Department of Computer Engineering, Bilkent University

CS319 Object Oriented Software Engineering

TX-D11 Software Quartet

Use Case NFR Tech Stack Report - D1

Section 1 - Anıl Koyuncu

25/02/2024

Kağan Rehber - 22101777

Furkan Komaç - 22102165

Seçkin Alp Kargı - 22001942

Erkan Can Arslan - 22103948

Table of Contents

1. Tech Stack

Frontend: Developed with the React.js library and TypeScript language to ensure a dynamic and type-safe user interface that enhances the educational experience.
Backend: Utilized Django framework alongside Python language to handle communication hub, user management, course content delivery, and data processing efficiently.
Database: MySQL, for its robustness and capability to manage complex data structures required for storing courses, user information, assessments, and more.
Design & Planning: Figma for UI/UX design to create an intuitive and accessible user interface, and Git for source code management and version control, ensuring that development progresses smoothly and collaboratively.

2. Non Functional Requirements

2.1. User Interface and Human Factors

- Design an intuitive interface with easily accessible features for course management, assessments, and communication.
- Implement an accessible color scheme and layout conforming to WCAG 2.0 guidelines, ensuring usability across diverse user needs.
- Responsive design to provide a seamless experience on desktops, tablets, and mobile devices. (Large Desktops: For screens 1200px and wider, Standard Desktops: Ranges from 992px to 1199px, Mobile Devices: Spanning 480px to 599px, covers 320px to 479px)
- Implement a sidebar or top navigation menu with clear icons and labels for "Courses," "Assessments," and "Messages" to ensure users can easily move between different sections of the application.

- Use a color scheme with sufficient contrast between text and background, such as blue on white for normal text and yellow on black for highlighted information, to aid users with visual impairments.

2.2 Performance

Achieve quick load times and efficient interaction handling to support academic activities without delays.Ensure the system supports a high number of concurrent users, reflecting a busy university environment.

- Implement lazy loading for course content and multimedia resources to ensure that pages load within 2 seconds under standard network conditions.
- Use asynchronous API calls for submitting assignments and fetching grades, designed to complete within 50 milliseconds, to keep the interface responsive.
- Deploy an efficient database system architecture that allows maintaining system responsiveness for up to 2000 concurrent users.

2.3. Security and Privacy

Use advanced encryption for data at rest and in transit, focusing on protecting student and faculty information.Implement strict access controls and authentication measures to safeguard user data and privacy.

- Using encryption for all stored data and TLS for data in transit, ensuring protection against data breaches.
- Implement role based account control with detailed permissions, ensuring that only instructors can enter and modify grades, while students can only view their personal information and course materials.
- Ensure that all personal data is encrypted and anonymized when used for analytics.

2.4. Error Handling

Develop a comprehensive error-handling framework to minimize disruptions in the learning process and provide clear guidance to users when issues occur.

- Display contextual error messages, such as "This file format is not supported for assignment uploads. Please use .pdf, .doc, or .docx formats," to guide users in resolving issues.
- Integrate an error logging system that captures and categorizes errors, allowing for real-time monitoring and rapid response to critical issues affecting user experience.

2.5. Scalability

Design the system for easy horizontal scaling to accommodate growth in user numbers and data volume, ensuring the system remains responsive during peak times.

- Using proper database system ensuring that the system can handle a sudden influx of logins and submissions at the start of an exam period without degradation in performance.,
- Implement sharding and replication strategies for the database to distribute the load and ensure quick access to data, even as the volume grows by 300% or more.

2.6. Reliability

Aim for high availability to guarantee continuous access to educational materials and functionalities.

- Using synchronization to ensure that maintenance or failures in one data base do not disrupt the service.
- Schedule nightly backups of the entire database to minimize the risk of data loss due to catastrophic events.

2.7. Compliance and Standards

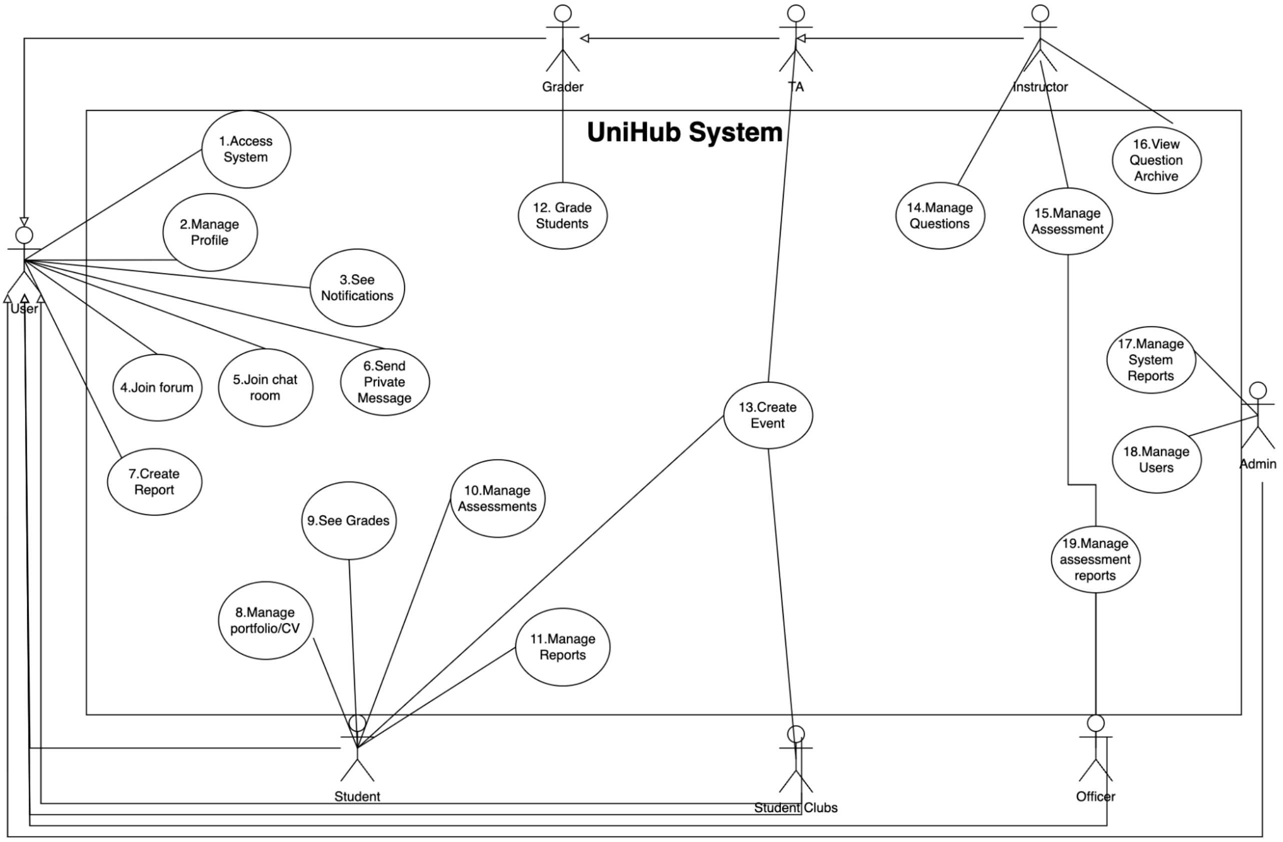
The app should follow the industry guidelines and standards. Such as:

- The application follows GDPR standards for protecting user rights and privacy.

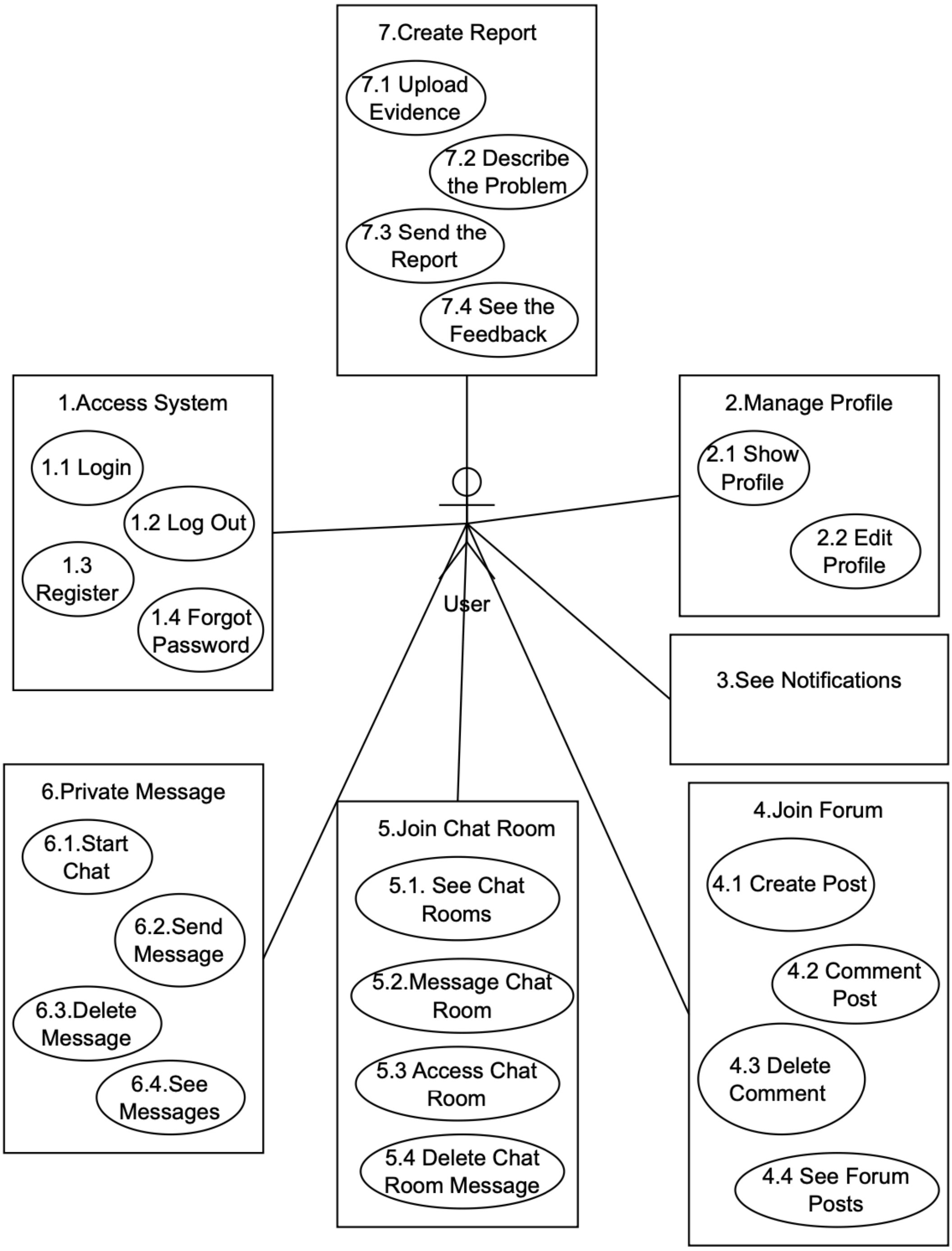
- The application follows WCAG 2.1 standards for accessibility and to be easy to use app.

