**CENG 3545: Mobile Application Development**

|  |  |
| --- | --- |
| **Project Name** | Roxid |
| **Team Members and Emails** | Ozan USLAN: uslanozan@gmail.com  Rıdvan Barış ÖZDEN:barkarabulut123@gmail.com |
| **Project Topic** | A Multi-Purpose Platform for Group Communication and Collaboration |
| **Project Description** | This project involves the development of a real-time communication platform designed to connect communities and facilitate seamless collaboration. Inspired by a popular platform like Discord, the app enables users to create and join chat rooms, engage in group conversations, and communicate via text and voice. |

**Scenario:** Sarah, a university student, is part of a study group working on a team project. Due to conflicting schedules, they need a way to communicate and collaborate virtually.

#### **1. Organizing the Study Group**

Sarah creates a *club* called "Project Group" and invites her teammates. They create *text channels* for different topics they also create a *voice channel* for discussing:

**Text Channels:**

* **Research and Resources**
* **Presentation Slides**
* **Final Report**

**Voice Channel:**

* **Discussing**

Each member joins the relevant channels and contributes without cluttering the conversation.

#### **2. Real-Time Messaging and File Sharing**

* Sarah shares links to research articles in the "Research and Resources" channel.
* John uploads a draft of the PowerPoint in the "Presentation Slides" channel for feedback.
* The team discusses their progress in real-time through text chat.

