

## Use Case Diagram for the Use Case Scenarios:



## Use Case 1: Logging into an Account

### **Iteration: 1**

**Primary Actor:** A user who owns an already registered account and wants to log into the multiplayer game platform

**Goal in context:** The user should be able to enter a username and password of the profile into a “password” and “username” text-field. If there is an account with a matching username and password saved in a database linked to the multiplayer game platform then upon clicking the log in button it will successfully log into the account.

**Preconditions:** There must already be a registered account with a matching username and password entered the database. The user attempting to log in must know the username and password of the account they are attempting to log into. The user must not be currently logged into an account and will have to log out of the account they are currently in if they wish to log into a different account.

**Trigger:** The user presses the “log in” button below the “username” and “password” text-field while there is a valid username and password entered in the “username” and “password” text-field. If there is an account within the database with that corresponding username and password that matches what was entered, then the platform will log the user into that account and follow it with a “welcome (name of account)” message.

### **Scenario:**

- User opens the multiplayer platform and goes to the log in screen
- The user will enter a username and password into the “username” and “password” text-field
- The user will click a button that says “log in” once the username and password has been entered
  - o If there is a valid account, there will be a “welcome (name of account)” message displayed to the user, and they will be logged in

- If there is not a valid account, then it will give an error message depending on the error that occurred
- Once logged in, the user can access the other features of the multiplayer platform that is available to all users

### **Exceptions:**

- **The username entered is incorrect:** Upon clicking the “log in” button there will be an error message displayed which states “this is not a valid account you are trying to log into”. No feedback will be given about the password entered.
- **The username entered is correct, but the password is incorrect:** Upon clicking the “log in” button there will be an error message displayed which states “this is not a valid account you are trying to log into”. No feedback will be given about the password entered.

**Priority:** This is a high priority feature as logging into an account is essential to being able to access the multiplayer platform for security and usability reasons.

**When available:** This feature is automatically entered into once the multiplayer platform is opened.

**Frequency of use:** This will occur whenever the multiplayer platform is opened and there is no account currently logged in.

**Channel to actor:** There will be various feedback messages shown to the user depending on the action that was performed. These will be shown within the multiplayer platform window.

**Secondary actors:** External database storing all the account information

**Channel to secondary actors:** An API built into the platform that allows the log in method to communicate with the external database allowing it to request and update the required information

### **Open issues:**

- Should there be a “remember user” feature implemented within the platform where once a user is logged in and the platform is closed the same user will be logged in whenever the platform is reopened?
- If a user was successfully logged into the platform, should it remember the account that was logged in and only require a password to relog into that specific account if that user is currently logged out?
- Should the platform allow the same account to be logged in on multiple devices at the same time?

### Use Case 2: Creating a New Account

#### **Iteration: 1**

**Primary Actor:** A brand new user who would like to register a new account on the multiplayer game platform

**Goal in context:** The user should be able to click a “register new account” button when they open the app for the first time. When there they will encounter a prompt that says, “enter a new username” and “enter a new password” with text-fields below them. Once a new username has been entered, the system will compare it with the database of existing accounts to see if the username is already taken. Once a new password has been entered, the system will check it meets all the requirements for a valid password (certain number of letters, must contain a symbol, must contain a number etc.). When a valid username and password has been entered, the user can click a “confirm” button and the account will be registered and added to the database of existing accounts.

**Preconditions:** There must be no account currently logged into the system to make a new account. The username entered by the user must not have been taken by another valid account beforehand. The password entered must also be a valid password that follows the requirements set beforehand (certain number of characters, must contain a symbol, must contain a number etc.).

**Trigger:** The user presses a “register new account” button and enters the required information into the text-fields depending on the prompt provided. Once all the text-fields have been filled and are valid options the user can click a “confirm” button and they are now a registered account and the system will follow it up with a “welcome (name of account)” message.

**Scenario:**

- User opens the multiplayer platform and goes to the log in screen
- The user does not have an existing account so instead of attempting to log in they click the “register new user” button
- The user is provided 2 text-fields with a prompt that states “enter a new username” and “enter a new password”
- The user will enter a username into the “username” text-field
  - o If the username entered is available (not matched with an already existing account) a green checkmark will display next to the text-field
  - o If the username is taken (there is an already existing account with that username) a red “x” symbol will display next to the text-field
- The user will enter a password into the “password” text-field
  - o If the password entered matches all the requirements for a valid password a green checkmark will display next to the text-field
  - o If the password does not match the requirements for a valid password a red “x” symbol will display next to the text-field and below the text-field it will state which requirement it does not meet
- Once the “username” and “password” are valid they can click a “confirm” button that will register the new account
- There will be a “welcome (name of account)” message displayed to the user, and they will be logged into their account
- Once logged in, the user can access the other features of the multiplayer platform that is available to all users

