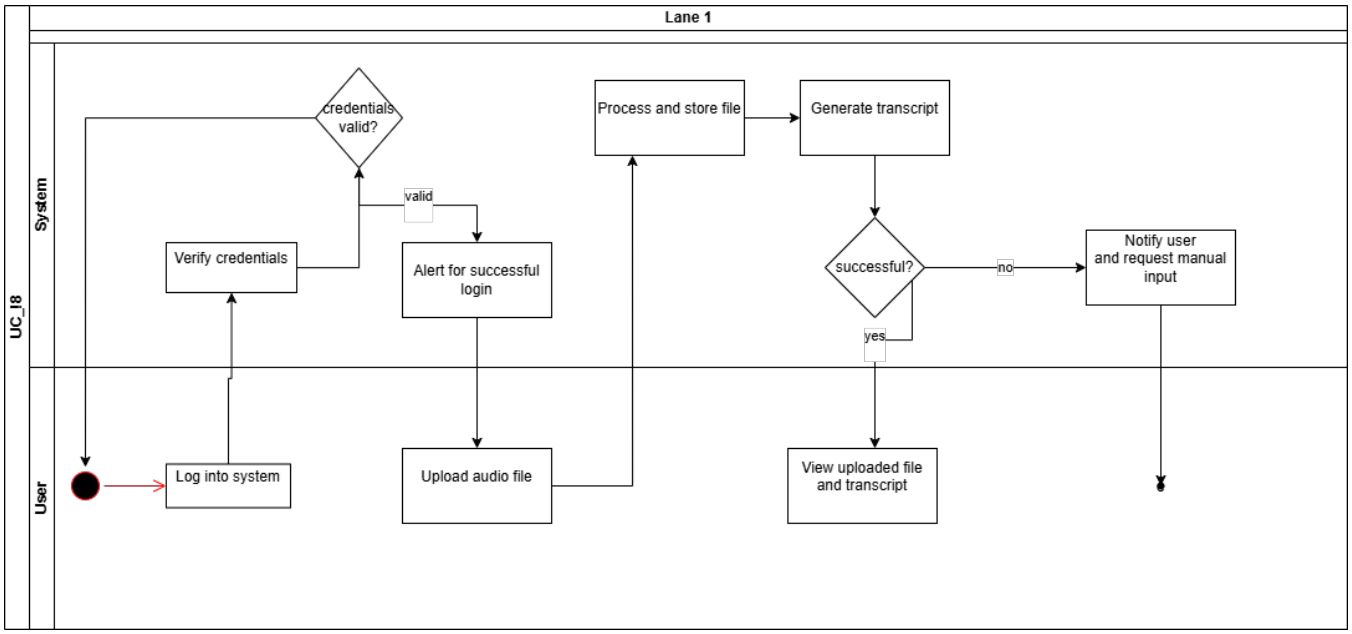
UC_16 - Marvi


Purchase Paid Notes via Multiple Payment Methods
System
User
User selects paid note for purchase
System displays available payment methods: bank transfer, credit card, online wallets
User selects a payment method
System processes payment via selected method
Is payment successful?
System confirms successful transaction
System notifies user of payment failure
System grants user access to paid note
System offers retry or other payment method

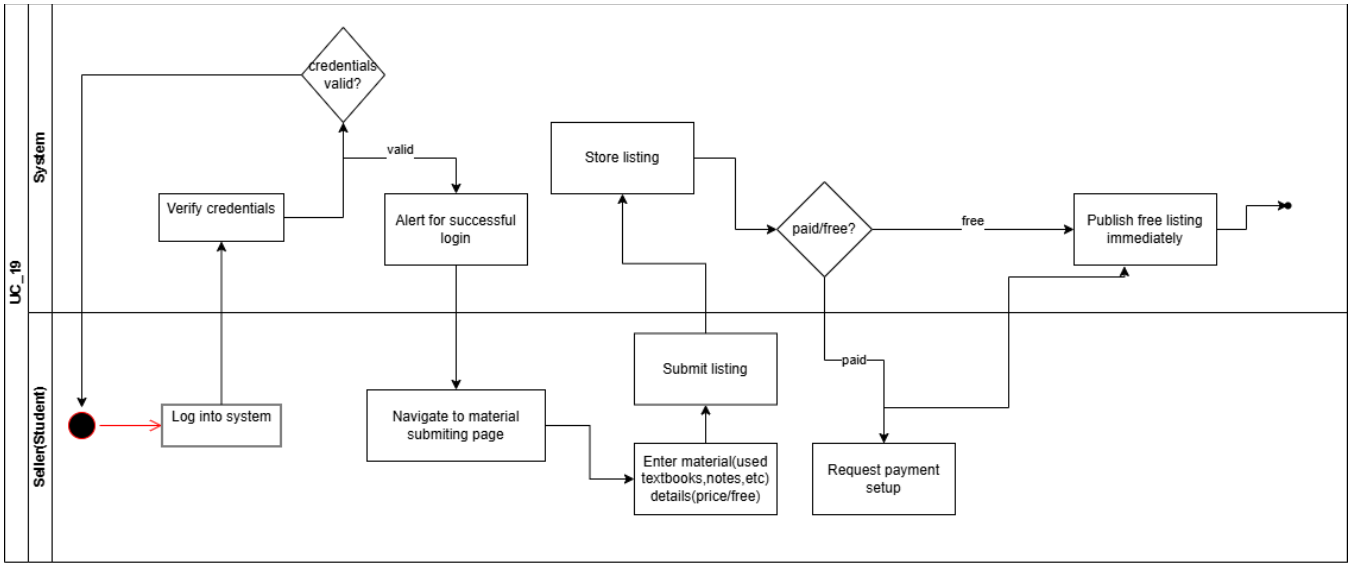
UC_17 - Marvi


Upload Recorded Lessons
System
User
Mentor navigates to "Upload Recorded Lesson" section
Mentor selects recorded lesson file from their device
Mentor provides metadata (title, description, subject
Mentor submits the upload request
System validates file format and metadata
Is file format and metadata valid?
System uploads recorded lesson and stores it in the cloud
System displays error message for invalid file or missing metadata
System confirms successful upload and makes lesson available for students
Mentor corrects issue and retries upload

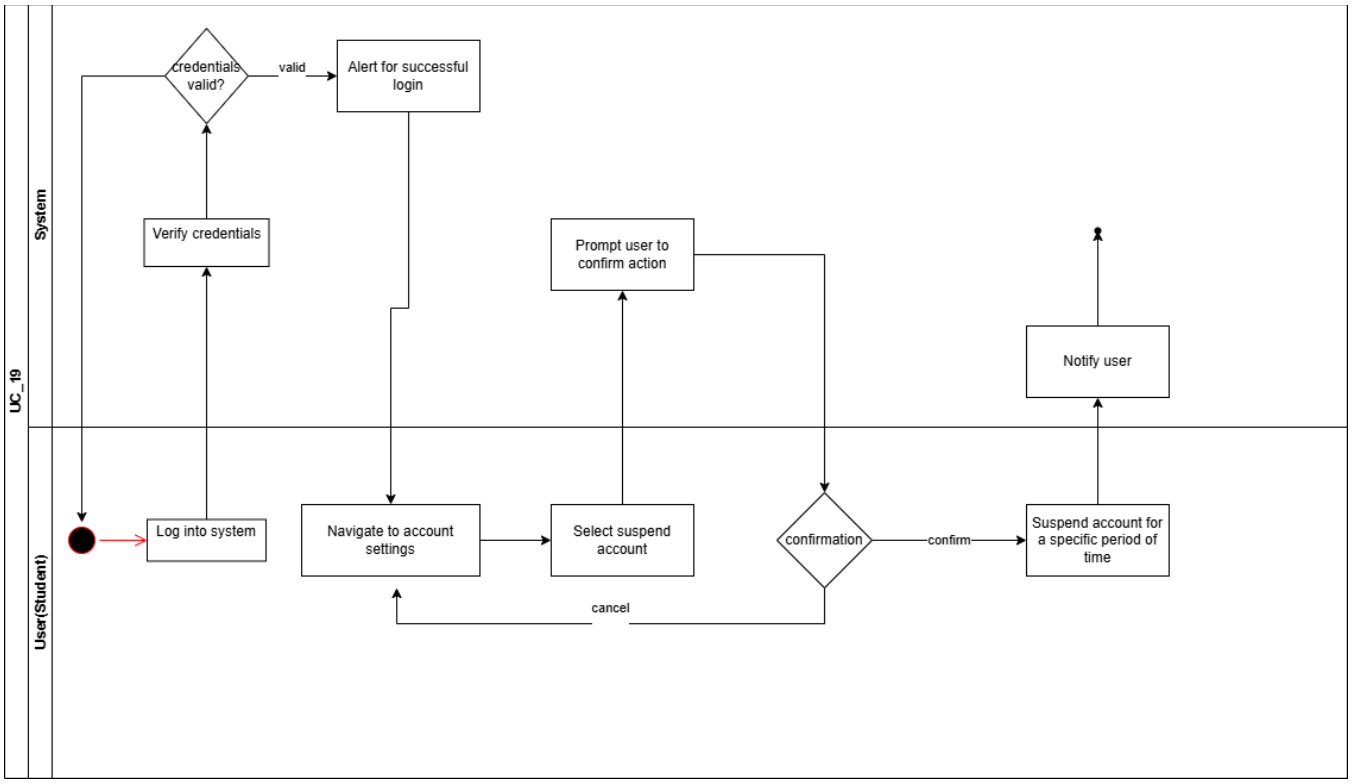
UC_18 - Dea



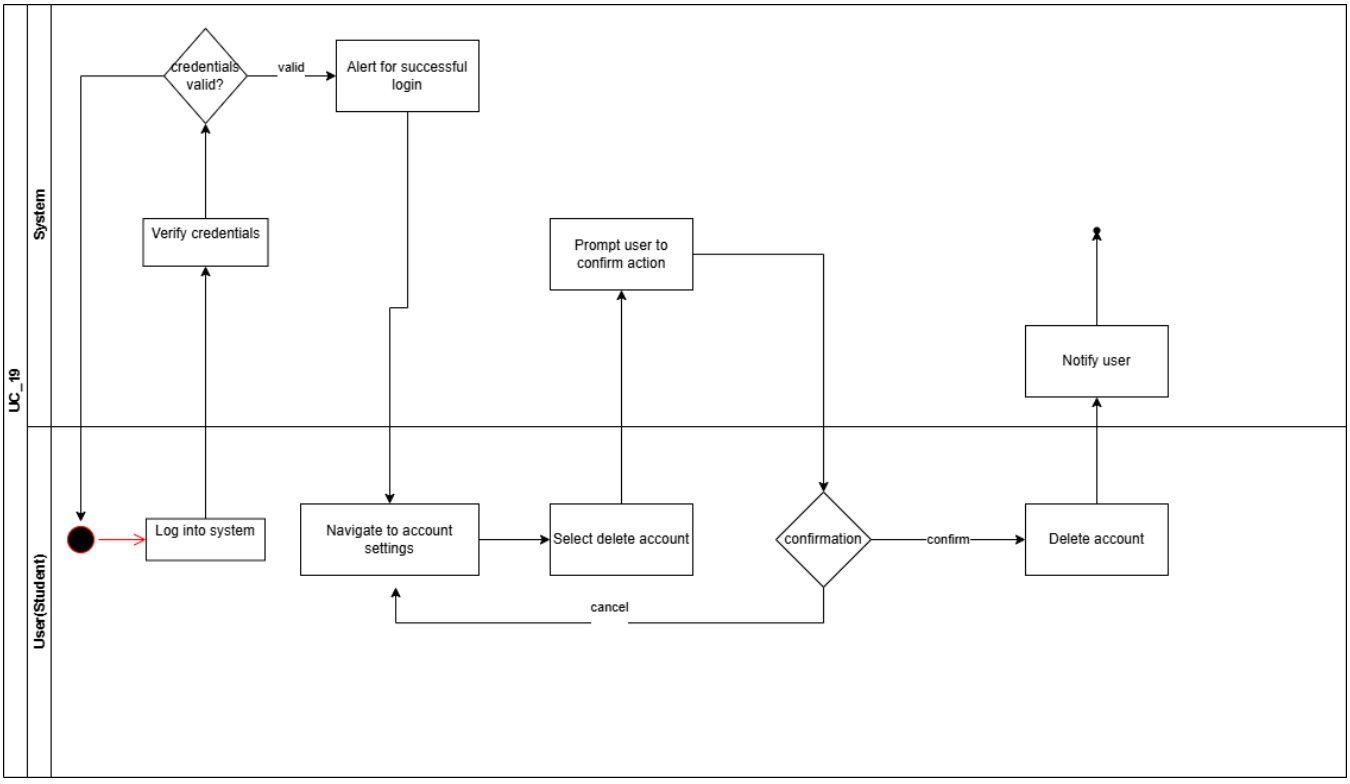
UC_19 - Dea



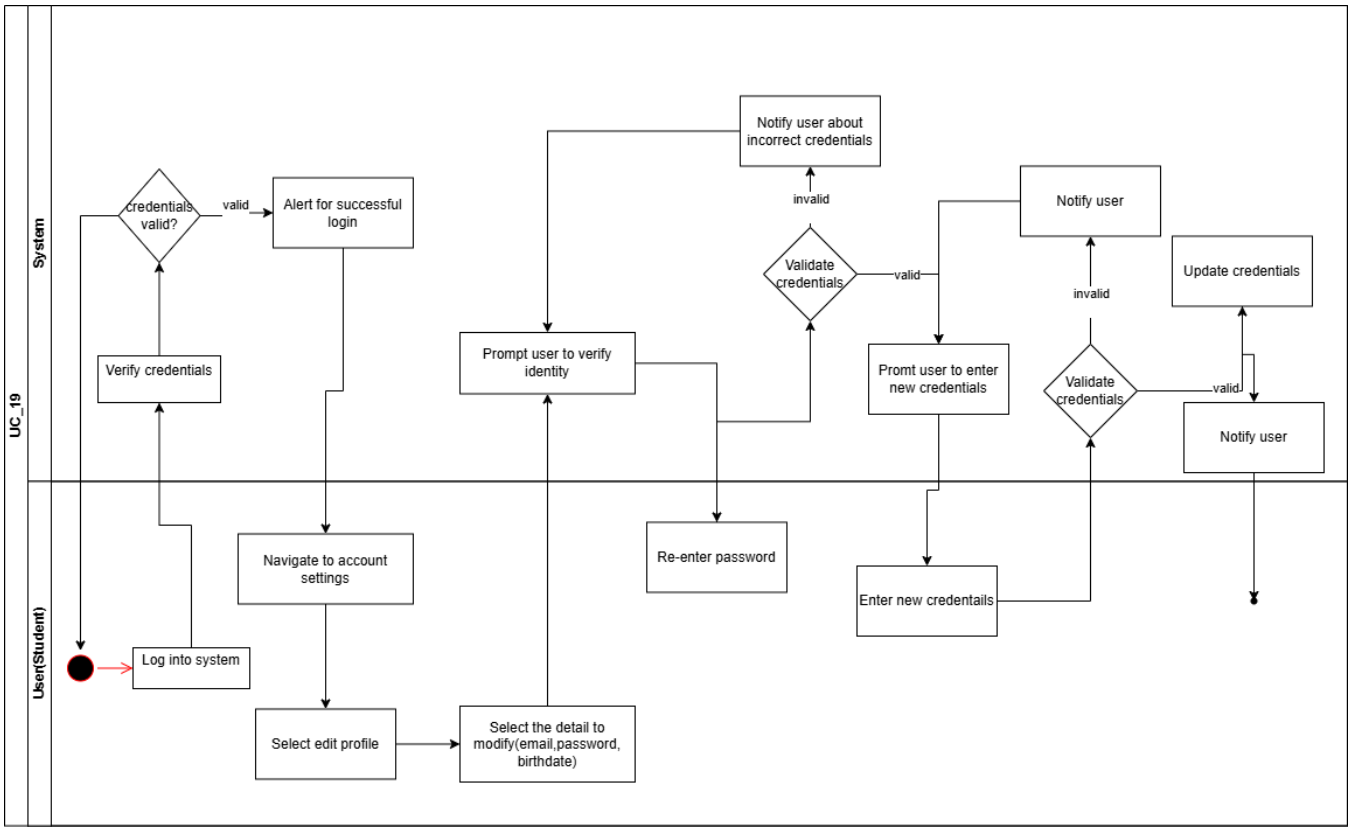
UC_20 - Dea



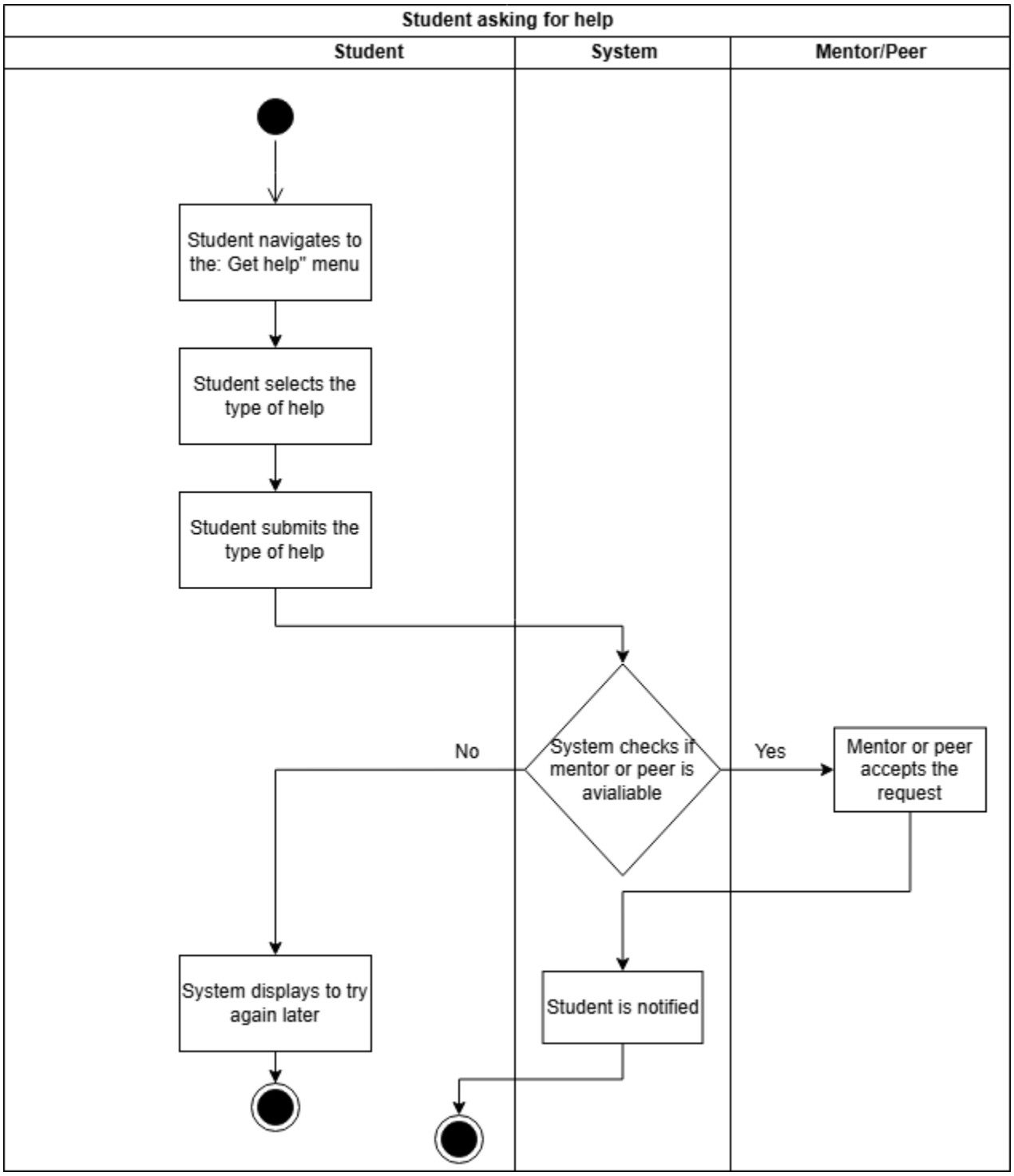
UC_21 - Dea



UC_22 - Dea

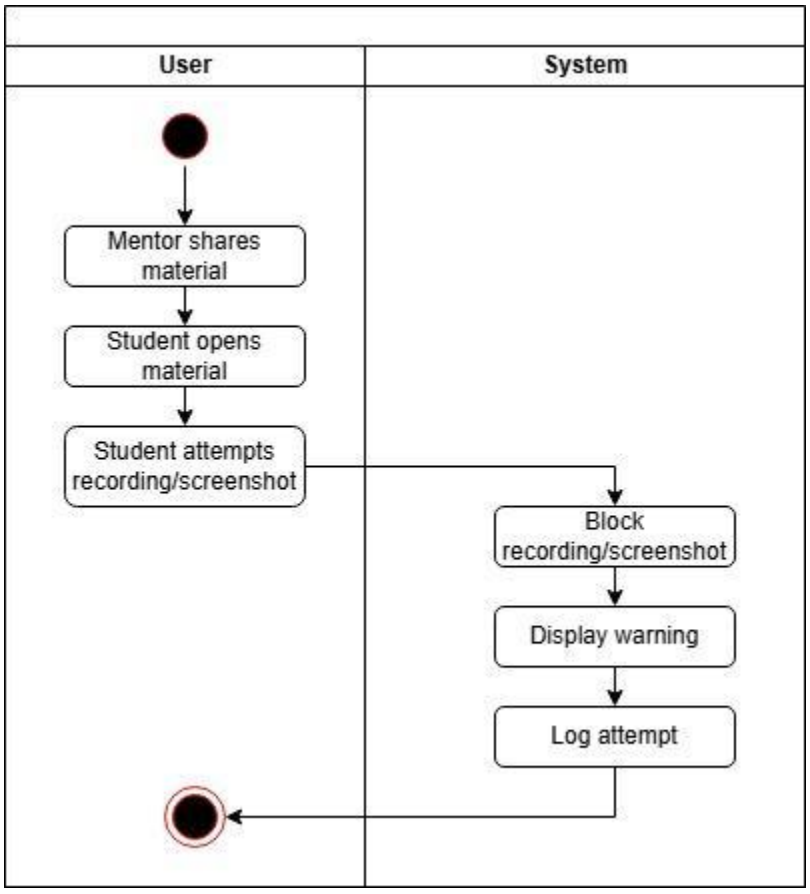

UC_19
System
User(Student)
credentials valid?
valid
Alert for successful login
Notify user about incorrect credentials
invalid
Validate credentials
valid
Notify user
Update credentials
invalid
Verify credentials
Prompt user to verify identity
Promt user to enter new credentials
Validate credentials
valid
Notify user
Navigate to account settings
Re-enter password
Enter new credentails
Log into system
Select edit profile
Select the detail to modify(email,password, birthdate)