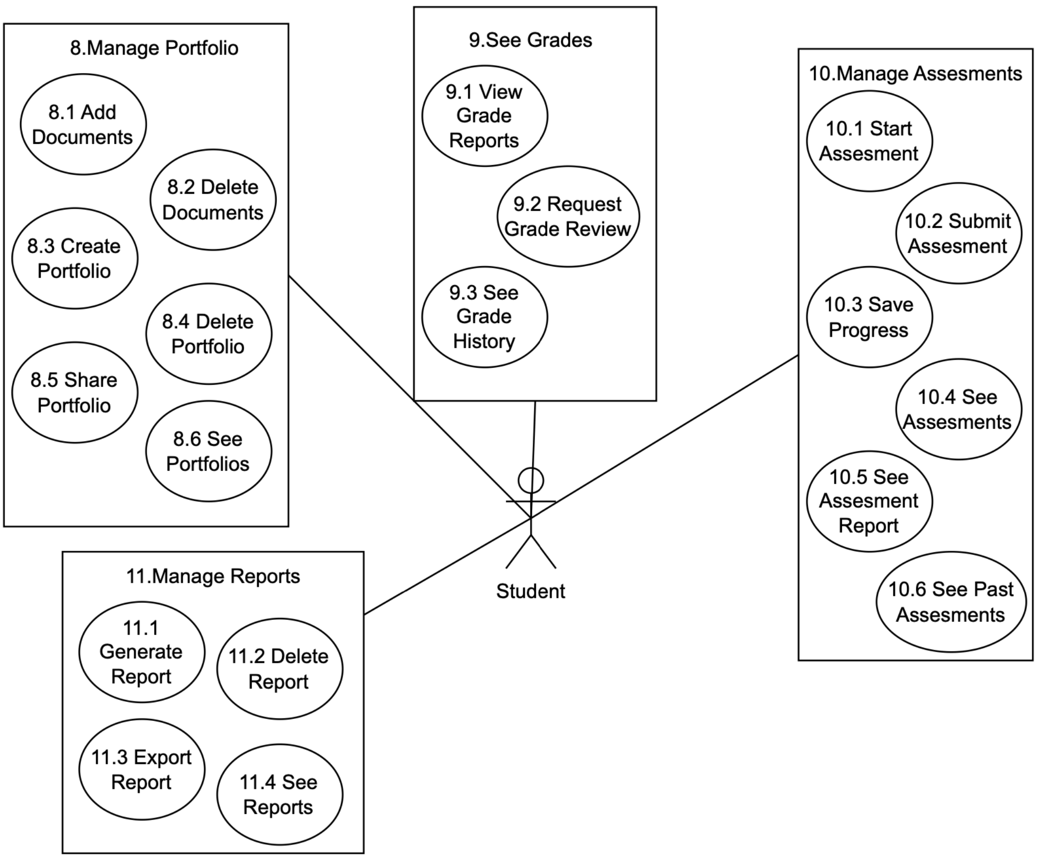
3. Use Cases

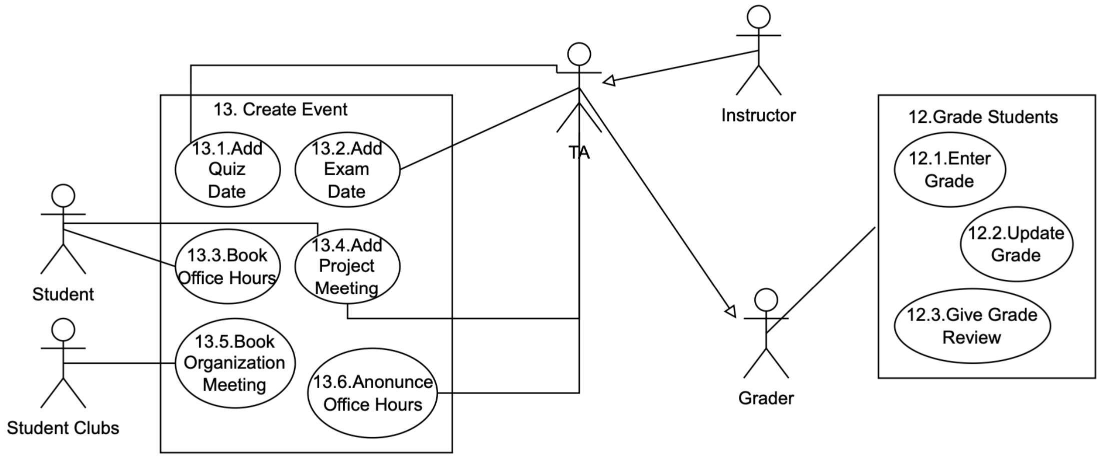
3.1. Level 1 Use Case Diagram

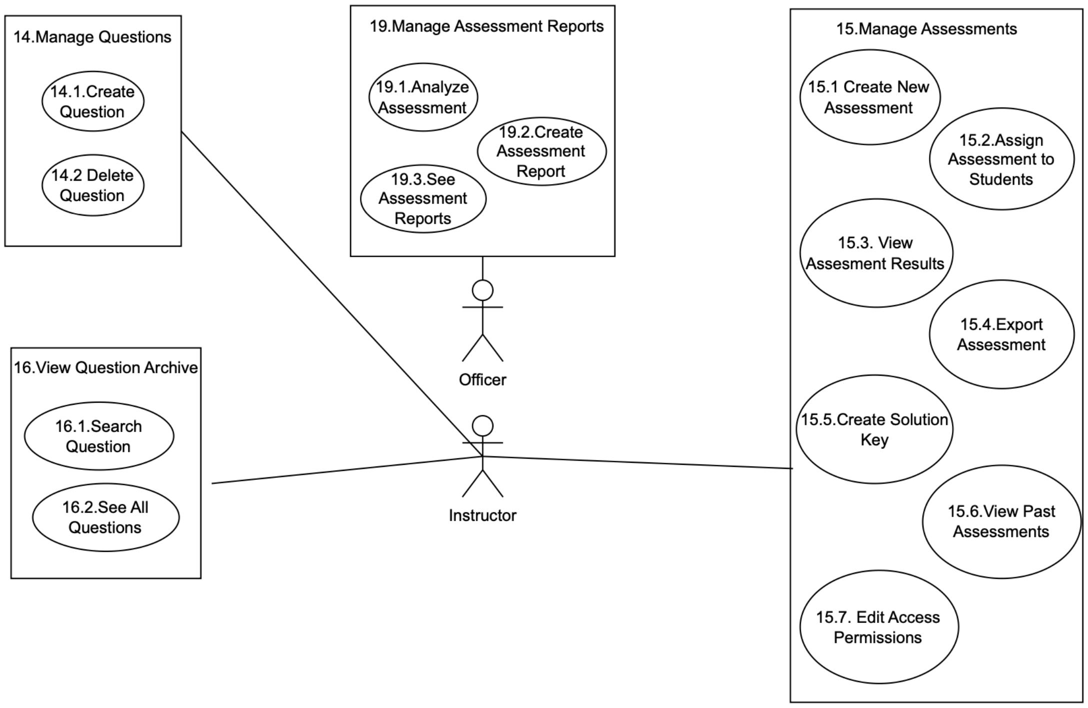


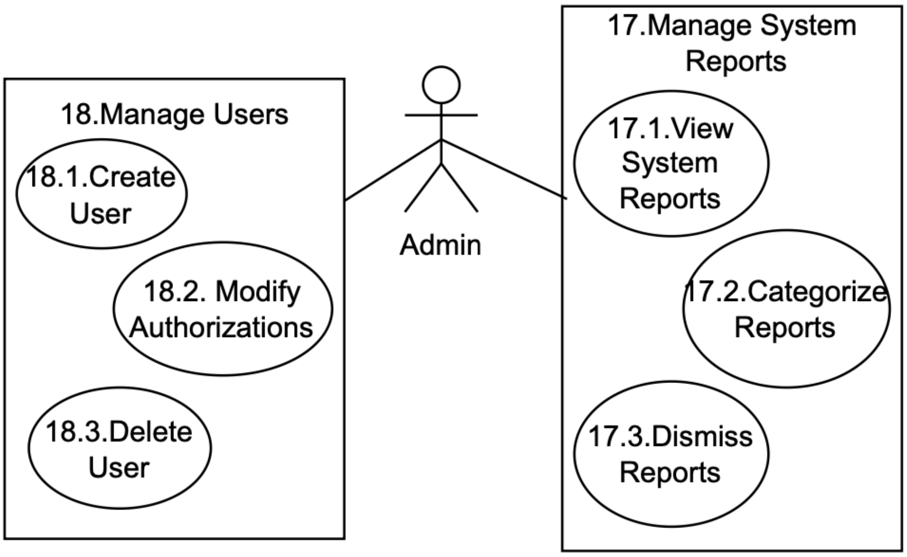
3.2. Level 2 Use Case Diagrams




8.Manage Portfolio
8.1 Add Documents
8.2 Delete Documents
8.3 Create Portfolio
8.4 Delete Portfolio
8.5 Share Portfolio
8.6 See Portfolios
9.See Grades
9.1 View Grade Reports
9.2 Request Grade Review
9.3 See Grade History
10.Manage Assesments
10.1 Start Assesment
10.2 Submit Assesment
10.3 Save Progress
10.4 See Assesments
10.5 See Assesment Report
10.6 See Past Assesments
Student
11.Manage Reports
11.1 Generate Report
11.2 Delete Report
11.3 Export Report
11.4 See Reports


TA
Instructor
13. Create Event
13.1.Add Quiz Date
13.2.Add Exam Date
13.3.Book Office Hours
13.4.Add Project Meeting
13.5.Book Organization Meeting
13.6.Anonunce Office Hours
Student
Student Clubs
Grader
12.Grade Students
12.1.Enter Grade
12.2.Update Grade
12.3.Give Grade Review


14.Manage Questions
14.1.Create Question
14.2 Delete Question
19.Manage Assessment Reports
19.1.Analyze Assessment
19.2.Create Assessment Report
19.3.See Assessment Reports
Officer
15.Manage Assessments
15.1 Create New Assessment
15.2.Assign Assessment to Students
15.3. View Assesment Results
15.4.Export Assessment
15.5.Create Solution Key
15.6.View Past Assessments
15.7. Edit Access Permissions
16.View Question Archive
16.1.Search Question
16.2.See All Questions
Instructor


18.Manage Users
18.1.Create User
18.2. Modify Authorizations
18.3.Delete User
Admin
17.Manage System Reports
17.1.View System Reports
17.2.Categorize Reports
17.3.Dismiss Reports

3.3. Actor Explanation

User: Anyone who interacts with the system is considered a user. A user must register into the system in order to use the system first. All users can use features like accessing the system, managing their profile, sending private or chat room messages and creating technical reports about the system.