**Exceptions:**

- **The username entered is unavailable:** Once something has been entered into the text-field, a red “x” symbol will display next to the text-field and it will not go away until what was entered does not match the username of an existing account in the database
- **The password entered does not match the requirements:** Once something has been entered into the text-field, a red “x” symbol will display next to the text-

field and it will not go away until all of the requirements for a valid password have been met. Below the text-field there will be a list of requirements for a valid password. These will be red in colour if they are not met and green if they are met.

**Priority:** This is a high priority feature as everyone trying to access the multiplayer platform will have to register an account at some point to use it.

**When available:** This feature is automatically entered into once the multiplayer platform is opened.

**Frequency of use:** This will occur whenever the “register new account” button is clicked.

**Channel to actor:** The screen in the multiplayer platform window will change if the “register new account” button is clicked. There will be various messages and symbols of the colours green and red to denote to the user whether something is valid or invalid. A feedback message will be displayed to the user once a new account has been registered.

**Secondary actors:** External database storing all the account information

**Channel to secondary actors:** An API built into the platform that allows the create new account method to communicate with the external database allowing it to request and update the required information

**Open issues:**

- Should there be suggested usernames that are similar to the username entered but are currently not taken by another account shown to the user?
- Should there be a “password strength” bar shown to the user to give them a rough idea of how strong their password is?

## Use Case 3: Changing an Account Username

### **Iteration: 1**

**Primary Actor:** An existing user who wants to change the username of the account they have already registered on the multiplayer game platform

**Goal in context:** The user should be able to go to their profile settings after logging in and click a setting that says, “change your username”. If it is clicked a prompt will appear that says “enter your new username” and an empty text-field will appear. Once a text has been entered into the text-field the database will check if the username is available. If the username is available, the user can proceed to the next step and enter their current password in the text-fields below for verification purposes. The user can then click a “confirm” button to update the username of the account to the new one that was just entered.

**Preconditions:** There must already be a registered account on the system and the account must be currently logged into the multiplayer platform. The new username that the user is attempting to change to must be available and not currently taken by an existing account.

**Trigger:** The user enters their account settings and presses a “change your username” button and then enters the required information into the text-fields depending on the prompt provided. Once all the text-fields have been filled and are valid options the user can click a “confirm” button and they have now updated their username. A feedback message will show to the user indicating to them that their username has been updated upon exiting.

### **Scenario:**

- User clicks the “profile settings” and the “change username” option
- The user is provided 3 text-fields with prompts that state “enter a new username”, “enter your password” and “confirm password”
- The user will enter a new username into the “username” text-field

- If the username entered is available (not matched with an already existing account) a green checkmark will display next to the text-field
- If the username is taken (there is an already existing account with that username) a red “x” symbol will display next to the text-field
- The user will enter their account password into the “password” text-fields
  - If both passwords entered match the password associated with the account then upon clicking the “confirm” button the changing of the username will proceed
  - If either password entered does not match the password associated with the account then upon clicking the “confirm” button the changing of the username will not proceed and the system will give a feedback message that “the password entered is incorrect”
- Once the new “username” is entered and the “password” is a match the user can click a “confirm” button that will update the old username to the new username
- There will be a “your username has been updated” message displayed to the user, and they will be sent back to the profile settings screen

### Exceptions:

- **The username entered is unavailable:** Once something has been entered into the text-field, a red “x” symbol will display next to the text-field and it will not go away until what was entered does not match the username of an existing account in the database
- **The password entered does not match the saved one:** Once something has been entered into the text-field and the “confirm” button has been pressed, if the password entered does not match the saved password they will be sent back to the “change username” screen with a feedback message saying “the password entered is incorrect”

**Priority:** This should not be as high a priority as some features but would be a nice quality of life feature to implement.

**When available:** This feature becomes available when the profile settings are opened.

**Frequency of use:** This will occur whenever the “change username” option is selected in the profile settings of an account



**Channel to actor:** The screen in the multiplayer platform window will change if the “change username” button is clicked. There will be various prompts as well as feedback messages to denote to the user whether something is a valid username or not. A feedback message will be displayed to the user once the username has been updated.