UC_23 - Ameraldo



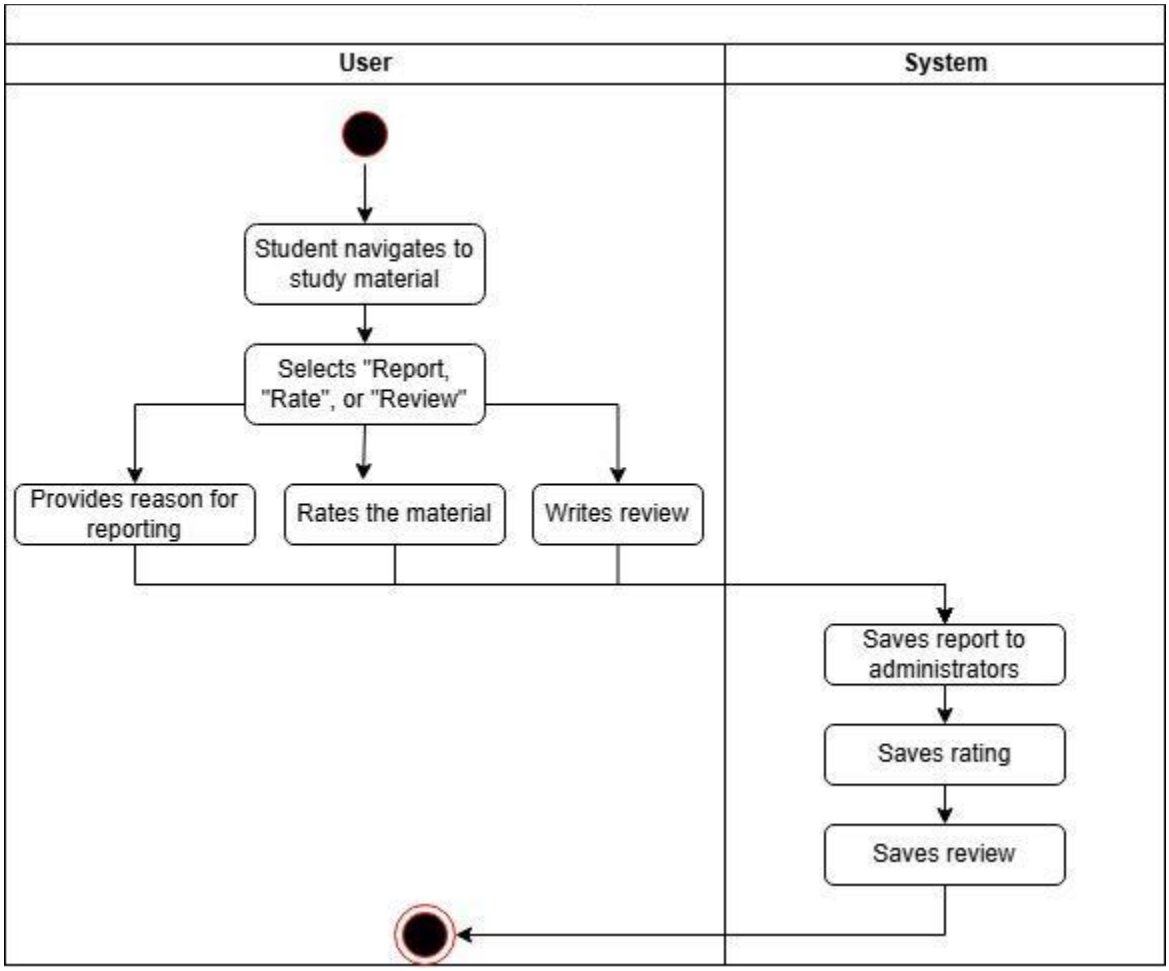
UC_24 - Ameraldo

Screen Recording/Screenshot prevention Requirement



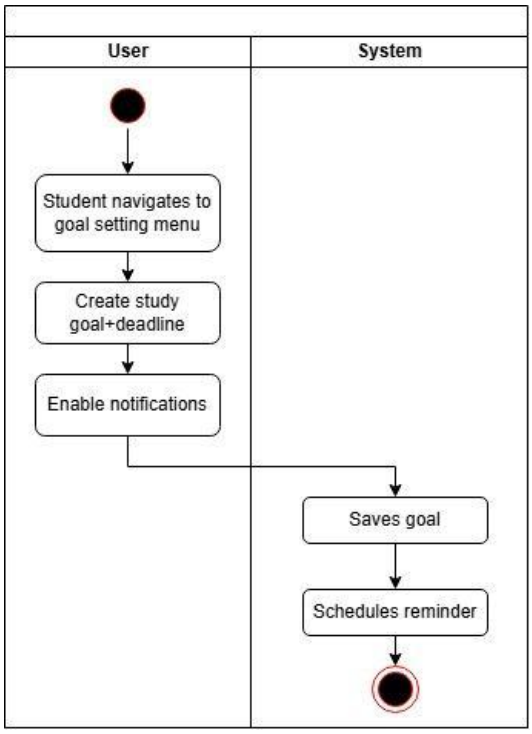
UC_25 - Ameraldo

Rate/Review/Report Requirement



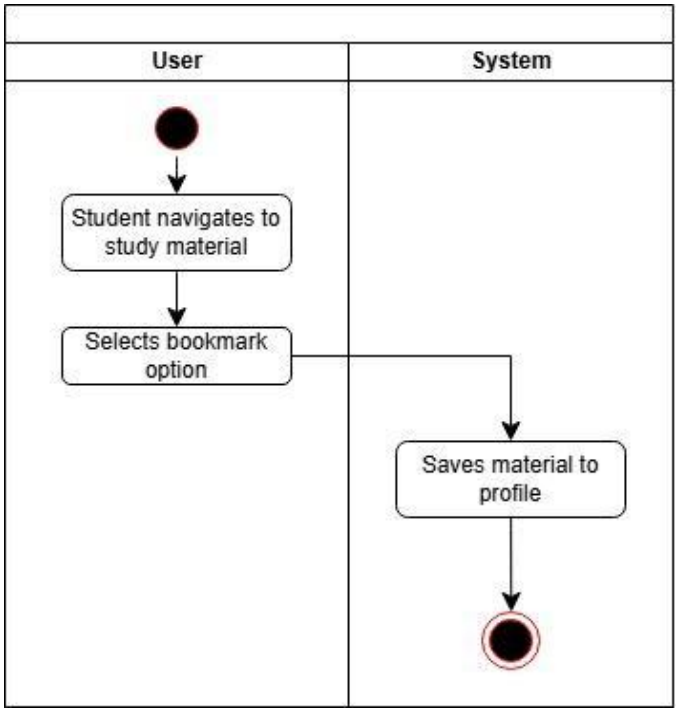
UC_26 - Ameraldo

Goal setting Requirement

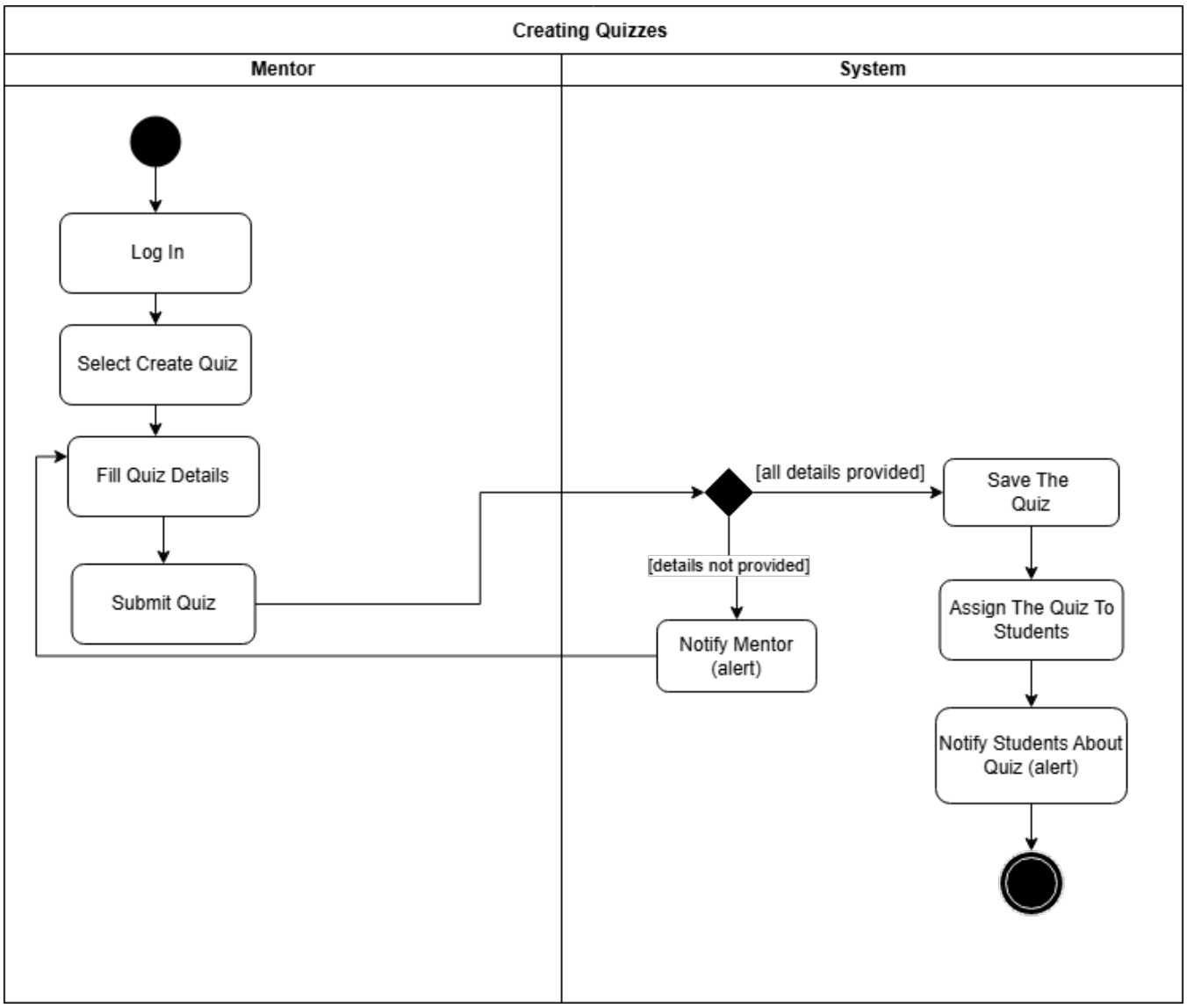


UC_27 - Ameraldo

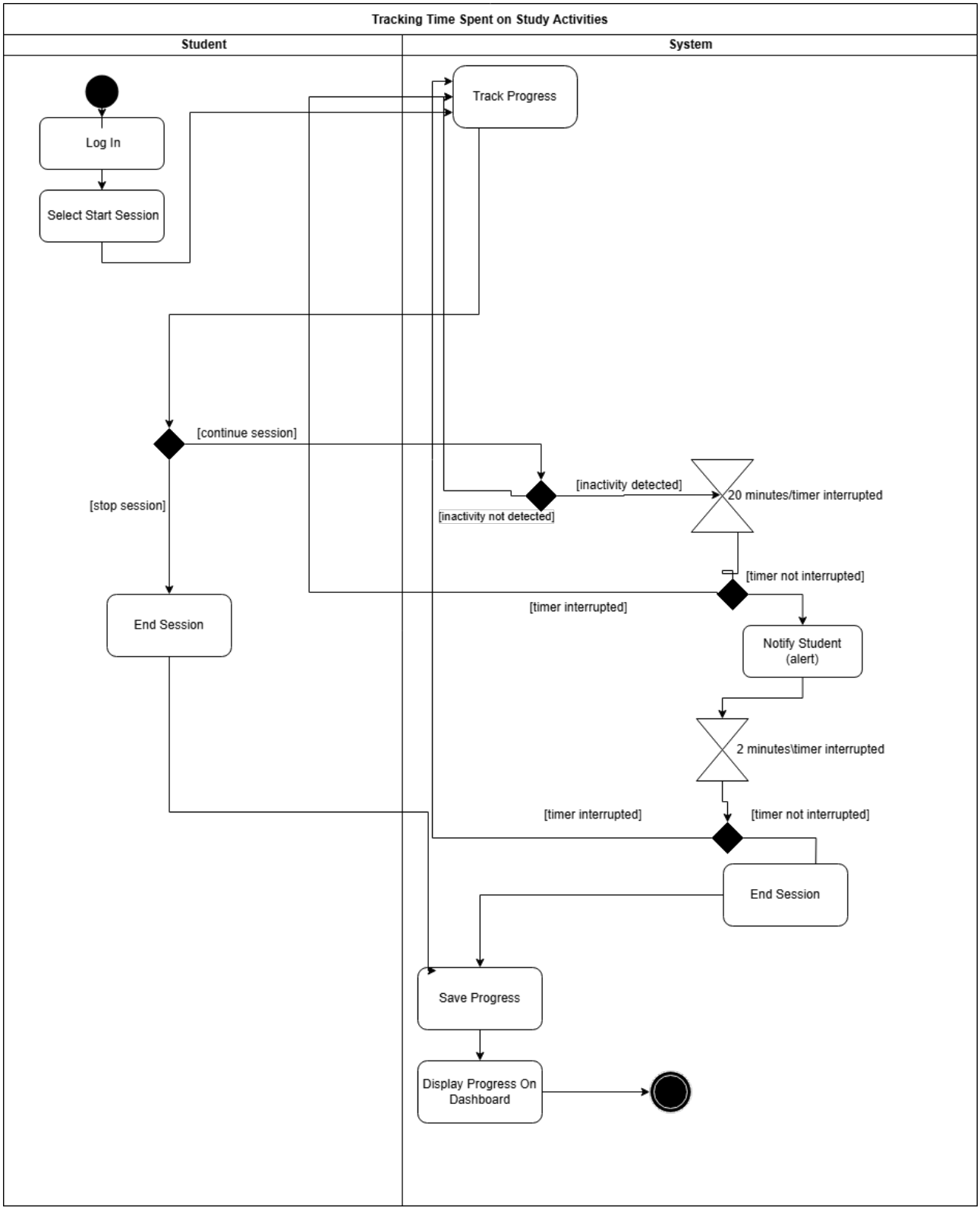
Student bookmarks material Requirement



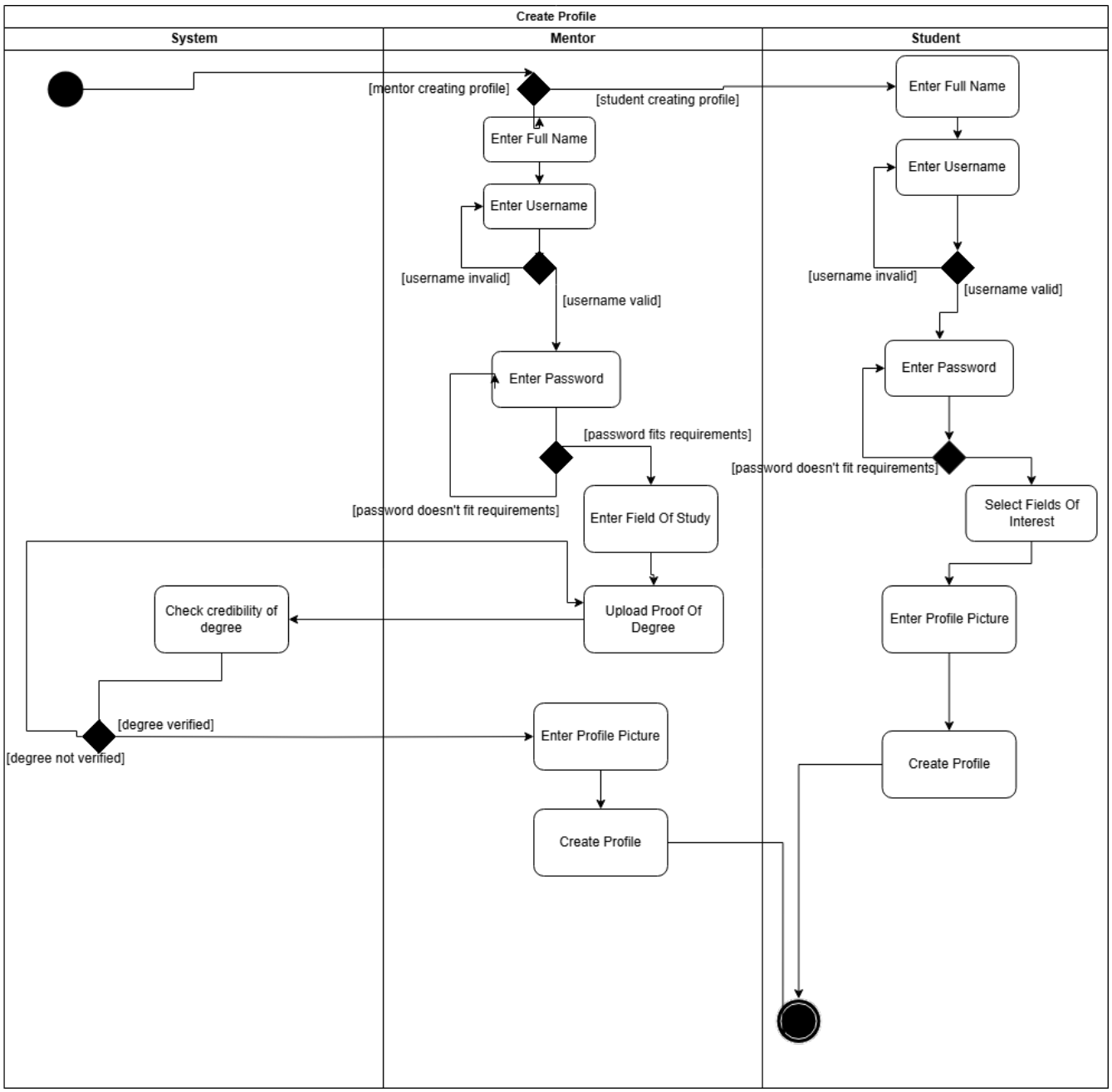
UC_28 - Daniela



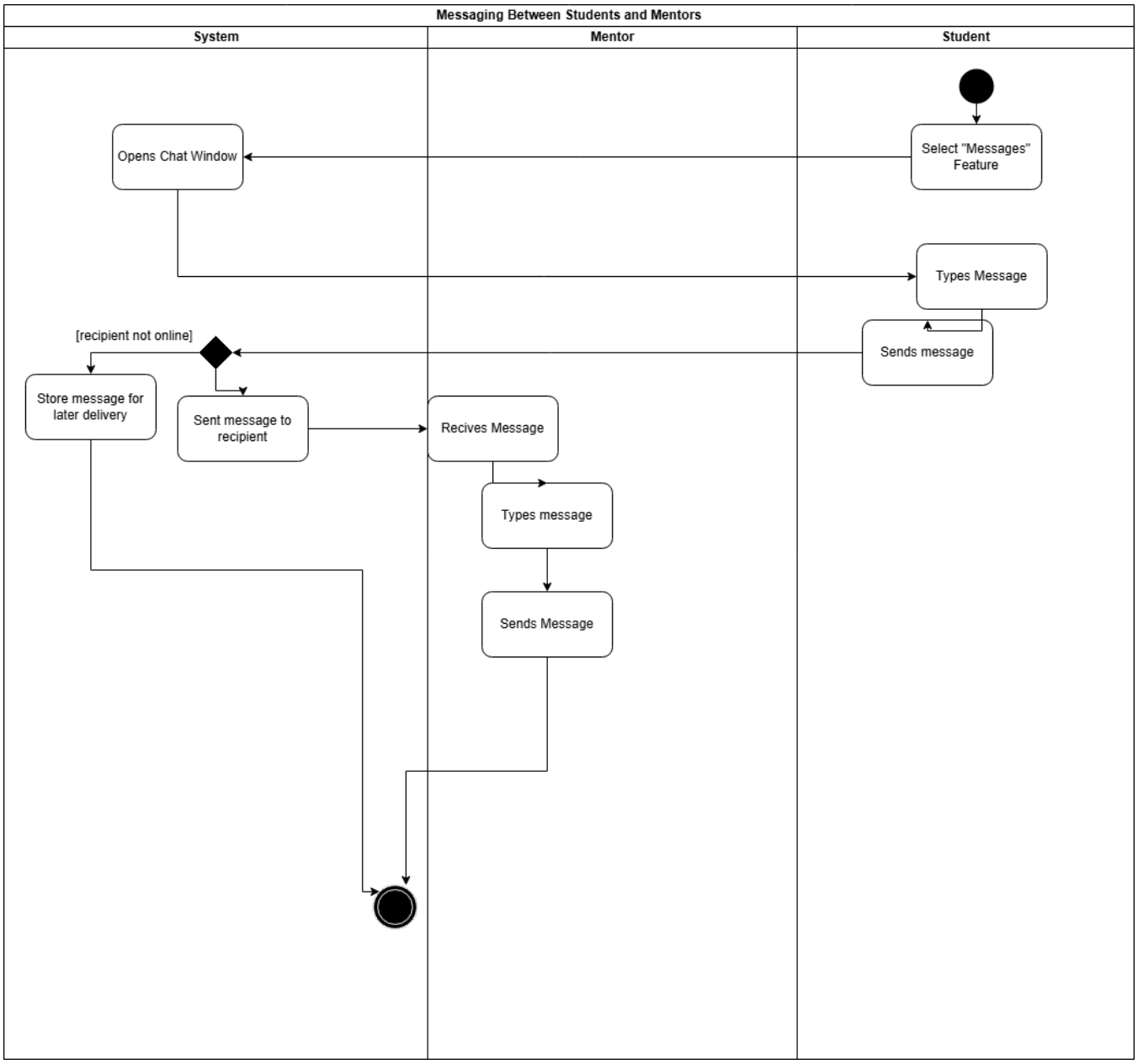
UC_29 - Daniela



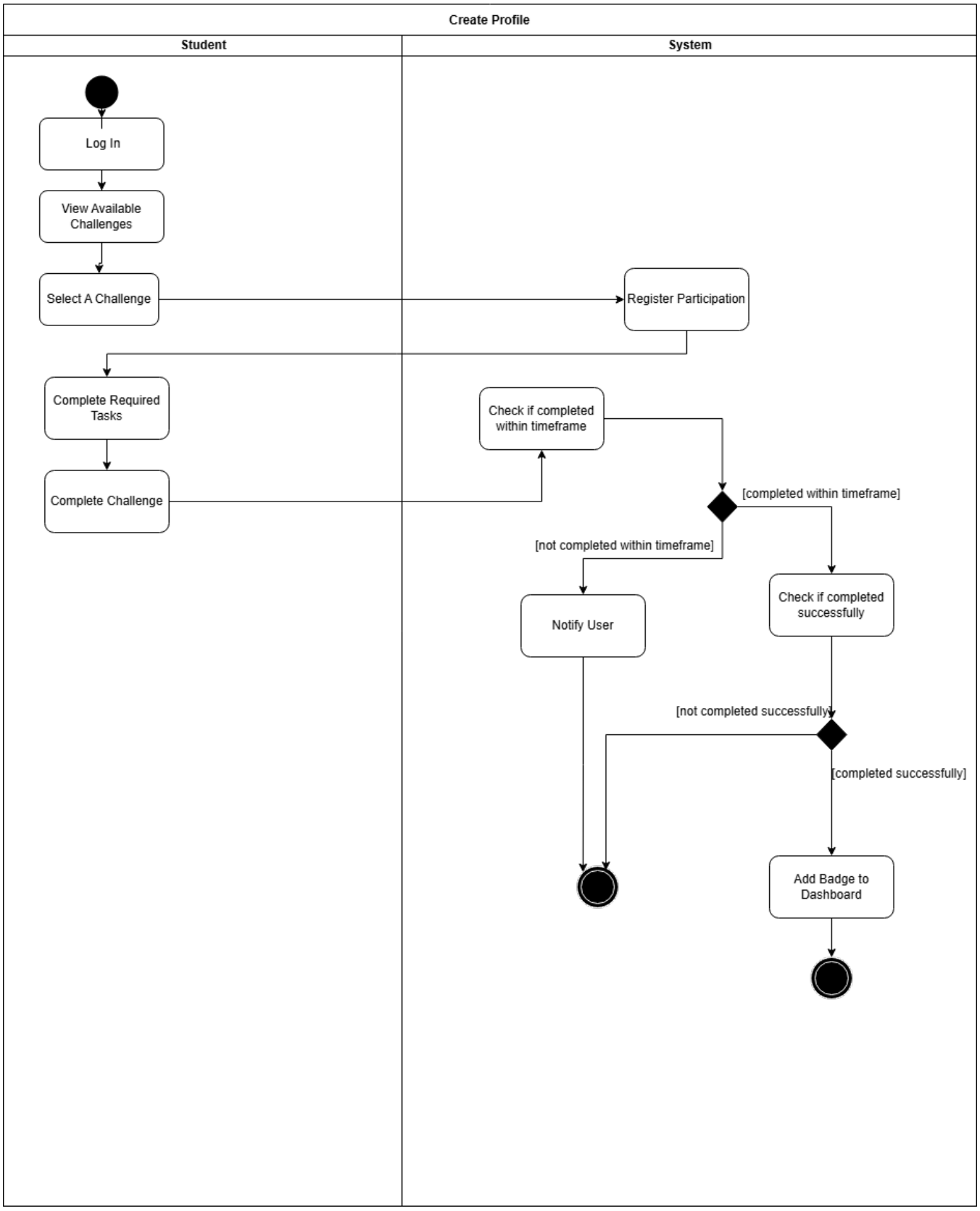
UC_30 - Daniela



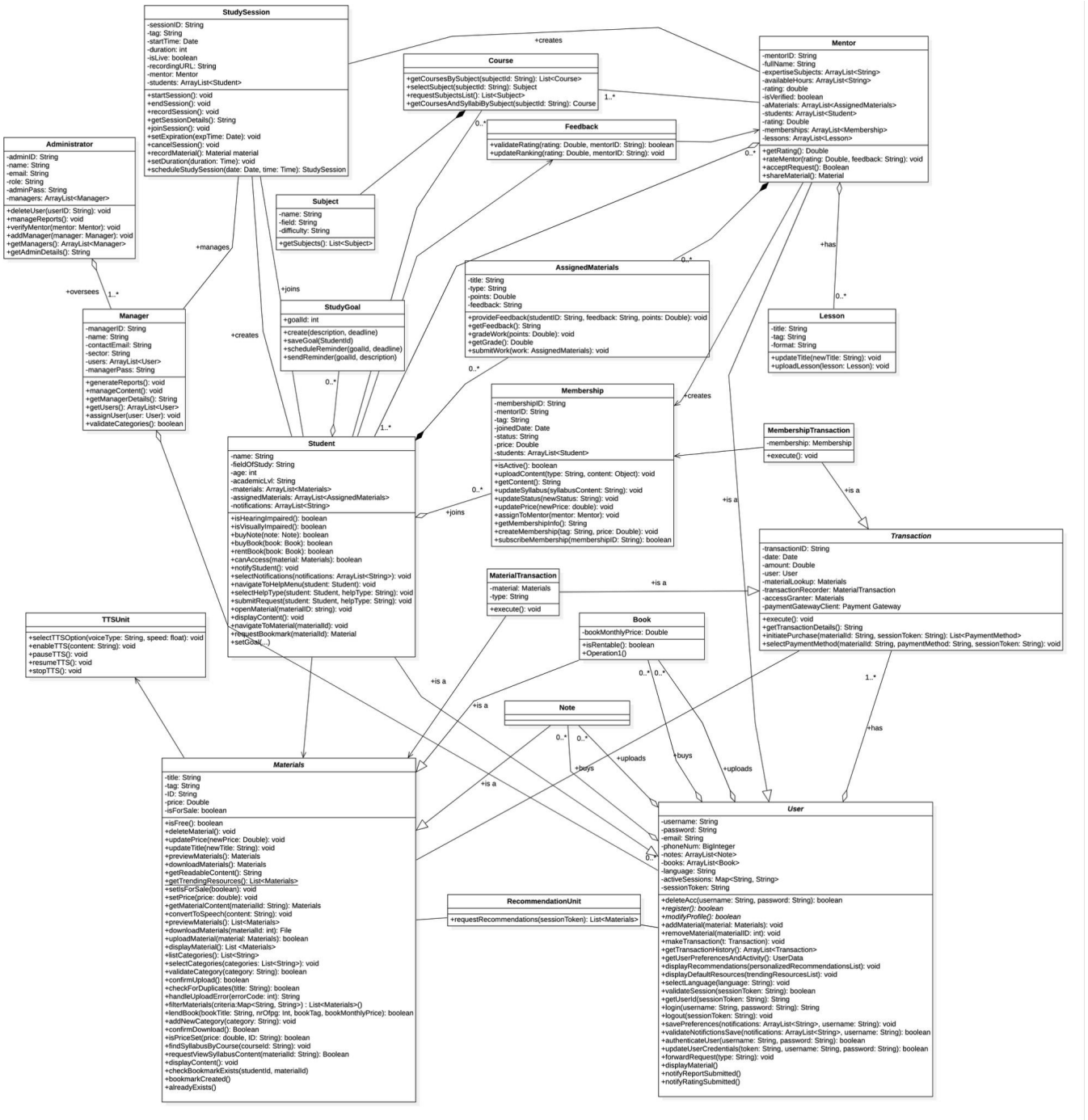
UC_31 - Daniela



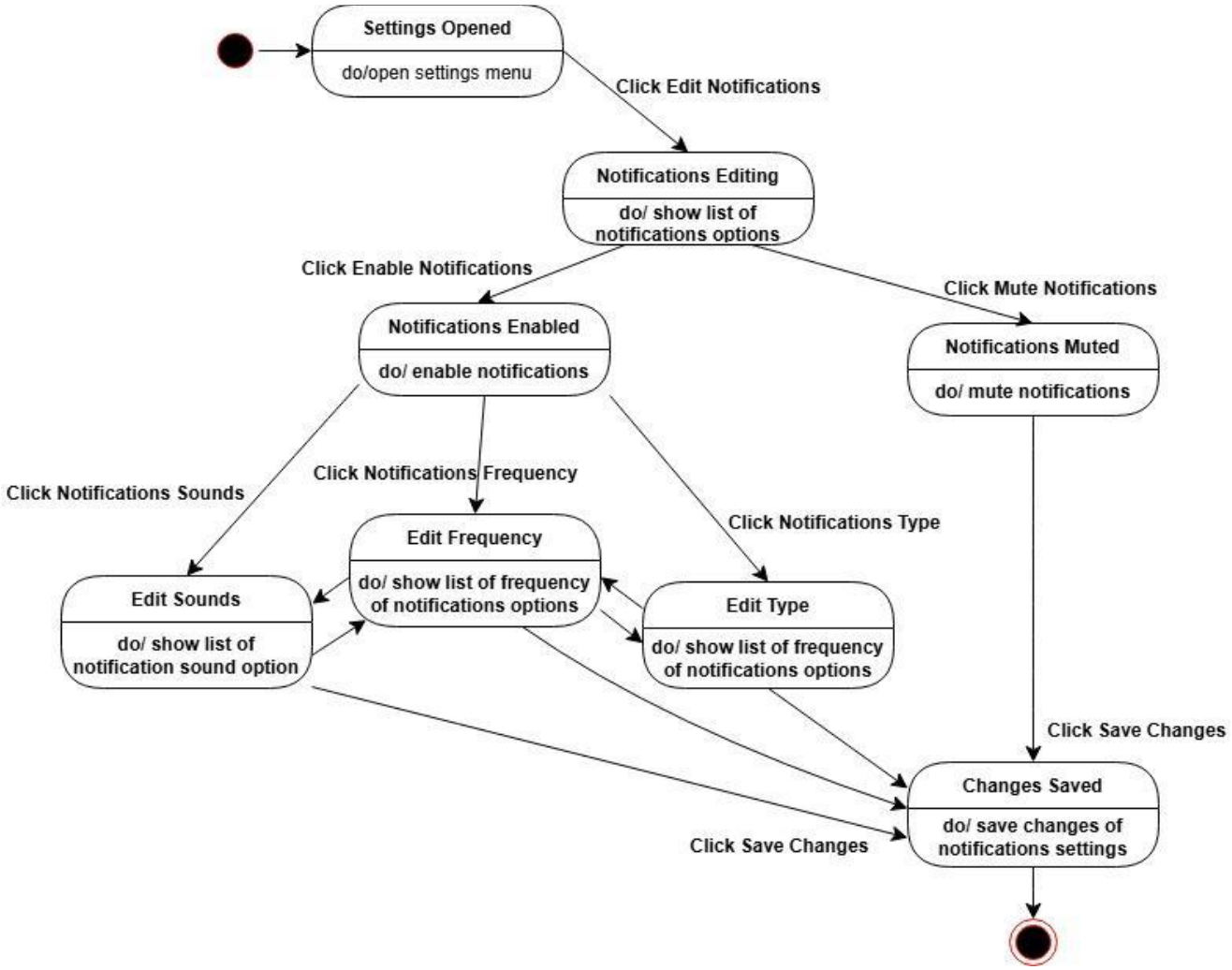
UC_32 - Daniela

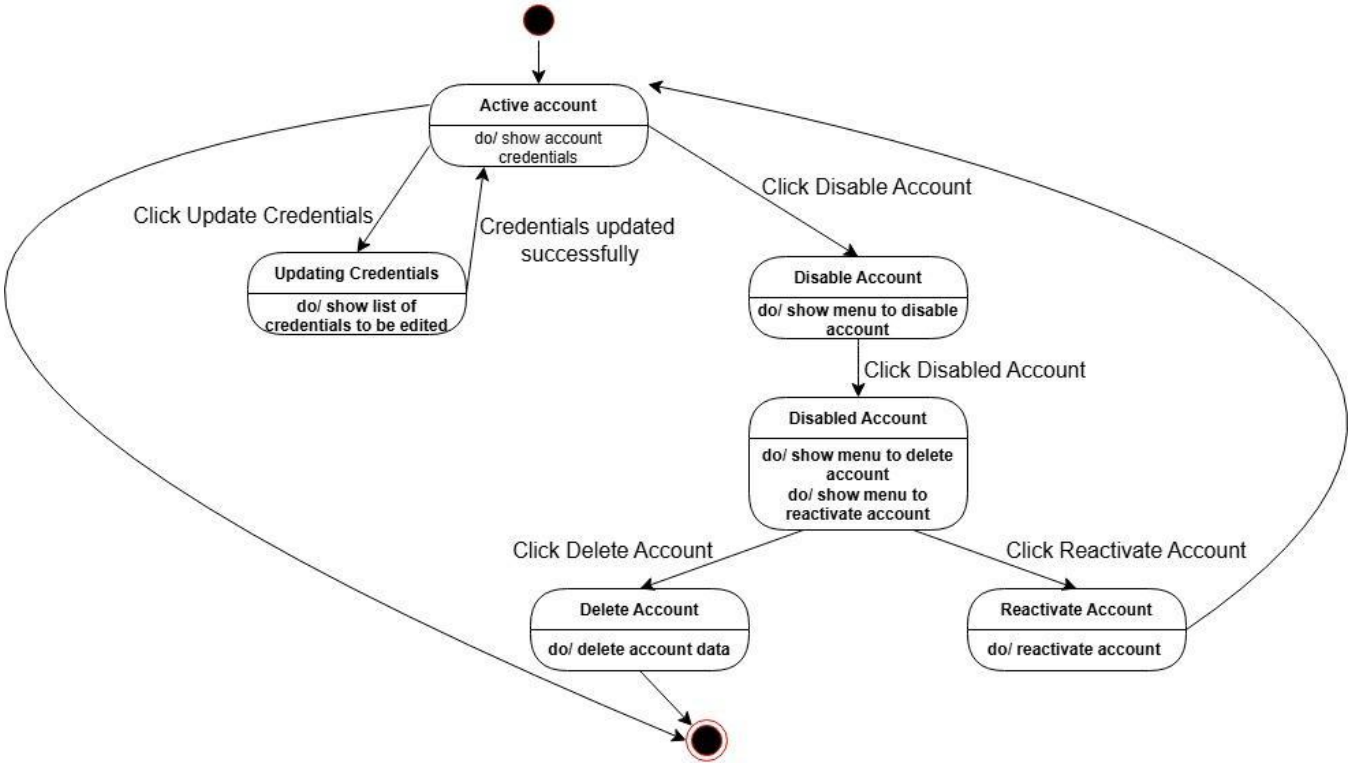


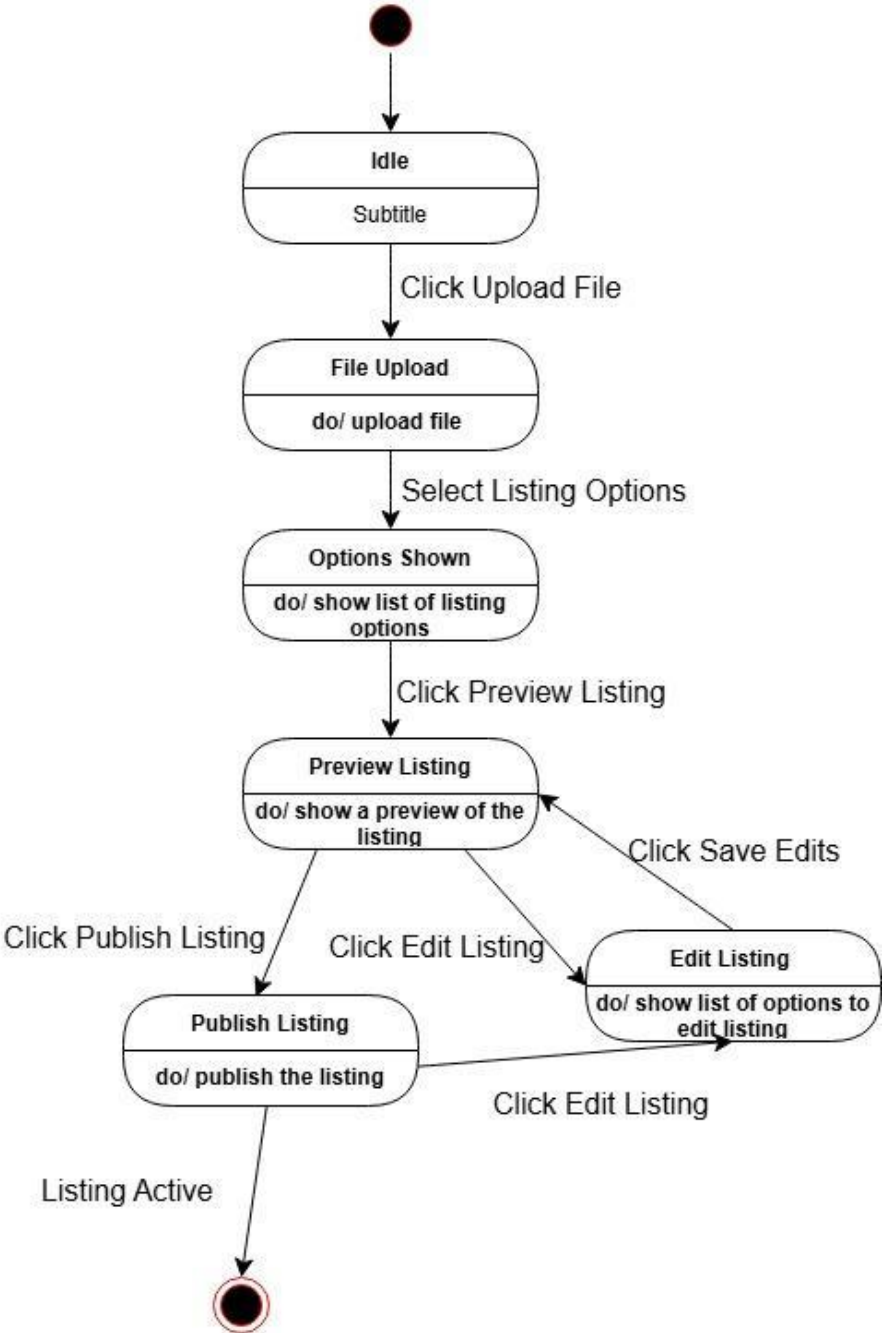
5.4 Class Diagram

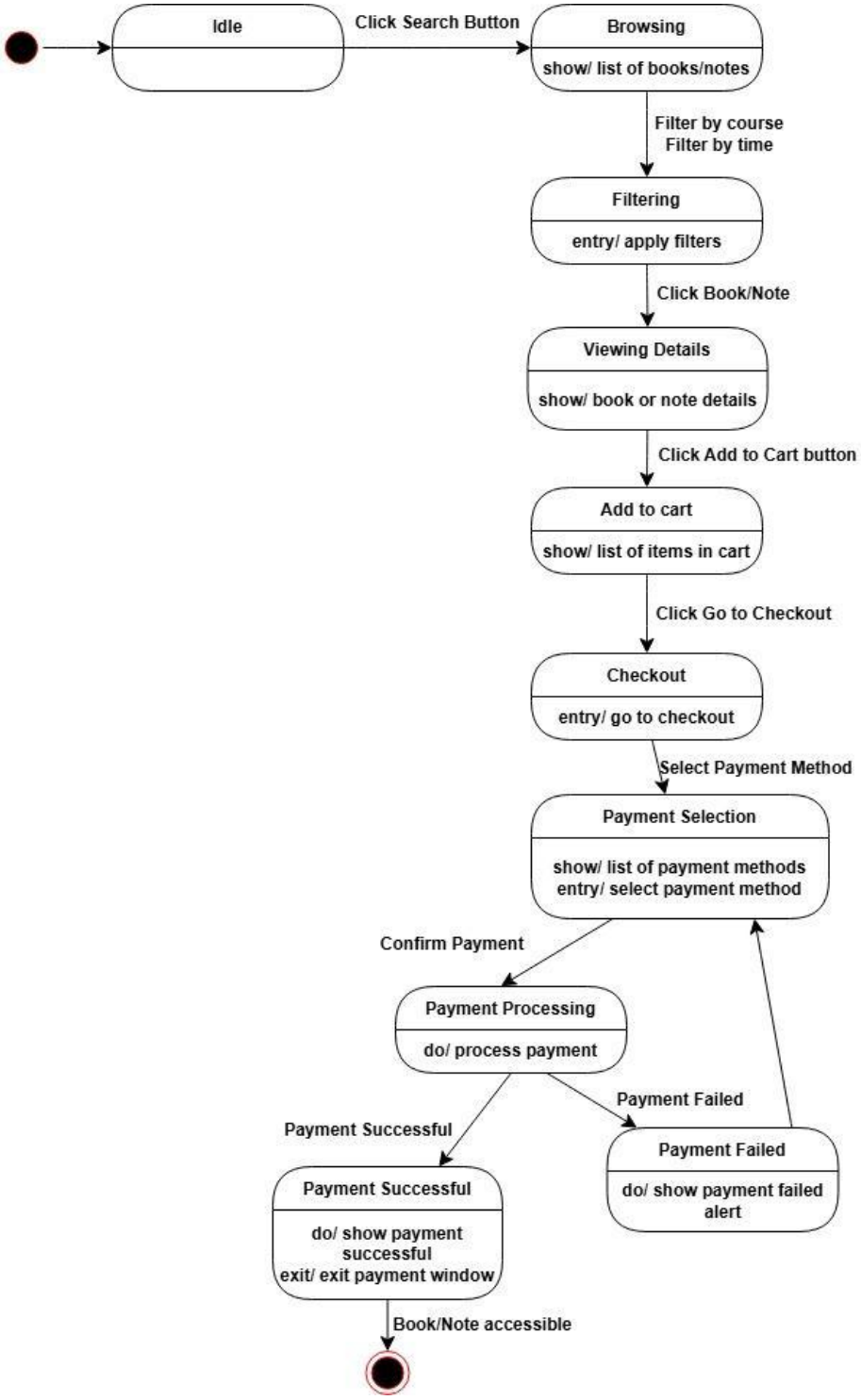


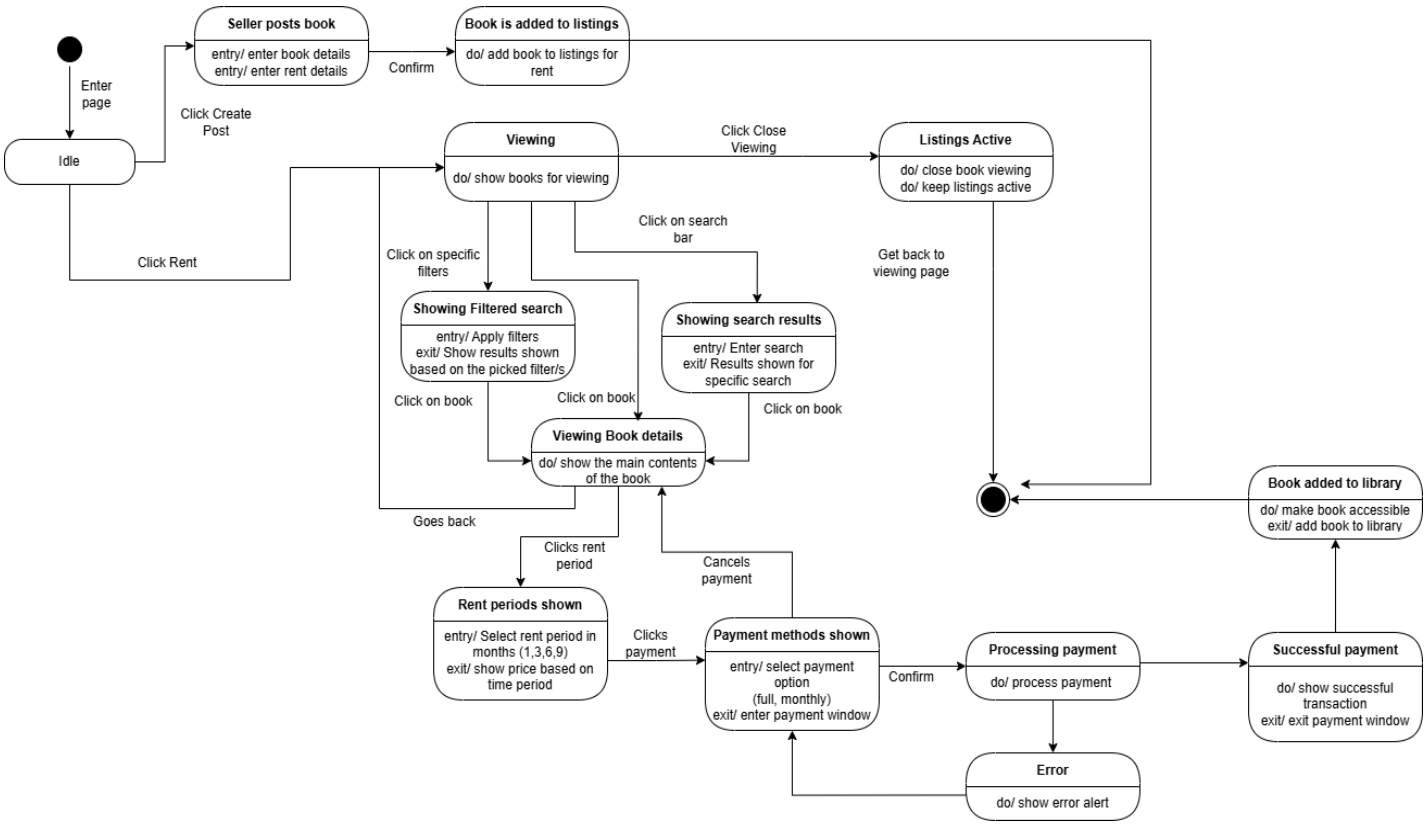
5.5 State Diagrams

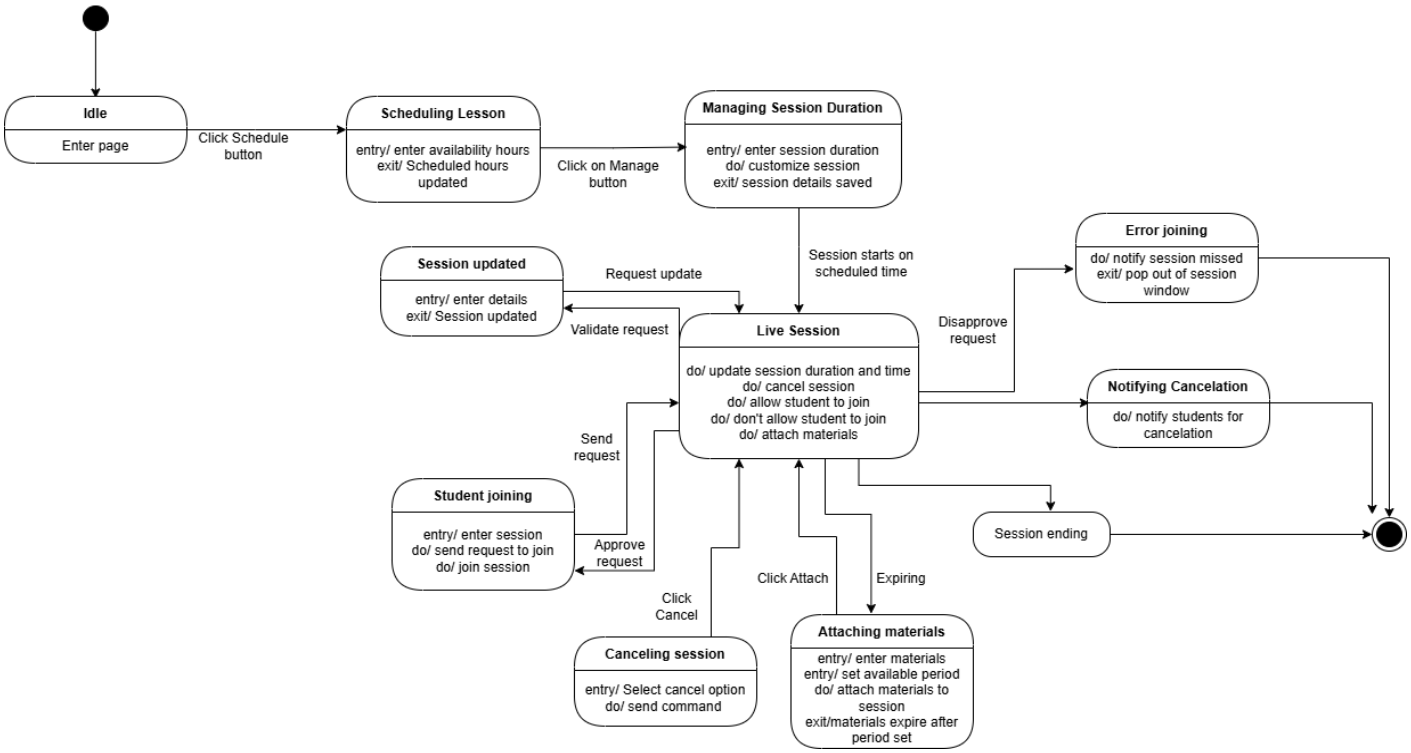



Active account
do/ show account credentials
Click Update Credentials
Credentials updated successfully
Updating Credentials
do/ show list of credentials to be edited
Click Disable Account
Disable Account
do/ show menu to disable account
Click Disabled Account
Disabled Account
do/ show menu to delete account
do/ show menu to reactivate account
Click Delete Account
Click Reactivate Account
Delete Account
do/ delete account data
Reactivate Account
do/ reactivate account