Student: The student actor describes the students that can access the university system. The students can use the system for checking their grades, create or edit portfolios for further uses, take assessments assigned to them by their instructor and create/view performance reports created from the assessment test they have taken. Also students can book an office hour with their teaching assistant or instructor to discuss any problem they might have.

Student Clubs: The student clubs actor represents the student clubs. Only feature they can use is to post about the events that they organize and send notifications to students that want to be notified by their chosen student clubs.

Officer: The officer actor is responsible for generating and managing reports on the assessments taken by the students. The content of these reports are usually on the student's performance on certain assessments. (i.e. How many students gave a wrong answer for x question etc.). The officer can create, delete or edit these reports.

Grader: The grader actor is simply responsible for grading students. The grader can enter grades for students, change the grade of a student or respond to student's grade review requests. Grade review requests are simply student's objections to their grade. The grader can either choose to review the grade or refuse it.

TA (Teaching Assistant): The TA is the assistant of the instructors. TA can do everything the graders can do. Besides, the TAs can announce quiz/exam dates, arrange their office hours and can accept student's office hour booking requests.

Instructor: The instructor is mainly responsible for the assessment of the students. The instructor inherits all the features the grader and the TA has. Besides, the instructor simply handles the assessments. They can create questions to add it to an assessment or store them in the question database for further uses as well as deleting the questions they created. The instructor can also create assessment tests and assign it to students for them to take the test.

Admin: Admin is separate from all of the users and is simply responsible for maintaining the system. The admin can manage the authorization of the other users, can create users as well as deleting them and block some users from the system. The admin can also see the technical system reports created by any of the users and fix the system accordingly. They can also give feedback about the technical system reports.

3.4. Use Case Narratives

Explanation Format
Use case name: *Name of the Use case*
Actors: *List of Actors that use the use case.*
Description:*Description of the use case.*
Preconditions: *Preconditions for use case working.*
Basic Flow: *Describing how use cases work systematically.*
Postconditions: *Postconditions for use case working.*
Exception Flow: *Exceptions of the use case.*

Use Case 1.1 Login

Actors:
- User

Description:
Describes the process by which User gains access to their account within the System by entering their credentials.

Preconditions:
- The System is operational.
- User has already been registered and has valid login credentials.

Basic Flow:
1. User navigates to the login interface of the System.

2. User enters their ID and password into the related fields.
3. User submits the login request by clicking the "Login" button.
4. The System validates the entered credentials according to the database.
5. Upon successful validation, the system grants access to the user, and the user is directed to their dashboard or homepage.

Postconditions:

- User has access to their account and is able to interact with the system's features.

Alternate Flow:

- If User enters incorrect login credentials:

 a. The system displays an error message indicating the issue.

 b. User has the opportunity to re-enter their credentials.

 c. After a certain number of failed attempts, the system may lock the account and advise users to reset their password or contact support.

Exception Flow:

- If the system fails to read data from database and terminate the session properly:

 a. The system displays an error message.

 b. The User is advised to close their browser to end the session and ensure security.

Use Case 1.2 Log Out

Actors:

- User

Description:

Outlines the steps taken by User to securely exit their account in the System.

Preconditions:

- User is logged into the system.

Basic Flow:

1. User clicks on the "Log Out" button or link within the system's interface.
2. The System terminates the user's session.

3. User is redirected to a confirmation page or the system's login page, indicating that they have been successfully logged out.

Postconditions:

- User's session in the system has been securely closed.
- The system logs User's action of logging out for security and auditing purposes.

Exception Flow:

- If the system fails to terminate the session properly:
  a. The system displays an error message.
  b. The User is advised to close their browser to end the session and ensure security.

Use Case 1.3 Register

Actors:

- User

Description:

Outlines the steps for a prospective User to create a new account in the System.

Preconditions:

- The System is operational.
- User does not already have an account.

Basic Flow:

1. User navigates to the registration interface of the System.
2. User enters required information, which may include personal details, ID, a password etc.
3. The System may require the user to agree to terms of service or privacy policies.
4. User submits the registration form.
5. The System validates the provided information and checks for validness of ID and email.
6. The system confirms account creation, and User is directed to the login page to access their new account.

Postconditions:

- A new user account is created in the system.
- User becomes an authorized user with access to the system.

Alternate Flows:

- If User enters invalid or incomplete information:
  a. The system displays an error message detailing the issues.
  b. User is prompted to correct the mistakes and resubmit the form.

- If the chosen ID or email is already in use or invalid:
  a. The system informs User to select a different ID or email.
  b. User modifies the required fields and resubmits the form.

Exception Flow:

- If the system fails to terminate the session properly:
  a. The system displays an error message.
  b. The User is advised to close their browser to end the session and ensure security.

Use Case 1.4: Forgot Password

Actors:

- User

Description:

This use case describes the process through which a User can recover access to their account in the University Student System after forgetting their password.

Preconditions:

- The User has an existing account with the system.
- The User has forgotten their account password.

Basic Flow:

1. The User selects the "Forgot Password" option on the login page.
2. The User is prompted to enter their registered email address or username.
3. The University Student System validates the provided information against its records.
4. Upon successful validation, the system sends a password reset link or code to the User’s registered email address.

5. The User checks their email and clicks on the provided password reset link or enters the provided code in the given field.
6. The User is redirected to a secure page where they can set a new password for their account.
7. After setting a new password, the User is informed that their password has been reset successfully and is prompted to log in with the new password.

Postconditions:

- The User has regained access to their account using a new password.
- The old password is invalidated and cannot be used to access the account anymore.

Alternate Flow:

- If the User enters an email address or username not associated with an account:
  a. The system displays an error message indicating the issue.
  b. The User is prompted to try again or to contact customer support for further verification..

Exception Flows:

- If the User does not receive the password reset email due to a system error or other issue:
  a. The User can request the email to be resent.
  b. The User may contact support for assistance

- If the User cannot reset the password due to a system error or other issue:
  a. The system notifies the User of the failure.
  b. The User may be advised to try the process again later or to contact support for help.

Use Case 2.1 Show Profile

Actors:

- User

Description:

Outlines the steps taken by the user to view their profile information in the System.

Preconditions:

- The System is operational.
- User is logged into the system.

Basic Flow:

1. User selects the option to view their profile within the system.
2. The System retrieves User's profile information from the database.
3. User's profile information is displayed, including personal data, academic details, and any other relevant information allowed by the system's privacy settings.
4. User reviews their profile information.
5. If User wants to update any information, they proceed to "Edit Profile"

Postconditions:

- User has accessed and reviewed their profile information.
- The system logs the action of User viewing their profile for security and auditing purposes.

Exception Flow:

- If the system cannot retrieve User's profile information due to a system error or connectivity issue:
  a. The system displays an error message to User.
  b. The User may try to refresh the page or log out and log in again.
  c. If the problem persists, User is advised to contact technical support.

Use Case 2.2 Edit Profile

Actors:

- User

Description:

Describes the process by which User updates or modifies their profile information within the System.

Preconditions:

- The System is operational.
- User is logged into the system and has accessed their profile (refer to the "Show Profile" use case).

Basic Flow:

1. The User selects the option to edit their profile from their profile page.

2. The System presents the editable fields of User's profile, which may include personal data, contact information, academic details, etc.
3. The User makes the desired changes to their profile information in the provided fields.
4. User submits the changes by selecting the save or update option.
5. The System validates the updated information according to the system's rules and constraints.
6. Upon successful validation, the system updates User's profile information in the database.
7. The system displays a confirmation message to User that their profile has been updated.

Postconditions:
- User's profile information is updated in the system.
- The system maintains a log of the changes made to User's profile for security and auditing purposes.

Alternate Flows:
- If User attempts to enter invalid information:
  a. The system displays an error message indicating which fields are invalid.
  b. The User is prompted to correct the fields with invalid information.
  c. User resubmits the changes.
- If User does not have permission to edit certain fields:
  a. Those fields are either not displayed or are disabled.
  b. The system informs User which fields are not editable and why.

Exception Flow:
- If the system encounters an issue while saving the updated profile:
  a. The system displays an error message indicating that the update failed.
  b. The User is advised to try again later or contact technical support if the issue persists.

Use Case 3: See Notifications

Actors:
- User

Description:

Describes how User receives and views notifications within the System which may include alerts about new messages, forum replies, system updates, or other relevant activities.

Preconditions:

- User is logged into the system.
- Notifications are generated by the system as a result of certain events or actions.

Basic Flow:

1. The User is alerted to the presence of new notifications through a visual indicator, such as a badge or pop-up, on the system's interface.
2. User clicks on the notification icon or link in the interface.
3. The System displays a list of recent notifications. This list may include information such as the type of notification, the associated event, and the time of occurrence.
4. User reviews the notifications to understand recent activities or actions required.
5. User can click on individual notifications to be taken directly to the related item or page, such as a new forum post or a message thread.
6. After viewing, the system may automatically mark the notifications as read or provide User with an option to clear them.

Postconditions:

- The User is informed about recent and relevant activities within the system.
- The User can take action on notifications if necessary.

Alternate Flow:

- If there are no new notifications:

  a. The system displays a message indicating there are no new notifications.

Exception Flow:

- If the notification service is unavailable:

  a. The system displays an error message indicating that notifications cannot be retrieved at the moment.

  b. Users may be advised to try again later.

Use Case 4.1: Create Post

Actors:

- User

Description:

Outlines the process by which a User can create a new post in the forums provided by the System.

Preconditions:

- Users must be logged into the system.

Basic Flow:

1. User navigates to the forum section of the System.
2. User selects the appropriate forum category to create a new post.
3. User clicks on the "Create Post" button or link.
4. The system presents a form where User can enter the title and body of the post.
5. User enters the relevant content and may attach files or links if the system permits.
6. User submits the post by clicking the "Post" button.
7. The System saves the post and displays it in the selected forum category.

Postconditions:

- The new forum post is visible to other Users.
- User may receive notifications of replies if the system supports this feature.

Alternate Flow:

- If User attempts to post in a restricted category:
  a. The system displays an error message indicating the lack of permission.

Exception Flow:

- If there is a system error while submitting the post:
  a. The system informs User of the error and may suggest the user to retry.

Use Case 4.2: Comment Post

Actors:

- User

Description:

Describes how a user can add a comment to an existing forum post within the System.

Preconditions:

- User is logged into the system.
- The forum post to be commented on exists.

Basic Flow:

1. User navigates to the forum post they wish to comment on.
2. User clicks on the "Comment" or "Reply" button associated with the post.
3. The system presents a text input field for User to write their comment.
4. User enters their comment into the field.
5. User submits the comment by clicking the "Submit" or "Reply" button.
6. The System adds the comment to the post's thread.

Postconditions:

- User's comment is visible to others viewing the post.
- The original poster and other subscribers may be notified of the new comment.

Alternate Flow:

- If the post is closed for comments:
  a. The system displays a message that the post cannot be commented on.

Exception Flow:

- If there is a submission error:
  a. User is informed and may be asked to try again.

Use Case 4.3: Delete Comment

Actors:

- User

Description:

Illustrates the process by which a User can delete their own comment from a forum post in the System.

Preconditions:

- The User must be logged into the system.
- User has previously posted a comment on a forum post.

