Project Group4 Task2 - Requirements Analysis

User Classes and their use cases

Anonymous User
An anonymous user is an individual who has not presented the system with credentials. They are only able to access the login screen.
UC – AU – 1: Login

Logged-in Student
A student is any individual who has an account. They can create personal challenges, receive and accept event invitations, and use the instant messaging system.
UC – S – 1 : Create a challenge
UC – S – 2 : Receive and accept event invites
UC – S – 3 : Access and use instant messaging
UC – S – 4 : Create a group chat
UC – S – 5 : Search for other users
UC – S – 6 : Set event reminder time
UC – S – 7 : Submit an assignment
UC – S – 8 : Download a lecture
UC – S – 9 : Lend/borrow an item
UC – S – 10: Form a group calendar

Logged-in Coordinator User
A Coordinator User has the same privileges as a student, in addition to access to other systems, such as event creation.
UC – CU - 1: Create an event
UC – CU - 2: Delete an event

Operating Environment

Student life shall be compatible with all standard smartphones (Apple, Samsung, Google, Huawei and LG). The system shall be supported on devices running iOS 11 or Android OS 9 or later.

Design and Implementation Constraints

CO-1: Commercially available systems shall be used for the messaging and calendar features of the system, limiting the amount of unique code that needs to be created.

CO-2: The commercially available systems must be able to integrate with existing university systems.

User Documentation

The developer shall provide the necessary user documentation on how to use the system, system features and solutions to common problems for users through the Help menu tab.

Assumptions and Dependencies

Multiple functions of the system are dependent upon the integration of other software systems such as the calendar system and university assignment submission system.

Functional Requirements

Use cases for Anonymous Users

UC – AU – 1: Login

Description: An anonymous user will be able to login to the client.

System.login - (Priority=H)

Upon accessing the client, the system shall prompt the anonymous user for login credentials. Prior to the login, the anonymous user shall have no access to the rest of the system.

Use cases for Logged-in Students

UC – S – 1: Create a challenge

Description: A logged-in student shall be able to create a challenge for themselves from the system.

Challenge.create - (Priority=L)

When prompted, the system shall allow the logged-in student to create challenges using a variety of templates and customization tools.

Challenge.info: - (Priority=L)

During an active challenge, the system shall provide detailed information about the progress of the challenge from the challenge's page.

UC – S – 2: Receive and accept event invites

Description: A logged-in student shall be able to receive and accept event invites from other users.

Event.invite.notify: - (Priority=H)

The user shall receive a notification to join an event which they have been invited to. The user shall have the option to request more information, and to accept or decline.

Event.info: - (Priority=M)

The event invitation shall provide a link to a page with the event information. This should include information about other staff members, the time and date, the coordinator, and general information about the event.

Event.accept: - (Priority=H)

If the user selects the accept option, the system shall create an entry in their calendar, and add the user to the allocated group chat, giving access to further event information.

Event.decline - (Priority = H)

If the user selects the decline option, the system shall delete the invite.

UC – S – 3: Access and use instant messaging
Description: A logged-in student shall be able to access and use the instant messaging system.

User.message: - (Priority=L)
While logged in, the user shall be able to message other users.

UC – S – 4: Create a group chat
Description: A logged-in student shall be able to create a group chat with other users.

Group.create: - (Priority=H)
The system shall allow users to create group chats with class groups, clubs, teams, and fellow peers.

UC – S – 5: Search for other users
Description: A logged-in student shall be able to search for other users.

User.search: - (Priority=M)
The system shall allow users to search for other students and staff using their full names or ID numbers. If the other student or staff member is not found, the system should display "User not found".

User.info: - (Priority=L)
When a user is selected the system shall display the profile of the chosen user in a popup tab. The profile shall show the user's profile picture, current degree and year of study, and any clubs/teams they are a part of.

UC – S – 6: Set event reminder time
Description: A logged-in student shall be able to set a reminder for an upcoming event.

Event.set.reminder: - (Priority=L)
When creating an event, the system should allow the user to select when they would like a reminder for an upcoming event. The reminder time should be able to be set to up to 24 hours before an event begins.

Event.notify: - (Priority=L)
If a reminder time has been set, the system shall notify the user via a device notification at the set time of the reminder.

UC – S – 7: Submit an assignment
Description: A logged-in student shall be able to submit an assignment to the university from the system.

Assignment.info: - (Priority=M)
If posted, the system shall provide a link to the assignment page with the assignment information, from the assignments list page.

Assignment.submit: - (Priority=M)
The system shall provide a page that allows users within the assignment's course to submit files for the assignment. The maximum file size and deadline for submissions will be set by the course coordinator.

UC – S – 8: Download a lecture
Description: A logged-in student shall be able to download available lectures from their classes.

Lecture.download - (Priority=M)
If the lecture video has been uploaded and the user's device has more than 1 GB of available storage, the system shall allow the user to download lectures onto their device.

Lecture.info: - (Priority=M)
When the download has been prompted, the system shall display information about the lecture. This should include the file size, file name and the date the lecture was recorded.

UC – S – 9: Lend/borrow an item
Description: A logged-in student shall be able to lend and borrow items from other users.

Item.post: - (Priority=L)
The system shall allow users to publicly post any items they are willing to lend to other students.

Item.request: - (Priority=L)
The system shall allow users to publicly request any items they want to borrow.

Item.borrow: - (Priority=L)
The borrower should receive a request from the lender to borrow their item. If the borrower accepts the request, the item should be listed as "borrowed" in the borrower's "items" view and "lent" in the lender's "items" view.

Item.lend: - (Priority=L)
The lender shall have the ability to select another student (the borrower) to lend an item to. The borrower will receive a request to have the item be lent to them. The lent/borrowed status shall be visible to both parties in the transaction for the duration of the lending/borrowing. The lending/borrowing tracking shall be ended when the item is manually marked as returned by the lender.

UC – S – 10: Form a group calendar
Description: A logged-in student shall be able to form a group calendar that tracks the events of the users in the group.

Group.calendar.create: - (Priority=M)
If a user is in a group, the system shall provide an option for a group calendar to be created in which students from that group can add events to.

Group.calendar.info: - (Priority=M)
If a group calendar has been created, students in the group shall be able to see the information that the calendar contains.

Use cases for Logged-in Coordinator Users

UC – CU – 1: Create an event

Description: A logged-in coordinator user shall be able to create an event, set the details of the event and invite others to it.

Event.create: - (Priority=H)
The system shall provide an option that coordinator users can access in which they create an event by inputting the required details of the event.

Event.invite: - (Priority=H)
The system shall provide an option on the event coordinator's event page that allows them to invite other users by searching for them.

UC – CU – 2: Delete an event

Description: A logged-in coordinator user can delete events which they have created. This will remove the event from the system, and notify all other staff members who accepted the invitation of the deletion.

Event.delete - (Priority=H)
The system shall provide an option for the creator of the event to remove it from the system. Prior to deletion, a second prompt shall ask for confirmation of the deletion. If the confirmation is accepted, the system will purge the event from the Coordinator User's account, and from the staff members' accounts. The system will also delete any pending invites for the deleted event.