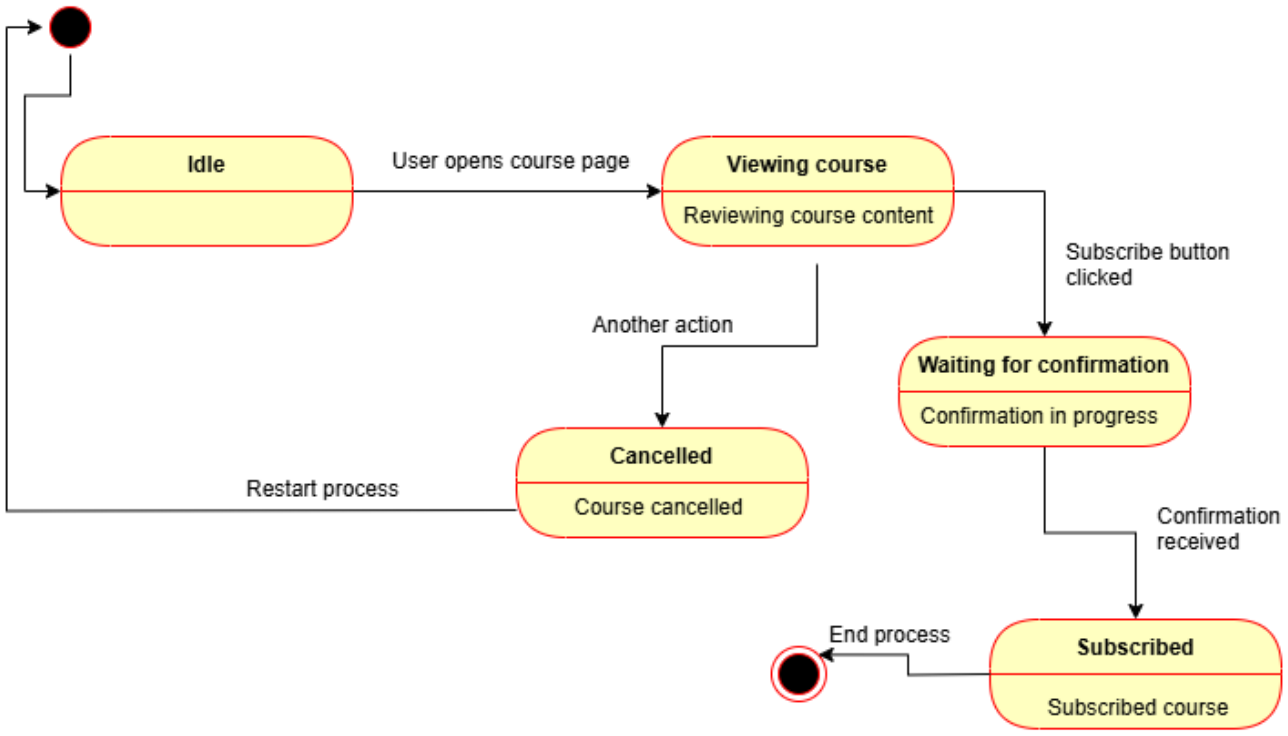

Idle
Subtitle
Click Upload File
File Upload
do/ upload file
Select Listing Options
Options Shown
do/ show list of listing options
Click Preview Listing
Preview Listing
do/ show a preview of the listing
Click Save Edits
Click Publish Listing
Click Edit Listing
Edit Listing
do/ show list of options to edit listing
Publish Listing
do/ publish the listing
Click Edit Listing
Listing Active


Idle
Click Search Button
Browsing
show/ list of books/notes
Filter by course
Filter by time
Filtering
entry/ apply filters
Click Book/Note
Viewing Details
show/ book or note details
Click Add to Cart button
Add to cart
show/ list of items in cart
Click Go to Checkout
Checkout
entry/ go to checkout
Select Payment Method
Payment Selection
show/ list of payment methods
entry/ select payment method
Confirm Payment
Payment Processing
do/ process payment
Payment Failed
Payment Successful
Payment Failed
do/ show payment failed alert
Payment Successful
do/ show payment successful
exit/ exit payment window
Book/Note accessible


Seller posts book
entry/ enter book details
entry/ enter rent details
Confirm
Book is added to listings
do/ add book to listings for rent
Enter page
Click Create Post
Idle
Viewing
do/ show books for viewing
Click Close Viewing
Listings Active
do/ close book viewing
do/ keep listings active
Click Rent
Click on specific filters
Click on search bar
Get back to viewing page
Showing Filtered search
entry/ Apply filters
exit/ Show results shown based on the picked filter/s
Showing search results
entry/ Enter search
exit/ Results shown for specific search
Click on book
Click on book
Click on book
Viewing Book details
do/ show the main contents of the book
Goes back
Book added to library
do/ make book accessible
exit/ add book to library
Clicks rent period
Cancels payment
Rent periods shown
entry/ Select rent period in months (1,3,6,9)
exit/ show price based on time period
Clicks payment
Payment methods shown
entry/ select payment option
(full, monthly)
exit/ enter payment window
Confirm
Processing payment
do/ process payment
Successful payment
do/ show successful transaction
exit/ exit payment window
Error
do/ show error alert

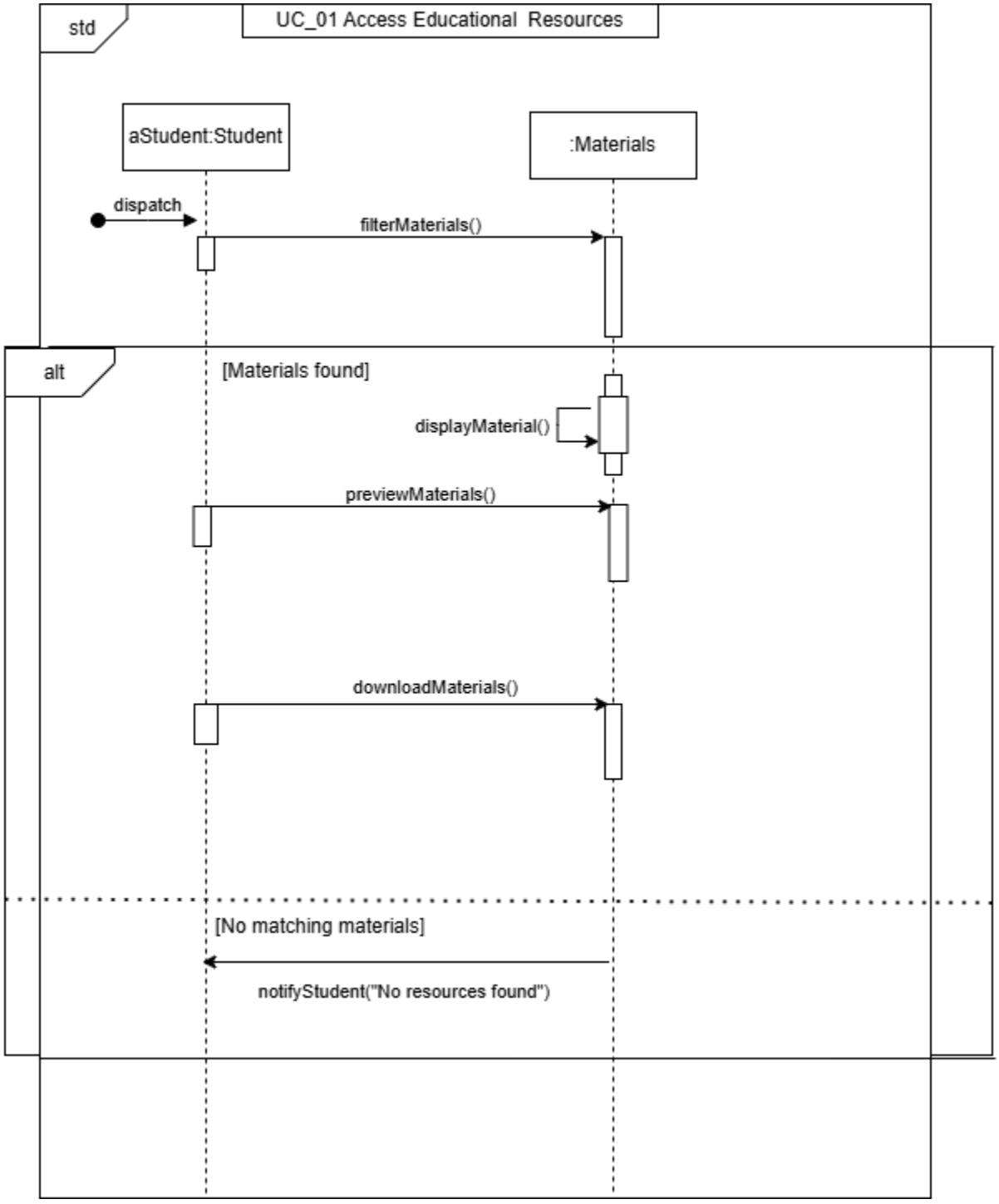

Idle
Enter page
Click Schedule button
Scheduling Lesson
entry/ enter availability hours
exit/ Scheduled hours updated
Click on Manage button
Managing Session Duration
entry/ enter session duration
do/ customize session
exit/ session details saved
Session starts on scheduled time
Error joining
do/ notify session missed
exit/ pop out of session window
Session updated
entry/ enter details
exit/ Session updated
Request update
Validate request
Live Session
do/ update session duration and time
do/ cancel session
do/ allow student to join
do/ don't allow student to join
do/ attach materials
Disapprove request
Notifying Cancelation
do/ notify students for cancelation
Send request
Student joining
entry/ enter session
do/ send request to join
do/ join session
Approve request
Session ending
Click Attach
Expiring
Click Cancel
Canceling session
entry/ Select cancel option
do/ send command
Attaching materials
entry/ enter materials
entry/ set available period
do/ attach materials to session
exit/materials expire after period set

Marvi

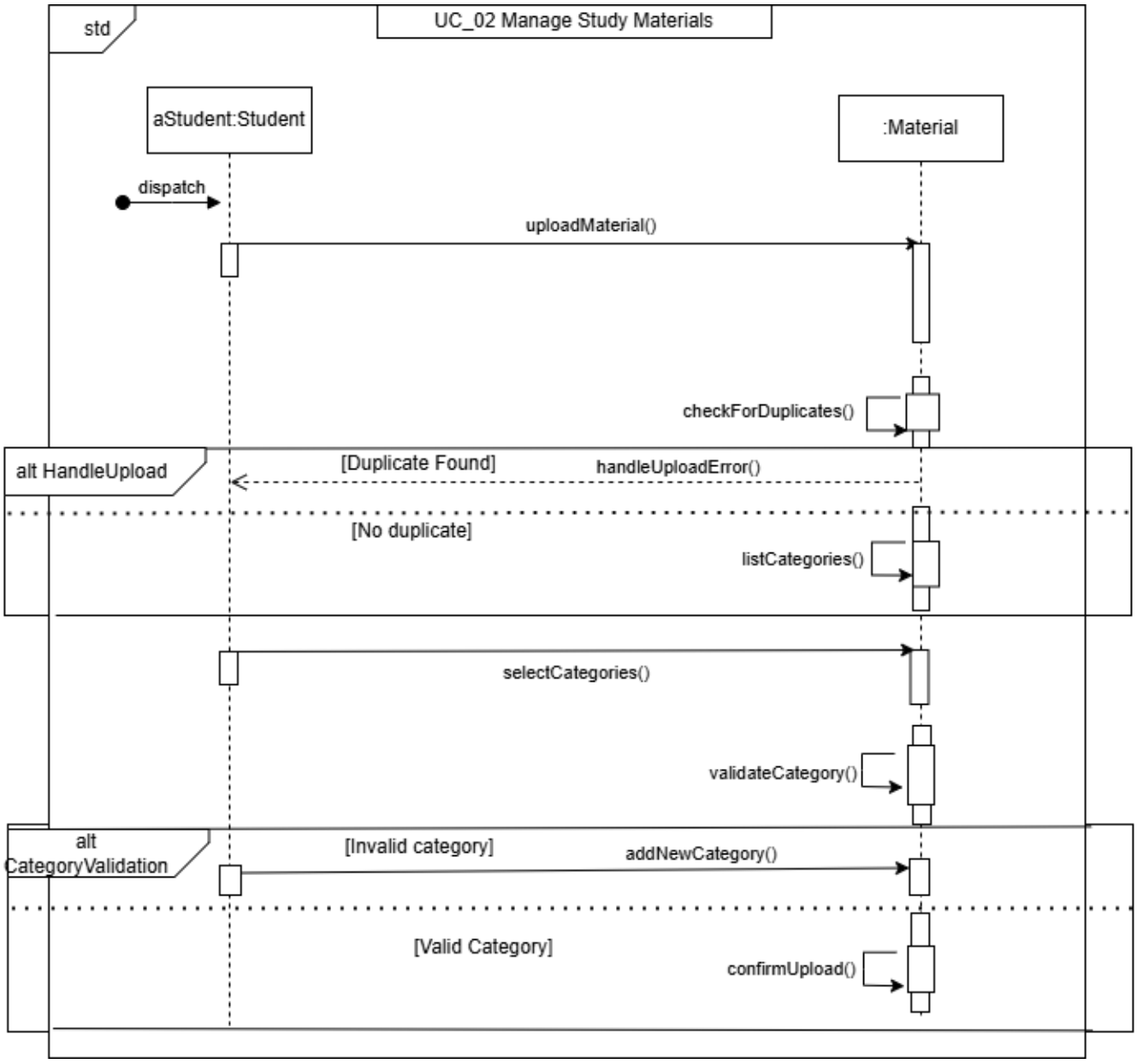

Idle
User opens course page
Viewing course
Reviewing course content
Subscribe button clicked
Another action
Waiting for confirmation
Confirmation in progress
Cancelled
Course cancelled
Restart process
Confirmation received
End process
Subscribed
Subscribed course

5.6 Sequence Diagrams

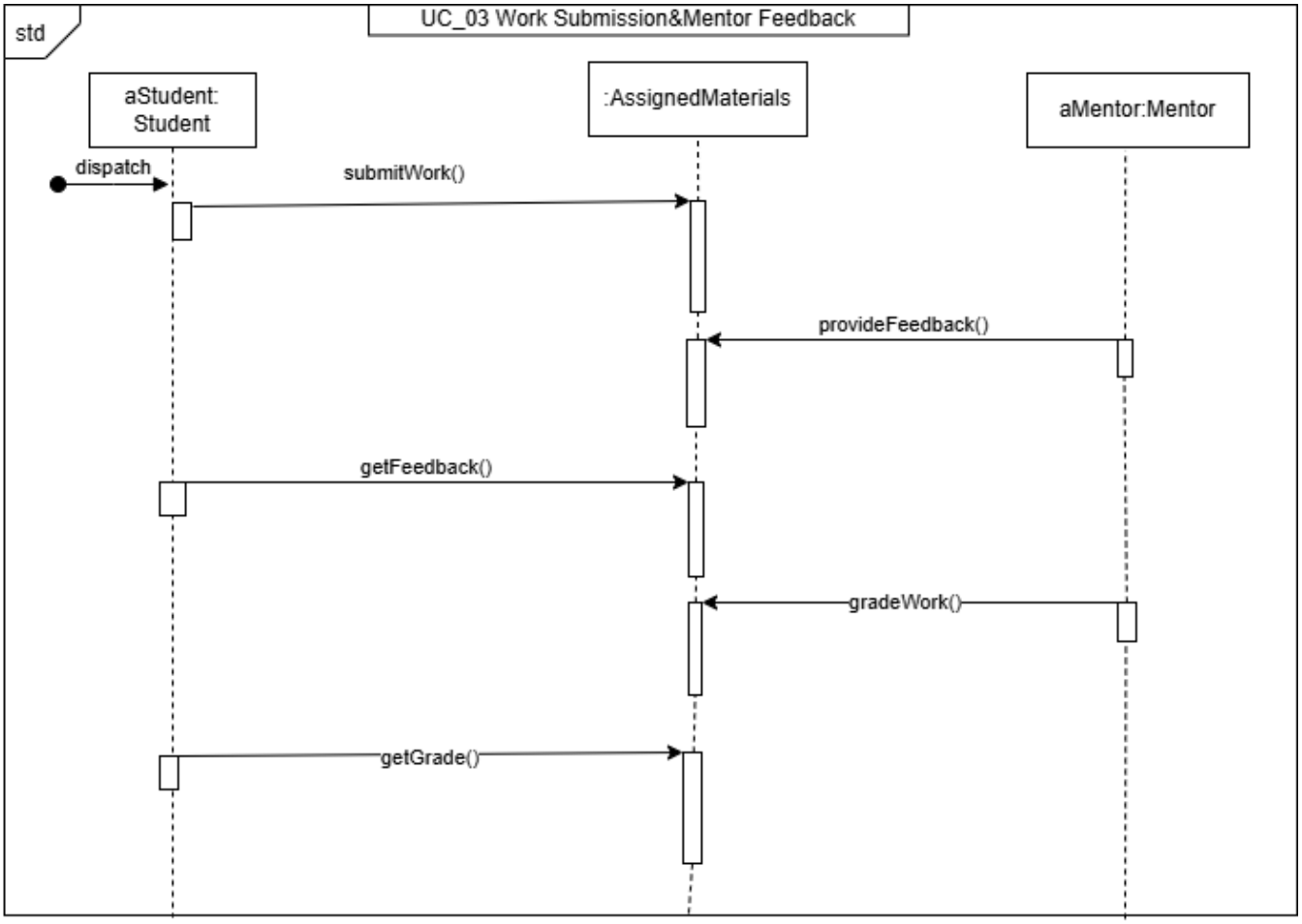
UC_01- Jagoda Andrea



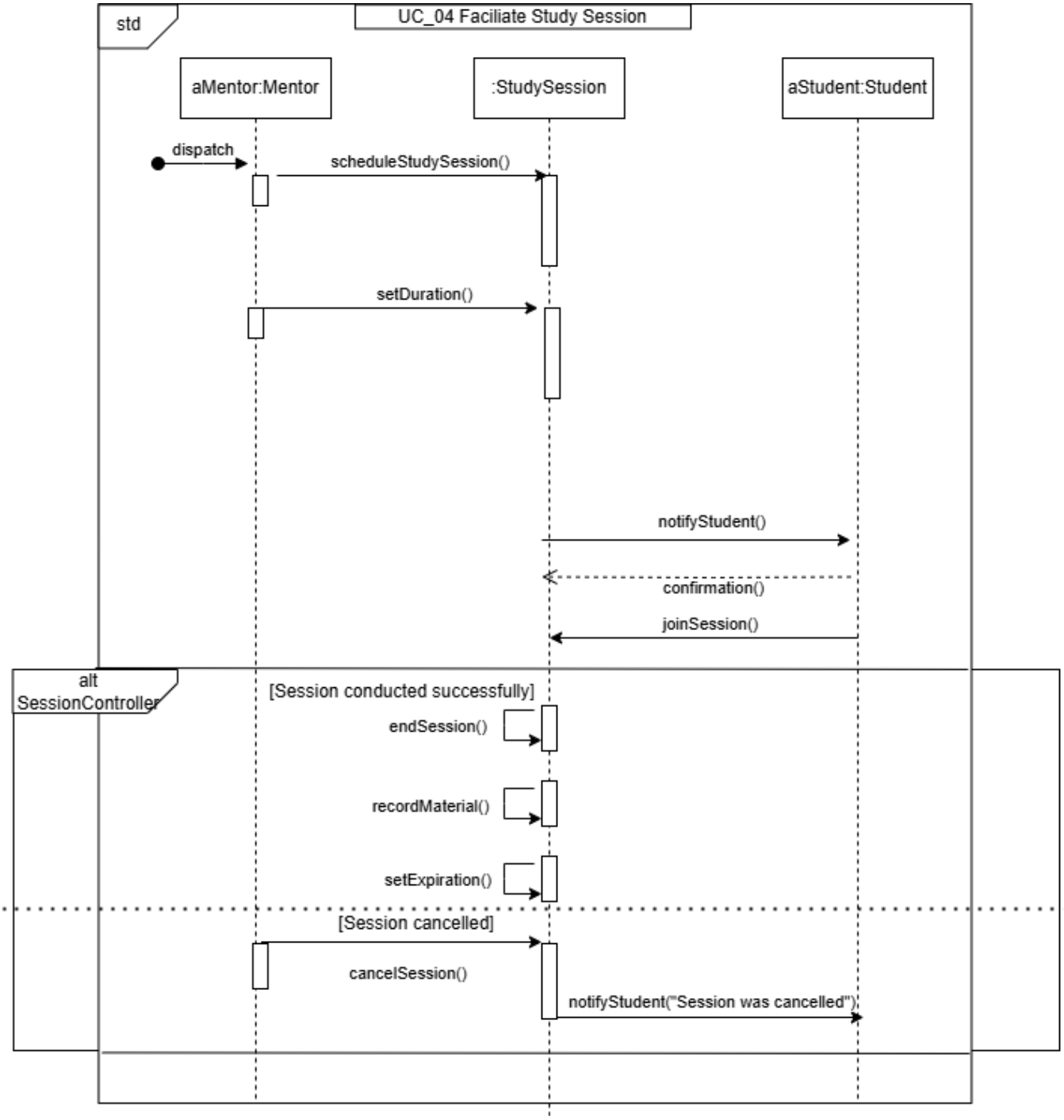
UC_02-Jagoda Andrea



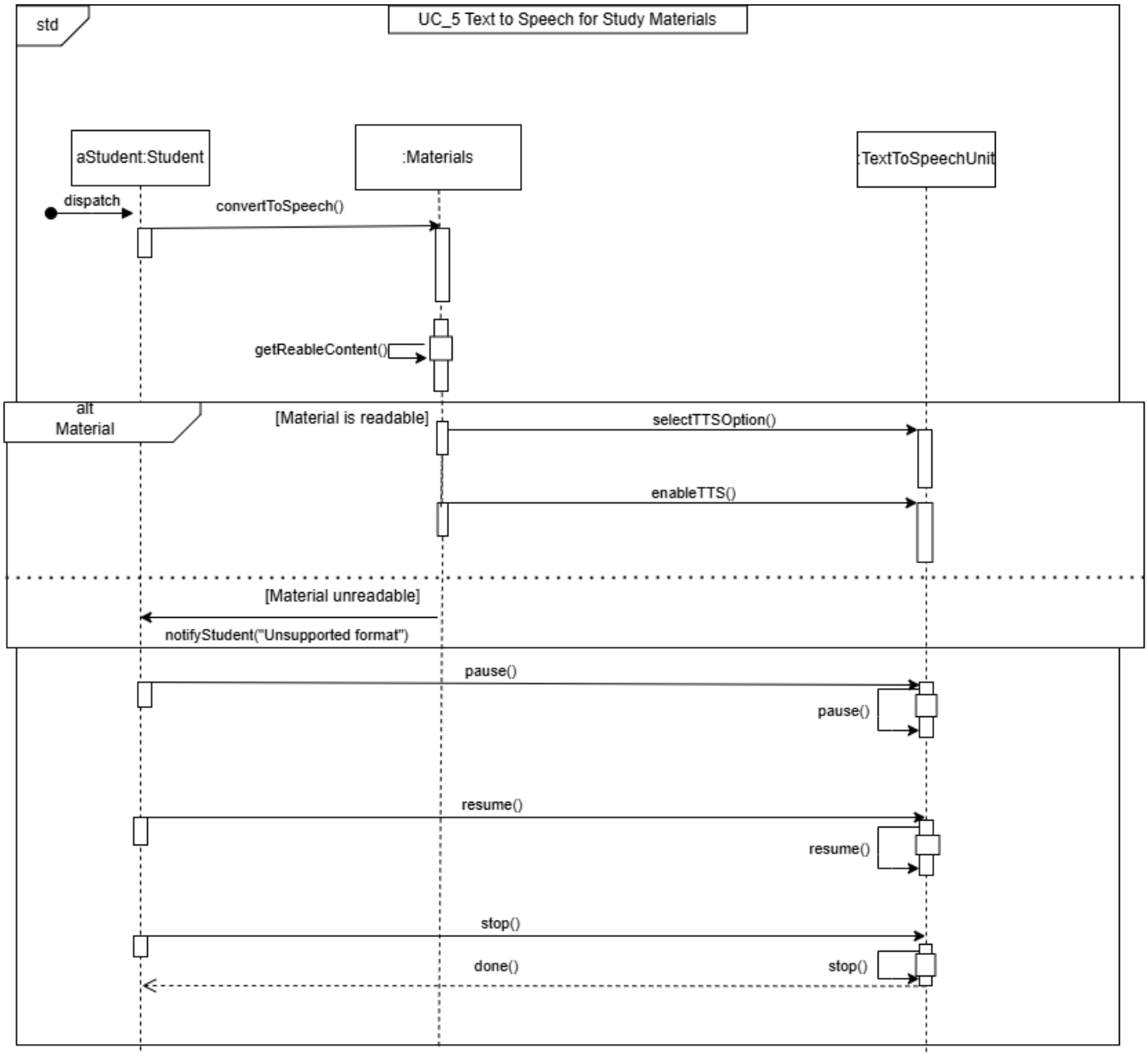
UC_03 -Jagoda Andrea



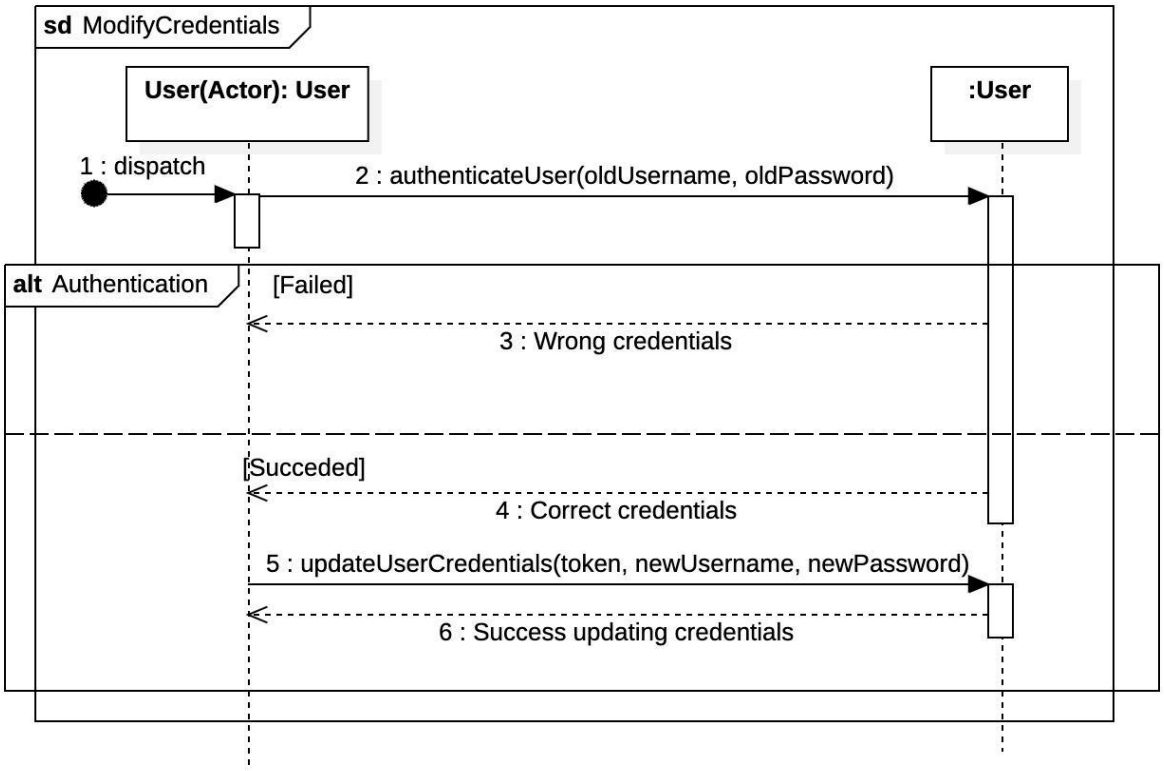
UC_04-Jagoda Andrea



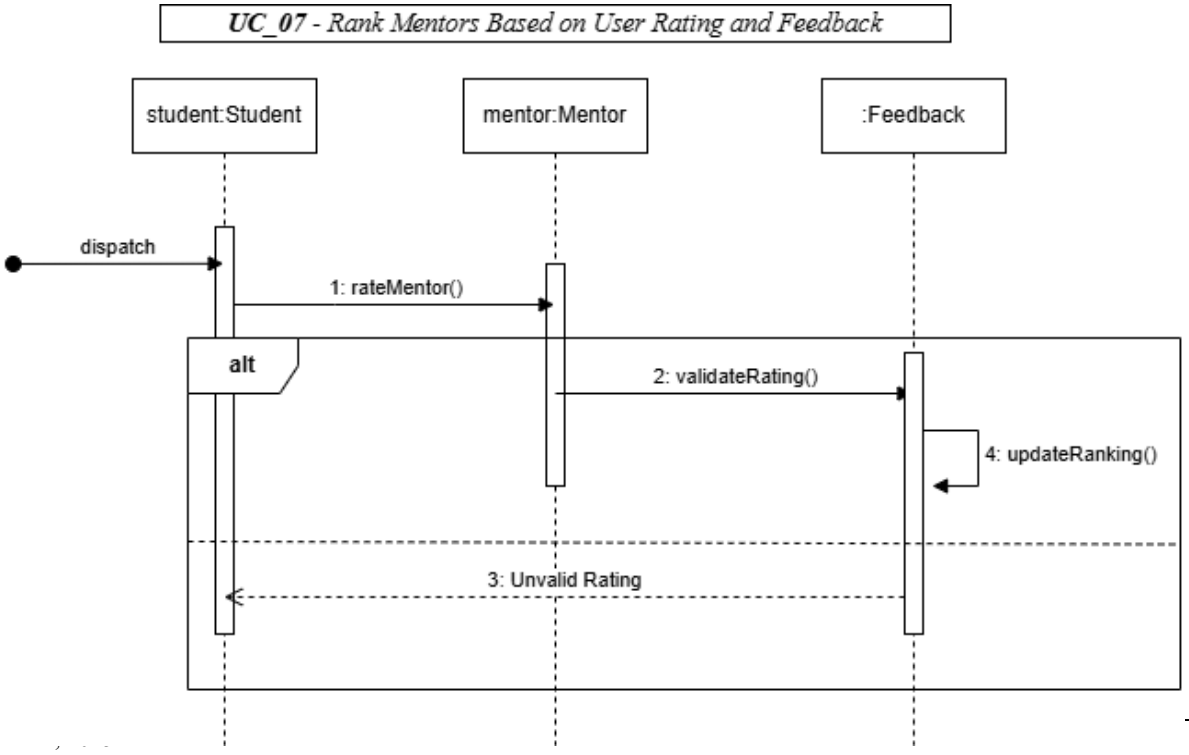
UC_05- Jagoda Andrea



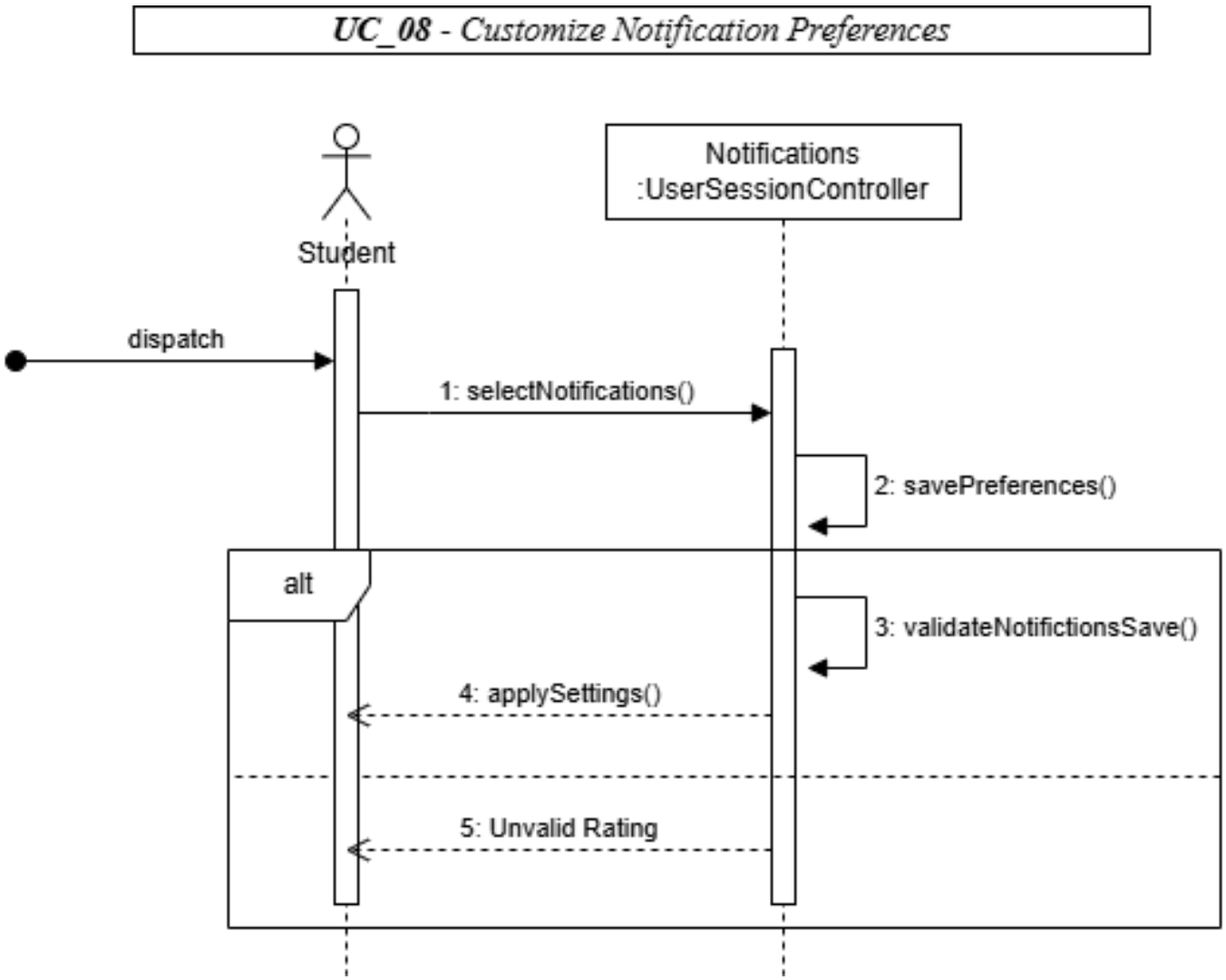
UC_6 – Halil



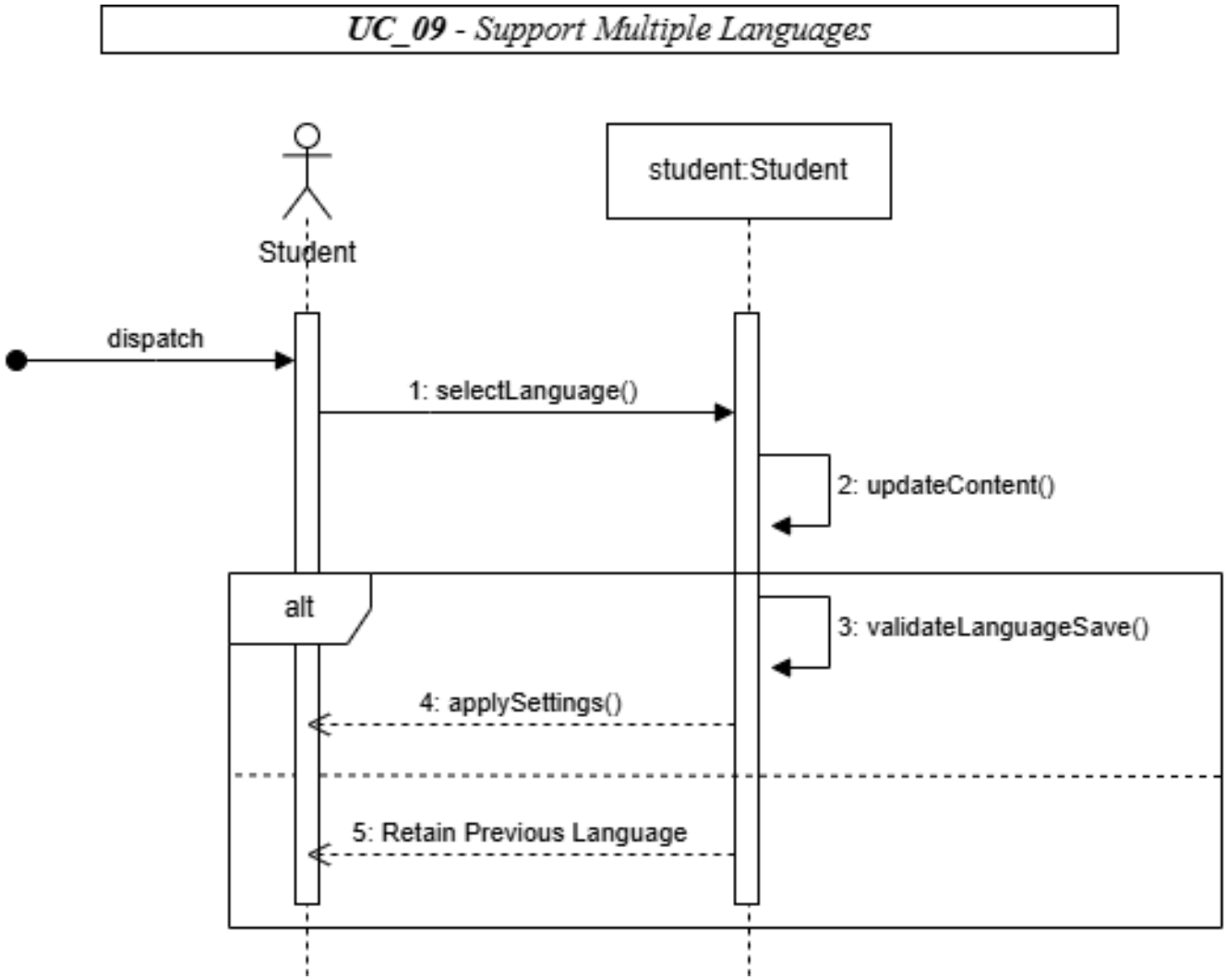
UC_7 - Dionis



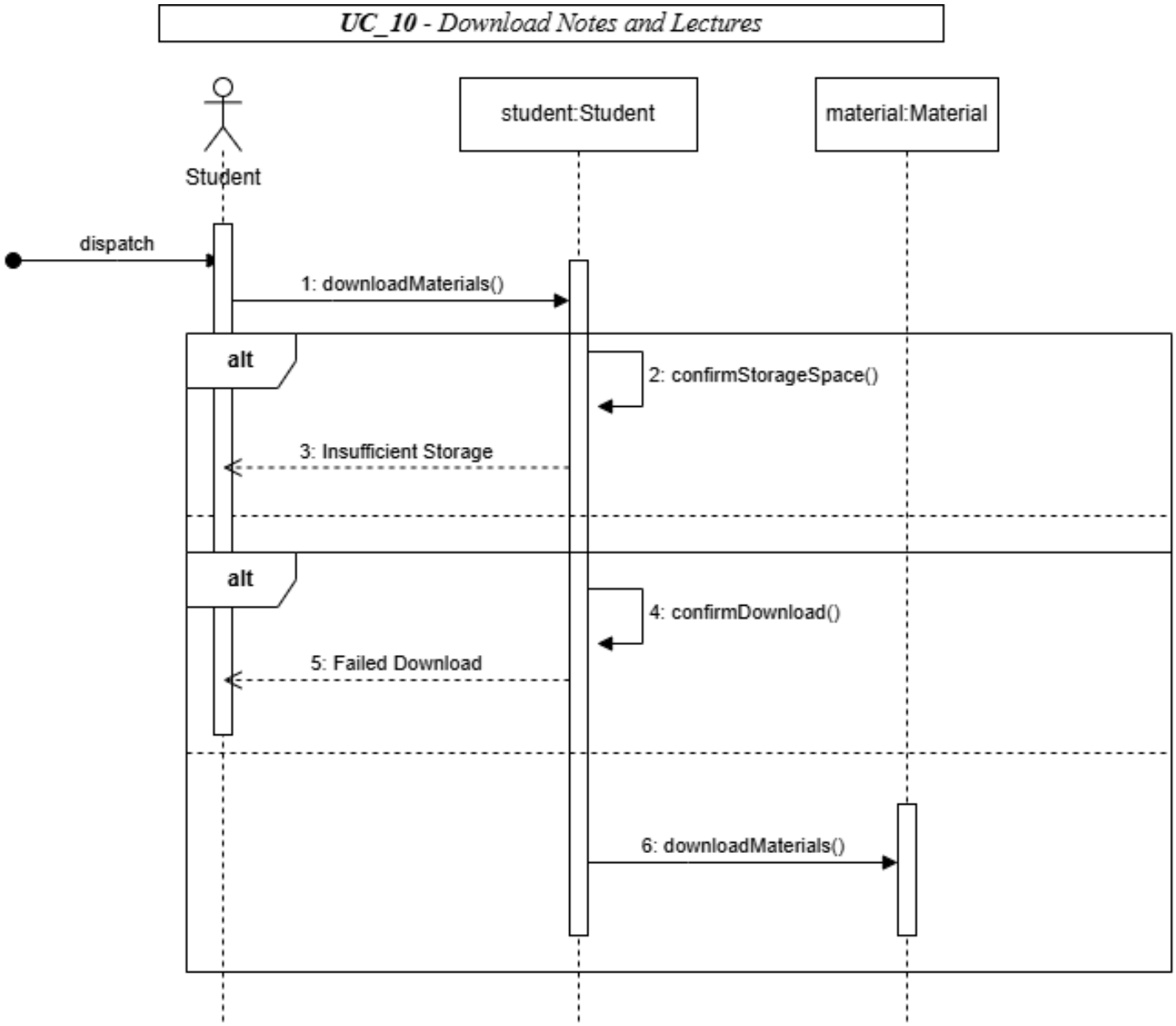
UC_8 - Dionis


UC_08 - Customize Notification Preferences
Student
Notifications
:UserSessionController
dispatch
1: selectNotifications()
2: savePreferences()
alt
3: validateNotifictionsSave()
4: applySettings()
5: Unvalid Rating