Basic Flow:

1. User navigates to the comment they wish to delete.
2. User clicks on the "Delete" option associated with their comment.
3. The system may ask User to confirm the deletion.
4. Upon confirmation, the System removes the comment from the forum post.

Postconditions:

- The comment is no longer visible in the forum post thread.

Alternate Flow:

- If User does not have permission to delete the comment:
  a. The system displays an error message indicating that the action is not allowed.

Exception Flow:

- If there is a system error during deletion:
  a. User is informed of the failure and may be asked to try again.

Use Case 4.4: See Forum Posts

Actors:

- User

Description:

This use case details how a User can view posts on various forums within the University Student System.

Preconditions:

- The User must be logged into the University Student System.
- Forums must exist within the system and contain posts.

Basic Flow:

1. The User navigates to the forum section of the University Student System.
2. The User selects a specific forum they are interested in from a list of available forums.
3. The system displays a list of topics within the selected forum.
4. The User selects a topic to view detailed posts.

5. The system presents all the posts within the topic, including the original post and all subsequent responses.
6. The User can read through the posts to gain information, insights, or join in the discussion by posting a reply if they wish to.

Postconditions:

- The User has successfully viewed the posts in a forum.
- The User has the option to engage with the forum by posting replies or starting a new topic.

Alternate Flows:

- If the User attempts to access a forum they do not have permissions for:
  a. The system displays a message indicating they do not have access.
  b. The User is prompted to either request access if possible or return to the list of other forums.
- If there are no posts in the selected topic or forum:
  a. The system informs the User that there are no posts to display.
  b. The User may be given the option to create the first post or choose another topic.

Exception Flow:

- If the system fails to load the posts due to a technical issue:
  a. An error message is displayed to the User.
  b. The User may be asked to refresh the page or try accessing the posts at a later time.

Use Case 5.1: See Chat Rooms

Actors:

- User

Description:

Details the automatic process where a user is entered into a chat room relevant to their interests or current course enrollment within the System.

Preconditions:

- User must be logged into the system.
- The chat room is associated with a course or group User is part of.

Basic Flow:

1. Upon logging in or navigating to the chat room section, the system automatically detects User’s relevant associations.
2. The system automatically enters The User into chat rooms that are linked to their courses or groups.
3. The User is notified that they have been added to the chat room(s).

Postconditions:

- The User is present in a chat room related to their academic interests or group affiliations and can participate in discussions.

Exception Flow:

- If the system cannot add User to a chat room due to a system error:
  a. The system informs User of the issue.
  b. The User may attempt to access the chat room manually or contact support.

Use Case 5.2: Message Chat Room

Actors:

- User

Description:

Describes how a User can send a message to a chat room within the System.

Preconditions:

- The User has been added to the chat room either automatically or manually.

Basic Flow:

1. User navigates to the desired chat room.
2. The User types their message into the chat input field provided in the chat room interface.
3. User sends the message by clicking the "Send" button or pressing enter.
4. The system broadcasts the message to all members present in the chat room.
5. The message is visible to all members of the chat room in real-time or upon their next access.

Postconditions:

- User’s message is successfully sent and is visible in the chat room conversation.

Exception Flow:

- If there is a failure in sending the message:
  a. The system displays an error message to User.
  b. The User may try to resend the message.

Use Case 5.3: Access Chat Room

Actors:

- User

Description:

Outlines the steps a User takes to view messages within a chat room in the System.

Preconditions:

- User is a member of the chat room.

Basic Flow:

1. User opens the chat room from their chat rooms list.
2. The System displays the chat interface along with the history of messages sent in the room.
3. User scrolls through the messages to read previous discussions.
4. The User can interact with the messages if the system allows, such as reacting to messages or replying to specific ones.

Postconditions:

- User is up to date with the conversations that have taken place in the chat room.

Exception Flow:

- If the messages fail to load due to a system error:
  a. The system notifies User of the error.
  b. The User may try to reload the chat room or contact technical support for assistance.

Use Case 5.4: Delete Chat Room Message

Actors:

- User

Description:

This use case outlines the process by which a User or a Chat Room Moderator/System Administrator can delete a message from a chat room within the University Student System.

Preconditions:

- The User is logged into the University Student System and has joined a chat room.
- The User has the permission to delete messages in the chat room. If not, a Moderator or System Administrator has the necessary permissions.

Basic Flow:

1. The User navigates to the chat room from which they wish to delete a message.
2. The User locates the message to be deleted.
3. The User clicks on an option to delete the message (e.g., a trash bin icon or a "delete" button next to the message).
4. The system asks the User to confirm the deletion of the message.
5. Upon confirmation, the system deletes the message from the chat room.
6. The system updates the chat room to reflect the deletion and may leave a placeholder note indicating a message was deleted, depending on the system's configuration.

Postconditions:

- The message is permanently removed from the chat room.
- The chat history reflects the deletion of the message.

Alternate Flows:

- If the User does not have permission to delete the message:
  a. The system displays an error message indicating a lack of permission.
  b. The User is directed back to the chat room without the message being deleted.

- If the message has already been deleted or does not exist:
  a. The system notifies the User that the message is no longer available.
  b. The User is redirected back to the current chat room messages.

Exception Flow:

- If the message cannot be deleted due to a system error:

  a. The User is informed of the failure to delete the message.

  b. The User may be asked to try again later or contact a Moderator or System Administrator.

Use Case 6.1: Start Chat

Actors:

- User

Description:

Details the steps required for a User to initiate a private chat with another User within the System.

Preconditions:

- The User must be logged into the system.

Basic Flow:

1. User navigates to the private messaging or chat section of the System.
2. User selects the option to start a new chat or to find a User to chat with.
3. The system presents a search or selection mechanism for User to find the intended recipient.
4. User selects the recipient for the new chat.
5. The system opens a new chat window or dialogue box.
6. The User types their initial message into the chat interface.
7. User sends the message, which initiates the chat conversation.

Postconditions:

- A private chat session is established between User and the selected recipient.
- The initial message is visible in the chat window for both parties.

Alternate Flows:

- If the selected recipient cannot be found or is not available for chat:

  a. The system informs User about the issue.

- If User does not have the necessary permissions to initiate a chat:

a. The system displays an error message indicating the restriction.

Exception Flow:

- If the system fails to start chat due to a technical issue:
  a. An error message is displayed to the User.
  b. The User may be asked to refresh the page or try accessing the posts at a later time.

Use Case 6.2: Send Message

Actors:

- User

Description:

Describes the process through which a User sends a message in an ongoing private chat within the System.

Preconditions:

- The User must be logged into the system.
- An existing chat session must be open.

Basic Flow:

1. The User types their message into the active chat input field.
2. User clicks the "Send" button or presses the enter key to dispatch the message.
3. The system transmits the message and displays it in the chat window.
4. The recipient receives the message in real-time if they are online or once they access the chat session.

Postconditions:

- The sent message is added to the chat history and is visible to both User and the recipient.

Exception Flow:

- If there is a network or system error preventing message delivery:
  a. The system informs User of the error.
  b. The system may automatically retry sending the message or prompt User to send it again.

Use Case 6.3: Delete Message

Actors:
- User

Description:
Outlines the procedure for a User to delete a message they have sent in a private chat within the System.

Preconditions:
- The User must be logged into the system.
- User has previously sent a message in a private chat.

Basic Flow:
1. User navigates to the message they wish to delete within the chat session.
2. User selects the "Delete" option for the message.
3. The system asks User to confirm the deletion of the message.
4. Upon confirmation, the system removes the message from the chat history.

Postconditions:
- The deleted message is no longer visible in the chat window.
- Depending on the system's features, the recipient may or may not be able to see that a message was deleted.

Exception Flow:
- If there is a system error during the deletion process:
  a. User is informed of the error.
  b. The system may offer the option to attempt the deletion again.

Use Case 6.4: See Messages

Actors:
- User

Description:
The See Messages use case describes the process where a User views their private messages within the System.

Preconditions:

- The User must be logged into the University Student System.
- User has an existing account with the system that can receive messages.

Basic Flow:

1. The User navigates to the private messaging section of the University Student System.
2. The system displays a list of conversations or message threads that the User is a part of.
3. The User selects a conversation to view the full exchange of messages with another student or group.
4. The system presents the selected message thread, showing the exchanged messages in chronological order.
5. The User reads through the messages, scrolling up or down through the history as desired.

Postconditions:

- The User has accessed and read their private messages.
- The User's status in the system may be updated to reflect that new messages have been read.

Alternate Flow:

- If the User has no messages:
  a. The system informs the User that there are no messages to display.
- If the User attempts to access a message thread that has been deleted or is no longer available:
  a. The system notifies the User of the unavailability.
  b. The User is redirected back to the list of available message threads.

Exception Flow:

- If the system fails to load the messages due to a technical error:
  a. The system displays an error message to the User.
  b. The User is advised to try accessing the messages again later or to contact technical support if the problem persists.

Use Case 7.1: Upload Evidence

Actors:

-User

Description:

-This use case describes the process for a User to upload evidence supporting a report about a system issue or problem.

Preconditions:

-User must be logged into the system.

-The User has identified a system issue or problem they wish to report.

Basic Flow:

1. User navigates to the "Create Report" or "Report Issue" section within the system.
2. User selects the "Upload Evidence" option to attach supporting documentation for the issue.
3. User browses their device and selects files (e.g., screenshots, log files) that illustrate the problem.
4. The system verifies the file format and size before uploading.
5. Upon successful upload, the system attaches the files to the draft report.
6. A confirmation message is displayed, indicating the successful upload of evidence.

Postconditions:

-The evidence is successfully attached to the draft report, ready for further action by the User.

Exception Flow:

-If the file format or size is not supported:

a. The system displays an error message detailing the issue.

b. The User is prompted to select a different file or modify the file to meet requirements.

Use Case 7.2: Describe the Problem

Actors:

User

Description:

-Details the steps for a User to describe the problem they are experiencing with the system, as part of creating a report.

Preconditions:

-User must be logged into the system and initiated the report creation process, possibly after uploading evidence.

Basic Flow:

1. The User is prompted to enter a detailed description of the issue, including actions leading up to the problem, what went wrong, and any error messages seen.
2. User fills in the problem description in the provided input area.
3. The system saves the detailed description as part of the report.
4. User reviews the description for accuracy and completeness.

Postconditions:

-The problem description is successfully added to the report, ready for submission.

Exception Flow:

-If the description is left blank or is insufficient:

a. The system prompts the User to provide more detailed information before proceeding.

Use Case 7.3: Send the Report

Actors:

User

Description:

-Outlines the process for a User to send a report about a system issue to the support team, including uploaded evidence and a detailed problem description.

Preconditions:

-User must be logged into the system.

-User has prepared a report with evidence and a detailed problem description.

Basic Flow:

1. User selects the "Send Report" option after completing the report.
2. The system prompts for confirmation to ensure User intends to send the report.

3. Upon User's confirmation, the system sends the report to the support team and provides a report ID for tracking.
4. A confirmation message is displayed to the User, acknowledging the report's submission.