**Secondary actors:** External database storing all the account information

**Channel to secondary actors:** An API built into the platform that allows the change username method to communicate with the external database allowing it to request and update the required information

**Open issues:**

- Should there be a limit to the number of times a username can be changed in a certain period?
- Should there be a cooldown that makes it so the user is unable to change their username for a certain period of time once the username has been updated?
- Should the user be notified that their username has been changed if there is a linked phone number or e-mail?

Use Case 4: Changing an Account Password

**Primary Actor:** An existing user who wants to change the password of the account they have already registered on the multiplayer game platform

**Goal in context:** The user should be able to go to their profile settings after logging in and click a setting that says, “change your password”. If it is clicked a prompt will appear that says “enter your new password” and an empty text-field will appear. Once text has been entered into the text-field the system will check if the text entered meets all the requirements for a valid password. If the check passes the user can then enter their current password twice in the corresponding fields for security purposes. Once all the fields have been filled the user can then click a “confirm” button to update the password of the account to the new one that was just entered.

**Preconditions:** There must already be a registered account on the system and the account must be currently logged into the multiplayer platform. The new password that the user is attempting to change must meet the requirements for a valid password. The user must know their current password to type it into the required fields.

**Trigger:** The user enters their account settings and presses a “change your password” button and then enters the required information into the text-fields depending on the prompt provided. Once all the text-fields have been filled and are valid options the user can click a “confirm” button and they have now updated their password. A feedback message will show to the user indicating to them that their password has been updated upon exiting.

**Scenario:**

- User clicks the “profile settings” and the “change password” option
- The user is provided 3 text-fields with prompts that state “enter your old password”, “enter a new password” and “confirm new password”
- The user will enter the old password into the “old password” text-field
  - If the password matches, then there is no error when the “confirm” button is pressed
  - If the password does not match, then an error prompt will be shown to the user upon clicking the “confirm” button
- The user will enter their new password into the “new password” text-fields
  - If the password entered matches all the requirements for a valid password a green checkmark will display next to the text-field
  - If the password does not match the requirements for a valid password a red “x” symbol will display next to the text-field and below the text-field, it will state which requirement it does not meet
  - If both “new password” text-fields do not match, then an error prompt will be shown to the user upon clicking the “confirm” button
- Once all the fields have been filled the user can click a “confirm” button that will update the old password to the new password
- There will be a “your password has been updated” message displayed to the user, and they will be sent back to the profile settings screen

**Exceptions:**

- **The old password entered does not match the saved one:** Once something has been entered into the text-field and the “confirm” button has been pressed, if the password entered does not match the saved password they will be sent back to the “change password” screen with a feedback message saying “the password entered is incorrect”
- **The new password does not meet the requirements for a password:** Once something has been entered into the text-field, a red “x” symbol will display next to the text-field and it will not go away until all of the requirements for a valid password have been met. Below the text-field there will be a list of requirements for a valid password. These will be red in colour if they are not met and green if they are met.
- **The “confirm new password” field does not match what was entered in the “new password” field:** Once both text-fields have been filled and the “confirm” button has been pressed, if the fields entered do not match each other they will be sent back to the “change password” screen with a feedback message saying “the new password entered does not match”

**Priority:** This should be a high priority feature as account security should be a focus to be able to run a successful multiplayer platform

**When available:** This feature becomes available when the profile settings is opened.

**Frequency of use:** This will occur whenever the “change password” option is selected in the profile settings of an account

**Channel to actor:** The screen in the multiplayer platform window will change if the “change password” button is clicked. There will be various prompts such as the usage of green/red text as well as feedback messages to denote to the user whether something is a valid password or not. A feedback message will be displayed to the user once the password has been updated.

**Secondary actors:** External database storing all the account information

**Channel to secondary actors:** An API built into the platform that allows the change

password method to communicate with the external database allowing it to request and update the required information

**Open issues:**

- Should the user be notified that their password has been changed if there is a linked phone number or e-mail?
- If the user enters a password they have used previously, should it remain a valid password?
- If the password has not been changed for a long period of time, should the user be prompted to update their password?

Use Case 5: Setting an Account Profile Picture

**Iteration: 1**

**Primary Actor:** An existing user who wants to change the profile picture of the account they have already registered on the multiplayer game platform

**Goal in context:** The user should be able to go to their profile settings after logging in and click a setting that says, “change your profile picture”. If it is clicked the file system will open and the user can select an image in a valid format (.jpeg or .png). Once selected and clicking a “confirm” button, the profile picture of the account will be updated to that image. The user can then click a “save” button to save the picture as their new account profile picture.