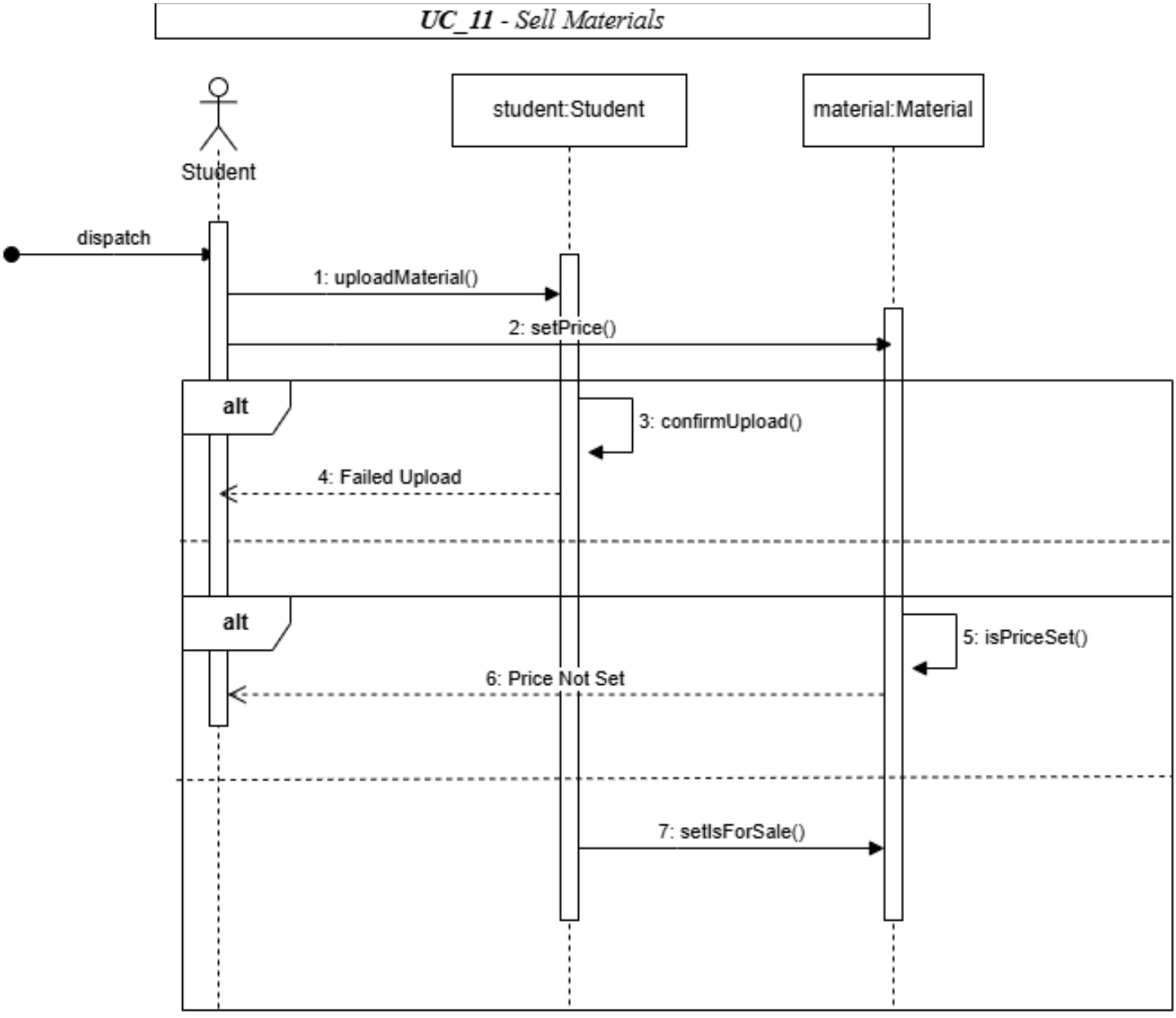
UC_9 - Dionis


UC_09 - Support Multiple Languages
Student
student:Student
dispatch
1: selectLanguage()
2: updateContent()
alt
3: validateLanguageSave()
4: applySettings()
5: Retain Previous Language

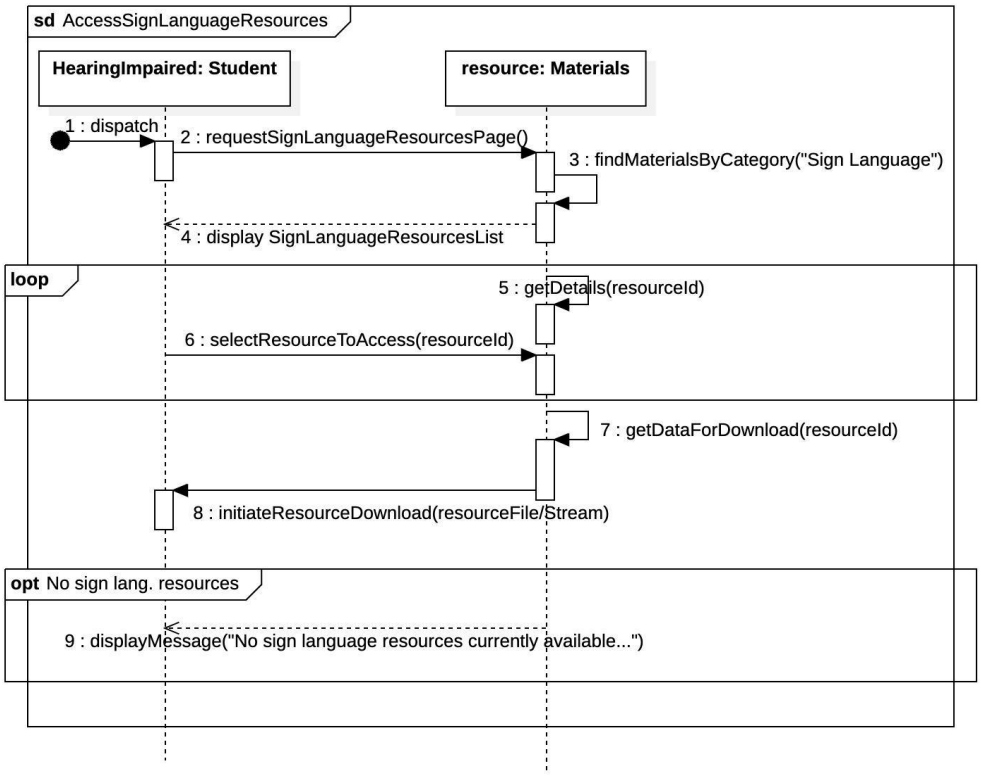
UC_10 - Dionis


UC_10 - Download Notes and Lectures
Student
student:Student
material:Material
dispatch
1: downloadMaterials()
alt
2: confirmStorageSpace()
3: Insufficient Storage
alt
4: confirmDownload()
5: Failed Download
6: downloadMaterials()

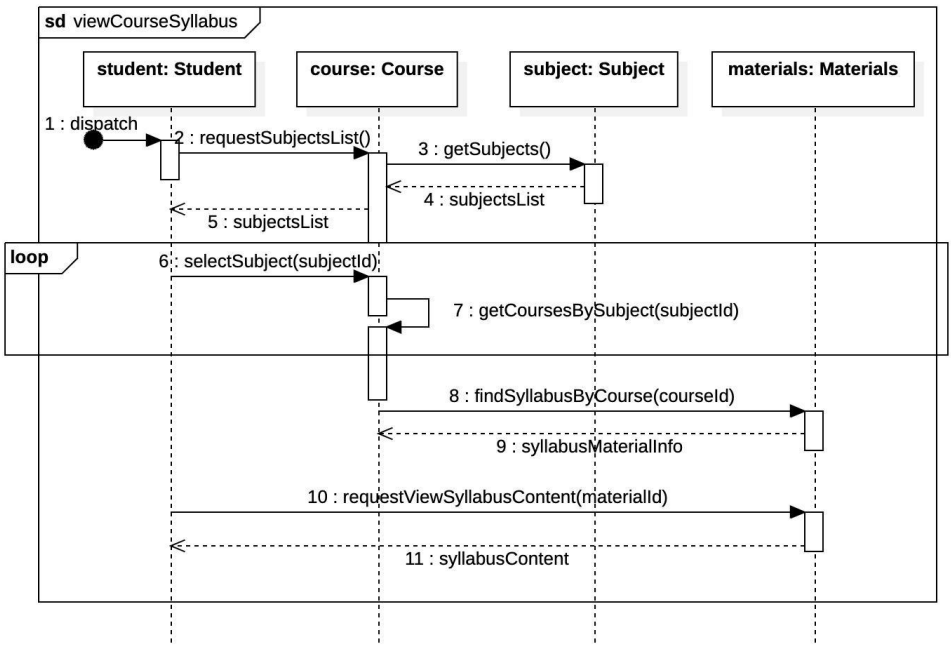
UC_11 - Dionis


UC_11 - Sell Materials
Student
student:Student
material:Material
dispatch
1: uploadMaterial()
2: setPrice()
alt
3: confirmUpload()
4: Failed Upload
alt
5: isPriceSet()
6: Price Not Set
7: setIsForSale()

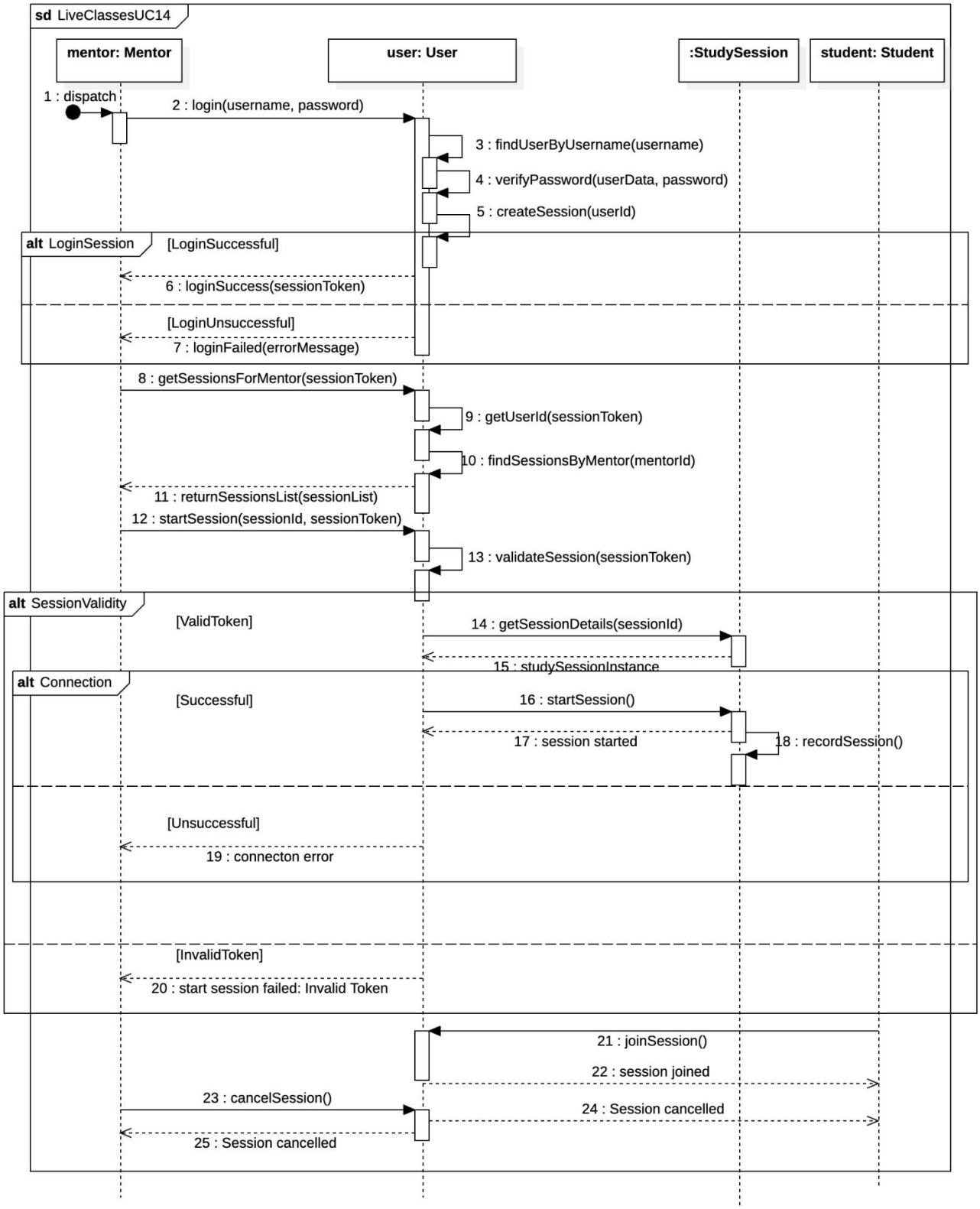
UC_12 - Marvi



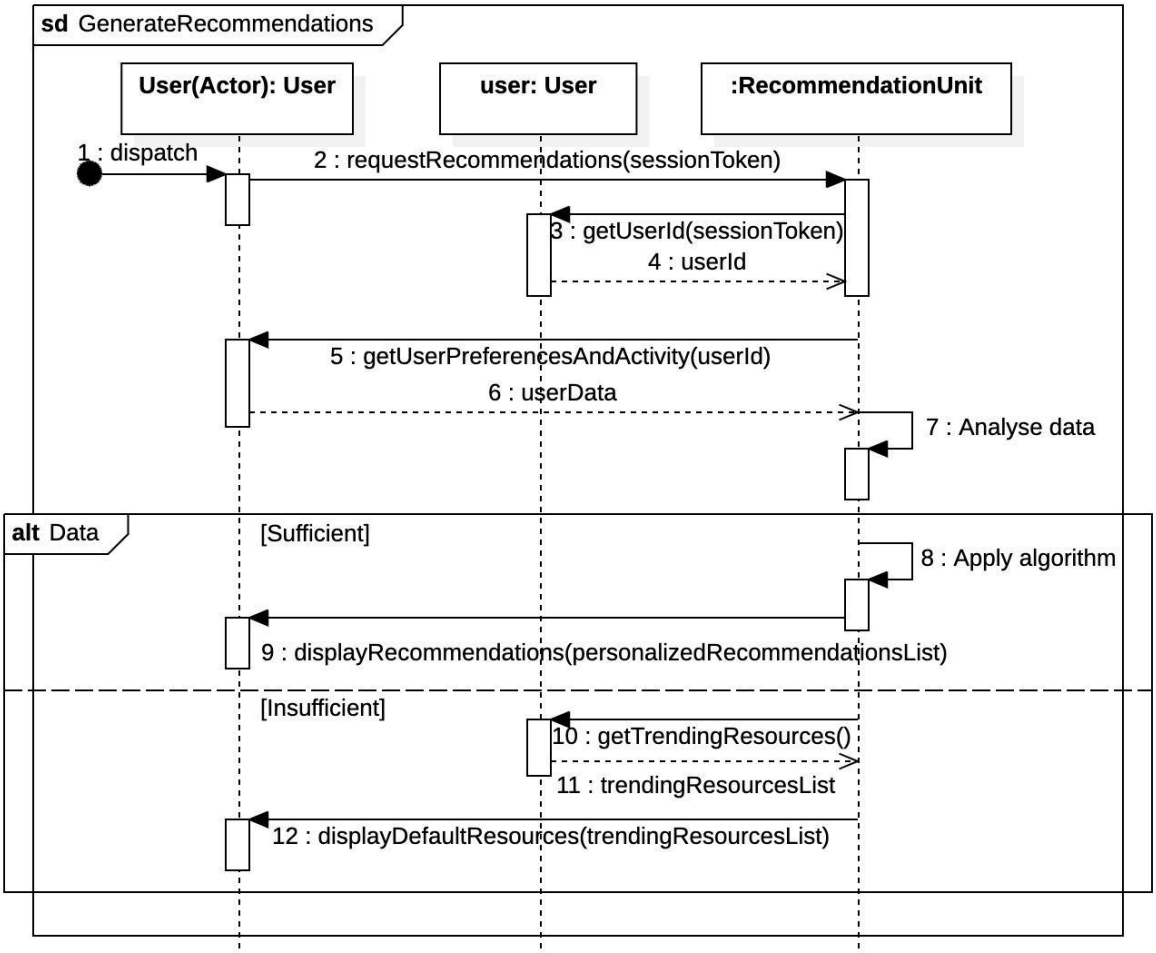
UC_13 - Marvi



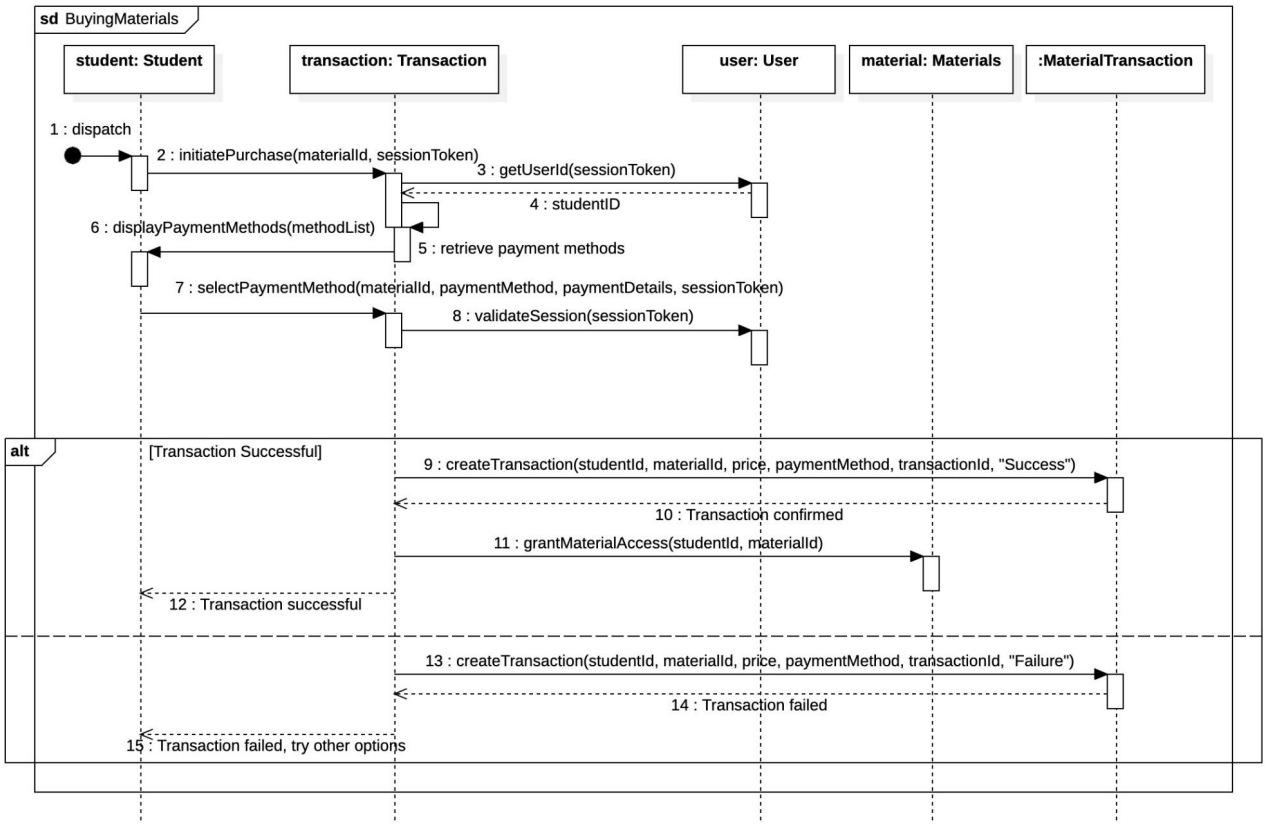
UC_14 - Halil


sd LiveClassesUC14
mentor: Mentor
user: User
:StudySession
student: Student
1 : dispatch
2 : login(username, password)
3 : findUserByUsername(username)
4 : verifyPassword(userData, password)
5 : createSession(userId)
alt LoginSession
[LoginSuccessful]
6 : loginSuccess(sessionToken)
[LoginUnsuccessful]
7 : loginFailed(errorMessage)
8 : getSessionsForMentor(sessionToken)
9 : getUserId(sessionToken)
10 : findSessionsByMentor(mentorId)
11 : returnSessionsList(sessionList)
12 : startSession(sessionId, sessionToken)
13 : validateSession(sessionToken)
alt SessionValidity
[ValidToken]
14 : getSessionDetails(sessionId)
15 : studySessionInstance
alt Connection
[Successful]
16 : startSession()
17 : session started
18 : recordSession()
[Unsuccessful]
19 : connecton error
[InvalidToken]
20 : start session failed: Invalid Token
21 : joinSession()
22 : session joined
23 : cancelSession()
24 : Session cancelled
25 : Session cancelled

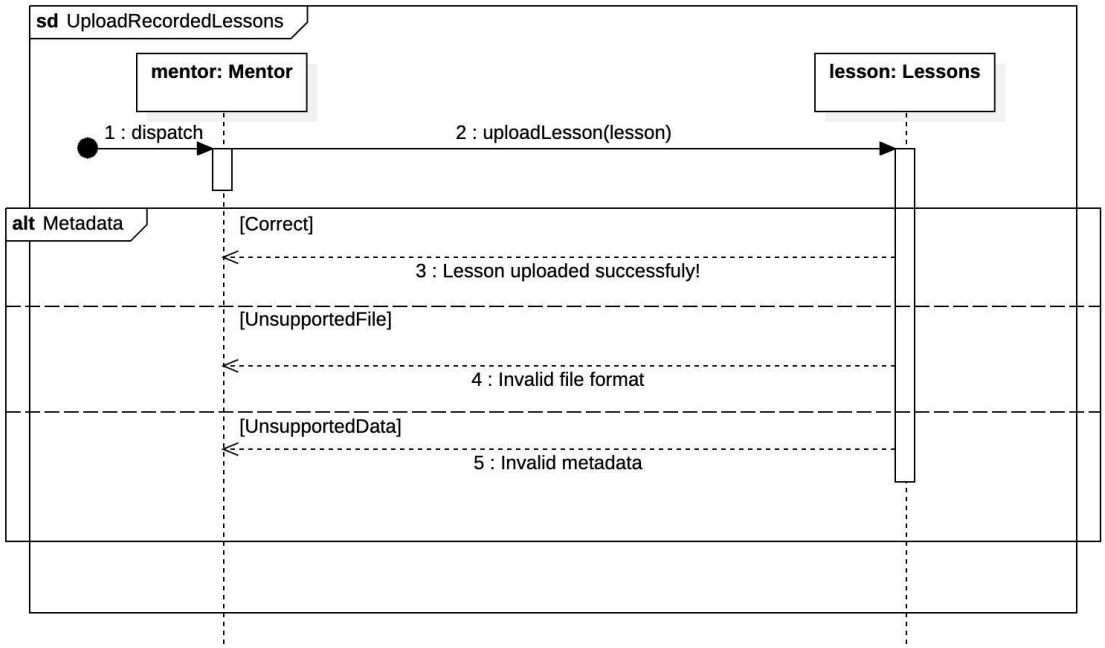
UC_15 - Halil


sd GenerateRecommendations
User(Actor): User
user: User
:RecommendationUnit
1 : dispatch
2 : requestRecommendations(sessionToken)
3 : getUserId(sessionToken)
4 : userId
5 : getUserPreferencesAndActivity(userId)
6 : userData
7 : Analyse data
alt Data
[Sufficient]
8 : Apply algorithm
9 : displayRecommendations(personalizedRecommendationsList)
[Insufficient]
10 : getTrendingResources()
11 : trendingResourcesList
12 : displayDefaultResources(trendingResourcesList)

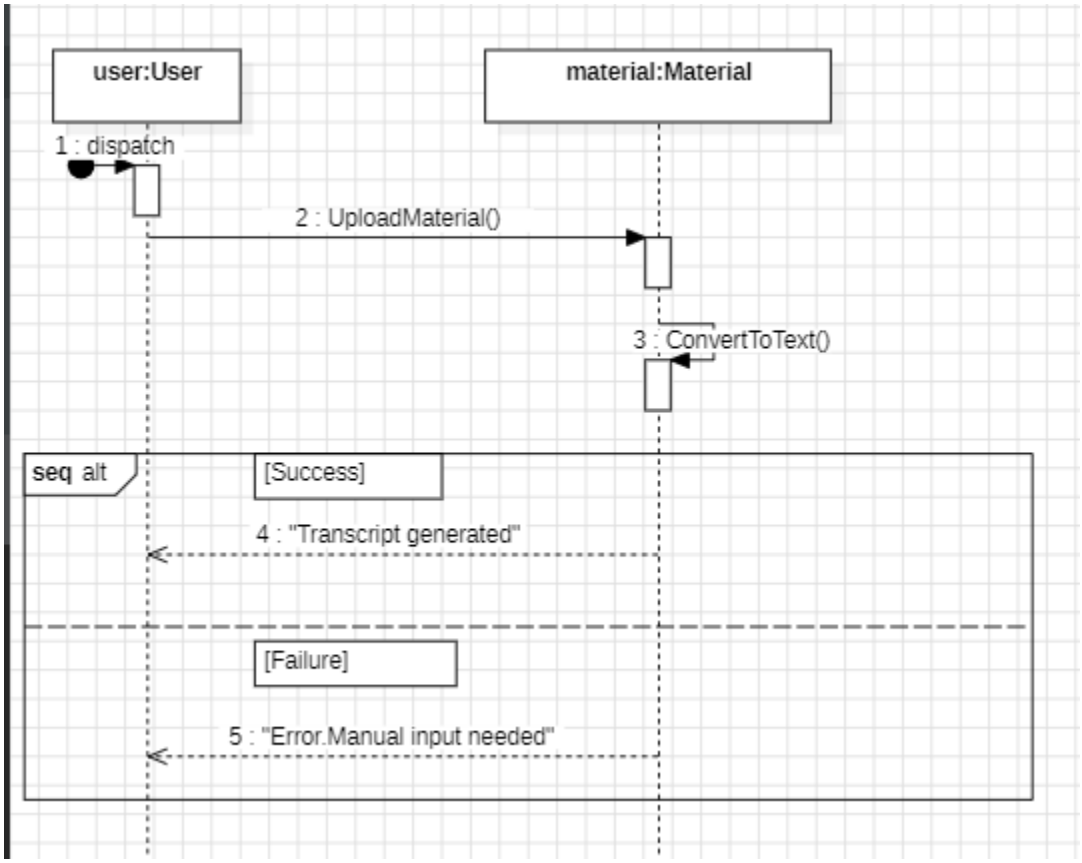
UC_16 - Purchase Materials via multiple payment methods – Halil



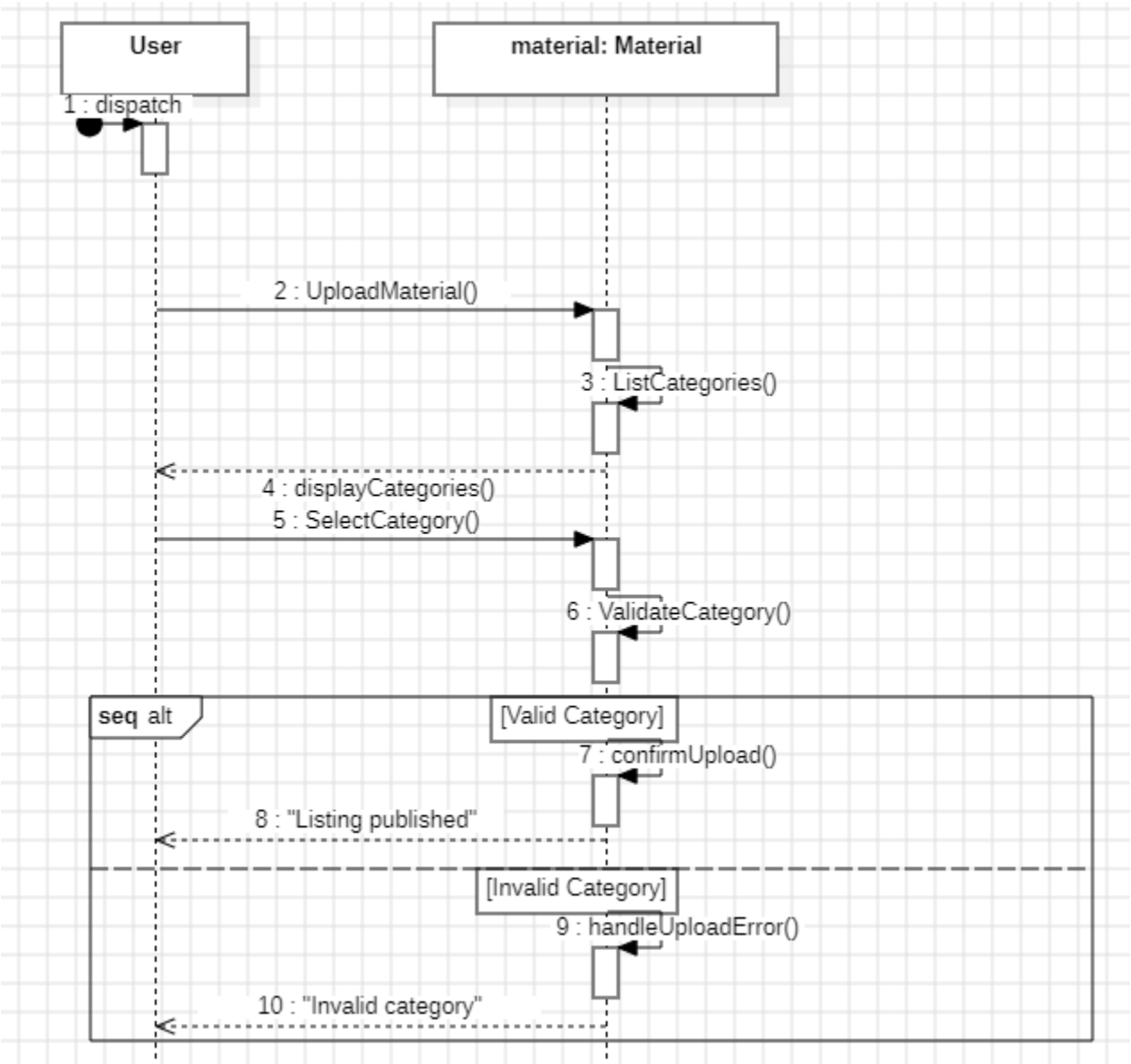
UC_17 - Halil



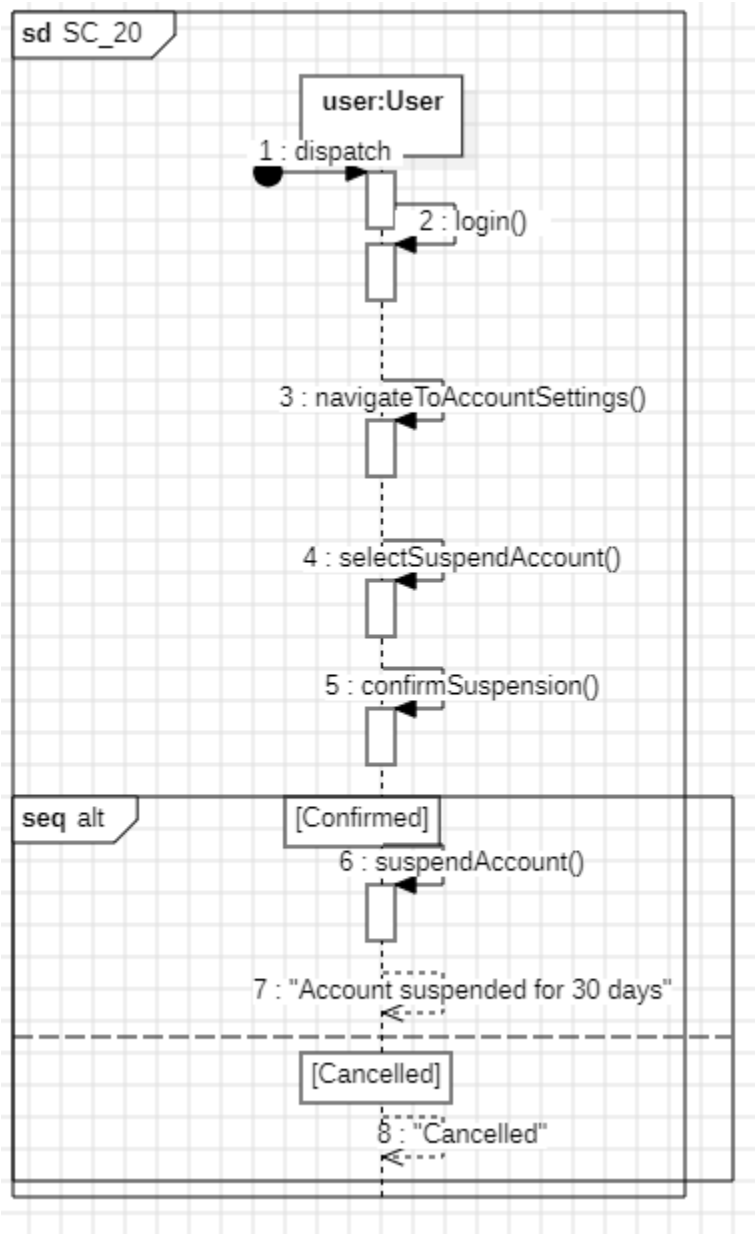
UC_18 - Dea


user:User
material:Material
1 : dispatch
2 : UploadMaterial()
3 : ConvertToText()
seq alt
[Success]
4 : "Transcript generated"
[Failure]
5 : "Error.Manual input needed"