Postconditions:

-The report is successfully submitted to the support team for review and action.

Exception Flow:

-If there is a problem sending the report (e.g., network issue):

a. The system notifies the User of the issue.

b. User is given the option to try sending the report again.

Use Case 7.4: See the Feedback

Actors:

-User

Description:

-Describes the procedure for a User to view feedback on a report they submitted regarding a system issue.

Preconditions:

-User must be logged into the system.

-User has submitted a report and been notified that feedback is available.

Basic Flow:

1. The User accesses the "My Reports" section to find their submitted report.
2. User selects the report to view the feedback provided by the support team.
3. The system displays the feedback, including any resolutions, workarounds, or requests for further information.
4. The User can respond to the feedback if further dialogue with the support team is necessary.

Postconditions:

-User is informed of the actions taken in response to their report or any further steps required.

Exception Flow:

-If feedback is not accessible due to a system error:

a. User is informed of the error and advised to try again later.

b. If the issue persists, The User may contact support directly for assistance.

Use Case 8.1 Add Documents

Actors:

-Student

Description:

-Outlines the procedure for a Student to add documents to their academic portfolio within the system.

Preconditions:

-The Student must be logged into the system.

Basic Flow:

1. Student navigates to their academic portfolio section within the system.
2. Student selects the "Add Documents" option.
3. Student uploads the chosen document(s) from their device.
4. The system validates the format and size of the document(s).
5. Upon successful validation, the document(s) are added to the portfolio.
6. The system displays a confirmation message indicating successful upload.

Postconditions:

-The student's portfolio is updated with the newly added document(s).

Exception Flow:

-If an unsupported document format or file size is detected:

a. The system displays an error message.

b. Student is prompted to select a different file.

Use Case 8.2 Delete Documents

Actors:

-Student

Description:

-Details the steps for a Student to delete documents from their academic portfolio.

Preconditions:

-The Student must be logged into the system and have documents in their portfolio.

Basic Flow:

1. The Student accesses their portfolio and selects the document(s) they wish to delete.
2. The Student clicks the "Delete" option.
3. The system asks for confirmation to delete the selected document(s).
4. Upon confirmation, the document(s) are removed from the portfolio.
5. A success message confirms the deletion.

Postconditions:

-The selected document(s) are no longer present in the student's portfolio.

Additional Flow:

-If deletion is canceled after the confirmation prompt:

a. No changes are made to the portfolio

.

Use Case 8.3 Create Portfolio

Actors:

-Student

Description:

-Describes the process for a Student to create a new academic portfolio in the system.

Preconditions:

-The Student must be logged into the system.

Basic Flow:

1. Student goes to the portfolio management section.
2. The Student selects "Create Portfolio."
3. Student names the new portfolio and optionally provides a description.
4. The system creates the portfolio and confirms its creation.
5. A confirmation message is displayed to the student.

Postconditions:

-A new portfolio is created and available to the student.

Exception Flow:

-If the portfolio name already exists:

a. The system prompts the student to choose a different name.

Use Case 8.4 Delete Portfolio

-Actors:

Student

Description:

-Outlines the procedure for a Student to delete an entire academic portfolio.

Preconditions:

-The Student must be logged into the system and own at least one portfolio.

Basic Flow:

1. The Student selects the portfolio they wish to delete.
2. Student clicks "Delete Portfolio."
3. The system requests confirmation for the deletion.
4. Upon confirmation, the portfolio and its contents are deleted.
5. A success message confirms the portfolio's deletion.

Postconditions:

-The selected portfolio is completely removed from the system.

Additional Flow:

-If deletion is canceled at the confirmation stage:

a. The portfolio remains unchanged.

Use Case 8.5 Share Portfolio

Actors:

-Student

Description:

-Details the steps for a Student to share their academic portfolio with an Instructor.

Preconditions:

-The Student is logged into the system.

-The portfolio intended for sharing exists.

Basic Flow:

1. Student selects the portfolio to share.
2. The Student initiates the "Share Portfolio" process.
3. Student enters the Instructor's email or username.
4. The system grants access to the Instructor and notifies them.
5. A confirmation message is displayed to the Student.

Postconditions:

-The Instructor has access to view the shared portfolio.

Additional Flow:

-If the entered email or username does not exist:

a. The system alerts the Student and requests correct information.

Use Case 8.6 See Portfolio

Actors:

-Student (Primary Actor)

Description:

-This use case outlines the process for Students to view their academic portfolio within the system. It enables Students to access, review, and reflect on their compiled work, including projects, assignments, and any other documents they have added to their portfolio. Instructors may also view a Student's portfolio when given permission, typically for assessment or feedback.

Preconditions:

-The Student must be logged into the system.

-The portfolio must already contain documents or entries added by the Student.

Basic Flow:

1. The Student navigates to the "Portfolio" section within the system.
2. The system displays an overview of the Student's portfolio, organized by categories, dates, or custom arrangements made by the Student.
3. Student selects a specific section or document to view in detail.
4. The system presents the selected portfolio content, including documents, multimedia files, or links, along with any descriptions or reflections the Student has provided.
5. The Student reviews the contents of their portfolio, which may include reflecting on their growth, preparing for assessments, or gathering work to share with Instructors or peers.

Postconditions:

-The Student has accessed and reviewed their portfolio, gaining insights into their academic progress and achievements.

Exception Flow:

-If the portfolio is empty or specific content cannot be found, the system notifies the Student with a message indicating the absence of content or suggesting actions to add content.
If an Instructor attempts to access a Student's portfolio without permission, the system denies access and may notify the Student of the access attempt.

.

Use Case 9.1 View Grade Reports

Actors:

-Student

Description:

-This use case outlines the steps for a Student to view their grade reports for various courses and assessments within the system.

Preconditions:

-The Student must be logged into the system.

Basic Flow:

1. The Student navigates to the "Grades" section after logging in.
2. The Student selects "View Grade Reports" to access their grades.
3. The system lists all courses with the grades for each assessment.

4. The Student can delve into a specific course to see a detailed grade report, including component breakdowns and instructor feedback.
5. The detailed grade report is displayed for review by the Student.

Postconditions:

-The Student has accessed and reviewed their grade reports for selected courses.

Exception Flow:

-If reports are pending due to delays, the system informs the Student with a pending message.

Use Case 9.2 Request Grade Review

Actors:

-Student

Description:

-Describes the process for a Student to request a review of a grade they believe is incorrect or wish to dispute.

Preconditions:

-Student is logged into the system and has identified a disputable grade.

Basic Flow:

1. The Student finds a disputable grade in their grade report.
2. The Student initiates a "Request Grade Review" from the grades section.
3. Student fills out and submits a form detailing the grade dispute.
4. The request is forwarded to the relevant Instructor for evaluation.
5. The Instructor reviews the assessment and the request, making a decision on the grade.
6. If necessary, the grade is updated, and feedback is provided to the Student.
7. The Student is notified of the outcome and can view any feedback.

Postconditions:

-The grade review request is processed, and the Student is informed of the outcome.

Additional Flow:

-If further clarification or additional submissions are required from the Student, additional steps are initiated.

Use Case 9.3 See Grade History

Actors:

-Student

Description:

-Outlines the steps for a Student to view their historical grade data across all courses and terms.

Preconditions:

-The Student must be logged into the system.

Basic Flow:

1. The Student accesses the "Grades" section and selects "See Grade History."
2. The system displays a comprehensive view of the Student's grade history.
3. The Student can select specific courses or terms to examine detailed grades and trends.
4. The system presents grade history in a graphical or tabular format for easy analysis.
5. The Student utilizes this data to understand their academic progress over time.

Postconditions:

-The Student reviews their historical grade data, gaining insights into their academic performance.

Use Case 10.1 Start Assessment

Actors:

-Student

Description:

-Describes the procedure for a Student to initiate an assessment within the system.

Preconditions:

-The Student must be logged into the system.

-Assessments must be available for the Student to start.

Basic Flow:

1. TheStudent accesses the "Assessments" section.
2. The Student selects an assessment they wish to begin.
3. System details the assessment including instructions, duration, and rules.
4. The Student initiates the assessment by clicking "Start Assessment."
5. System begins the assessment, starts any applicable timers, and presents questions.

Postconditions:

-Student is actively engaged in completing the assessment.

Exception Flow:

-If the assessment is timed, the system periodically updates the Student on time remaining.

Use Case 10.2 Submit Assessment

Actors:

-Student

Description:

-Outlines the steps for a Student to submit their completed assessment.

Preconditions:

-Student has initiated an assessment.

Basic Flow:

1. The Student completes the assessment and reviews their answers.
2. The Student submits their responses by clicking "Submit Assessment."
3. System confirms submission and concludes the session.
4. A confirmation of successful submission is displayed to the Student.
5. System records the submission time and date.

Postconditions:

-Student's assessment responses are successfully submitted.

Exception Flow:

-If submission is attempted post time expiration, the system may apply penalties or deny submission based on rules.

Use Case 10.3 Save Progress

Actors:

-Student

Description:

-Details how a Student can save their progress during an assessment to continue later.

Preconditions:

-Student is logged into the system and participating in an assessment.

Basic Flow:

1. The Student opts to save their current progress by clicking "Save Progress."
2. System saves the responses and current state.
3. A confirmation message informs the Student of successful save.
4. The Student can exit and later resume the assessment from the saved point.

Postconditions:

-Student's progress within the assessment is saved.

Exception Flow:

-System might auto-save progress at certain intervals as a precaution.

Use Case 10.4 See Assessments

Actors:

-Student

Description:

-Describes how a Student views available, in-progress, and completed assessments.

Preconditions:

-The Student must be logged into the system.

Basic Flow:

1. The Student enters the "Assessments" section.
2. System displays a catalog of assessments by status.
3. The Student explores assessments to view details or to start/continue.
4. Student selects an assessment based on its current status.

Postconditions:

-Student is informed about their assessments and their statuses

.

Additional Flow:

-Students can filter or sort assessments for easier navigation.

Use Case 10.5 See Assessment Report

Actors:

-Student

Description:

-Explains the process for a Student to view the report of a completed assessment.

Preconditions:

-The Student is logged into the system.

-Assessments have been completed and graded.

Basic Flow:

1. The Student accesses the "Assessments" section post-grading.
2. Student views "Assessment Report" for a chosen assessment.
3. System presents the report including scores and feedback.
4. The Student reviews their performance and instructor feedback.

Postconditions:

-The Student gains insights into their assessment performance.

Additional Flow:

-If concerns exist, The Student may request a review or clarification on the report.

Use Case 10.6 See Past Assessments

Actors:

-Student

Description:

-This use case details the process for a Student to view their past assessments within the system, allowing them to review completed exams, quizzes, assignments, and their outcomes.

Preconditions:

-The Student must be logged into the system.

Basic Flow:

1. The Student accesses the "Assessments" or "History" section within the system.
2. System displays a list of past assessments, including exams, quizzes, and assignments.
3. The Student selects an assessment to view detailed information, such as the date taken, grade received, and any available feedback or answers.
4. System presents a detailed view of the selected assessment, including specific questions, the Student's answers, correct answers (if available), and any instructor feedback.
5. The Student reviews the information to understand their performance, areas of strength, and areas needing improvement.