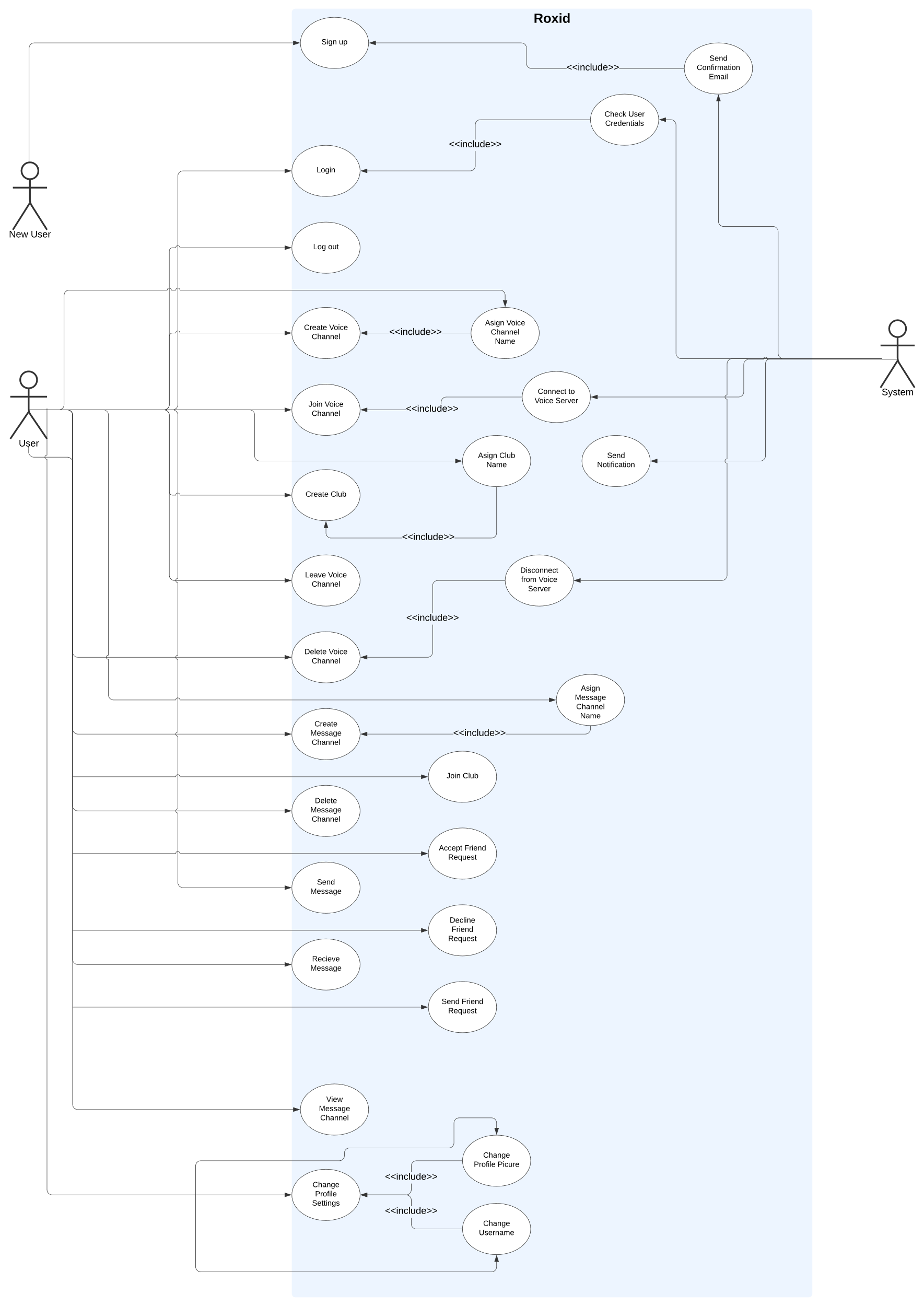
#### **3. Voice Collaboration**

Needing quick feedback on the report, Sarah and group enter the discussing voice channel, and the group discusses their edits live.

#### **4. Staying Updated**

The app sends push notifications when new messages or files are added, keeping everyone informed.

Use Case Diagram



|  |  |
| --- | --- |
| **Use Case Name** | Sign Up |
| **Participating Actors** | New User |
| **Flow of Events** | * A user navigates to the registration page. * The user selects the “Register” option. * The system presents a registration form. * The user enters required information, such as name, email address and password. * The user submits the registration form. * The system validates the provided information. * The system creates a new user account. * The system sends a verification email to the user’s provided email address. * The user clicks the verification link in the email. * The system verifies the email and activates the user account. * The system notifies the user of successful registration. * The user is now able to log in with the new account. |

|  |  |
| --- | --- |
| **Use Case Name** | Login |
| **Participating Actors** | User |
| **Flow of Events** | * The user navigates to the login page. * The user enters their registered email address or username and password. * The system validates the input:   + If the credentials are correct, proceed to next step.   + If the credentials are incorrect, display an error message and allow the user to re-enter the information. * The system verifies the user’s identity and session:   + If successful, create a secure session for the user.   + If unsuccessful (e.g., inactive account, incorrect password), display an appropriate error message. * The user gains access to the system and is redirected to the dashboard or a designated landing page. |

|  |  |
| --- | --- |
| **Use Case Name** | Logout |
| **Participating Actors** | User |
| **Flow of Events** | * The user selects the profile icon. * The user selects logout button. * System backs to login page. |

|  |  |
| --- | --- |
| **Use Case Name** | Change Profile Settings |
| **Participating Actors** | User |
| **Flow of Events** | * The user selects the profile icon. * System shows profile details (e.g., profile picture, nickname, background picture) * The user enters a new nickname and save. * The user select new background picture and profile picture. * The system checks format file extension:   + If the extension is valid profile picture is changed.   + If not valid, go back to previous step. |

|  |  |
| --- | --- |
| **Use Case Name** | Send Friend Request |
| **Participating Actors** | Users |
| **Flow of Events** | * The users selects friends icon. * The system shows friends and friend requests. * The user selects one of friends. * The system shows chat history. * The user selects add button and enter friend’s username:   + The system shows if user there is a matching username.   + If no matching, system show message. * A notification appears to other user:   + If user accepts friend request, they’re being friends and they see can each other’s icons.   + If user don’t accept request, nothing will happen. |

|  |  |
| --- | --- |
| **Use Case Name** | Create Club |
| **Participating Actors** | User |
| **Flow of Events** | * User selects the “+” button on the main menu. * 2 options pop up for the user to choose. * User chooses the “create club” button. * System prompts the user to enter a club name.   + If the club name is less than 50 characters the user creates the club.   + If the club name is more than 50 characters systems show an error message that the name should be less than or equal to 50 characters. |