**Preconditions:** There must already be a registered account on the system and the account must be currently logged into the multiplayer platform. There must be a saved .png or .jpeg image already saved on the device that the profile picture can be updated to.

**Trigger:** The user enters their account settings and presses a “change your profile picture” button and then selects the image they would like their profile picture to be.

Once the image has been selected the user can click a “confirm” button and they have now updated their profile picture.

**Scenario:**

- User clicks the “profile settings” and the “change profile picture” option
- The file menu will open and the user can select a photo in the .png or .jpeg format
- Once a photo has been selected the user can click a “confirm” button that will update the profile picture to the new one
- Upon clicking the “confirm” button the user will be sent back to the “profile settings” screen

**Exceptions:**

- **The selected file is not a .png or .jpeg:** If a non .png or .jpeg file is selected then an error message will be shown to the user indicating that it is not a “valid file type” and the user will be asked to select a new file”

**Priority:** This is not a high priority as it is purely an aesthetic feature

**When available:** This feature becomes available when the profile settings are opened.

**Frequency of use:** This will occur whenever the “change profile picture” option is selected in the profile settings of an account

**Channel to actor:** The screen in the multiplayer platform window will change if the “change profile picture” button is clicked. The profile picture icon will update once the new profile picture is selected

**Secondary actors:** External database storing all the account information

**Channel to secondary actors:** An API built into the platform that allows the change

profile picture method to communicate with the external database allowing it to request and update the required information

**Open issues:**

- Should there be a size limit applied to selected png and jpeg files?
- Should there be a cooldown applied to the number of times a profile picture can be updated?

Use Case 6: Setting the Status of an Account

**Iteration: 1**

**Primary Actor:** An existing user who wants to change the status of the account they have already registered on the multiplayer game platform

**Goal in context:** The user should be able to go to their profile settings after logging in and click a setting that says, “update your status”. If it is clicked a drop down menu will appear where the user can select various options (“appear offline”, “busy”, “idle” or the default option “online”). Once a status has been selected the user can close the drop down menu and the profile status will be updated.

**Preconditions:** There must already be a registered account on the system and the account must be currently logged into the multiplayer platform.

**Trigger:** The user enters their account settings and presses a “update your status” button and then selects the status they would like from the drop down menu. Once the status has been selected the user can click a “confirm” button and they have now updated their profile status.

**Scenario:**

- User clicks the “profile settings” and the “update status” option

- A drop down menu will open with various statuses the user can select (“appear offline”, “busy”, “idle” or the default option “online”)
- Once a status has been selected the user can click a “confirm” button that will update the status to the selected one
- Upon clicking the “confirm” button the user will be sent back to the “profile settings” screen with the updated status

**Exceptions:** N/a

**Priority:** This is not a high priority as it is purely an aesthetic feature

**When available:** This feature becomes available when the profile settings is opened.

**Frequency of use:** This will occur whenever the “update status” option is selected in the profile settings of an account

**Channel to actor:** The screen in the multiplayer platform window will change if the “change status” button is clicked. There will be a clear drop down menu shown to the user with various statuses. The selected status will update in real time once the new status is selected.

**Secondary actors:** External database storing all the account information

**Channel to secondary actors:** An API built into the platform that allows the update status method to communicate with the external database allowing it to request and update the required information

**Open issues:**

- Should there be an option implemented for custom statuses?

Use Case 7: Linking an Existing E-mail to Your Account

## **Iteration: 1**

**Primary Actor:** An existing user who wants to link an existing e-mail account they own to their already registered profile on the multiplayer game platform

**Goal in context:** The user should be able to go to their profile settings after logging in and click a setting that says, “link an e-mail”. If it is clicked the user will be given a text-field as well as a prompt that states “enter the e-mail you want to link” and below it a text-field that says “enter your password” which asks for the current account password for the profile. Once the text-fields are entered the user can click a “confirm” button and if the password matches, a verification e-mail will be sent to the e-mail containing a six digit code that needs to be entered into the system. Once the six digit code is entered successfully the user can click a “confirm” button and the e-mail that was entered will be linked to the platform account.

**Preconditions:** There must already be a registered account on the system and the account must be currently logged into the multiplayer platform. The user must have a valid e-mail address as well as access to this e-mail. This e-mail account must also not be currently linked to another profile.

**Trigger:** The user enters their account settings and presses a “link an e-mail” button and then enters the required information into the text-fields depending on the prompt provided. Once all the text-fields have been filled and the user has progressed through each verification step the user can click a “confirm” button and they have now linked an e-mail. A feedback message will be shown to the user indicating to them that an e-mail account has been linked to their profile.