Postconditions:

-The Student has accessed and reviewed their past assessments, gaining insight into their academic progress.

Exception Flow:

-If no past assessments are available (e.g., for a new student or at the beginning of a term), the system displays a message indicating there are no past assessments to view.

If the system encounters an error retrieving past assessments, it informs the Student of the issue and may suggest trying again later..

Use Case 11.1 Generate Reports

Actors:

-User (Primary Actor): This role can be filled by instructors, administrators, or students, each with varying levels of access and functionality within the system depending on their role.

Description:

-This use case outlines the process for Users to generate reports within the system, focusing on gathering and analyzing data related to assessments, course progress, or user activity, among other metrics.

Preconditions:

-User must be logged into the system with appropriate permissions to generate reports.

Basic Flow:

1. User navigates to the "Reports" section within the system.
2. User selects the "Generate Report" option.
3. User chooses the type of report and specifies necessary parameters, such as date range, courses, or user groups.
4. The system processes the request, compiles the data, and generates the report.
5. The generated report is displayed to the User for review.

Postconditions:

-A new report based on the User's specifications is created and available for review.

Exception Flow:

-If no data matches the specified criteria, the system notifies the User, suggesting adjustments to the parameters.

Use Case 11.2 Delete Reports

Actors:

-User (Primary Actor): Typically an administrator or a user with permissions to manage and delete reports.

Description:

-This use case details the steps for Users to delete unwanted or outdated reports from the system, helping manage the storage and relevance of generated reports.

Preconditions:

-The User must be logged into the system and have the necessary permissions to delete reports.

Basic Flow:

1. User accesses the "Reports" section and views the list of generated reports.
2. User selects the report(s) they wish to delete.
3. The system requests confirmation to ensure intentional deletion.
4. Upon User's confirmation, the selected report(s) are deleted from the system.
5. A confirmation message indicates the successful deletion of the report(s).

Postconditions:

-The selected report(s) are permanently removed from the system.

Exception Flow:

-If a report is currently in use or required for compliance, the system may restrict deletion and inform the User.

Use Case 11.3 Export Reports

Actors:

-User (Primary Actor): This role encompasses instructors, administrators, or students who need to export reports for external use or analysis.

Description:

-This use case describes the procedure for Users to export generated reports from the system into various formats such as PDF, Excel, or CSV for offline access, sharing, or further analysis.

Preconditions:

-The User must be logged into the system and have access to the report(s) they wish to export.

Basic Flow:

1. User selects a generated report from the "Reports" section.
2. User chooses the "Export Report" option.
3. User selects the desired format for the export.
4. The system processes the request and prepares the report for download.
5. The User is prompted to download the exported file to their device.

Postconditions:

-The User successfully exports and downloads the report in the selected format.

Exception Flow:

-If an error occurs during the export process, the system notifies the User and may suggest retrying or selecting a different format.

Use Case 11.4 See Reports

Actors:

-User (Primary Actor): Users in this context can be instructors, administrators, or students looking to access and review generated reports.

Description:

-This use case outlines the process for Users to view and interact with reports that have been generated within the system. It allows Users to review data and insights provided by various reports.

Preconditions:

-User must be logged into the system with permissions to view reports.

Basic Flow:

1. User navigates to the "Reports" section within the system.
2. The system displays a list of available reports, categorized by type, date, or other relevant criteria.
3. User selects a report to view detailed information, analysis, and insights.
4. The system presents the selected report, including any graphical data representations, tables, or textual analyses.
5. User reviews the report details within the system.

Postconditions:

-User accesses and reviews the desired reports, gaining insights and information relevant to their needs.

Exception Flow:

-If a report fails to load or display correctly, the system informs the User and may suggest troubleshooting steps or alternative actions.

Use Case 12.1 Enter a Grade

Actors:
-Instructor
-TA
-Grader

Description:
This use case outlines the steps taken by an instructor or grader to enter grades for students into the grade database.

Preconditions:

-The instructor or grader is logged in to the university hub.

-The instructor or grader has the necessary permissions to enter grades for the specific course, exam and students.

Basic Flow:
1. The instructor or grader navigates to the section within the university hub that displays their courses.
2. They select the specific course for which they want to enter grades.
3. The system displays the grades for the selected course, listing all enrolled students and their associated grades which will be blank if not entered before.
4. The instructor or grader can enter grades for each student using various methods : Entering grades directly in the table cells, Uploading a file containing grades in a specific format.
5. Once grades are entered, the instructor or grader saves their changes. The system prompts for confirmation before finalizing grade entry.

Postconditions:
-The entered grades are saved in the system's grade database for the selected course and students.
Students might be notified about the updated grades. The instructor or grader has the ability to view and edit previously entered grades within restricted timeframes (refer to 13.2 and 13.3 use cases).

Alternate Flow:

5a. If a required grade is missing or invalid (e.g., not a valid value based on the grading scheme), the system displays an error message prompting the instructor or grader to correct before saving.

Additional Flow:
Instructor might try to enter a grade for a student who did not take that exam. In this case the user will be notified about the situation.

Use Case 12.2 Update a Grade
Actors:
-Instructor
-TA
-Grader (if authorized)

Description:
This use case outlines the steps taken by an instructor, TA, or grader to modify previously entered grades in the grade database.

Preconditions:
-The instructor, TA, or grader is logged in.
- They have the necessary permissions to update grades for the specific course and students.
- The course grading scheme and assessment details are already set up in the system. Grades have already been entered for the students in the grade book (refer to Use Case 13.1: Enter a Grade).

Basic Flow:
1. The instructor, TA, or grader navigates to the section within the university hub that displays their courses.
2. They select the specific course for which they want to update grades
3. The system displays the grades table for the selected course, listing all students and their grades
4. The instructor, TA, or grader identifies the specific student and grade entry they want to modify
5. They modify the existing grade using the available methods, which might be similar to the options in Use Case 13.1 (e.g., direct editing, uploading files).

6. Once the update is made, the instructor, TA, or grader saves their changes. The system may prompt for confirmation before finalizing the update.

Postconditions:
-The updated grade is reflected in the system's grade database for the selected student and course.
-Students might be notified about the updated grade.
-A log or history may be saved within the system to save changes on grades (who made the update, when, etc.).

Alternate Flow:
If the update results in a missing or invalid grade (e.g., not a valid value based on the grading scheme), the system displays an error message prompting the instructor, TA, or grader to correct the issue before saving.

Exception Flow:
Errors in grade updates trigger a correction prompt.

12.3 Give Grade Review
Actors:
Instructor

Description:
Describes the process for instructors to review and provide feedback on student submissions within UniHub, enhancing communication and learning.

Preconditions:

The instructor is logged into UniHub and has courses or assignments to review.

Basic Flow:
Log into UniHub and access the review section.
Select a course or assignment for review.
Choose a student to view their submission and any current grades or feedback.
Review the submission and enter grades and/or feedback.
Submit the review, updating the student's record.

Postconditions:

Student submissions have been reviewed, with grades and feedback recorded in UniHub.

Exception Flow:

Encounter errors during review, prompting for error resolution or support contact.

Each of these use cases follows a structured format that clearly outlines the actors involved, the steps required to complete the task, preconditions for the task, the basic flow of activities, any alternate flows for handling exceptions, and the outcomes of the use case, ensuring a standardized approach to managing grades and feedback within the university's system.

Use Case 13.1 Add Quiz Date

Actors:

-TA (Teaching Assistant)
-Instructor

Description:

This use case outlines how a Teaching Assistant or Instructor add a quiz event to the calendar of UniHub.

Preconditions:

-The TA is logged in to UniHub with valid credentials and has the necessary permissions to manage course schedules or create events.

Basic Flow:

1. Event Creation: The TA navigates to a section that allows them to create a new quiz event.
2. Event Details: The TA fills in the quiz details, including: course, topics, date, time, place. Also could include specific details about the quiz (e.g., format, duration, notification settings and ways, who should be notified).
3. Save and Confirmation: The TA reviews the event details and submits their creation.
4. System checks whether this TA is allowed to create a quiz event for this particular course and if yes, saves the data and creates the event and notifies related users if wanted.

Alternate Flow:

-Missing or Invalid Information: If the TA leaves required fields blank or enters invalid information (e.g., past date), the system displays an error message prompting them to correct the issue.
-If the TA lacks the necessary permissions to create events or manage course schedules, the system displays an error message explaining the restriction.

Postconditions:
-The newly created event appears on the UniHub calendar, notifying students and potentially other users about the upcoming quiz for a specific course.
-The event may be linked to the relevant course within the system, providing easier access to details or associated resources.

Exception Flow:
-- If technical issues prevent event creation or display, an error message is shown, and the TA or instructor is advised to retry or report the issue.

Use Case 13.2 Add Exam Date
Actors:
-TA (Teaching Assistant)
-Instructor

Description:
This use case outlines how a Teaching Assistant or Instructor add an exam event to the calendar of UniHub.

Preconditions:
-The TA is logged in to UniHub with valid credentials and has the necessary permissions to manage course schedules or create events.

Basic Flow:
1. Event Creation: The TA navigates to a section that allows them to create a new exam event.
2. Event Details: The TA fills in the exam details, including: course, topics, date, time, place. Also could include specific details about the exam (e.g., format, duration, notification settings and ways, who should be notified).
3. Save and Confirmation: The TA reviews the event details and submits their creation.

4. System checks whether this TA is allowed to create an exam event for this particular course and if yes, saves the data and creates the event and notifies related users if wanted.

Alternate Flow:

-If the TA or instructor leaves required fields blank or enters invalid information (e.g., past date, invalid location), the system displays an error message prompting them to correct the issue before saving.

Postconditions:

-The newly created exam event appears on the UniHub calendar, informing students and other relevant users about the upcoming exam for a specific course.

Exception Flow:

- If technical issues prevent event creation or display, an error message is shown, and the TA or instructor is advised to retry or report the issue.

Use Case 13.3 Book Office Hours

Actors:

-Student

Description:

This use case outlines the steps taken by a student to schedule an appointment for an instructor's office hours using the University Hub (UniHub).

Preconditions:

-The student is logged in to UniHub with valid credentials. The student has access to the office hours schedule for the instructor they wish to meet with. This information might be displayed on the course webpage, UniHub announcements, or directly provided by the instructor.

Basic Flow:

1. The student navigates to a section within UniHub that displays office hours schedules for instructors.
2. The student identifies and selects the specific instructor from the list whose office hours they want to book.
3. The system displays the available time slots for the chosen instructor's office hours. This might be presented using a calendar interface or a list of time blocks.

4. The student selects the desired time slot from the available options.
5. Depending on the system's configuration, the student may have the option to provide a brief description of the meeting purpose. This allows the instructor to prepare beforehand if needed.
6. The student reviews the selected time slot and meeting purpose (if provided) and confirms their booking request.