|  |  |
| --- | --- |
| **Use Case Name** | Join Club |
| **Participating Actors** | User |
| **Flow of Events** | * User selects the “+” button on the main menu. * 2 options pop up for the user to choose. * User chooses the “join club” option. * System prompt user to enter a url of the club. * User enters the url of the club.   + If the URL is valid, the user joins the club.   + If the URL is not valid, the system shows an error message. |

|  |  |
| --- | --- |
| **Use Case Name** | Create Message Channel |
| **Participating Actors** | User |
| **Flow of Events** | * User chooses a club to enter at the main menu. * In the club screen the user chooses the “message channels” button. * On the message channels menu the user clicks the “+” button. * System prompts the user to enter the name of the message channel.   + If the message channel name is less than 50 characters the user creates the message channel.   + If the message channel name is more than 50 characters systems show an error message that the name should be less than or equal to 50 characters. |

|  |  |
| --- | --- |
| **Use Case Name** | Delete Message Channel |
| **Participating Actors** | User |
| **Flow of Events** | * User chooses a club to enter at the main menu. * In the club screen the user chooses the “message channels” button. * The user selects the settings icon near the message channel the user wants to delete. * The user selects the delete button. * System asks to confirm.   + If the user selects confirm, the message channel is deleted.   + If the user selects refuse the message channel is not deleted. |

|  |  |
| --- | --- |
| **Use Case Name** | Create Voice Channel |
| **Participating Actors** | User |
| **Flow of Events** | * User chooses a club to enter at the main menu. * In the club screen the user chooses the “voice channels” button. * On the message channels menu the user clicks the “+” button. * System prompts the user to enter the name of the voice channel.   + If the voice channel name is less than 50 characters the user creates the voice channel.   + If the voice channel name is more than 50 characters systems show an error message that the name should be less than or equal to 50 characters. |

|  |  |
| --- | --- |
| **Use Case Name** | Delete Voice Channel |
| **Participating Actors** | User |
| **Flow of Events** | * User chooses a club to enter at the main menu. * In the club screen the user chooses the “voice channels” button. * The user selects the settings icon near the voice channel the user wants to delete. * The user selects the delete button. * System asks to confirm.   + If the user selects confirm the voice channel is deleted.   + If the user selects refuse the voice channel is not deleted. |

|  |  |
| --- | --- |
| **Use Case Name** | Join Voice Channel |
| **Participating Actors** | User |
| **Flow of Events** | * User chooses a club to enter at the main menu. * In the club screen the user chooses the “voice channels” button. * The user selects the voice channel it wants to join. * The system connects the user to a voice server. * The user has joined a voice channel and can talk with any other person that also joined or joins the voice channel. |

|  |  |
| --- | --- |
| **Use Case Name** | Leave Voice Channel |
| **Participating Actors** | User |
| **Flow of Events** | * User clicks the disconnect button. * The system severs the connection to the voice channel the user is in. * The user is redirected to the “voice channels” screen. |

|  |  |
| --- | --- |
| **Use Case Name** | Send Message |
| **Participating Actors** | User |
| **Flow of Events** | * User chooses a club to enter at the main menu. * In the club screen the user clicks the “message channels” button. * The user selects the message channel it wants to join. * The user enters a message into a text bar. * The user clicks the send button. |

|  |  |
| --- | --- |
| **Use Case Name** | Receive Message |
| **Participating Actors** | User 1,User 2 |
| **Flow of Events** | * User 1 sends a message to User 2. * System delivers the message to User 2. * User 2 gets a notification about the message. * User 2 clicks the friends button on the main menu. * User 2 clicks the User 1’s icon on the friends menu. * User 2 reads the message User 1 sent. |

|  |  |
| --- | --- |
| **Use Case Name** | View Message History |
| **Participating Actors** | User |
| **Flow of Events** | * User chooses a club to enter at the main menu. * In the club screen the user clicks the “message channels” button. * User clicks the message channel they want to view the history of. * System loads all the messages that are sent in that channel. |