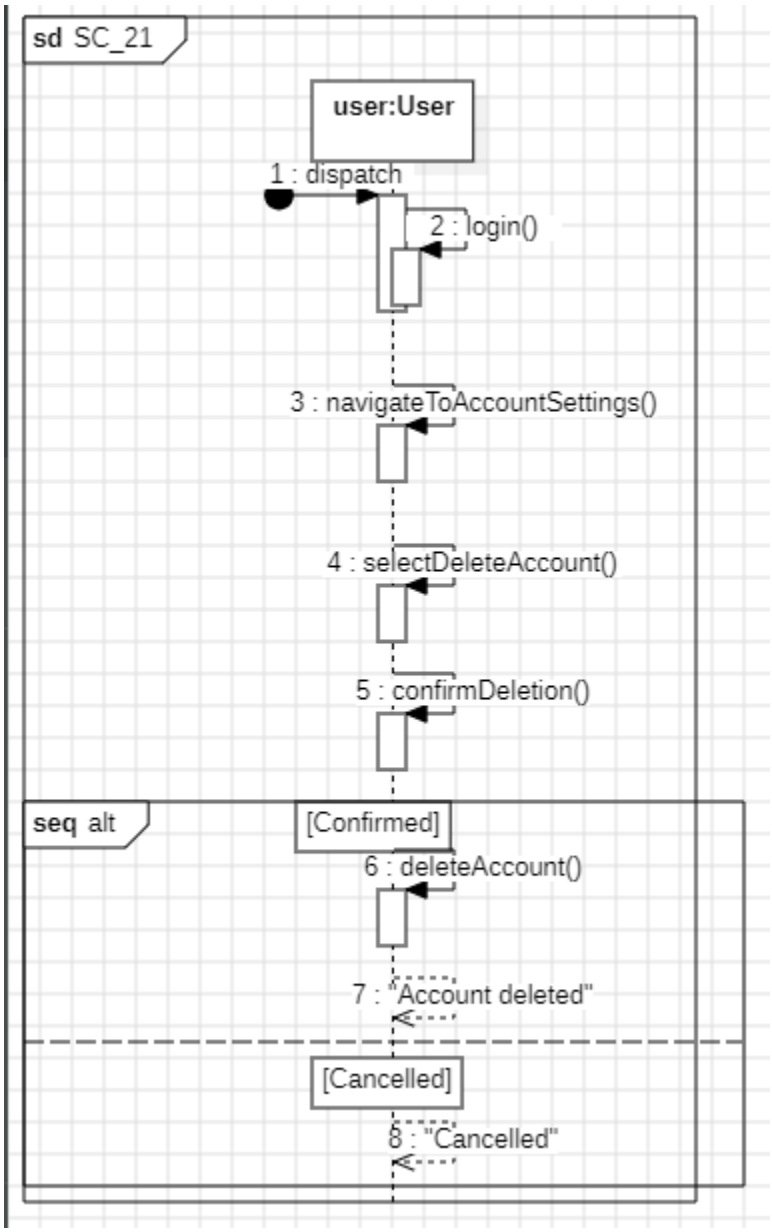
Uc_19 - Account deletion- Dea



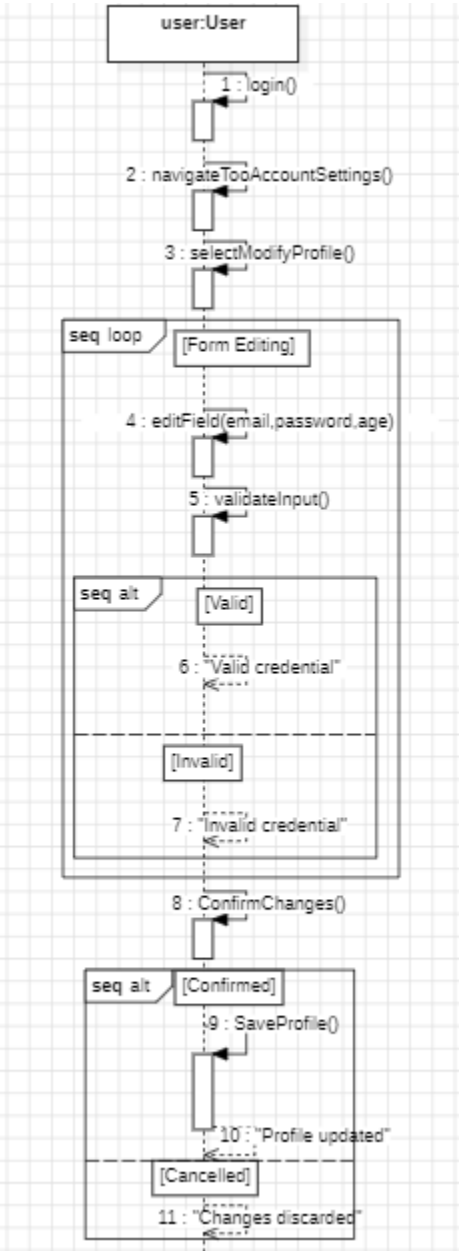
SC_20


sd SC_20
user:User
1 : dispatch
2 : login()
3 : navigateToAccountSettings()
4 : selectSuspendAccount()
5 : confirmSuspension()
seq alt
[Confirmed]
6 : suspendAccount()
7 : "Account suspended for 30 days"
[Cancelled]
8 : "Cancelled"

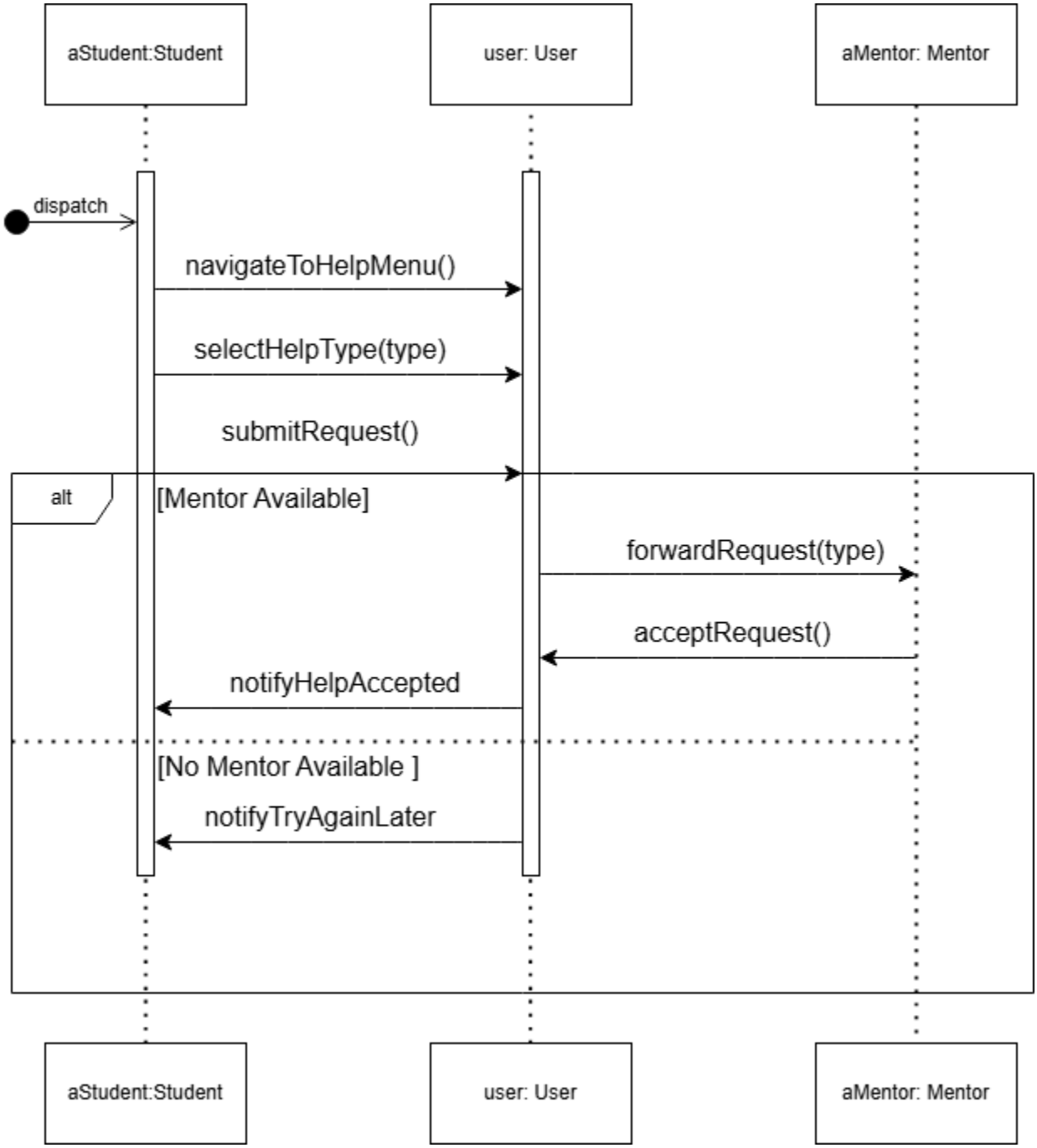
SC_21


sd SC_21
user:User
1 : dispatch
2 : login()
3 : navigateToAccountSettings()
4 : selectDeleteAccount()
5 : confirmDeletion()
seq alt
[Confirmed]
6 : deleteAccount()
7 : "Account deleted"
[Cancelled]
8 : "Cancelled"

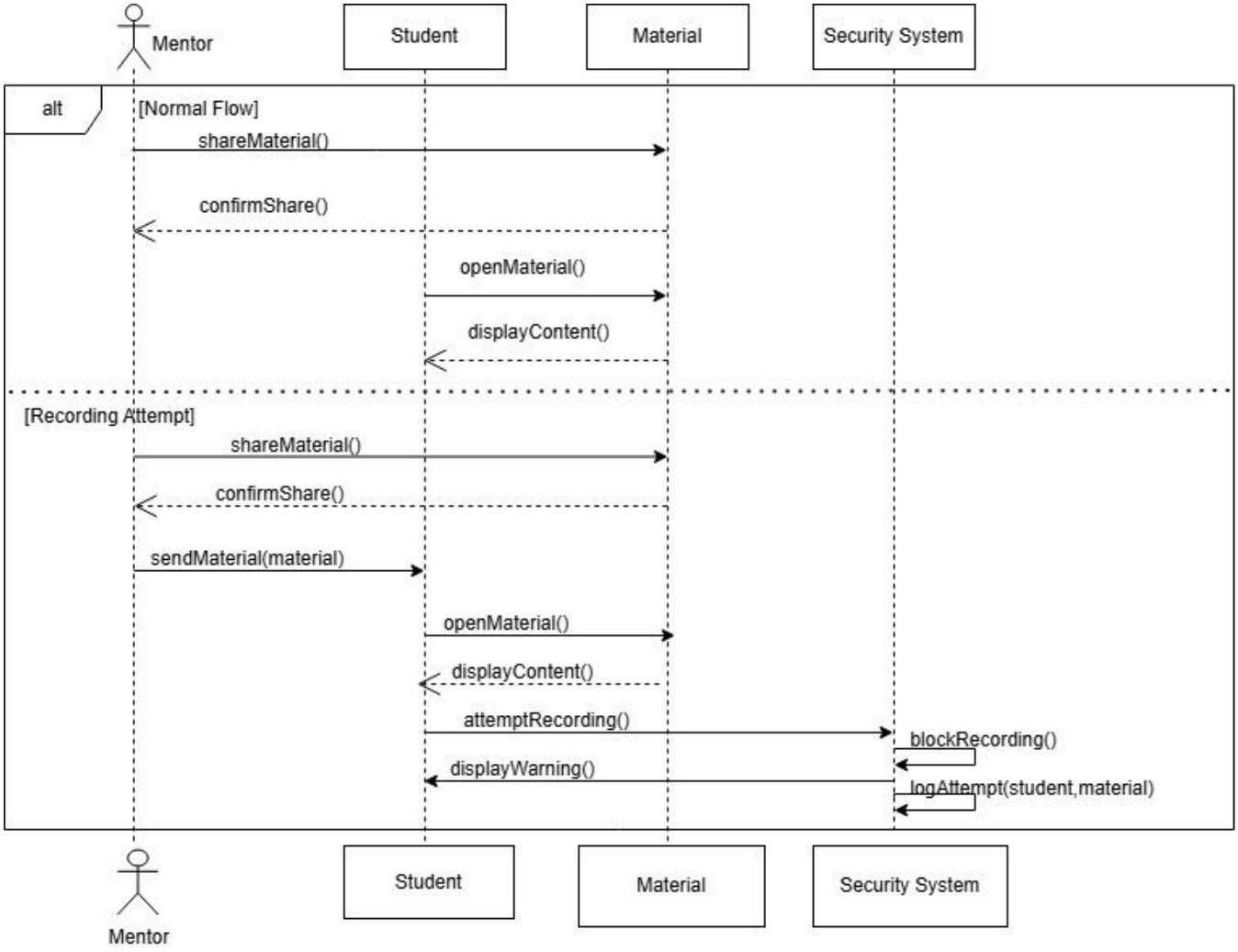
SC_22


user:User
1 : login()
2 : navigateTooAccountSettings()
3 : selectModifyProfile()
seq loop
[Form Editing]
4 : editField(email,password,age)
5 : validateInput()
seq alt
[Valid]
6 : "Valid credential"
[Invalid]
7 : "Invalid credential"
8 : ConfirmChanges()
seq alt
[Confirmed]
9 : SaveProfile()
10 : "Profile updated"
[Cancelled]
11 : "Changes discarded"