Alternate Flow:
- If the student attempts to submit a booking without selecting a time slot or fails to meet any other required information (e.g., meeting purpose, if applicable), the system displays an error message prompting them to complete the missing information.
-If all available time slots are already booked, the system informs the student and may provide options to join a waiting list or search for alternative office hours (with different instructors or times).
-If a student is trying to book an office hour from an Instructor who is not a lecturer of students current courses system may hold students to approve this appointment and ask instructors approval about the situation.

Postconditions:
-Upon successful booking, the system confirms the appointment and provides the student with details such as date, time, location (if applicable), and meeting link (if virtual). The chosen time slot is marked as booked in the instructor's office hours schedule, preventing someone else try to book the same slot.
-The student and the instructor may receive an email or notification within UniHub reminding them about the upcoming appointment based on notification choices of student and instructor.

Exception Flow:
- If technical issues prevent event creation or display, an error message is shown, and the TA or instructor is advised to retry or report the issue.

Use Case 13.4 Add Project Meeting
Actors:
-Student
-TA

Description:

This use case outlines how Students of a group or TA’s can create meeting events for semester projects.

Preconditions:

-The participant (student or TA) is logged in to UniHub.

-The participant has access to create events.

-The user is associated with the relevant project (e.g., enrolled in the course, assigned as TA).

Basic Flow:

1. The participant navigates to a section that allows them to create new events.
2. The participant fills in the event details, meeting aim:"Project Meeting" for the specific project (e.g., "Group X - Project Y Meeting").
3. Optionally include details about the meeting purpose, agenda, or specific topics to be discussed.
4. Selecting the date and time for the meeting, indicating the physical or online location for the meeting.
5. The participant may have the option to invite other participants to the meeting. This might include other group members, relevant TAs, or instructors.
6. The participant reviews the event details and submits their creation.

Postconditions:

-The newly created project meeting event is saved on the UniHub calendar, informing relevant users (including other project members, Ta’s) about the meeting.

-The event may be linked to the associated project within the system, allowing easy access to information and potential resources related to the meeting.

Exception Flow:

If technical issues prevent event creation or display, an error message is shown, and the participant is advised to retry or report the issue.

Use Case 13.5 Book an Organization Meeting

Actors:

-Student Clubs

Description:

This use case outlines how student clubs can add club events into UniHub calendar.

Preconditions:

-An authorized member of the student club is logged in to UniHub with valid credentials.

-The club is registered and recognized within the university system and has the necessary permissions to create events in UniHub.

-The authorized member has access to information about the meeting (date, time, purpose, target audience).

Basic Flow:

1. The authorized member navigates to a section within UniHub that allows creating new events.
2. The member fills in the event details, including: Title (e.g., "Club X Meeting - [Topic]") , providing details about the meeting agenda, activities, or any relevant information for potential attendees, selecting the date and time for the meeting, location: Indicating the physical or online location for the meeting. Target Audience: Depending on the system's configuration, the member might specify whether the event is open to all students, club members only, or a specific group within the university. Selecting the specific student club this event is associated with.
3. The member reviews the event details and confirms their creation.

Alternate Flow:

If the member lacks the necessary permissions to create events on behalf of the club, the system displays an error message explaining the restriction.

Postconditions:

-The newly created organization meeting event appears on the UniHub calendar, notifying relevant users.

Exception Flow:

If the club member encounters restrictions or errors, UniHub provides feedback on the necessary steps to resolve the issue.

Use Case 14.1 Create Question

Actors:

-Instructor

Description:

Create question use case demonstrates the steps taken by an instructor to create a new question to be added to an assessment or to the question database

Preconditions:

-The system is operational

-The instructor has logged into the system

-Database is operational

Basic Flow:

1. The instructor accesses the university system.
2. Instructor initiates the question creation process.
3. The instructor creates a question, chooses its category and difficulty level.
4. Instructor may choose to submit the question to either the question database or an actively preparing assessment.
5. The system updates the database when the question is submitted.

Alternate Flow:

4a. The instructor chooses to add the question to an actively preparing assessment test.

4b. The system updates the assessment test to include the new question.

4c. When the preparation of the assessment test is finished, the system updates the question database to include the new question.

Postconditions:

-The system keeps a record of the new question.

Exceptional Flow:

5a. If the system has a sort of technical problem submitting the new question, the system shows an error message.

5b. The instructor tries again submitting when the system is working again.

Use Case 14.2 Delete Question

Actors:

-Instructor

Description:

Delete question use case explains the steps taken when an instructor decides to delete an existing question from the system.

Preconditions:

-The system is operational

-The instructor has logged into the system

-Database is operational

-Question database has the question to be deleted.

Basic Flow:

1. The instructor accesses the university system.

2. Instructor enters the question database

3. The instructor finds the question they want to delete by either filtering or searching.

4. Instructor deletes the question from the database.

5. The system updates the database when the question is deleted.

Alternate Flow:

4a. The instructor decides to delete a question from an actively preparing assessment test.

4b. The instructor finds the question in the assessment test and deletes the question from the assessment test

4c. The system updates the assessment test.

Postconditions:

-The system keeps a record of the question deleted from the database.

Exceptional Flow:

5a. If the system has a sort of technical problem deleting the question from the database, the system shows an error message.

Use Case 15.1 Create New Assessment

Actors:

-Instructor

Description:

The steps taken by the instructor to create a new assessment from zero.

Preconditions:

-The system is operational

-The instructor has logged into the system

Basic Flow:

1. The instructor accesses the university system.

2. The instructor enters the assessment creation screen.

3. The instructor chooses the type of the assessment.

4. The instructor adds questions from the question database.

5. The system checks if the question is used in any of the past assessments and shows a warning message

6. When the instructor finishes adding questions, they submit the assessment to the system.

7. The system updates the database and saves the assessment created.

Alternate Flow:

4a. The instructor may choose to create a new question instead of choosing from the database.

4b. The instructor creates a new question and adds it to the assessment.

4c. When the assessment creation is finished, the instructor submits it to the system.

4d. The system updates the database and saves the assessment created.

Postconditions:

-The system holds the created assessment in the database for the later use.

Exceptional Flow:

7a. If the system has a sort of technical problem submitting the assessment to the database, the system shows an error message indicating that creating assessment was unsuccessful.

7b. The instructor tries submitting again when the system is working again.

Use Case 15.2 Assign Assessment to Students

Actors:

-Instructor

-Student

Description:

The steps taken by an instructor the assign the students with an assessment.

Preconditions:

-The system is operational

-The instructor has logged into the system

-There is an active assessment

Basic Flow:

1. The instructor accesses the university system.
2. Instructor enters the assessment assigning screen.
3. Instructor chooses which students/class will be assigned with the assessment.
4. Instructor chooses the assessment from the system/uploads an assessment document.
5. Instructor assigns the chosen class with the chosen assessment.
6. The system updates the students with the assigned assessment and sends a notification.

Alternate Flow: None

Postconditions:

-The system keeps the students assigned with the assessment until the deadline.

Exceptional Flow:

5a. If the system has a sort of technical problem assigning the assessment to the students, the system shows an error message.

Use Case 15.3 View Assessment Results

Actors:

-Instructor

Description:

View assessment results use case shows the step for instructor to take in order to see each student’s answers given on the assessment.

Preconditions:

-The system is operational

-The instructor has logged into the system

-The instructor has created and assigned an assessment to certain students/class

-The students took the test.

Basic Flow:

1. The instructor accesses the university system.
2. The instructor chooses an assessment that is assigned to certain students.
3. The instructor can see the answers of each student and their grade according to the solution key

Alternate Flow: None

Postconditions: None

Exceptional Flow: None

Use Case 15.4 Export Assessment

Actors:

-Instructor

Description:

Export assessment use case demonstrates the step for instructor to export an assessment from the system to their own device

Preconditions:

-The system is operational

-The instructor has logged into the system

-There is an assessment created in the system

Basic Flow:

1. The instructor accesses the university system

2. The instructor navigates to the assessment they want to export

3. The instructor chooses if they want to save the assessment to the device or export somewhere else (Google Drive etc.)

4. The instructor chooses the file type they want to export the assessment in

5. The system exports the assessment for the instructor

Alternate Flow:

3a. The instructor chooses to export it to somewhere else besides their device

3b. The instructor chooses the platform they want to export

3c. The instructor connects their account to the system.

3d. The system exports the assessment for the instructor

Postconditions: None

Exceptional Flow:

5a. If the system has a sort of technical problem or the instructor loses their internet connection during the exportation, the system shows an error message indicating the exportation process failed.

Use Case 15.5 Create Solution Key

Actors:

-Instructor

Description:

Create Solution Key use case indicates the steps the instructor takes in order to create a solution key for an existing assessment.

Preconditions:

-The system is operational

-The instructor has logged into the system

-The instructor has created an assessment and submitted it to the system.

Basic Flow:

1. The instructor accesses the university system
2. The instructor accesses the assessments they created
3. The instructor chooses the assessment they want to create a solution key for
4. The instructor enters the create solution key screen
5. The instructor enters the answer and solution for each question in the assessment
6. The instructor submits the solution key
7. The system updates the assessment to have the new created solution key

Alternate Flow: None

Postconditions:

-The system updates and keeps the track of the new created solution key

Exceptional Flow:

7a. If any technical issue caused by the system occurs or the instructor loses connection during the system update, the system shows an error message indicating that creating a solution key was unsuccessful.

Use Case 15.6 View Past Assessments

Actors:

-Instructor

-Officer

Description:

View Past Assessments indicates the steps needed to take to view the past assessments.

Preconditions:

-The system is operational

-The instructor has logged into the system

-Database is operational

-The system has assessments submitted by any of the instructors before

Basic Flow:

1. The instructor accesses the university system
2. The instructor accesses the assessment database
3. The instructor chooses the type of assessment and the year it was submitted to the system
4. The instructor finds the assessment and views it

Alternate Flow:

4a. The system does not have the specific assessment the instructor searching for or does not have any assessments meeting the specified criteria

4b. The system shows an error message indicating that assessment was not found.

Postconditions: None

Exceptional Flow: None

Use Case 15.7 Edit Access Permission

Actors:

-Instructor

-Student

Description:

Edit access permission use case describes the instructor’s steps to edit the access permission of students for a certain past assessment

Preconditions:

-The system is operational

-The instructor has logged into the system

-The database has past assessments stored

Basic Flow:

1. The instructor accesses the university system

2. The instructor views the past assessments database

3. The instructor chooses the assessment they want to edit permissions of

4. The instructor edits the permission of the assessment to be available for students to view

5. The system updates the permissions of the assessment

Alternate Flow:

4a. The instructor chooses an assessment that is available for students

4b. The instructor edits the permissions of the assessment to be hidden from the students

4c. The system updates the permissions of assessment.

Postconditions:

-The system updates the permissions of assessment and keeps the track of the update

-The students are notified about they are given permission to view the exam

Exceptional Flow:

5a. If the system has a sort of technical problem updating the permissions of the assessment, the system shows an error message.