|  |  |
| --- | --- |
| **Use Case Name** | Change Profile Picture |
| **Participating Actors** | User |
| **Flow of Events** | * User chooses a photo they want to change to. * System checks the file format of the photo.   + if it is jpeg or png , the system sets the new profile photo.   + if it is a other file format than jpeg or png the system shows an error message. |

|  |  |
| --- | --- |
| **Use Case Name** | Change Username |
| **Participating Actors** | User |
| **Flow of Events** | * User writes the username they want to change to. * User clicks the confirm button. * System checks the length of the name:   + if it is longer than 50 characters the system shows an error message.   + if it is less than 50 characters the username is changed. |

|  |  |
| --- | --- |
| **Use Case Name** | Send Confirmation |
| **Participating Actors** | System |
| **Flow of Events** | * System generates a confirmation token. * System constructs a confirmation email. * System sends a confirmation email to the email address the new user has entered. |

|  |  |
| --- | --- |
| **Use Case Name** | Check User Credentials |
| **Participating Actors** | System |
| **Flow of Events** | * User enters their username and password. * System receives the credentials. * System checks if the username exists.   + If the username is not found,the system shows an error message. * System retrieves the stored password for the corresponding username. * System compares the entered password with the stored password.   + If the password does not match,the system shows an error message. * System checks the credentials that are entered by the user:   + if the credentials are true the user is granted access to the system.   + if the credentials are false the user is not granted access to the system |

|  |  |
| --- | --- |
| **Use Case Name** | Assign Voice Channel Name |
| **Participating Actors** | User |
| **Flow of Events** | * User enters a voice channel name. * User clicks the confirm button:   + if it is longer than 50 characters the system shows an error message.   + if it is less than 50 characters the voice channel name is assigned. * User is redirected to the voice channels screen. |

|  |  |
| --- | --- |
| **Use Case Name** | Connect to Voice Server |
| **Participating Actors** | System |
| **Flow of Events** | * User clicks a voice channel they want to join. * System retrieves connection details for the selected voice server. * System attempts to establish a connection to the voice server. * System checks network availability and connection stability.   + If the network is unavailable, the system returns an error. * System verifies the connection with the voice server.   + If the connection fails, the system shows an error. * System establishes the voice communication session and sets up audio streams. |

|  |  |
| --- | --- |
| **Use Case Name** | Send Notification |
| **Participating Actors** | System |
| **Flow of Events** | * System detects a notification trigger. * System checks who triggered it. * System generates a notification. * System sends the notification to the user. * User receives the notification. |

|  |  |
| --- | --- |
| **Use Case Name** | Assign Club Name |
| **Participating Actors** | User |
| **Flow of Events** | * User enters a club name. * User clicks the confirm button:   + if it is longer than 50 characters the system shows an error message.   + if it is less than 50 characters the club name is assigned. |

|  |  |
| --- | --- |
| **Use Case Name** | Disconnect From Voice Server |
| **Participating Actors** | System |
| **Flow of Events** | * User clicks the disconnect button. * System gets the voice channel the user is in. * System severs the connection to that voice server. |

|  |  |
| --- | --- |
| **Use Case Name** | Assign Message Channel Name |
| **Participating Actors** | User |
| **Flow of Events** | * User enters a message channel name. * User clicks the confirm button.   + if it is longer than 50 characters the system shows an error message.   + if it is less than 50 characters the message channel name is assigned. |

|  |  |
| --- | --- |
| **Use Case Name** | Accept Friend Request |
| **Participating Actors** | User |
| **Flow of Events** | * User clicks the friends button on the main menu. * In the friends menu the user clicks the friend requests button. * On the friend requests menu the user clicks the accept button near the name of the user who sent the friend request. * Person who sent the friend request is added to the friends list. |

|  |  |
| --- | --- |
| **Use Case Name** | Decline Friend Request |
| **Participating Actors** | User |
| **Flow of Events** | * User clicks the friends button on the main menu. * In the friends menu the user clicks the friend requests button. * On the friend requests menu the user clicks the decline button near the name of the user who sent the friend request. * Person who sent the friend request is notified. |