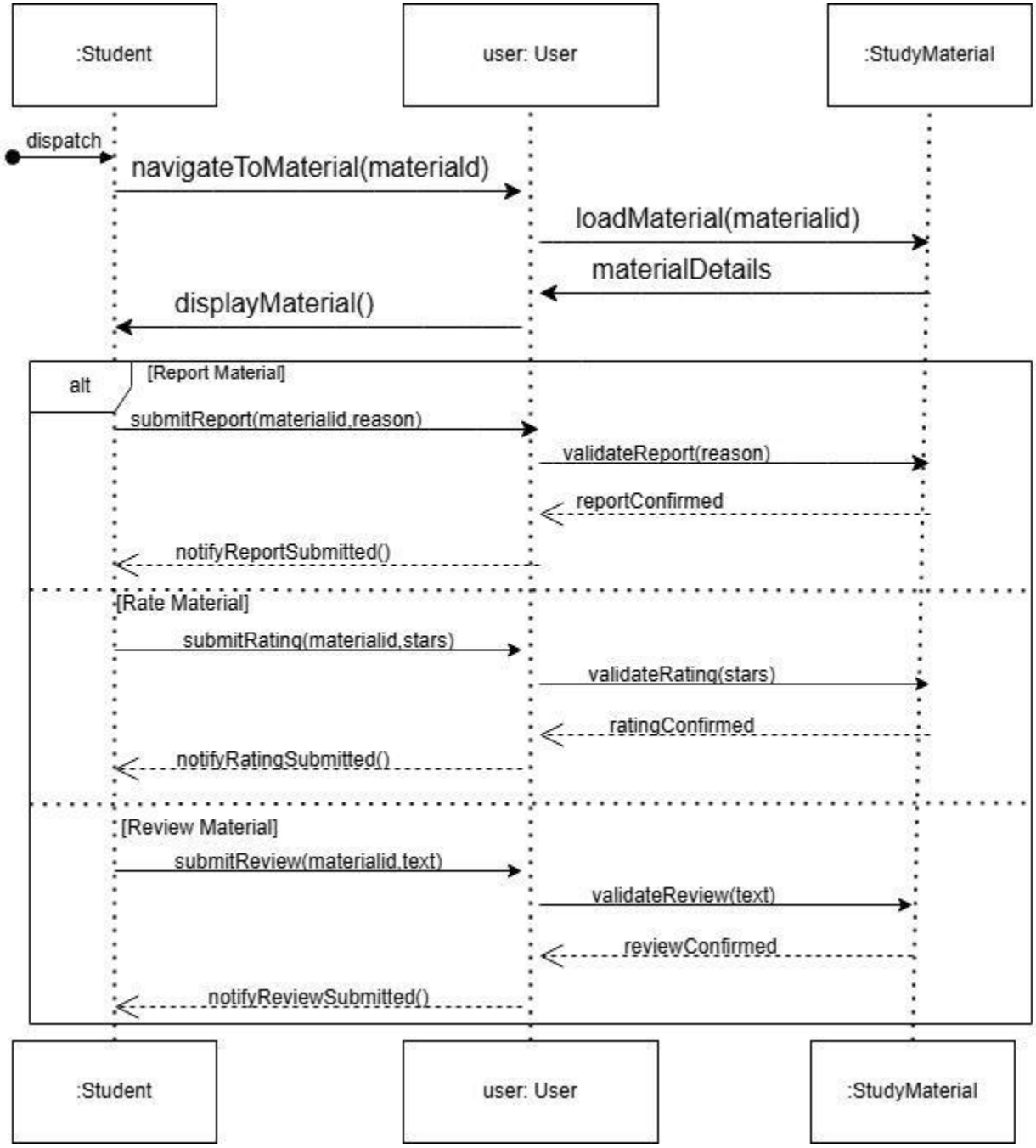
UC 23 Ameraldo



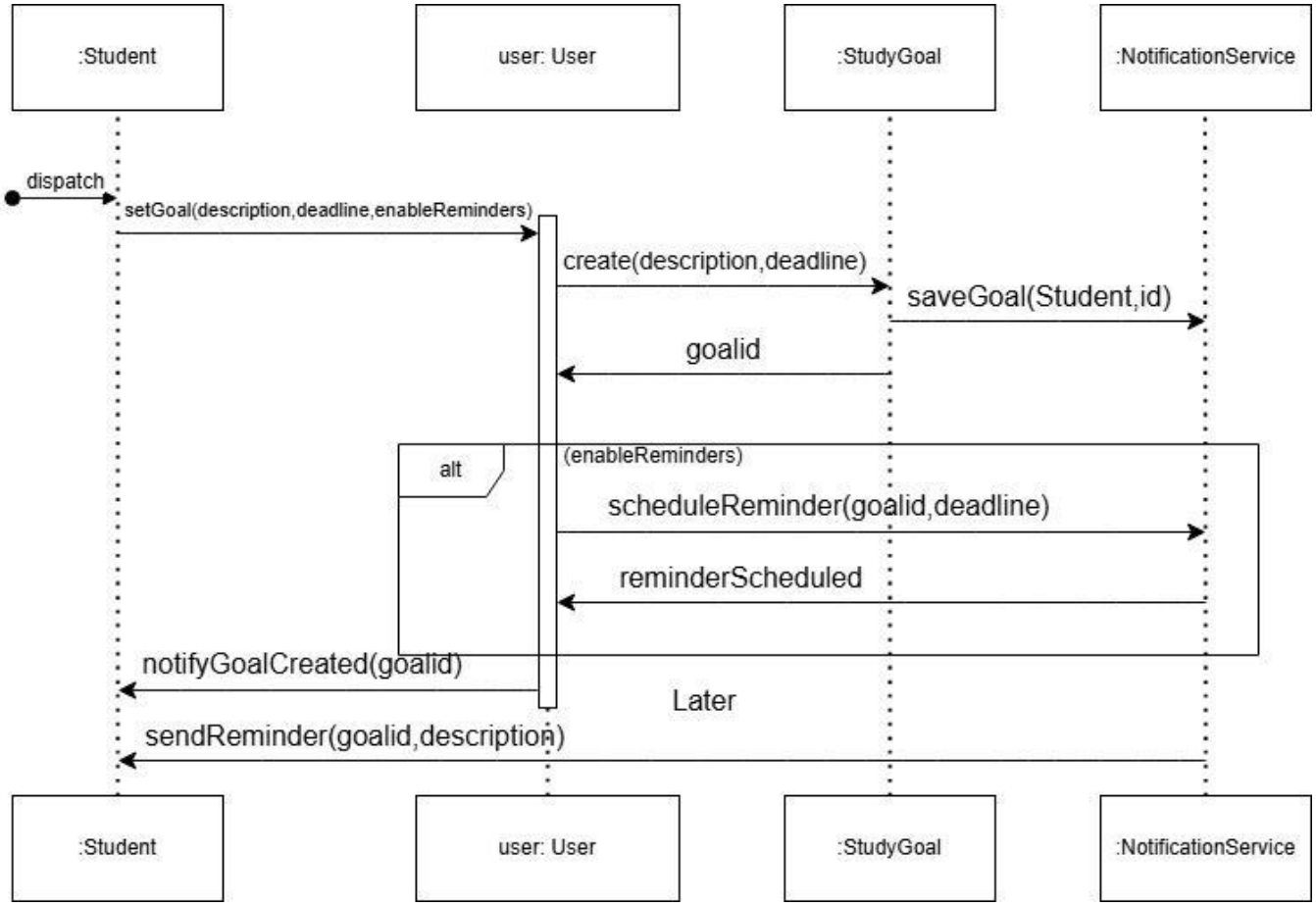
UC 24 - Ameraldo



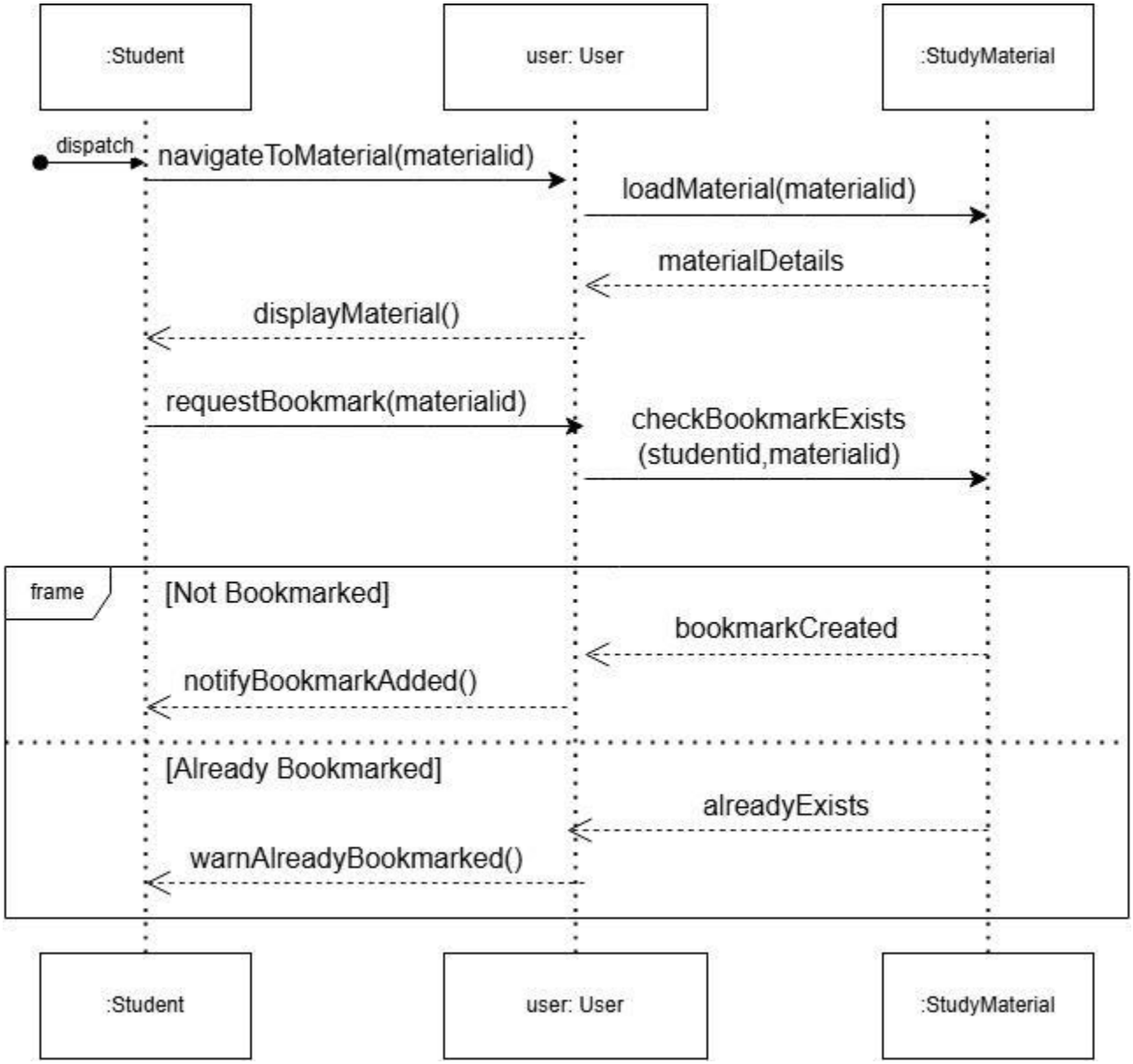
UC 25 Ameraldo



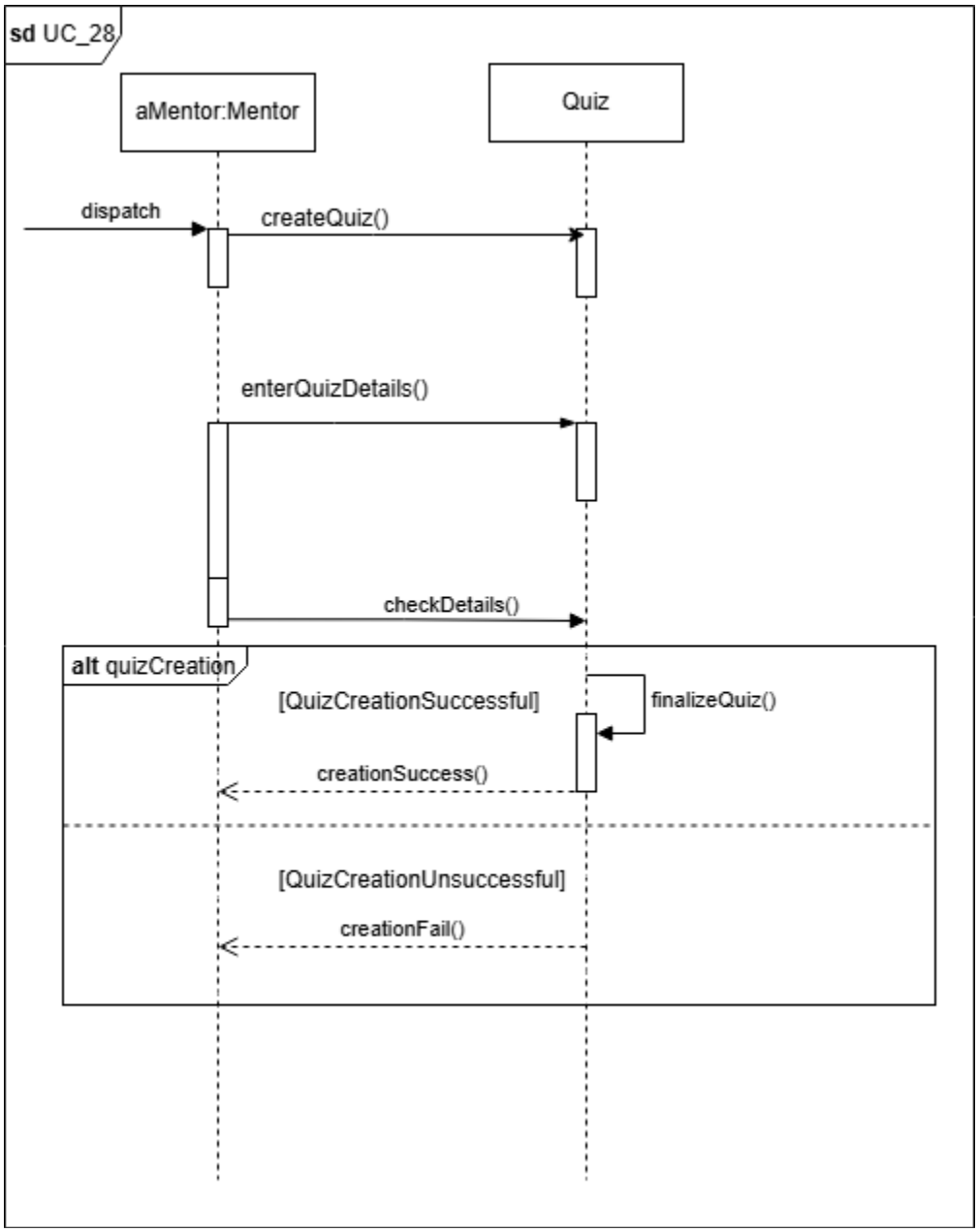
UC 26 Ameraldo



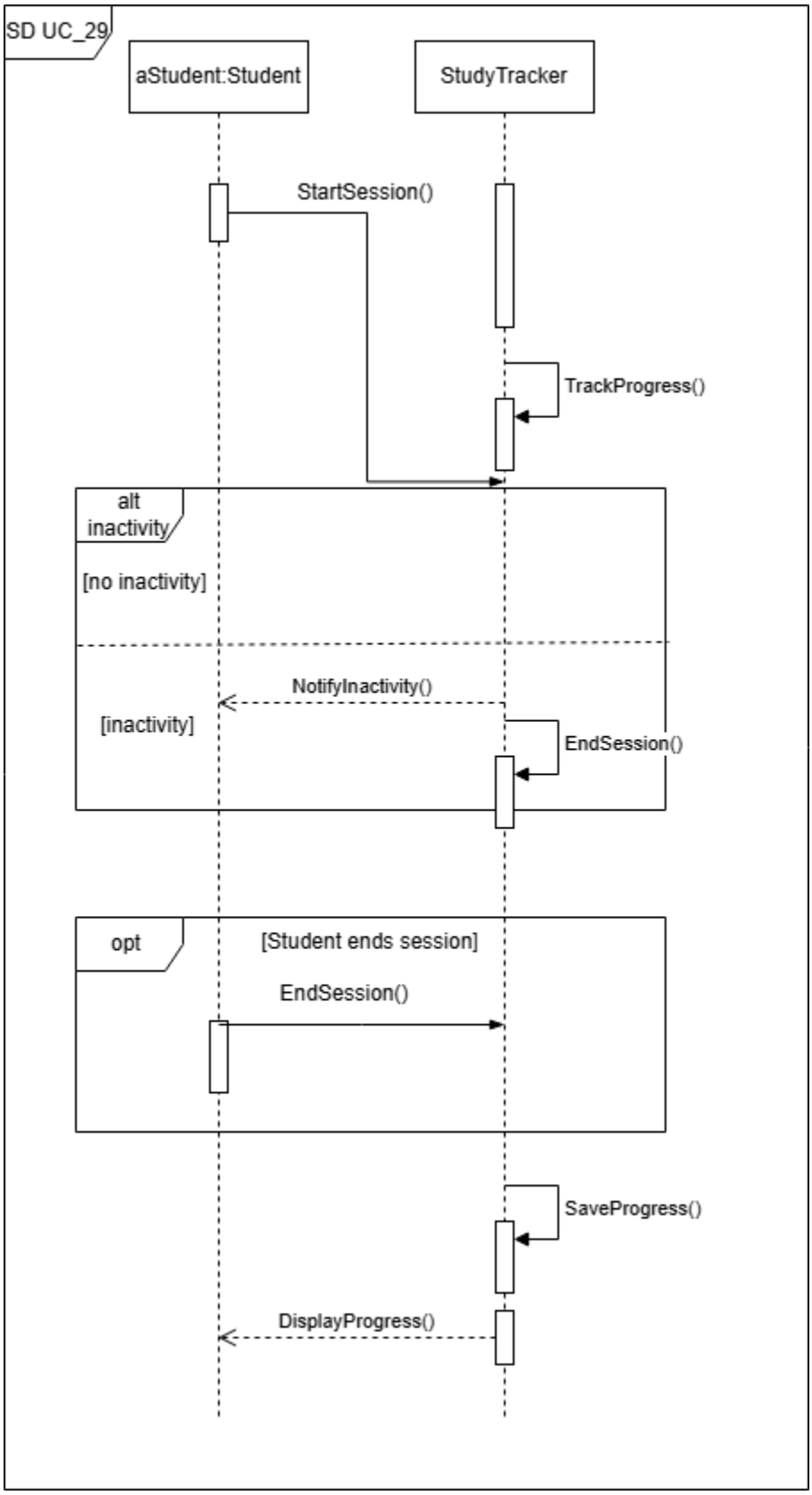
UC 27 Ameraldo



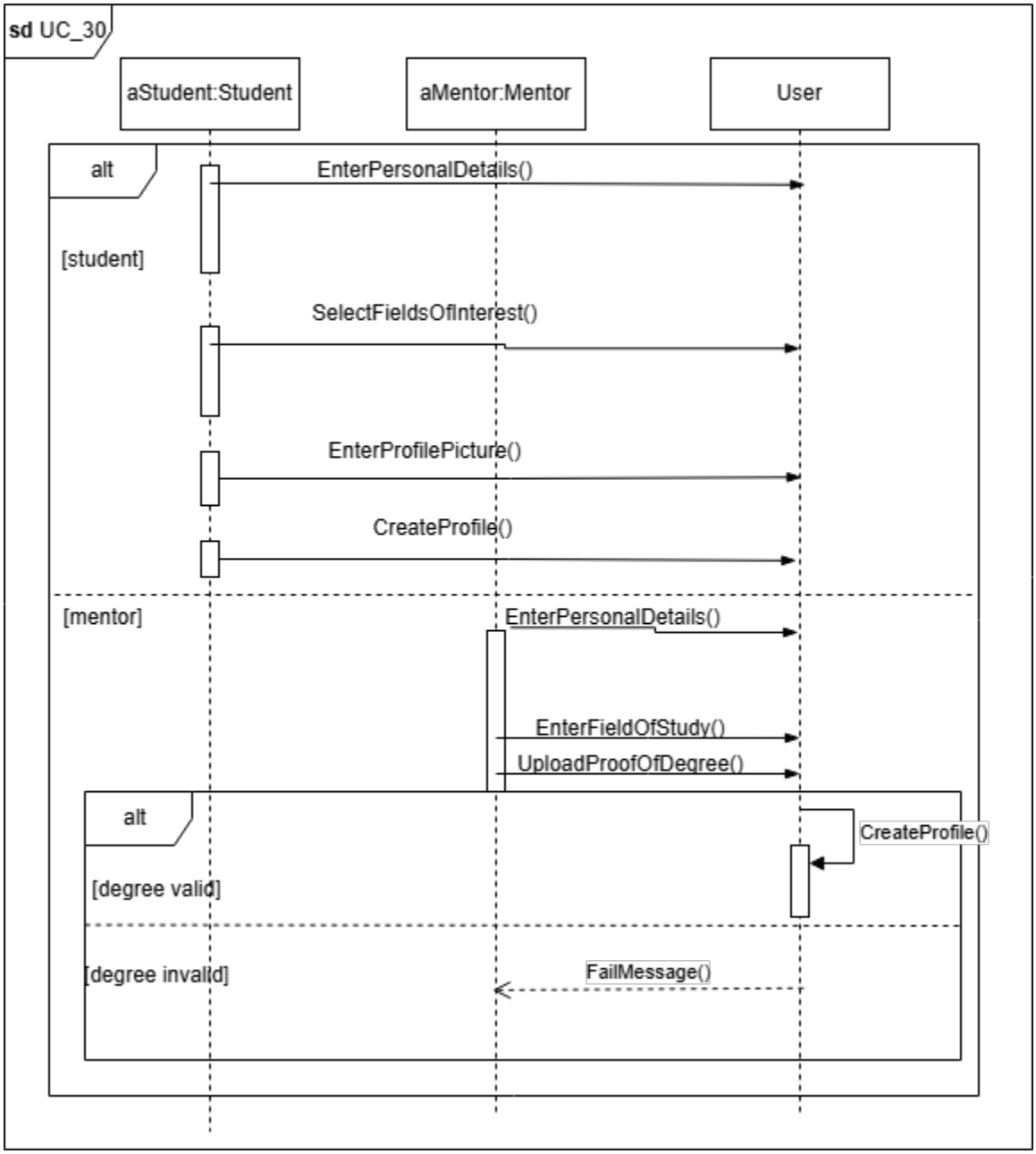
UC 28 Daniela



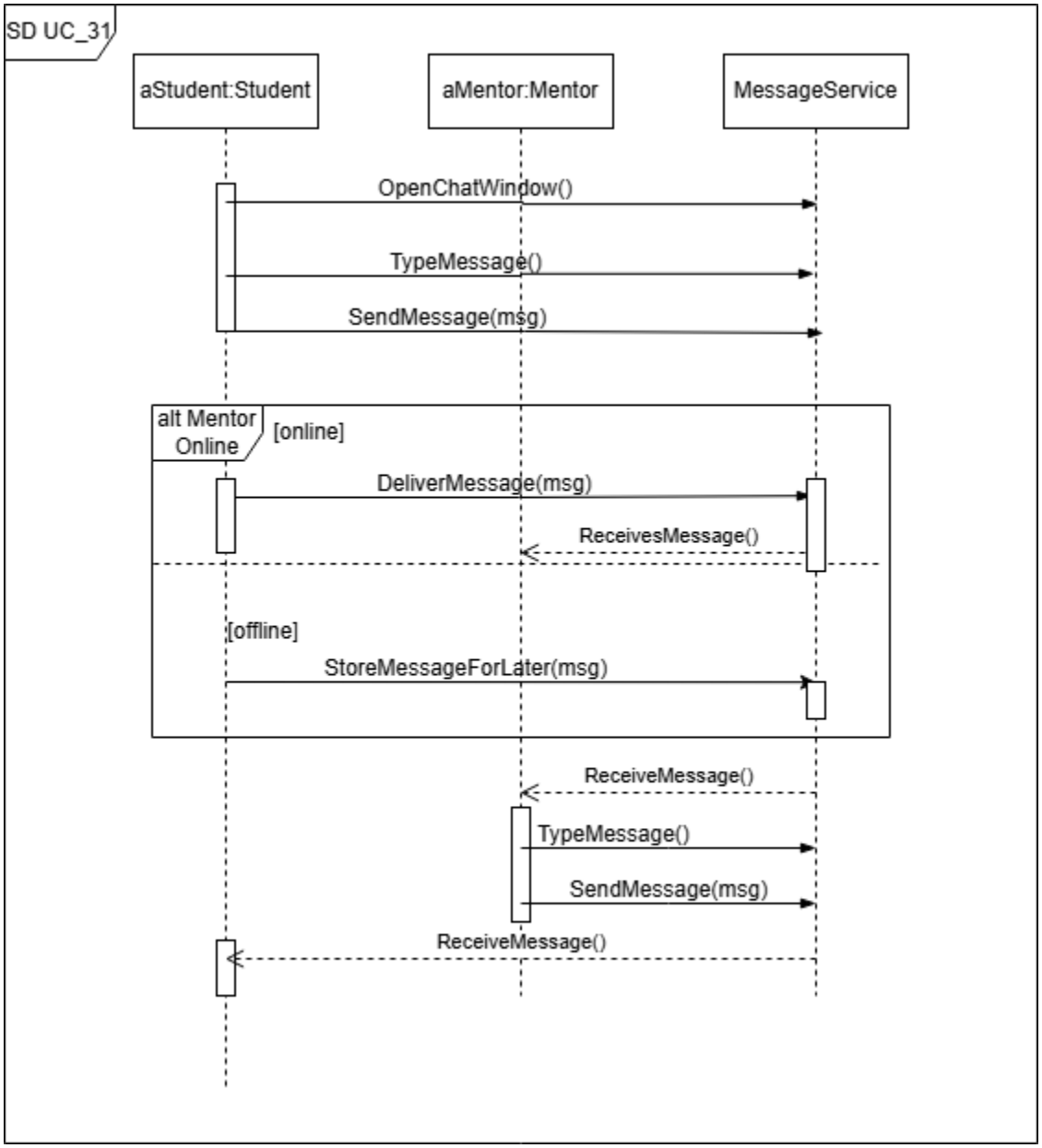
UC 29 Daniela



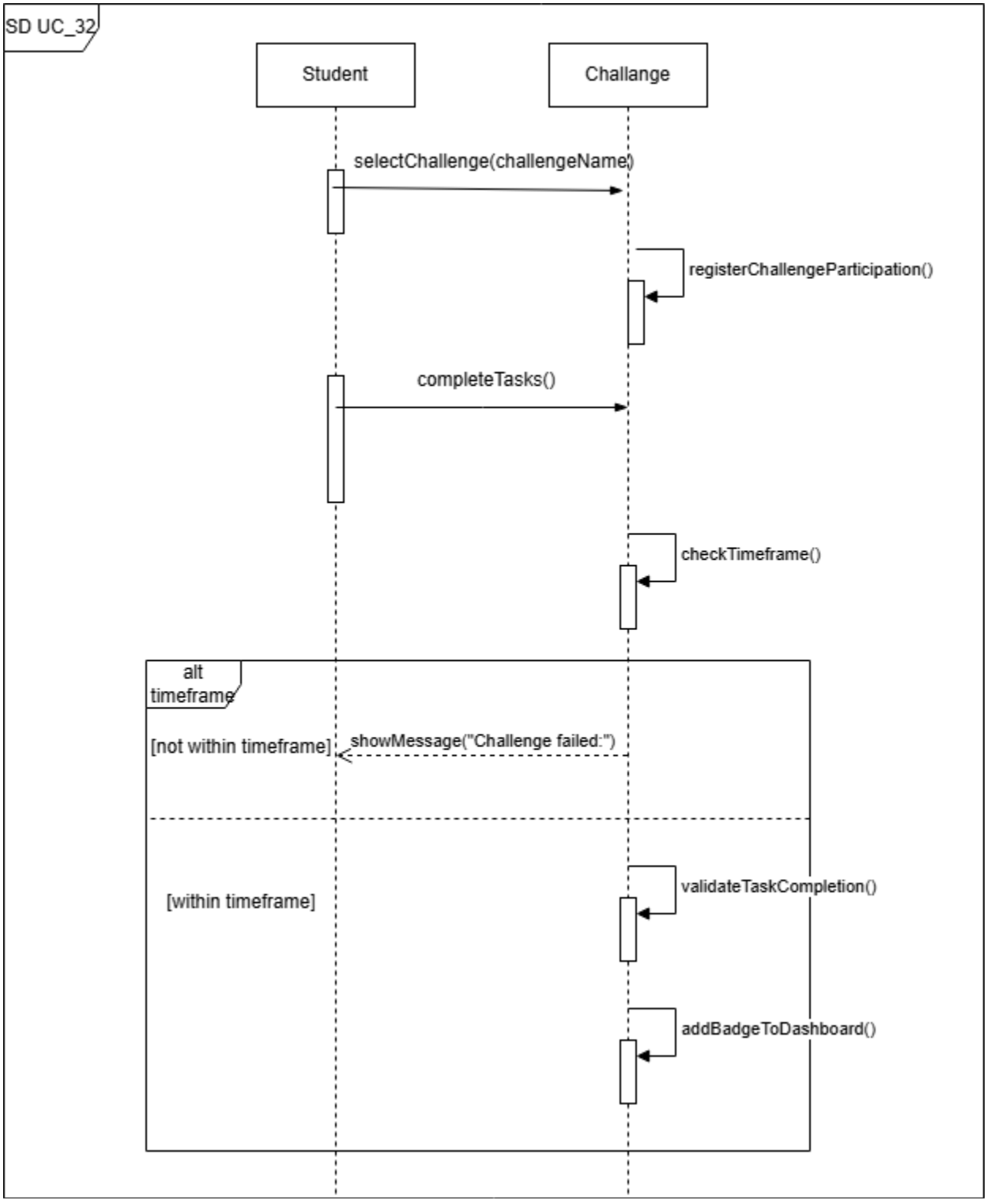
UC 30 Daniela



UC 31 Daniela

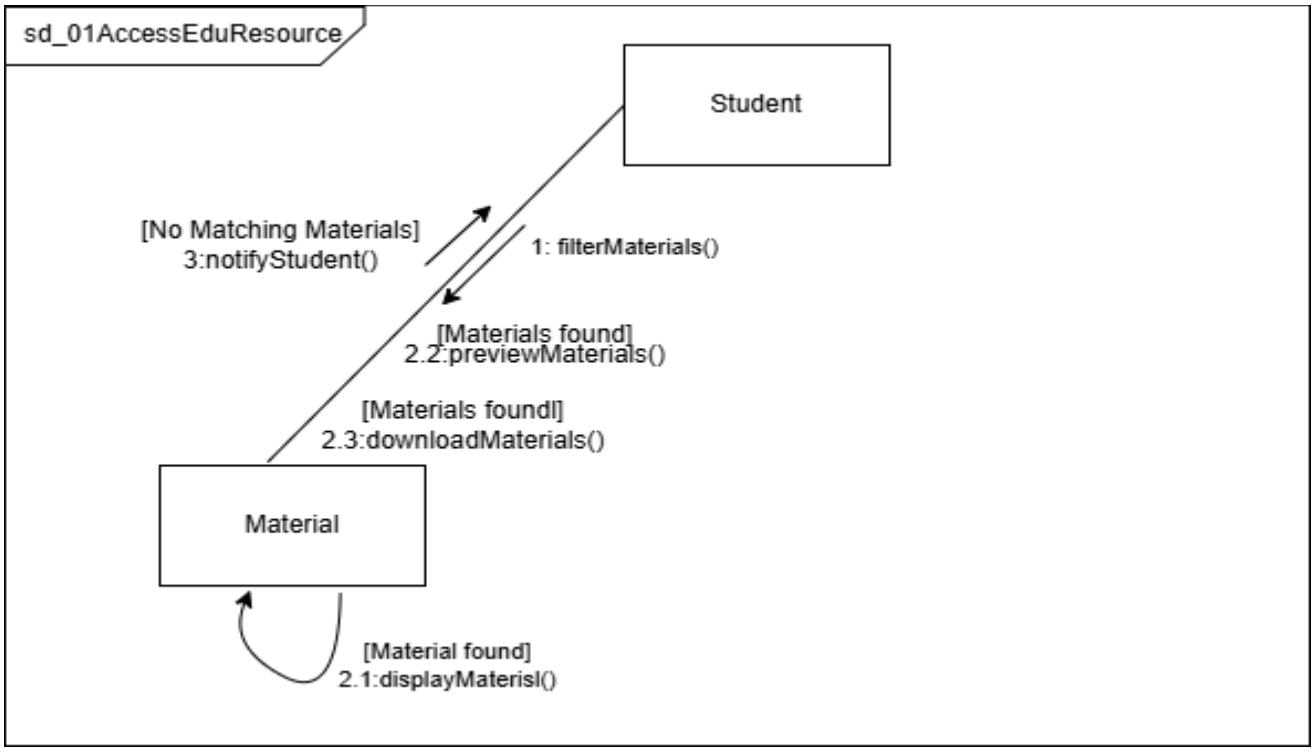


UC 32

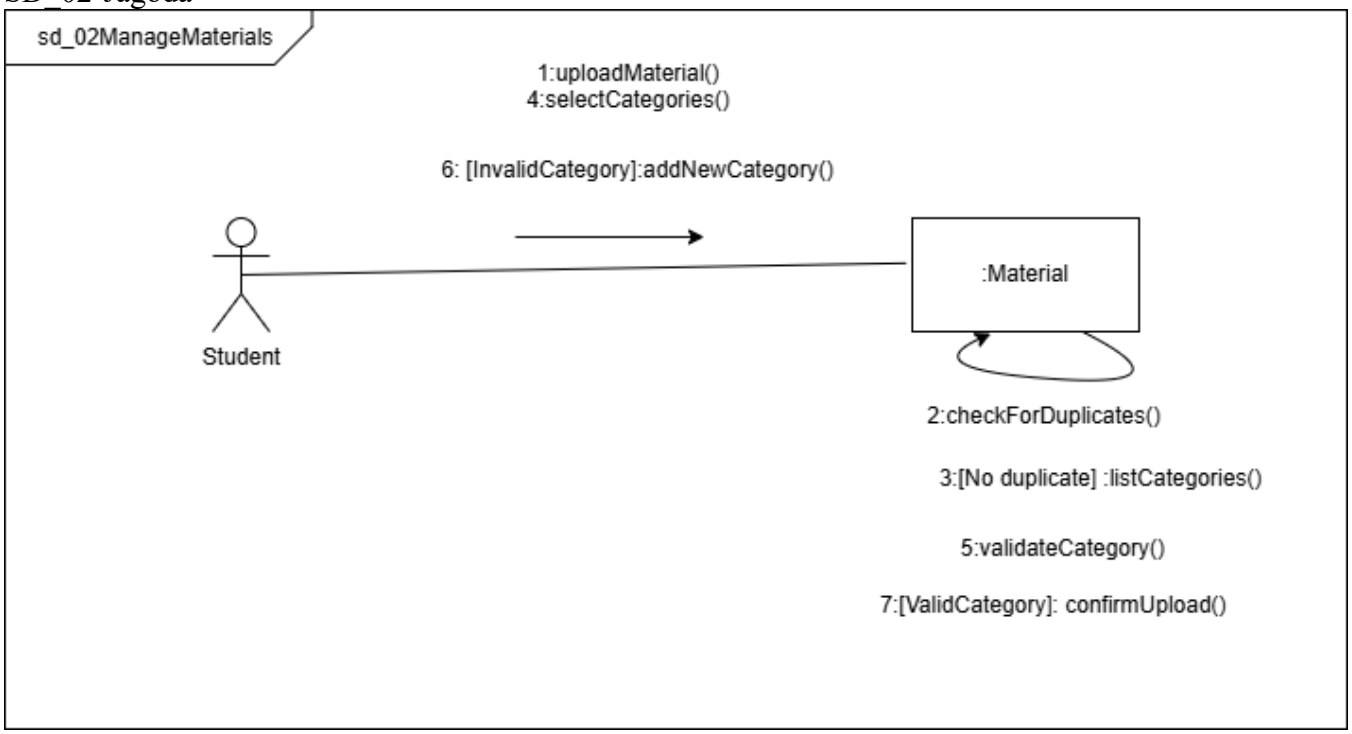

SD UC_32
Student
Challange
selectChallenge(challengeName)
registerChallengeParticipation()
completeTasks()
checkTimeframe()
alt
timeframe
[not within timeframe]
showMessage("Challenge failed:")
[within timeframe]
validateTaskCompletion()
addBadgeToDashboard()

5.7 Collaboration diagrams

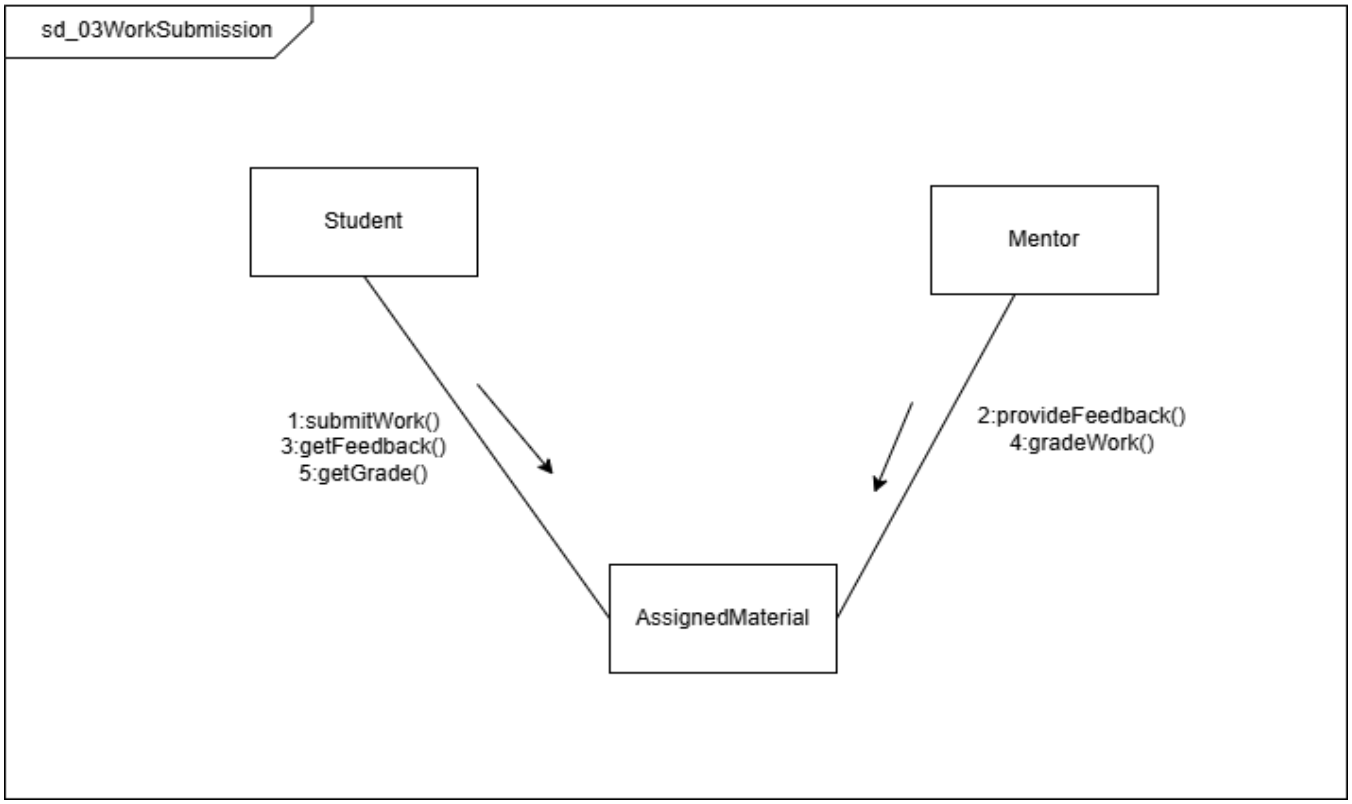
SD_01 -Jagoda



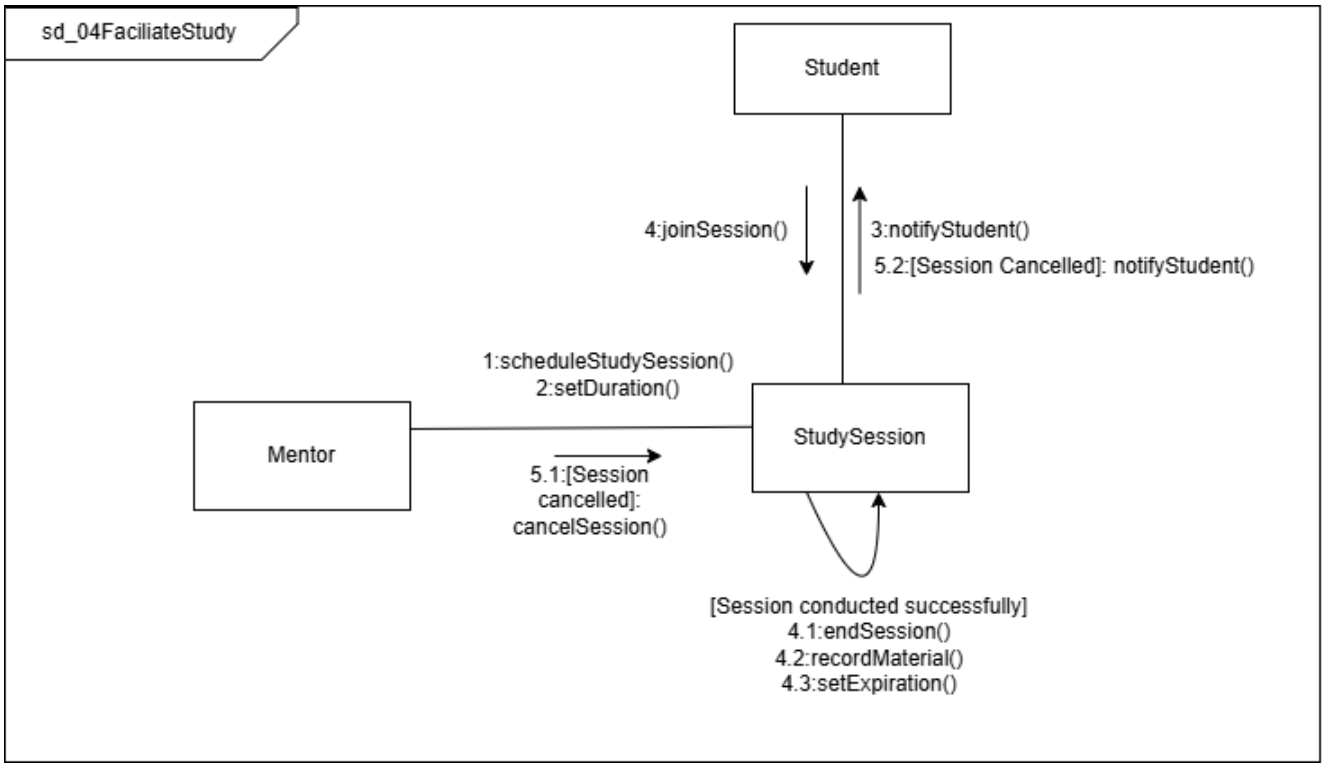
SD_02-Jagoda



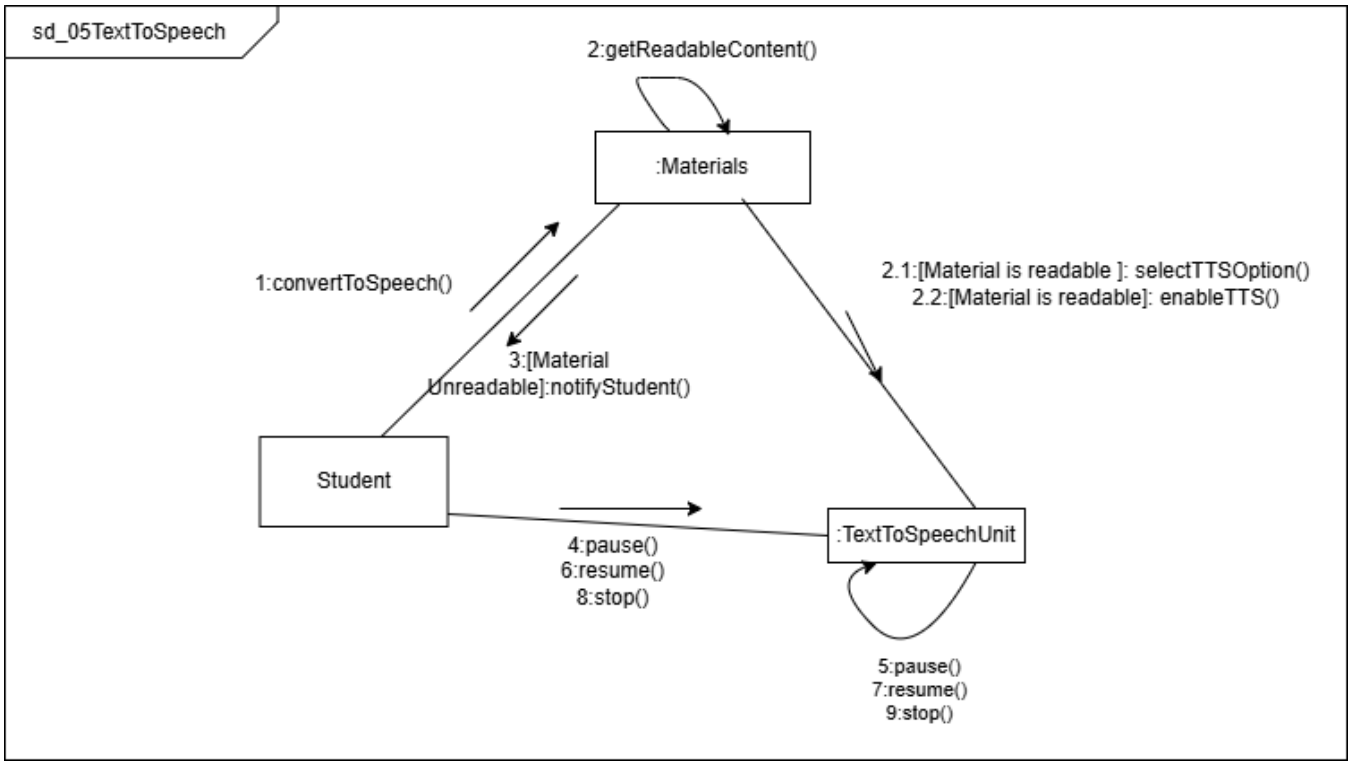
SD_03-Jagoda



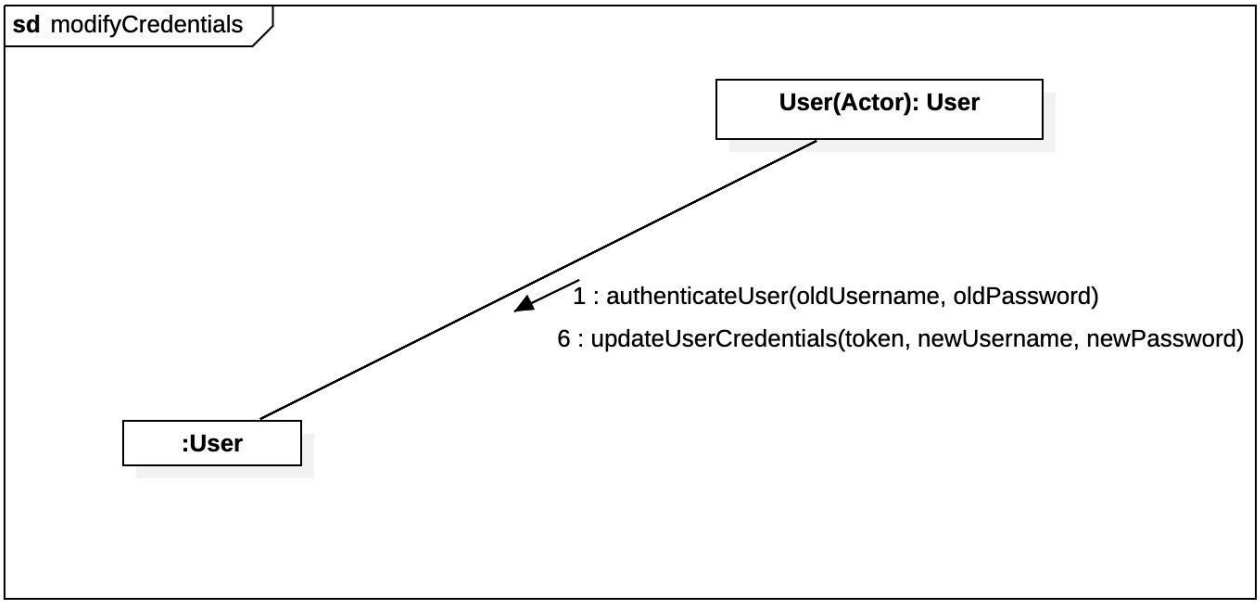
SD_04-Jagoda



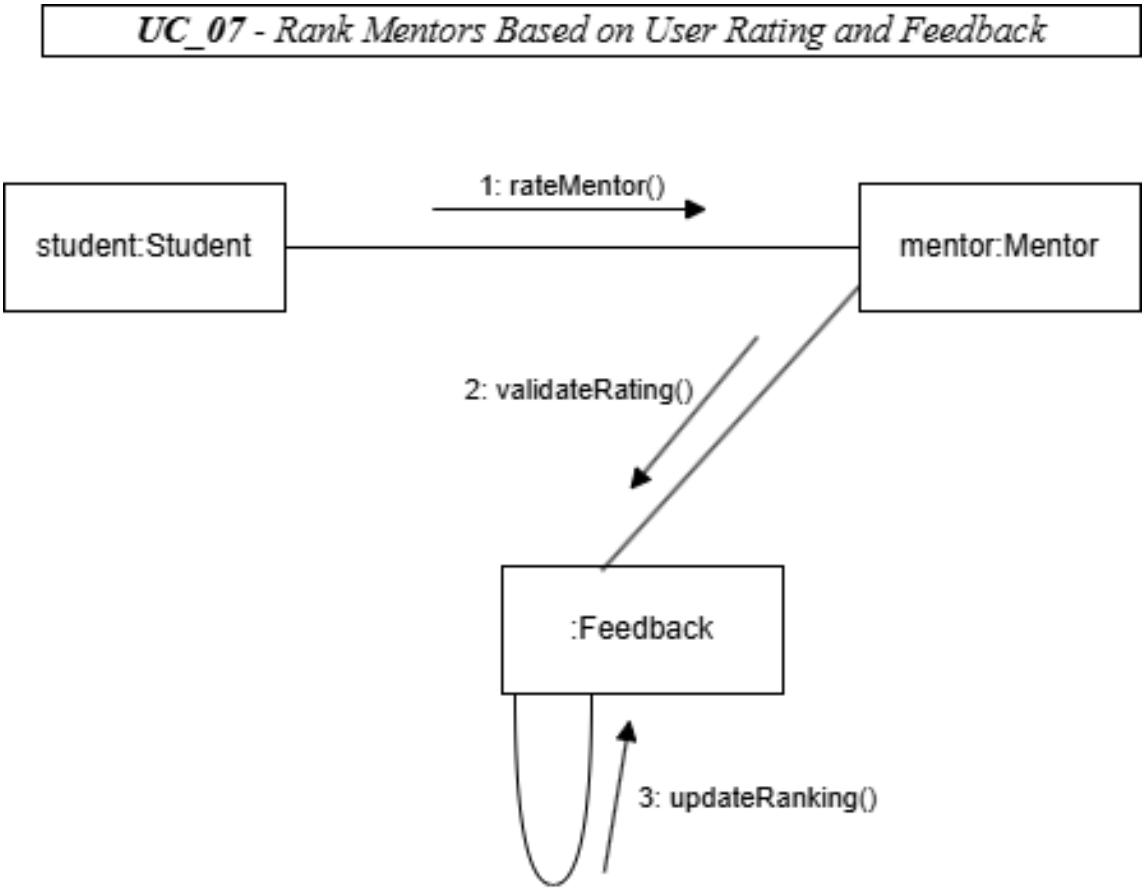
SD_05 -Jagoda



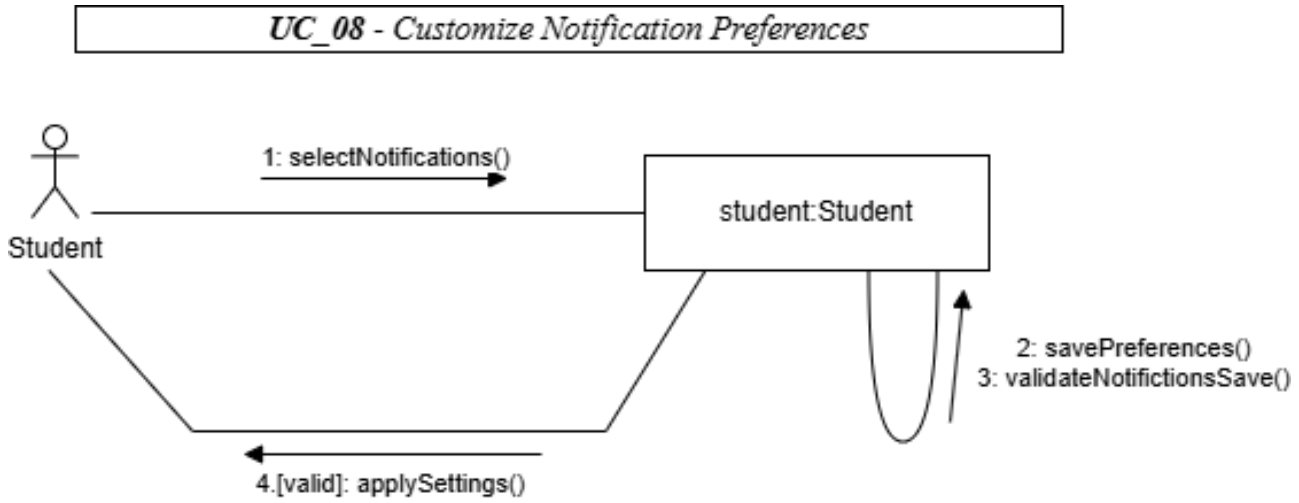
UC_6 - Halil



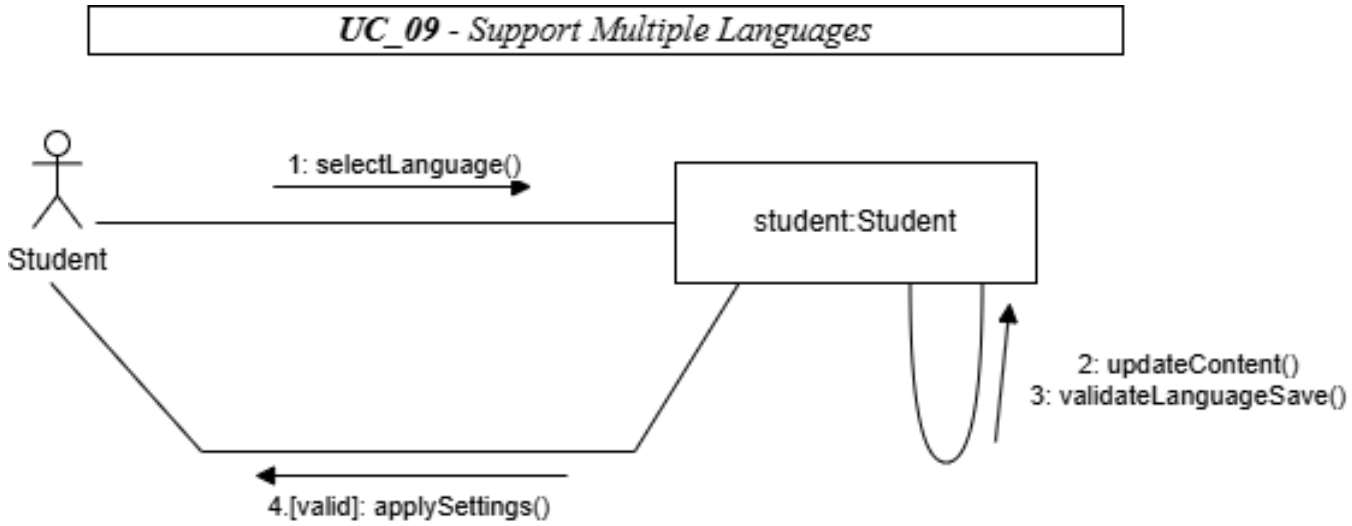
UC_7 - Dionis


UC_07 - Rank Mentors Based on User Rating and Feedback
student:Student
1: rateMentor()
mentor:Mentor
2: validateRating()
:Feedback
3: updateRanking()