Use Case 16.1 Search an Existing Question

Actors:

-Instructor

Description:

Search an Existing Question use case describes the steps the instructor should take in order to search the question database for a question

Preconditions:

-The system is operational

-The instructor has logged into the system

-The database is operational

-The database contains past questions

Basic Flow:

1. The instructor accesses the university system

2. The instructor enters the question database

3. The instructor either searches for a key word or applies filters suitable for their search

4. The system shows a list of question matching with the keyword or with the filters applied by the instructor

Alternate Flow:

4a. If the database has no question matching with the given keyword or filters applied, then shows an error message indicating that there are no questions in the database matching the search parameters

Postconditions: None

Exceptional Flow:

5a. If the system has a sort of technical problem searching for the questions or the instructor disconnects from the system during the search, the system shows an error message indicating that the searching process failed.

Use Case 16.2 See All Past Questions

Actors:

-Instructor

Description:

See All Past Question use case simply indicates the instructor’s access to the question database.

Preconditions:

-The system is operational

-The instructor has logged into the system

-The database is operational

-The database contains past questions

Basic Flow:

1. The instructor accesses the university system
2. The instructor navigates to the question database screen
3. Instead of searching for a certain question, the instructor chooses to list all past questions for a certain subject
4. The system lists and shows all the past questions about the certain subject

Alternate Flow: None

Postconditions: None

Exceptional Flow:

4a. If the database does not contain any past question about the subject, then the system shows an error message indicating that there are no questions to view.

Use Case 17.1 View System Reports

Actors:

-Admin

Description:

This use case outlines the steps taken by an administrator to wiev reports submitted by various users about bugs they have encountered while using the portal.

Preconditions:

-The admin is logged in to the system.

-The admin has the necessary permissions to access system reports.

Basic Flow:

1. The admin navigates to a section that allows access to system reports.
2. This section might be named "Reports," "System Health," or similar, depending on the specific system interface.
3. The system might offer various filtering and sorting options to help the admin find specific reports by offering filtering by category, date range, user type, priority.
4. Once the desired report(s) are selected, the admin can view details such as: Category: Identifying the type of issue reported, Date and time submitted: When the report was submitted. User who submitted: The user who encountered the bug. Description of the issue: A detailed explanation of the problem encountered by the user. Additional information, the report might also include screenshots, logs, or other relevant details.

Alternate Flow:

If there are no reports within the selected filters or timeframe, the system informs the admin of this and might offer options to expand the search criteria.

Postconditions:

-There is no action after this use case. Admin might analyse and modify source code in order to solve this problem.

Exceptional Flow:

None

Use Case 17.2 Categorize Reports

Actors:

-Admin

Description:

This use case outlines the steps taken by an admin on order to analyse and categorize submitted errors by users in order to classify issues.

Preconditions:

-The admin is logged in to the system with valid credentials.

-There are existing reports available in the system (refer to Use Case 18.1: View System Reports).

Basic Flow:

1. T1. The admin navigates to a section that allows access to system reports.

2. The admin identifies the specific report(s) they want to categorize. This might involve using the filtering and sorting options described in Use Case 18.1.

3. The system provides pre-defined categories to classifying reported issues (e.g., bug, security vulnerability, performance issue, user interface problem). The admin selects the most appropriate category that best reflects the issue.
4. Once the categorization is complete, the admin reviews and confirms their selection. The system updates the report with the assigned category and sub-category.

Alternate Flow:

3a. If the admin attempts to save the report without selecting a category or chooses an inappropriate one, the system might display a warning message prompting them to re-evaluate their selection.

Postconditions:

-The selected report(s) are categorized to achieve a better organization, analysis, and prioritization of reported issues.

-Categorized reports can be used to identify trends, common problems, and areas requiring focus for system improvements.

Exceptional Flow: None

Use Case 17.3 Dismiss Reports

Actors:

-Admin

Description:

This use case outlines the steps taken by admin to dismiss a report when the issue is solved or the report is not related to a issue it is based on users usage.

Preconditions:

-The admin is logged in to the system.

-There are existing reports available in the system, and some reports have been categorized (refer to Use Case 18.1: View System Reports and Use Case 18.2: Categorize Reports).

Basic Flow:

1. The admin navigates to a section that allows access to system reports.
2. The admin identifies the specific report(s) they want to dismiss.
3. This might involve using the filtering and sorting options (refer to Use Case 18.1) and considering the assigned categories (refer to Use Case 18.2).
4. The system asks the admin to select a reason for dismissing the report. This might involve pre-defined options like: Issue resolved: Indicating that the reported problem has been fixed through updates or troubleshooting, Not a valid issue: Clarifying that the report does not reflect an actual problem within the system and might be due to user error, misunderstanding, or outdated information, Optional Comment: The system might allow the admin to add an optional comment to provide further explanation or context for dismissing the report. This can be helpful for communication purposes, especially if the dismissal is due to "not a valid issue" and the user might need clarification or guidance.
5. The admin reviews the selected dismissal reason and optional comment (if provided) and confirms their decision. The system marks the report as dismissed and stores the dismissal reason and any comments for future reference.

Alternate Flow:

If the admin attempts to dismiss the report without selecting a reason, the system might display a prompt reminding them to choose a reason before proceeding.

Postconditions:

-The selected report are dismissed, indicating that the reported issue is no longer considered active or requiring further investigation.

- Dismissed reports may be archived or filtered out from the main view, but they should still be accessible for reference if needed.

Exceptional Flow: None

Use Case 18.1 Create User

Actors:

-Admin

Description:

This use case outlines the steps taken by an administrator to create new user accounts within the UniHub system.

Preconditions:

-The admin is logged in to the system.

Basic Flow:

1. 1. Admin navigates to a section within UniHub that allows user account management.
2. This section might be named "Users," "Accounts," or similar, depending on the specific system interface
3. The admin fills in the required information to create a new user account. This information typically includes: User ID: A unique identifier for the user within the system such as student id, instructor id etc based on the role of the new user. Username: The username chosen by the user for login purposes. (Username selection might be subject to specific rules or availability checks), Password: The user's password for secure access to the system, Full Name: The user's legal first and last name, Email Address: A valid email address for communication purposes, Role: Assigning a user role that determines their access privileges and functionalities within UniHub (e.g., student, faculty, staff, administrator)
4. Additional roles might be specific to different departments or functionalities.
5. The admin reviews the entered information and confirms its accuracy. Upon confirmation, the system creates the new user account and stores the information securely.

Alternate Flow:

2a. If the admin leaves required fields blank or enters invalid information, system displays an error message prompting the admin to correct the related issue.

3a. If the chosen username already exists in the system, the system prevents account creation and prompts the admin to choose a unique username.

Postconditions:

-A new user account is created in the UniHub system with the provided information.

-The system might generate a temporary password or require the user to set their password upon first login, depending on security protocols.

-The newly created user receives an email notification welcoming them to UniHub and providing them with login information and instructions to access their account.

Exceptional Flow: None

Use Case 18.2 Modify Authorizations

Actors:

-Admin

Description:

This use case outlines the steps taken by an administrator to modify the existing access privileges and functionalities of a user account within the University Hub (UniHub) system.

Preconditions:

-The admin is logged in to UniHub

-The administrator has identified the specific user whose roles need to be modified.

Basic Flow:

1. The admin navigates to section witch they can monitor and search by filtering all users registered in the system.
2. Admin identifies and selects the user account for which they want to modify roles.
3. The system displays the user's current role(s) and allows the admin to:
4. Assign new roles: Grant the user additional roles that expand their access and functionalities within portal, Remove existing roles: Revoke access to specific functionalities by removing roles from the user's account.
5. The admin reviews the changes made to the user's roles and confirms them.

6. The system updates the user's account accordingly.

Alternate Flow:

4a. If the selected user has a higher administrative level than the logged-in admin, the system displays an error message explaining that modifying their roles is not permitted.

Postconditions:

-The user's account is updated with the modified roles.

-The system might notify the affected user about the changes made to their account (depending on configuration of notifications).

Exceptional Flow: None

Use Case 18.3 Delete User

Actors:

-Admin

Description:

This use case outlines the steps taken by an administrator to delete a user account from the system.

Preconditions:

-The admin is logged in to system.

-The administrator has identified the specific user account that needs to be deleted.

Basic Flow:

1. The admin navigates to section witch they can monitor and search by filtering all users registered in the system.
2. The admin identifies and selects the user account they want to delete.

3. The system displays a clear confirmation message reminding the admin of the potential consequences of deleting a user account (e.g., loss of associated data, disruption of ongoing activities).
4. This message might also allow the admin to specify a reason for deletion if required.
5. Upon confirmation by the admin, the system permanently deletes the selected user account and its associated data (within the scope of the administrator's permissions and system policies).

Alternate Flow:

3a. If the admin chooses to cancel the deletion during the confirmation step, the process is aborted, and the user account remains intact.

Postconditions:

-The deleted user account is no longer accessible or within UniHub

Exceptional Flow:

If the user to be deleted has ongoing enrollment to courses, exams etc. Admin might be warned about these.

Use Case 19.1 Analyze Assessment

Actors:

-Officer

Description:

Analyze Assessment use case scenario describes the steps the officer should take in order to analyze a past assessment that has been taken by the students

Preconditions:

-The system is operational

-The instructor has logged into the system

-The database is operational

-The instructor assigned students with an assessment

-The students took the assessment test

Basic Flow:

1. The officer accesses the university system

2. The officer views the past assessments

3. The officer navigates the past assessment they want to analyze

4. Once they find the assessment and choose to analyze, the system creates statistical report based on the student's success on the questions (i.e. how many students answered a question correctly)

5. The officer can view the analysis, and save it to the system or their device

6. The system saves a copy of the created analysis in the database

Alternate Flow: None

Postconditions:

-The system keeps track of the analysis created

Exceptional Flow:

5a. If the system has a sort of technical problem analyzing the assessment or the instructor disconnects from the system during the analyze, the system shows an error message indicating that the analyzing process failed.

Use Case 19.2 Create Assessment Report

Actors:

-Officer

Description:

Create Assessment Report use case scenario demonstrates the step the officer needs to take in order to create an assessment report from the analyze of a past assessment.

Preconditions:

-The system is operational

-The instructor has logged into the system

-The database is operational

-The instructor assigned students with an assessment

-The students took the assessment test

-The officer created an analysis based off of a past assessment

Basic Flow:

1. The officer accesses the university system

2. The officer views the past assessments

3. The officer navigates the past assessment they want to analyze

4. Once they find the assessment and choose to analyze, the system creates statistical report based on the student's success on the questions (i.e. how many students answered a question correctly)

5. The officer can view the analyze, and save it to the system or their device

6. After the analyze is done, system shows a button for officer to "create an assessment report"

7. The officer chooses to create an assessment report from the analysis of the past assessment

8. The system creates a brief analysis about how students perform on each question and the assessment test overall.

9. The officer can edit the report if they wish to do so

10. The officer submits the report to the system when they finish editing the report.

11. The system updates the database to hold the report created by the officer

12. The officer can export the report if they wish to do so

Alternate Flow: None

Postconditions:

-The system keeps track of the analysis created

-The system keeps the report created in the database

Exceptional Flow:

11a. If the system has a sort of technical problem creating the assessment report the system shows an error message indicating that the report creation process failed.