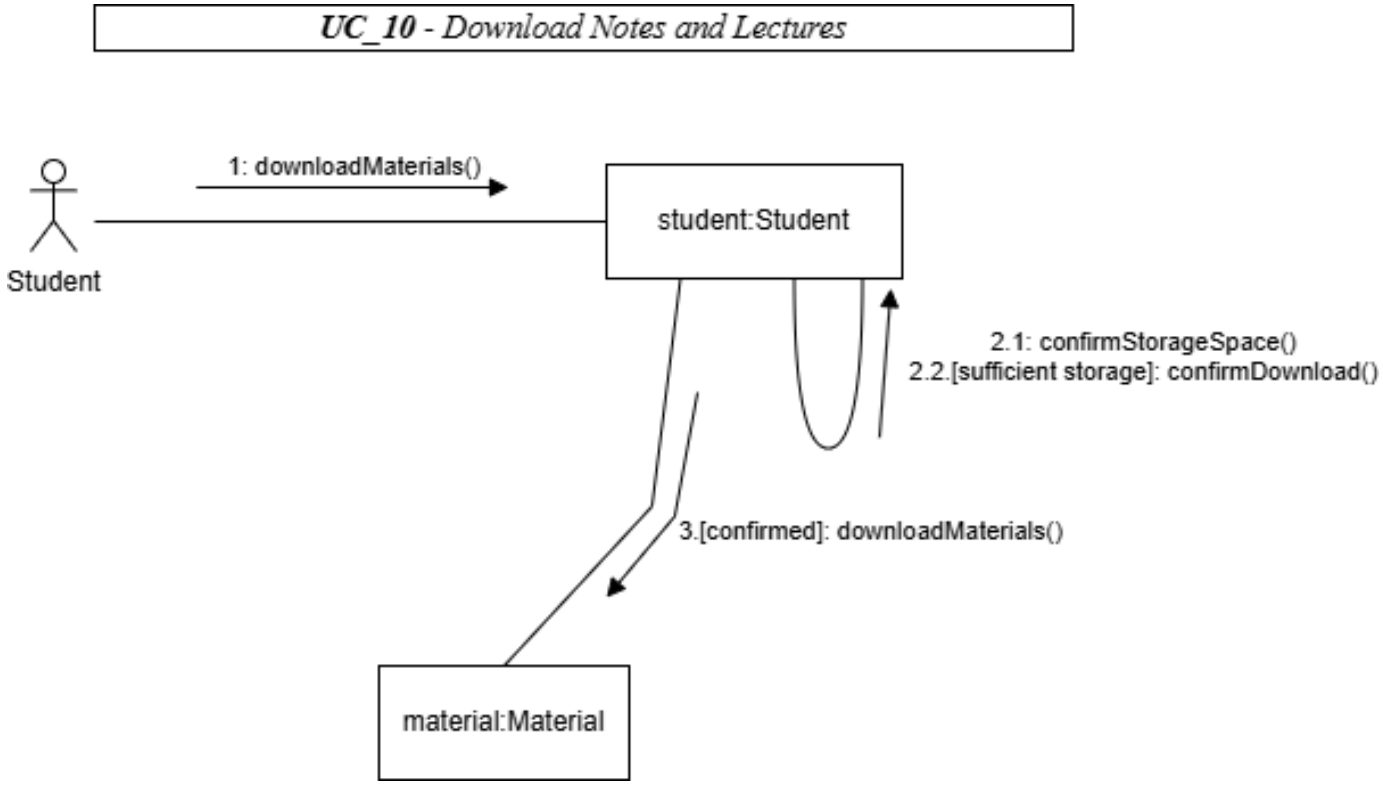
UC_8 - Dionis



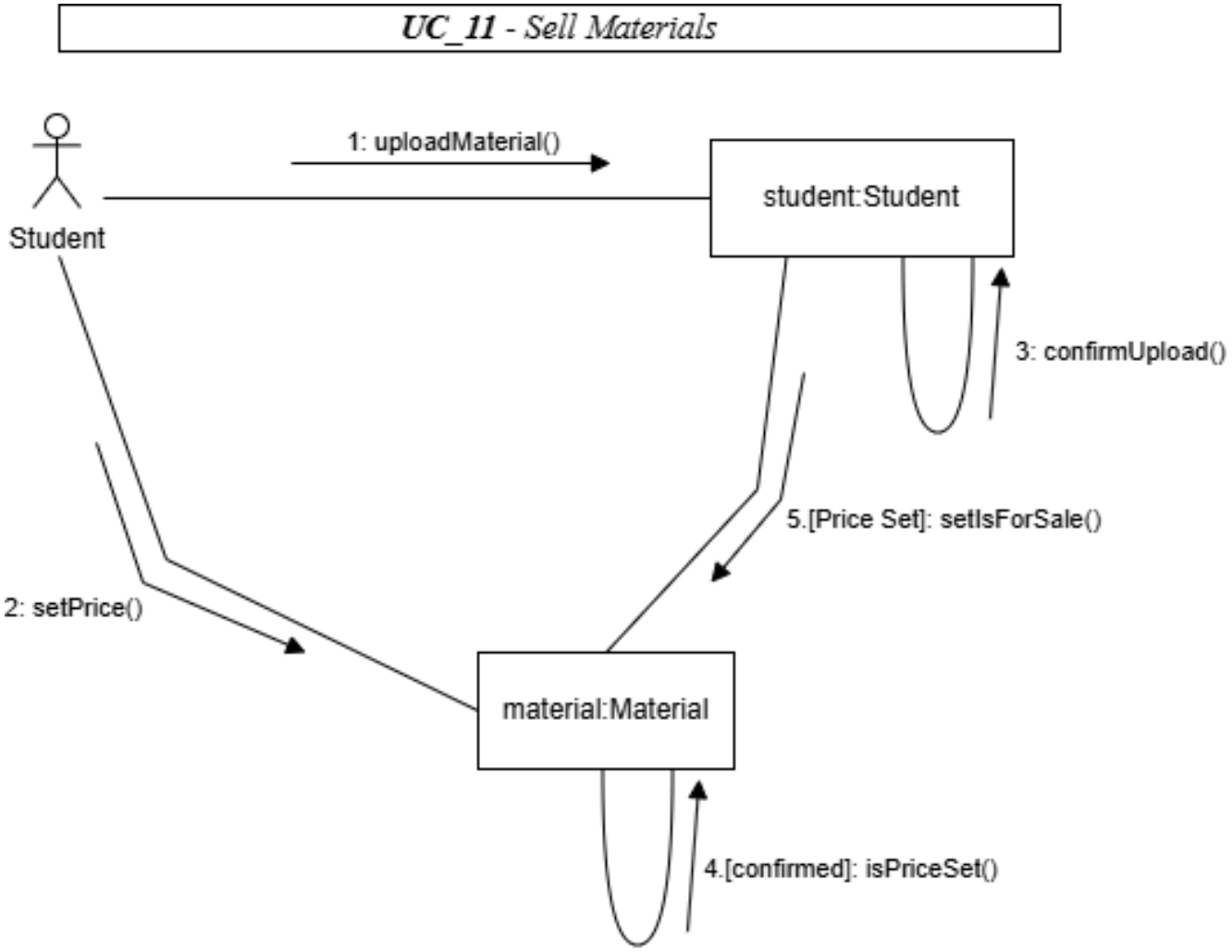
UC_9 - Dionis



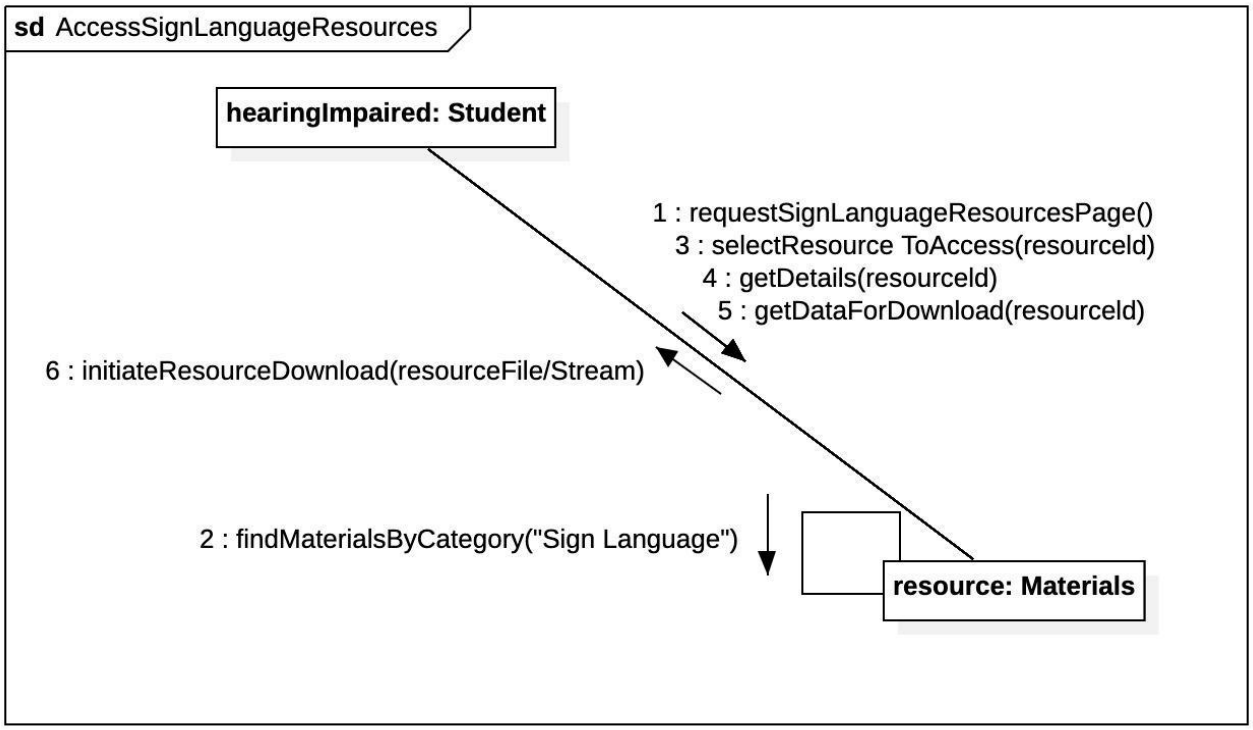
UC_10 - Dionis


UC_10 - Download Notes and Lectures
Student
1: downloadMaterials()
student:Student
2.1: confirmStorageSpace()
2.2.[sufficient storage]: confirmDownload()
3.[confirmed]: downloadMaterials()
material:Material

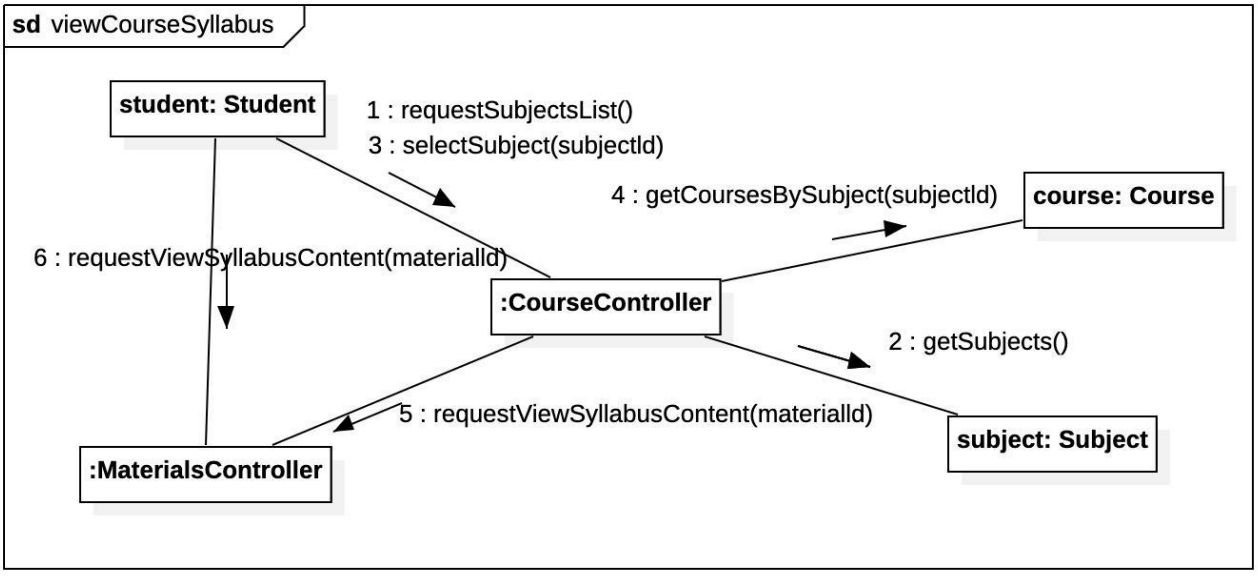
UC_11 - Dionis


UC_11 - Sell Materials
Student
1: uploadMaterial()
student:Student
3: confirmUpload()
5.[Price Set]: setIsForSale()
2: setPrice()
material:Material
4.[confirmed]: isPriceSet()

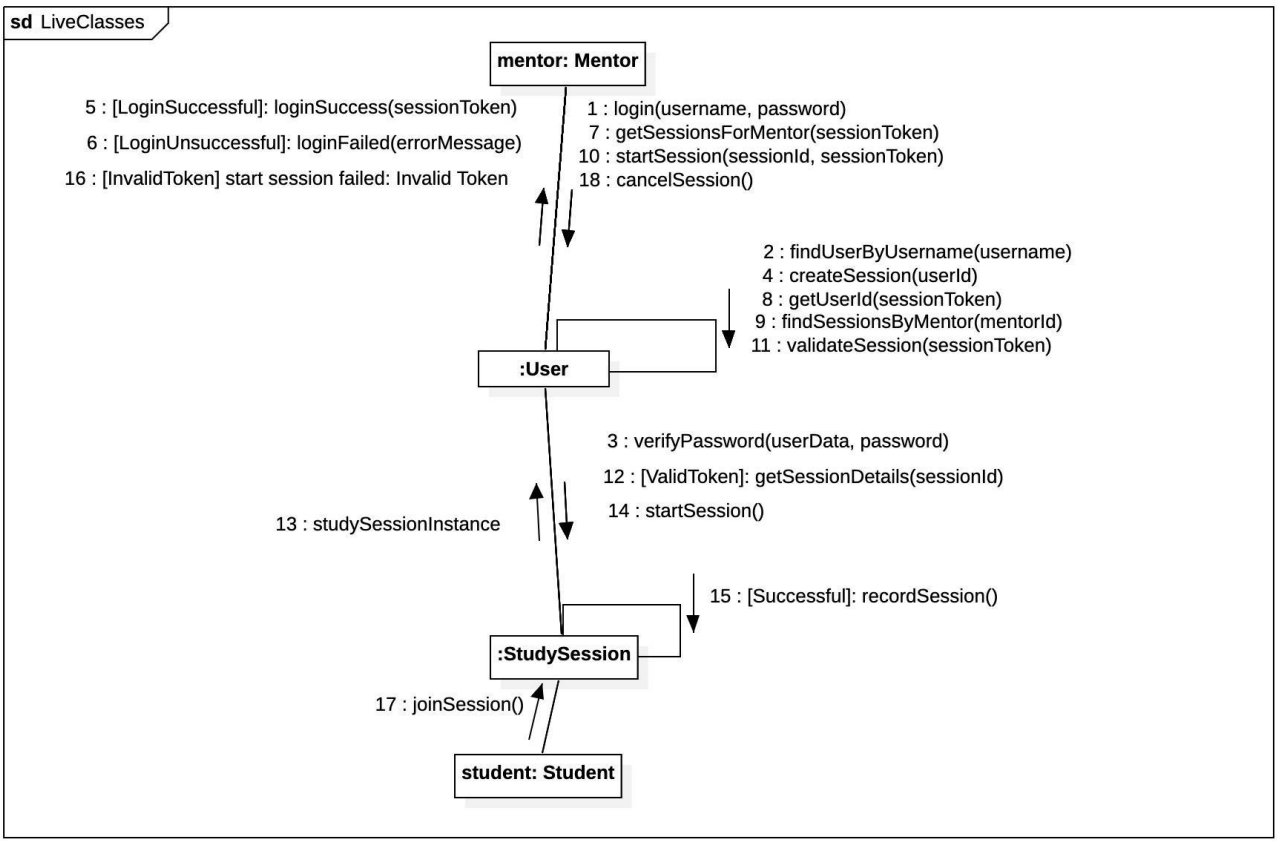
UC_12 - Marvi



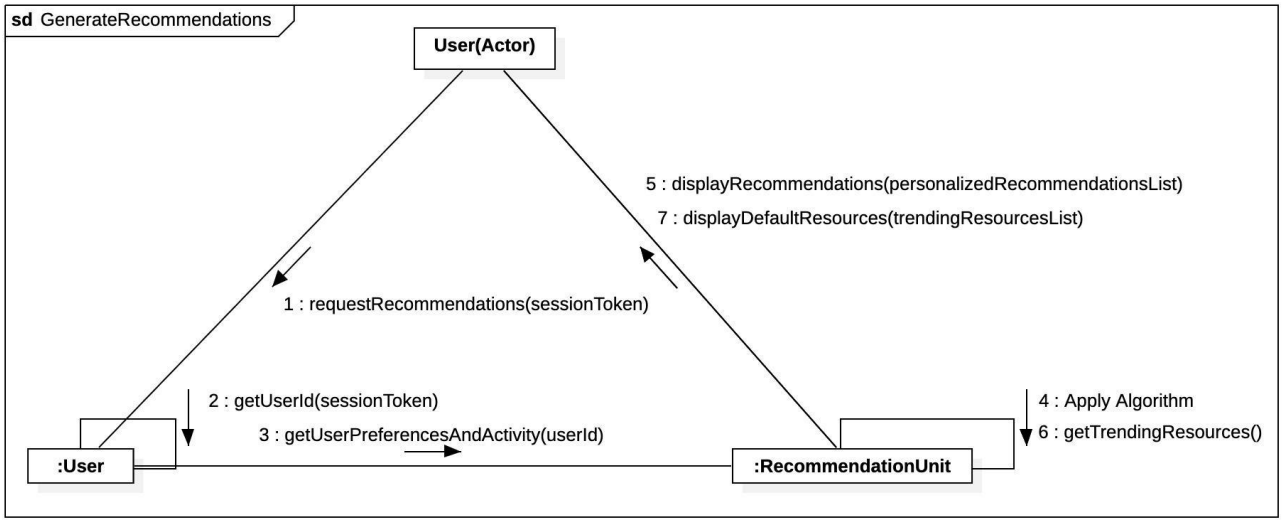
UC_13 - Marvi



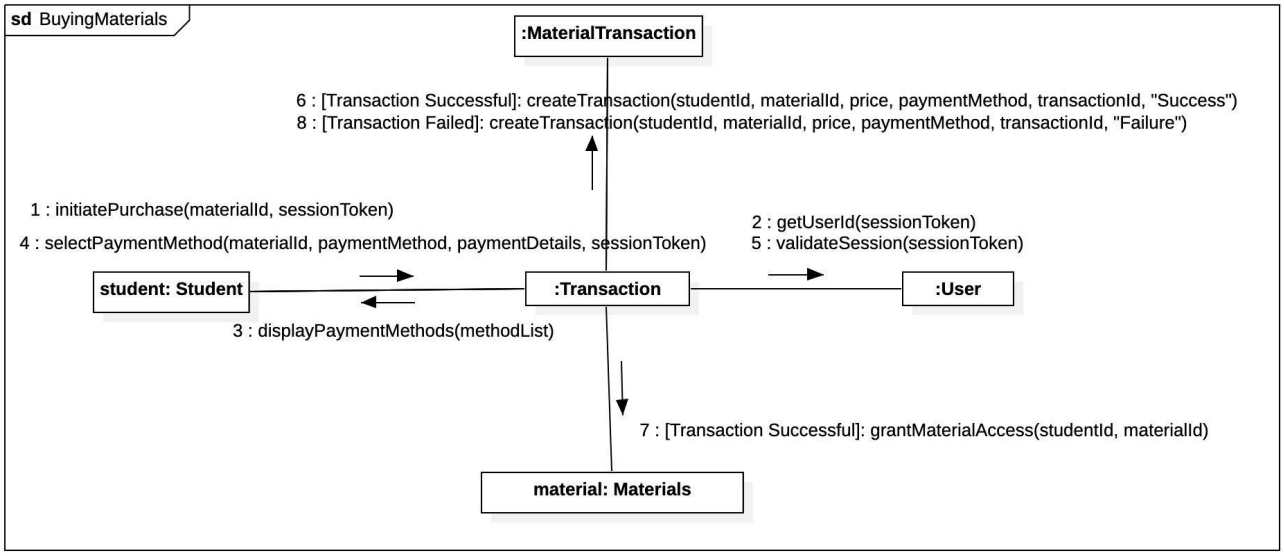
UC_14 - Halil



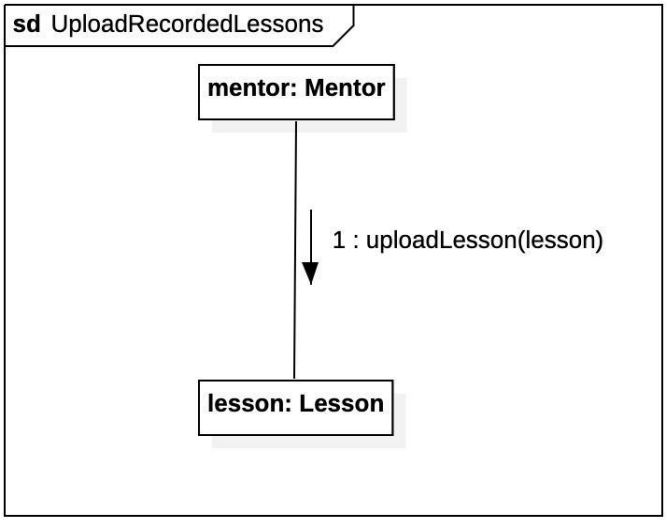
UC_15 - Halil



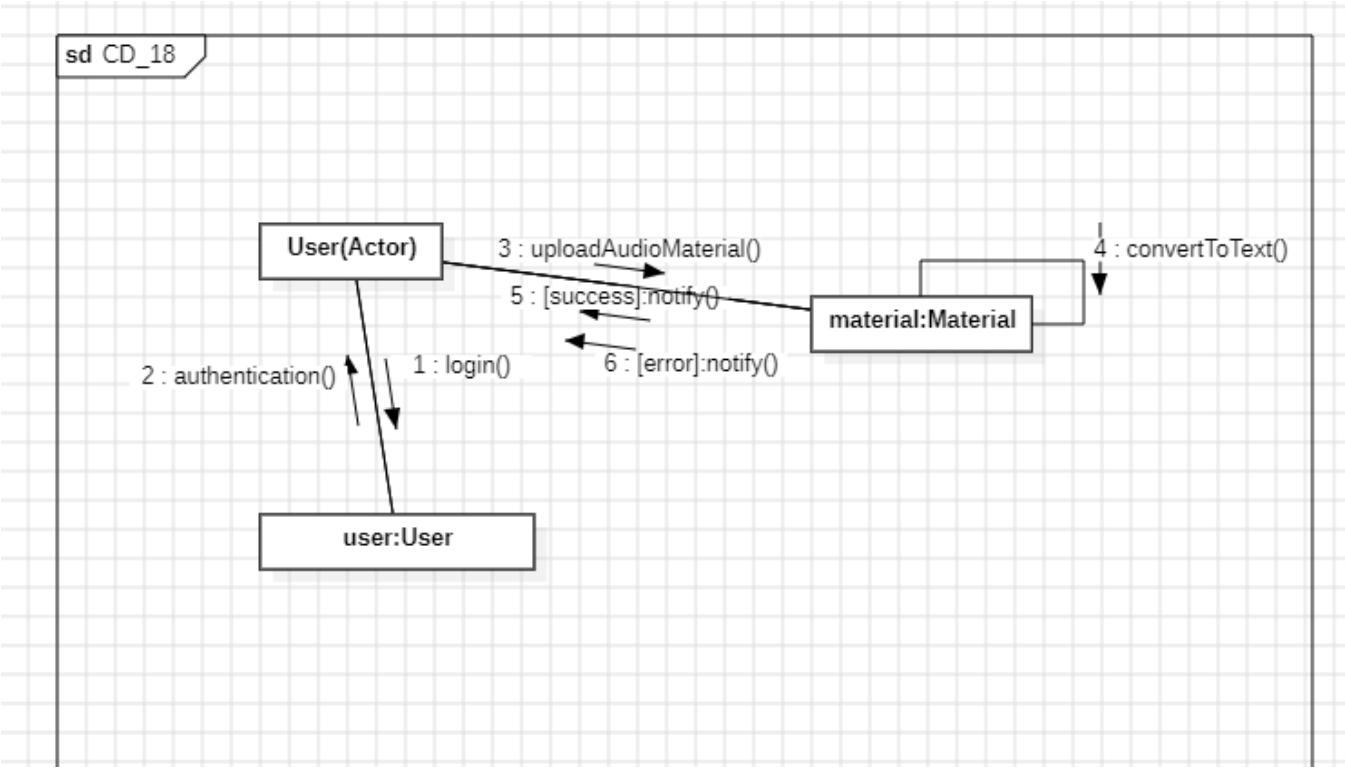
UC_16 - Halil



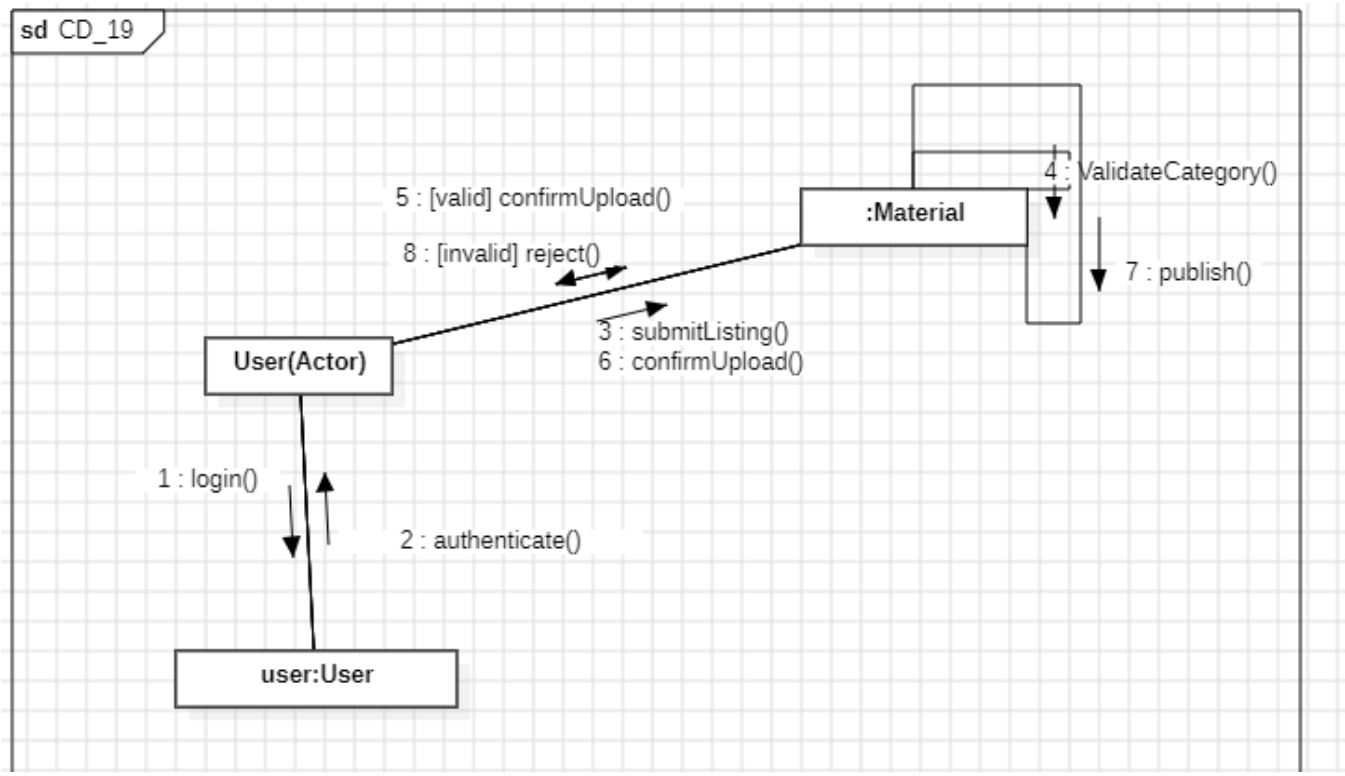
UC_17 - Halil



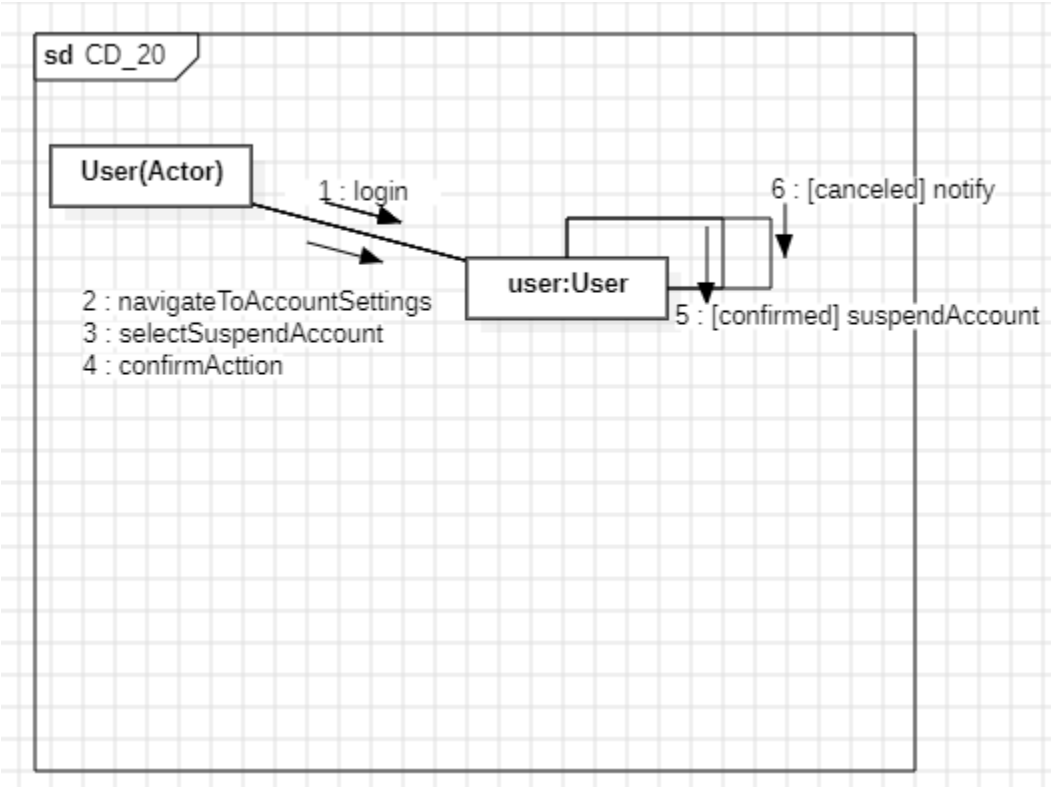
CD_18 - Dea



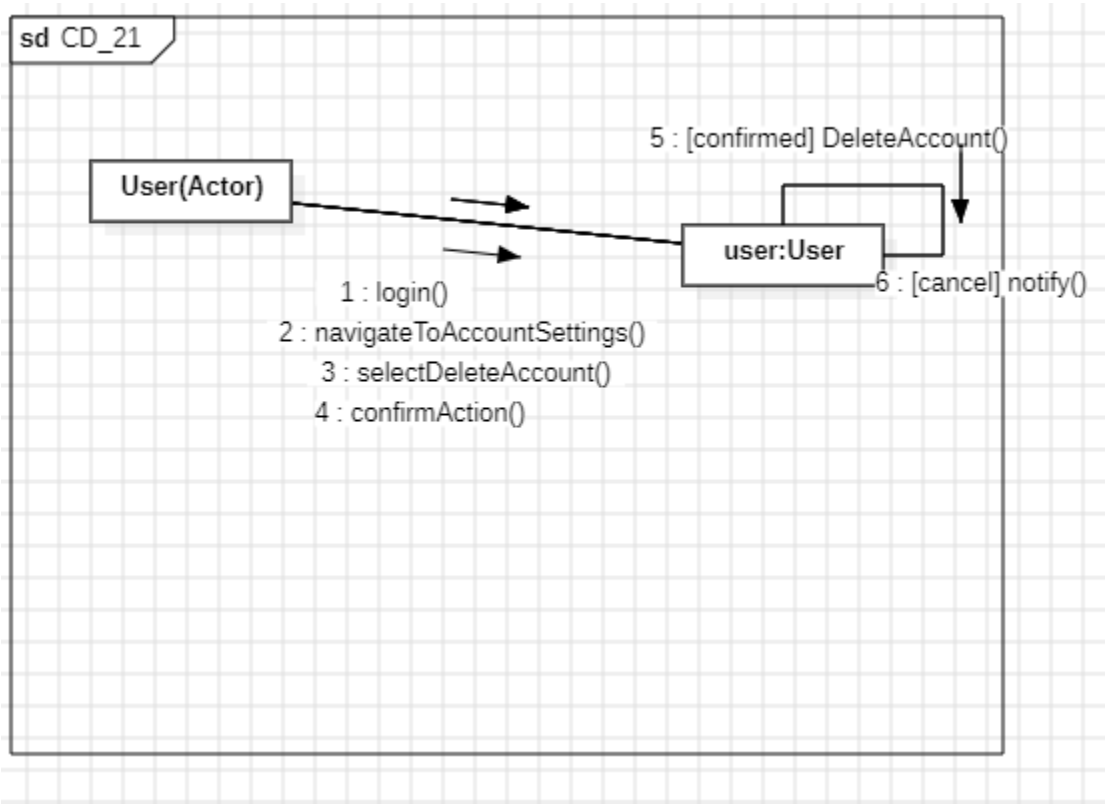
CD_19 - Dea



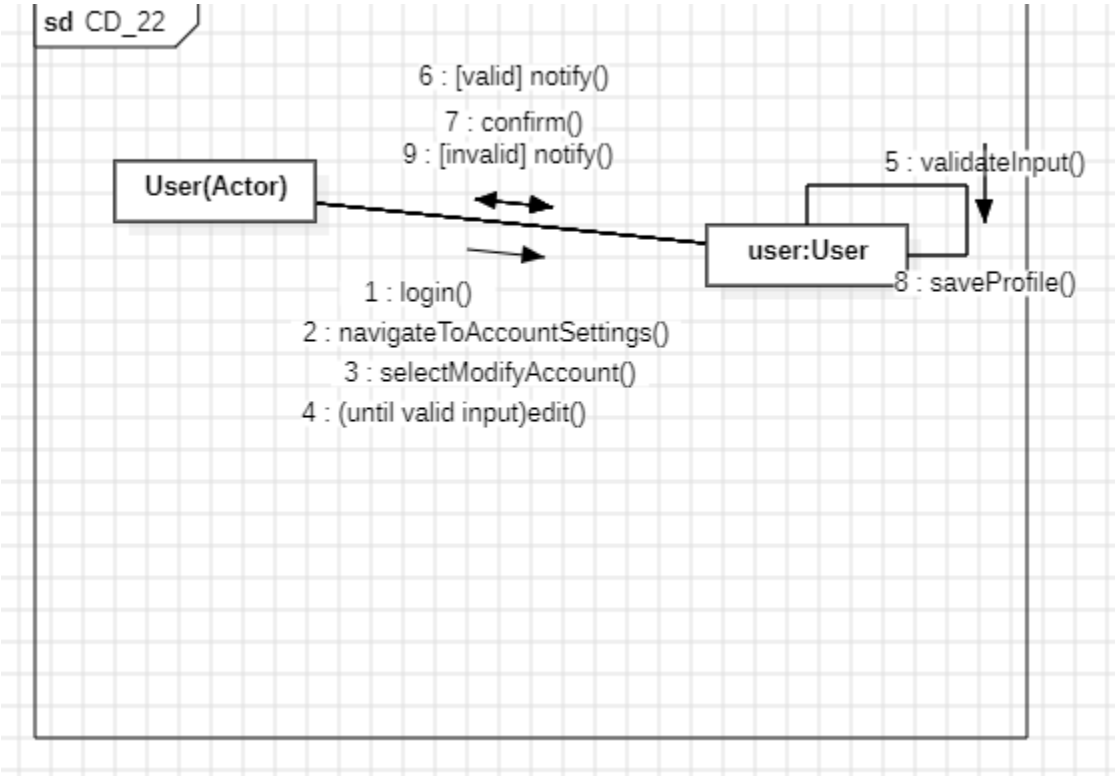
CD_20 - Dea



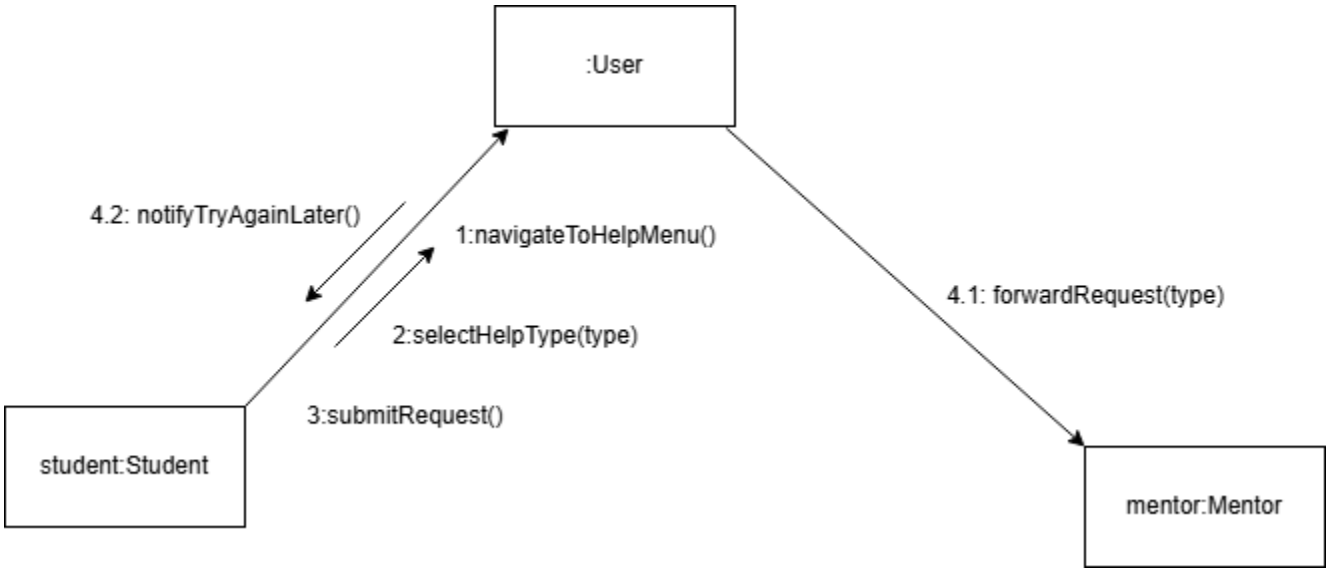
CD_21- Dea



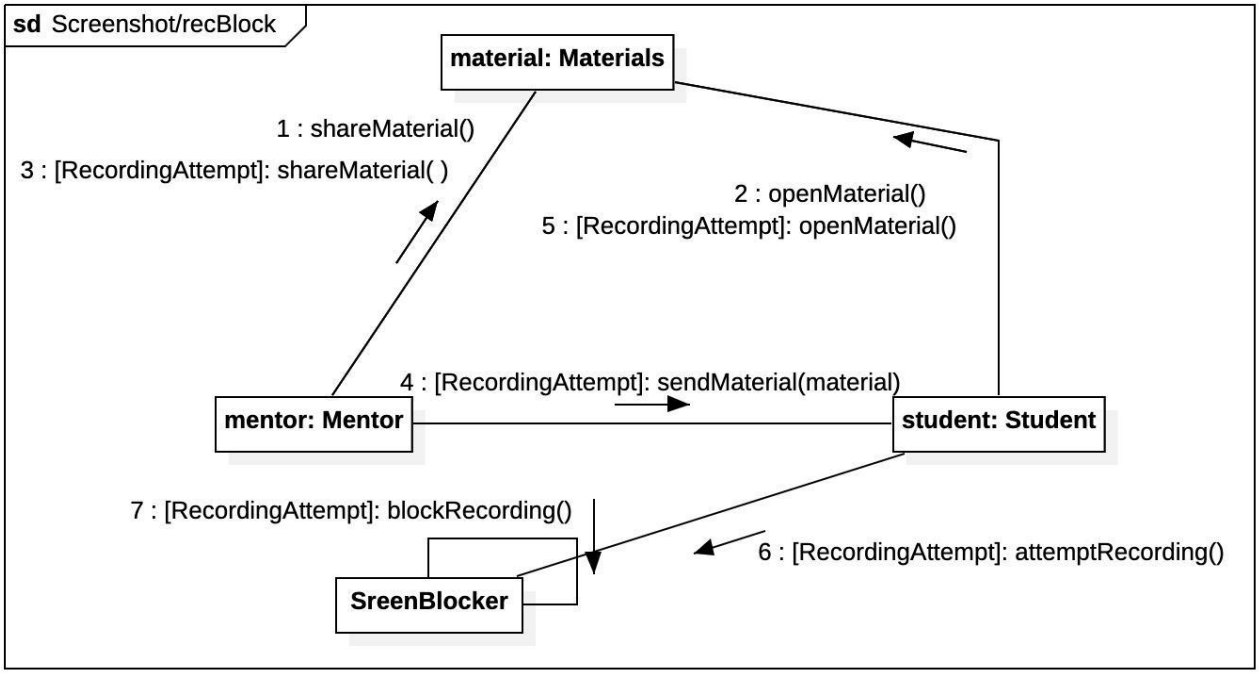
CD_22 - Dea



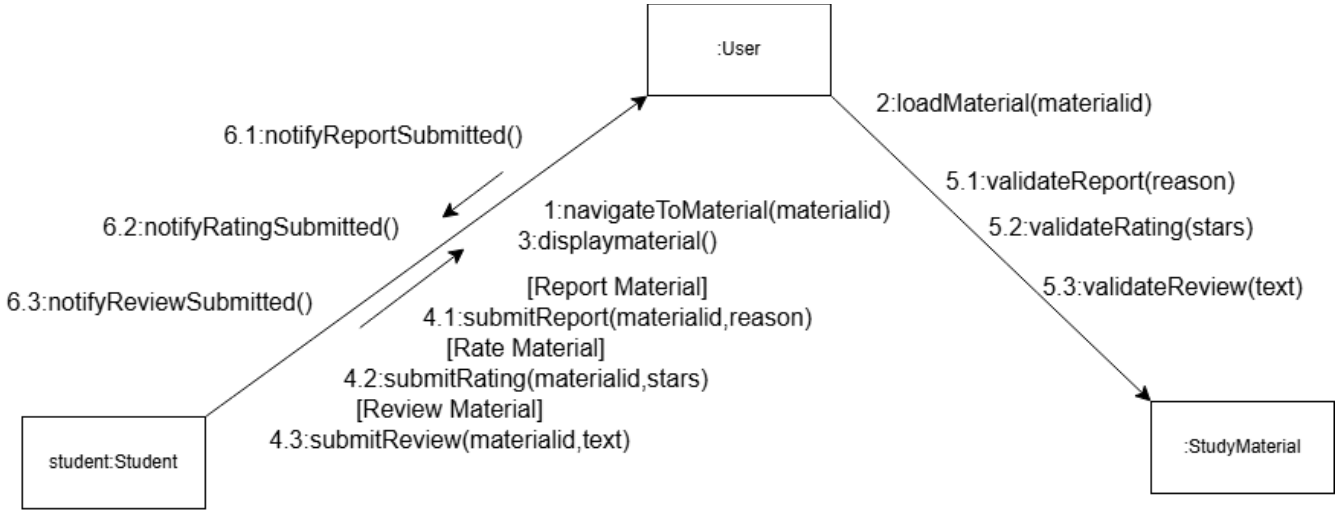
UC_23 - Ameraldo



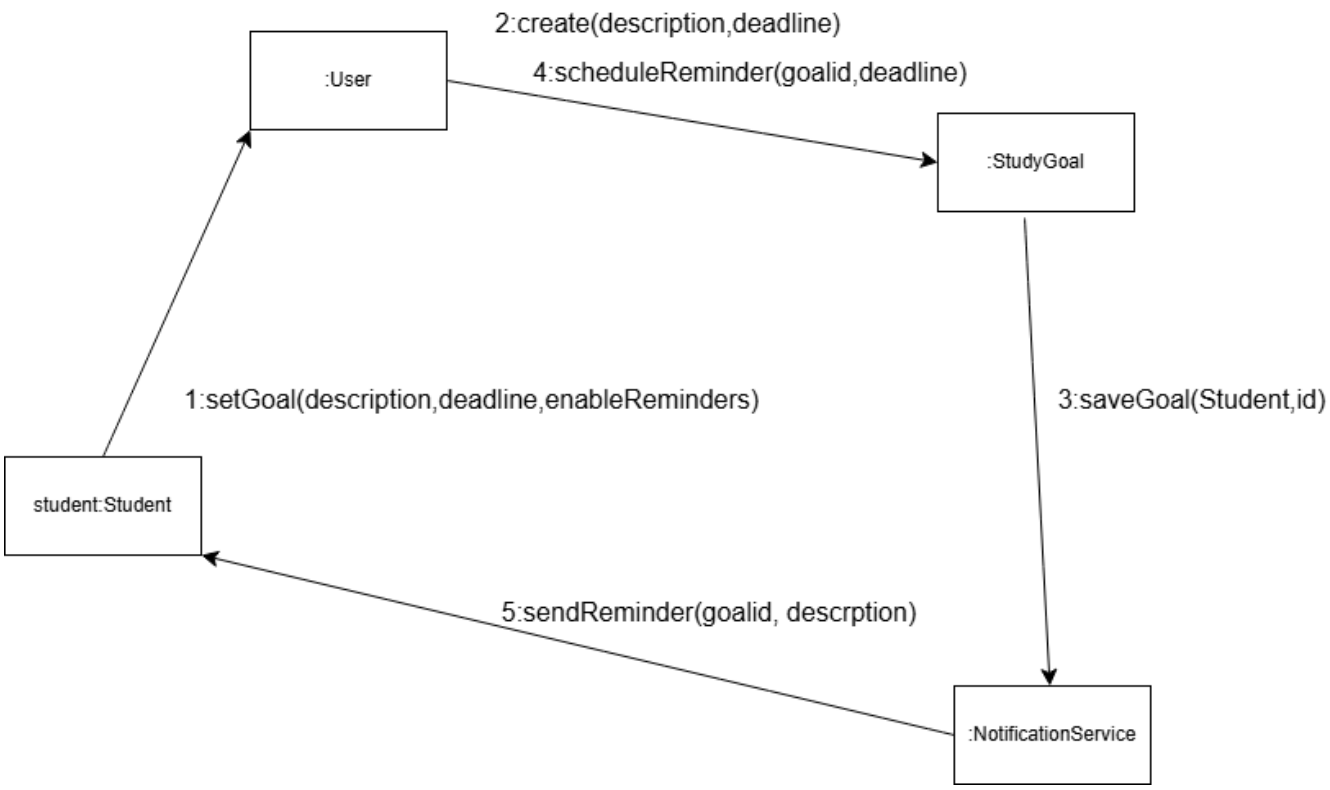
UC_24 – Ameraldo



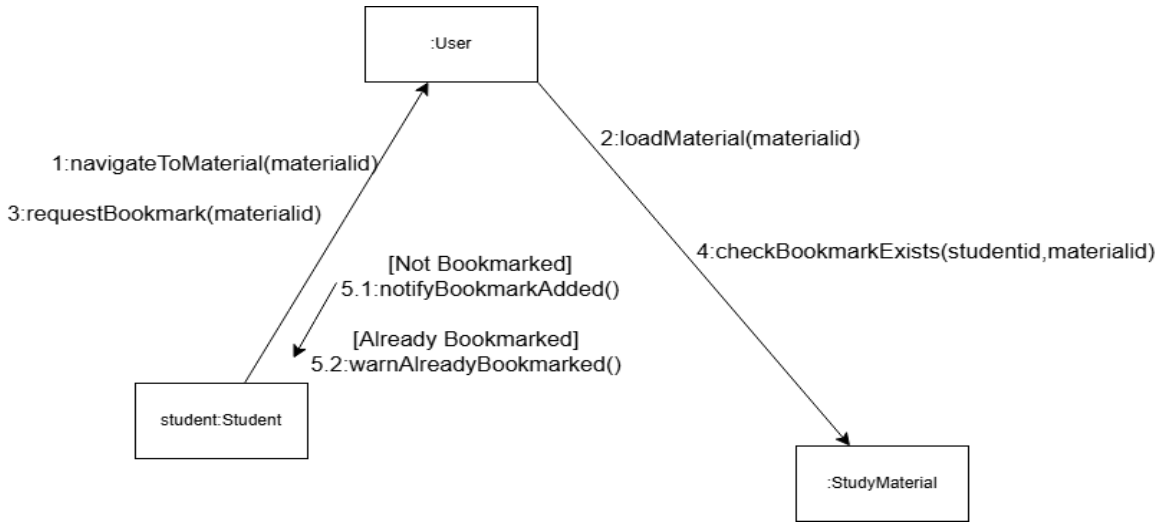
UC_25 – Ameraldo



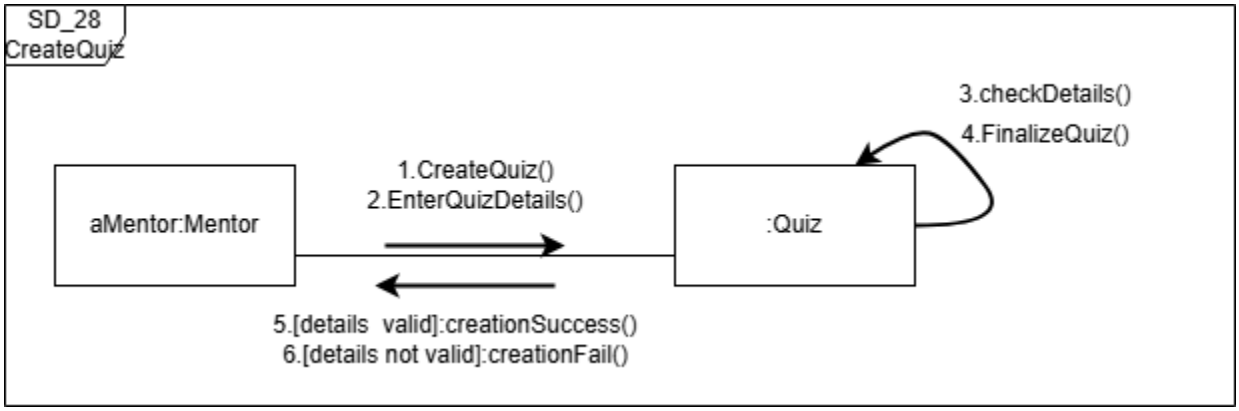
UC_26 - Ameraldo



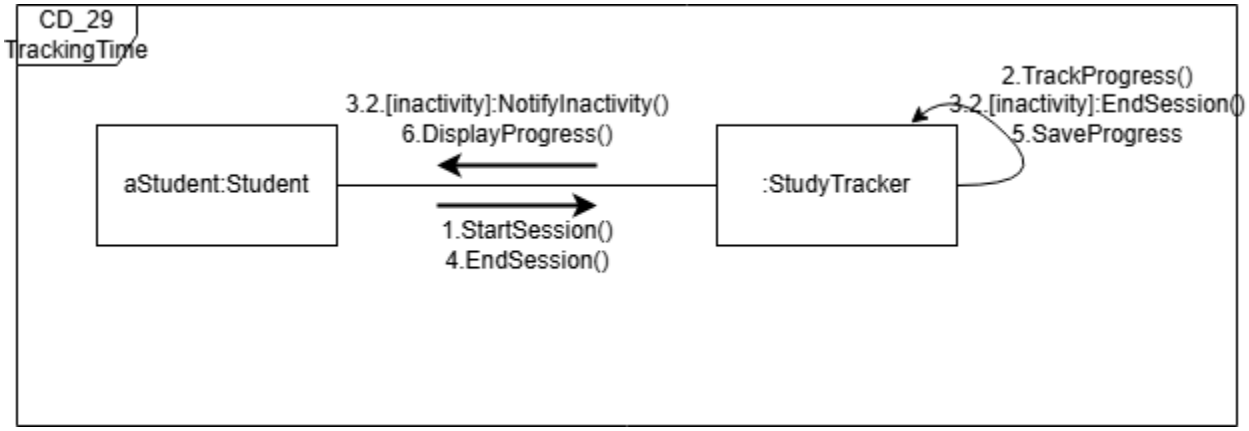
UC_27 - Ameraldo



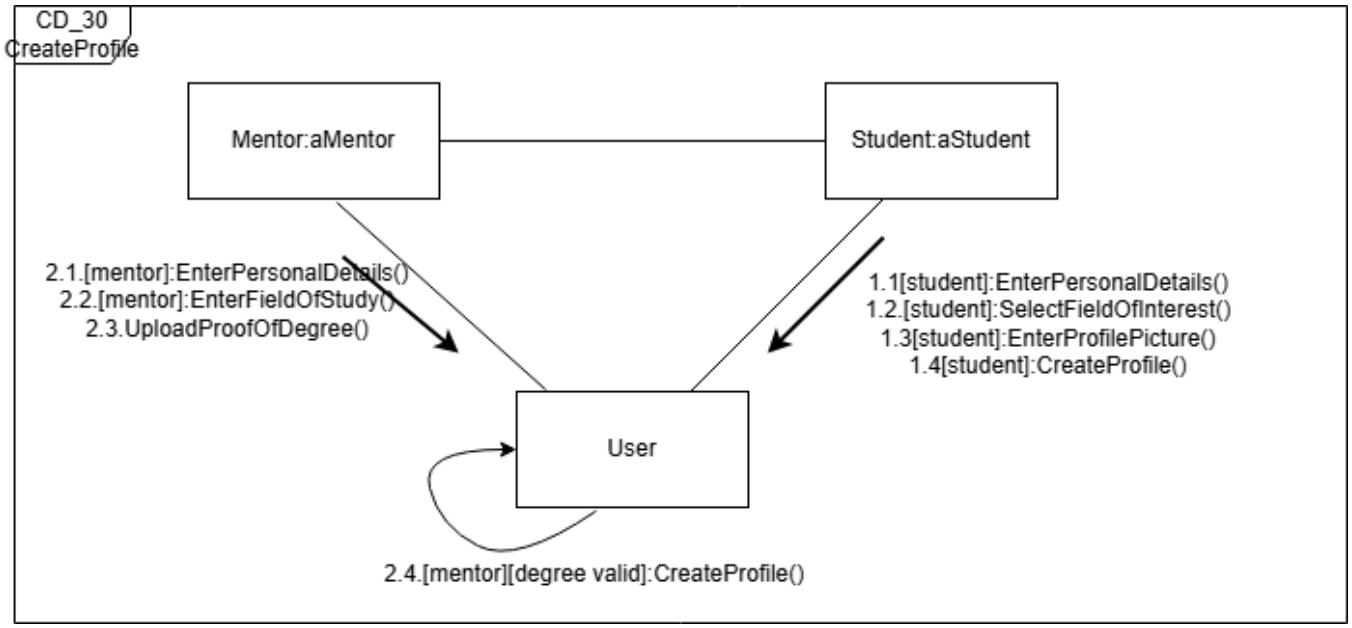
SD 28 - Daniela



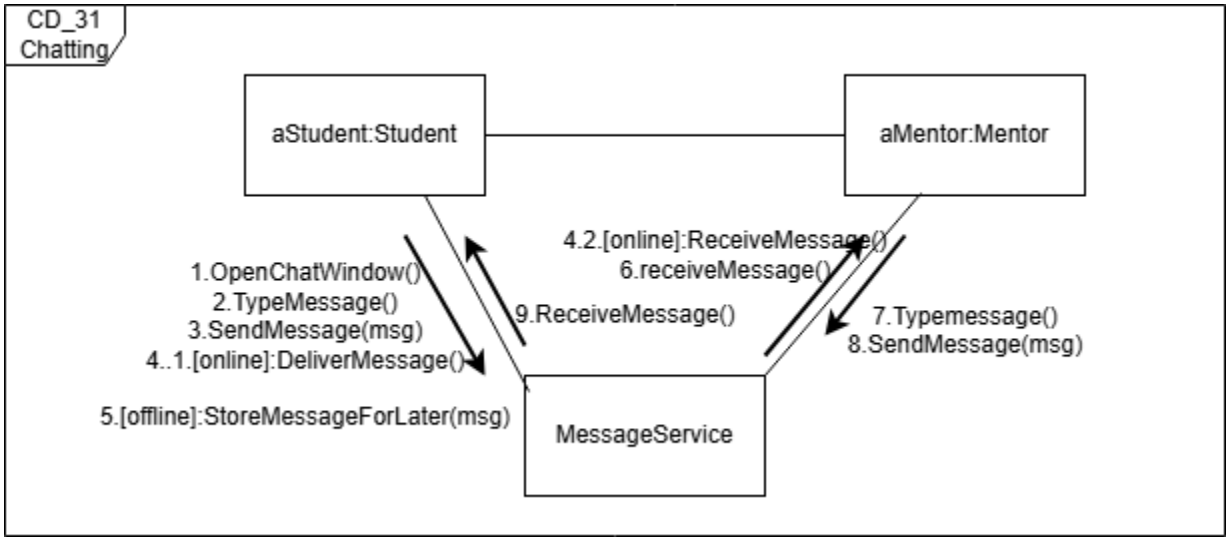
SD 29 - Daniela



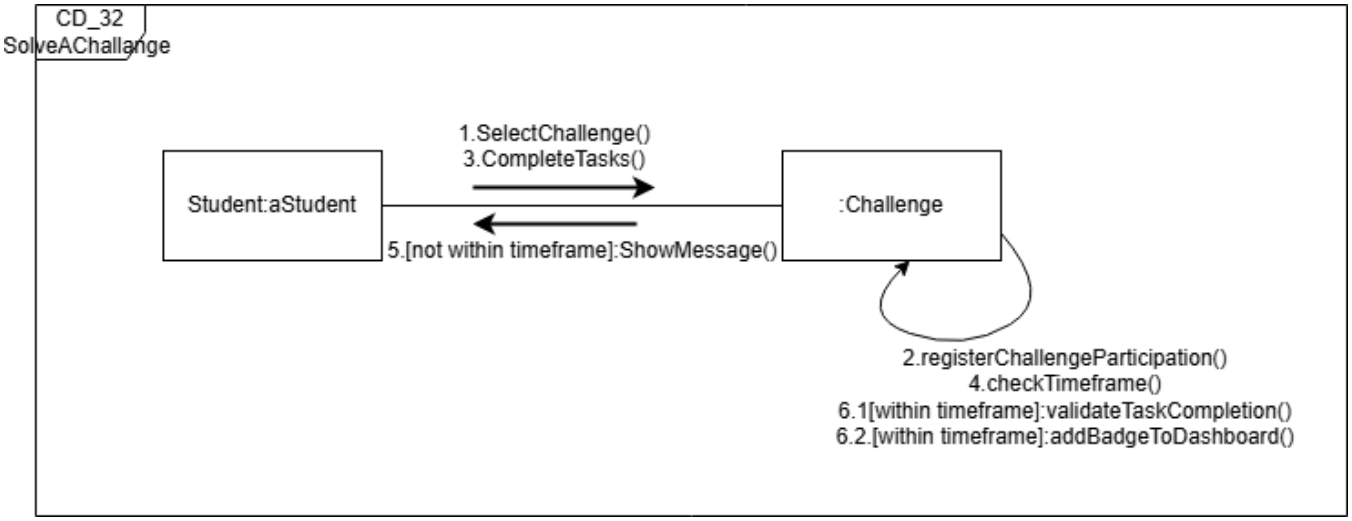
SD 30 - Daniela



SD 31 - Daniela



SD 32 – Daniela



6. Design patterns

6.1 Singleton Pattern

In the context of our Student Education Exchange System, the Singleton design pattern is applied to the RecommendationUnit class to ensure that there is only one centralized instance responsible for generating study material recommendations. Since recommendation logic should be consistent and shared across all students, creating multiple instances could lead to duplicated data, unnecessary computation, and inconsistent suggestions. By applying the Singleton pattern, we guarantee that all users access the same recommendation engine, maintaining global consistency, reducing memory overhead, and simplifying access. The class contains a private static instance variable and exposes a getInstance() method that initializes the object only once and returns the same instance every time.

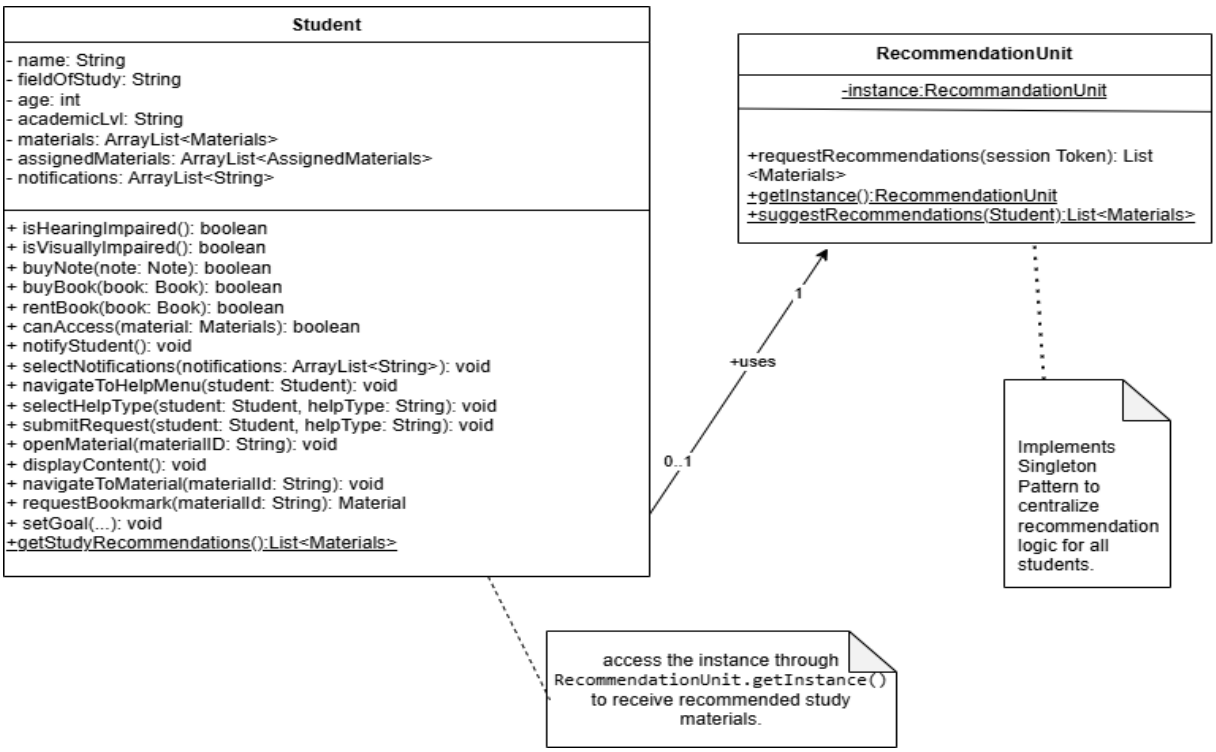
In RecommendationUnit class:

1. getInstance(): RecommendationUnit

2. suggestRecommendation(student: Student): List<Material>

In Student class:

1. getStudyRecommendations(): List<Material>



6.2 Observer Pattern

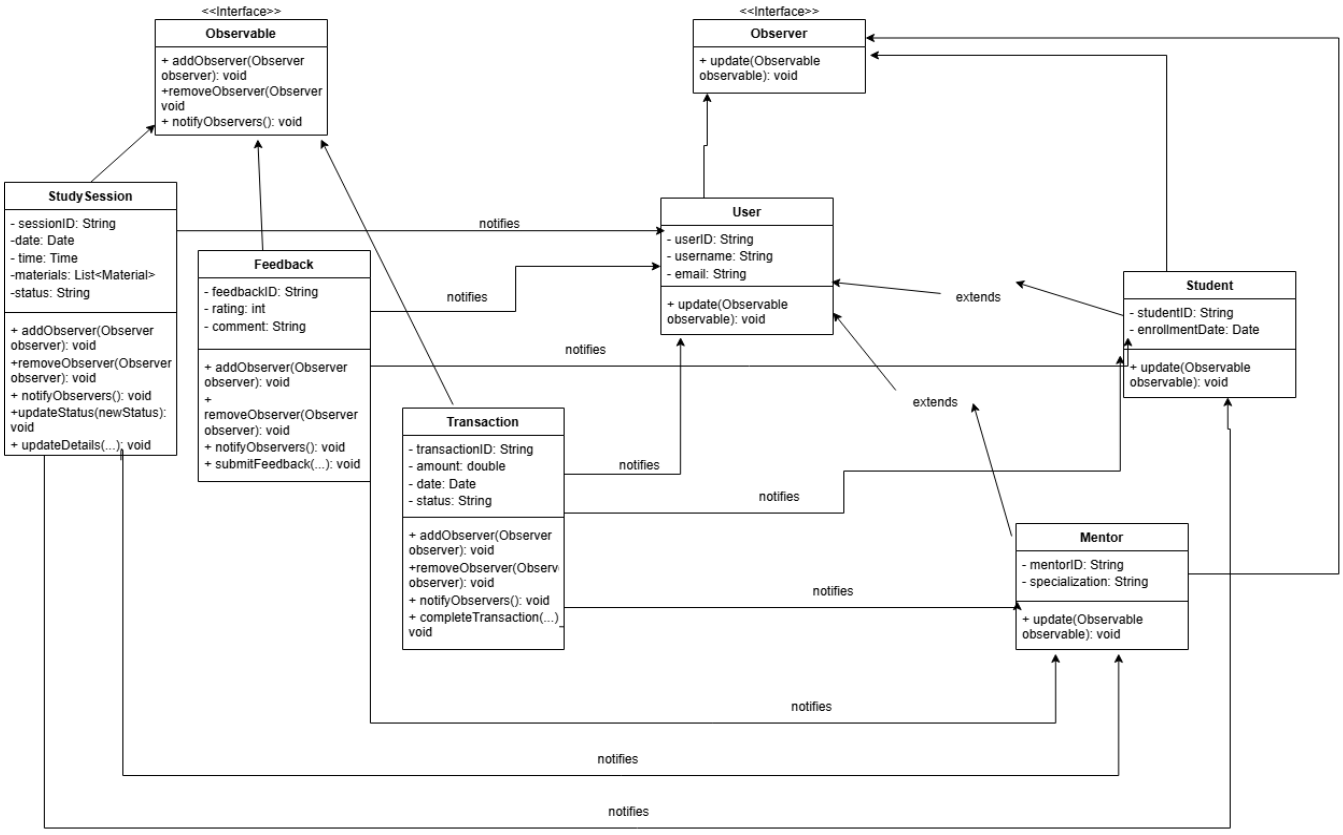
The Observer pattern's presence is justified by its ability to decouple the objects that change state (Subjects like StudySession, Feedback, Transaction) from the objects that need to react to those changes (Observers like User, Student, Mentor). This ensures that observers are automatically notified and updated when a subject's state changes, promoting a flexible and extensible architecture for handling notifications and data synchronization,

Registration: When a User (or Student/Mentor) needs to be notified about changes in a StudySession,Transaction or Feedback, they call the addObserver() method on the respective StudySession, Transaction or Feedback object, passing themselves as the observer.

State Change: When the state of a StudySession (e.g., status changes) or Feedback (e.g., new feedback submitted) changes, the ConcreteSubject (e.g., StudySession or Feedback) calls its notifyObservers() method.

Notification: The notifyObservers() method iterates through its list of registered Observers and calls their update() method.

Reaction: Each Observer's update() method then retrieves the necessary information from the Observable (e.g., the new StudySession status or Feedback details) and reacts accordingly (e.g., updating a UI, sending a notification, logging an event).

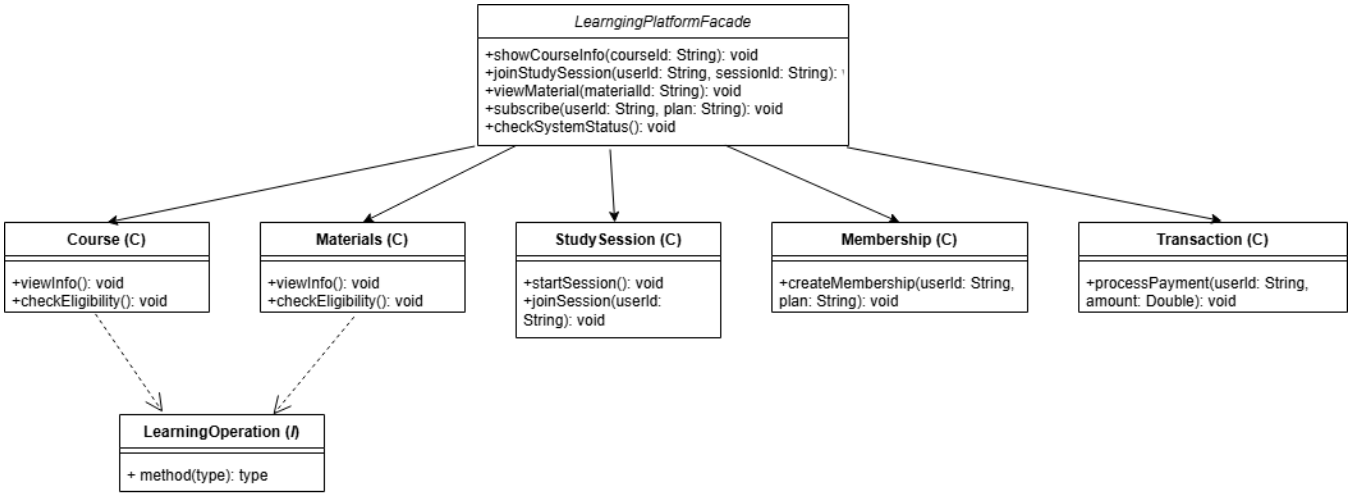


6.3 Facade Pattern

The Facade Design Pattern is used in the learning platform system to simplify access to its many subsystems. The **LearningPlatformFacade** provides a single, unified interface for interacting with components such as courses, study sessions, materials, memberships, and transactions. This prevents the client from needing to understand or manage the complexity of individual classes.

By introducing the facade, the system achieves loose coupling, meaning changes to internal components won't affect how external clients interact with the system. It also improves maintainability, as the facade centralizes key workflows, making it easier to implement logging, validation, or security checks. Additionally, the facade promotes reusability by coordinating common operations—like accessing materials or checking course eligibility—across multiple subsystems.

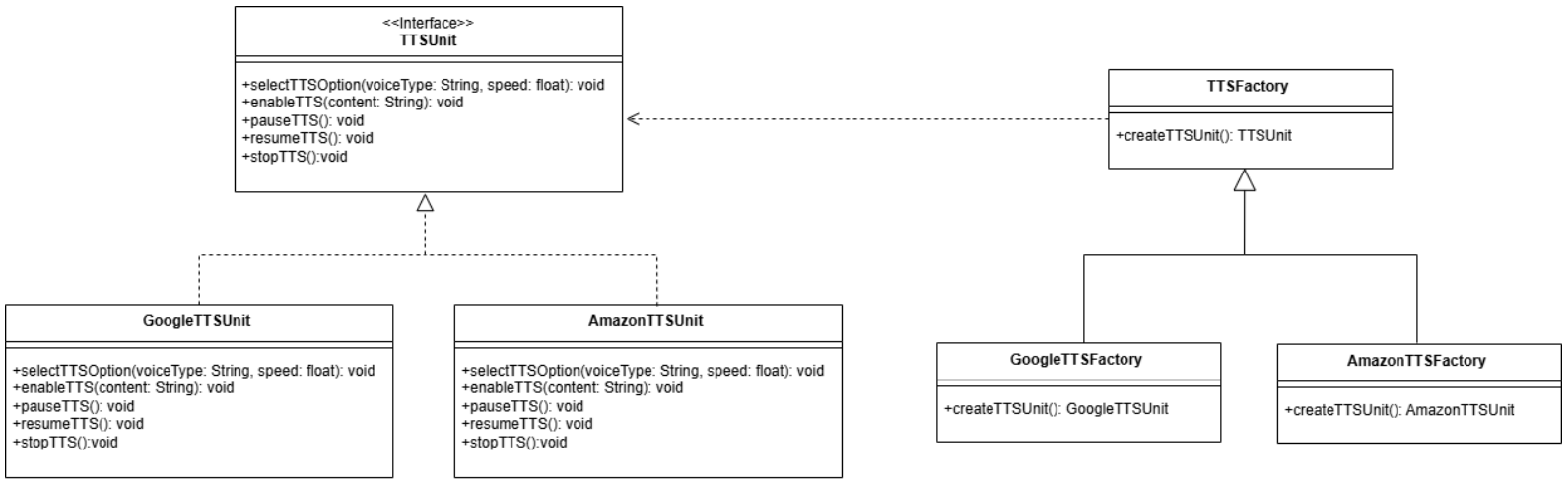
Overall, this pattern improves simplicity, supports future scalability, and ensures a consistent user experience across the learning platform.



6.4 Factory Method Pattern

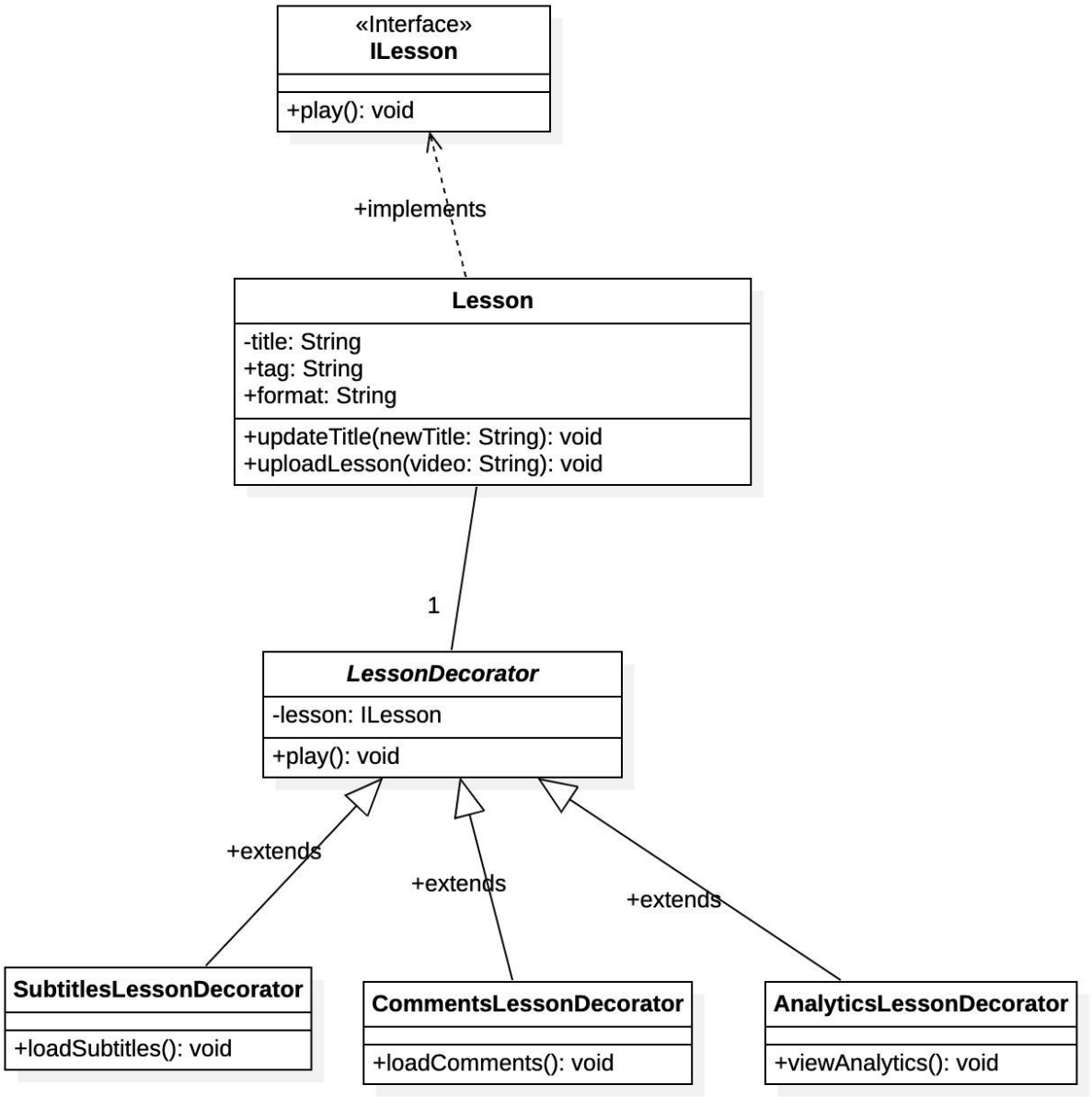
Implementing the Factory Method pattern for the TTSUnit class helps us manage the creation of different text-to-speech (TTS) engines in a clean and flexible way. Since the application might use various TTS services like Google TTS, Amazon TTS, or even a custom offline engine, we don't want to hardcode these dependencies directly into the client classes. Instead, we define a common interface (TTSUnit) and use factories to decide which specific TTS implementation to create. This makes it easy to switch between providers or add new ones without modifying the existing logic in the rest of the application.

The factory method improves maintainability and supports the Open/Closed Principle, because the code is open to extension (we can add new TTS types) but closed to modification (we don't need to change the main code). It also makes unit testing easier, because we can create a MockTTSUnit through a mock factory and test the system without depending on actual TTS APIs. Overall, the factory method fits well into our class diagram and project architecture by keeping responsibilities separate, reducing coupling, and making the system more adaptable for future needs.



6.5 Decorator Pattern

In this system, the **Decorator Pattern** is an ideal choice for enhancing the Lesson class with optional, modular features such as subtitles, comments and analytics tracking — without modifying the core class structure. Since each lesson may require different combinations of these features based on user roles (e.g., students vs. mentors) or course types, the decorator pattern allows for dynamic and flexible behavior composition. It adheres to the **Open/Closed Principle**, enabling the system to extend lesson functionality without altering existing code. This approach also avoids deep inheritance hierarchies, promotes clean separation of concerns, and ensures future scalability as new lesson features can be added independently through additional decorators.



6.6 Command Pattern

The Command Pattern is implemented in the platform to separate the logic of handling help requests from the user interface that triggers them. In particular, this pattern is applied in the context of student help requests, where a HelpRequestCommand encapsulates all the information needed to process a request: the student making the request, the type of help needed, and the service responsible for handling it.

This approach enables us to represent user actions (such as asking for help) as standalone command objects, which can be easily passed, queued, executed, or logged. By separating the request object from the invoker (HelpRequestInvoker) and the receiver (HelpService), we maintain a flexible, modular architecture that can support future features like undo functionality, request queues, or logging mechanisms.