## **Scenario:**

- User clicks the “profile settings” and the “link an e-mail option”
- The user is provided 2 text-fields with prompts that state “enter the e-mail account” and “enter your password”
  - If the password matches, then there is no error when the “confirm” button is pressed



- If the password does not match the saved password, then an error prompt will be shown to the user upon clicking the “confirm” button indicating the password was incorrect
- Once all the fields have been filled correctly the user can click a “confirm” button that will proceed to the next step
- A verification e-mail will be sent to the entered e-mail with a six digit code
- This code is then entered into the application once the e-mail has been accessed by the user
- Once the code has been entered correctly, the user can click a “confirm” button that will link the e-mail to the account currently logged in
- There will be a “an e-mail account has been successfully linked” message displayed to the user, and they will be sent back to the profile settings screen

#### **Exceptions:**

- **The e-mail entered is already linked to an account:** Once something has been entered into the text-field and the “confirm” button has been pressed, if there is already an account with this e-mail linked to it the user will be sent back to the “link an e-mail” screen with a feedback message saying “there is already an account linked with this e-mail”
- **The password entered does not match the saved password:** Once something has been entered into the text-field and the “confirm” button has been pressed, if the password entered does not match the saved password they will be sent back to the “link an e-mail” screen with a feedback message saying “the password entered is incorrect”

**Priority:** This should be a moderate priority as some features require an e-mail account to become available (e.g 2FA).

**When available:** This feature becomes available when the profile settings are opened.

**Frequency of use:** This will occur whenever the “link an e-mail” option is selected in the profile settings of an account

**Channel to actor:** The screen in the multiplayer platform window will change if the “link an e-mail” button is clicked. There will be various prompts as well as feedback messages (including an e-mail) to denote to the user whether something is happening

correctly or not. A feedback message will be displayed to the user once an e-mail has been linked to the account.

**Secondary actors:** External database storing all the account information

**Channel to secondary actors:** An API built into the platform that allows the link an e-mail method to communicate with the external database allowing it to request and update the required information

**Open issues:**

- Should this be a mandatory feature immediately after creating an account?
- Once a new e-mail has been updated, should the previous e-mail be notified via e-mail that the account in question is no longer linked to that e-mail?
- Once an e-mail has been successfully linked, should there be a cooldown period where for a certain period the e-mail cannot be changed?
- When verifying the e-mail, should the six digit code require a limited duration before the code becomes invalid and a new code needs to be requested?

Use Case 8: Linking an Existing Phone Number to Your Account

**Iteration: 1**

**Primary Actor:** An existing user who wants to link an existing phone number they own to their already registered profile on the multiplayer game platform

**Goal in context:** The user should be able to go to their profile settings after logging in and click a setting that says, "link a phone number". If it is clicked the user will be given a text-field as well as a prompt that states "enter the phone number you want to link" and below it a text-field that says "enter your password" which asks for the current account password for the profile. Once the text-fields are entered the user can click a "confirm" button and if the password matches, a text message will be sent to the phone number containing a six digit code that needs to be entered into the system. Once the

six digit code is entered successfully the user can click a “confirm” button and the phone number that was entered will be linked to the platform account.

**Preconditions:** There must already be a registered account on the system and the account must be currently logged into the multiplayer platform. The user must have a valid phone number as well as access to this phone number. This phone number must also not be currently linked to another profile.

**Trigger:** The user enters their account settings and presses a “link a phone number” button and then enters the required information into the text-fields depending on the prompt provided. Once all the text-fields have been filled and the user has progressed through each verification step the user can click a “confirm” button and they have now linked a phone number. A feedback message will be shown to the user indicating to them that a phone number has been linked to their profile.

**Scenario:**

- User clicks the “profile settings” and the “link a phone number”
- The user is provided 2 text-fields with prompts that state “enter the phone number” and “enter your password”
  - If the password matches, then there is no error when the “confirm” button is pressed
  - If the password does not match the saved password, then an error prompt will be shown to the user upon clicking the “confirm” button indicating the password was incorrect
- Once all the fields have been filled correctly the user can click a “confirm” button that will proceed to the next step
- A text message will be sent to the entered phone number with a six digit code
- This code is then entered into the application once the text message has been read by the user
- Once the code has been entered correctly, the user can click a “confirm” button that will link the phone number to the account currently logged in
- There will be a “a phone number has been successfully linked” message displayed to the user, and they will be sent back to the profile settings screen

**Exceptions:**

- **The phone number entered is already linked to an account:** Once something has been entered into the text-field and the “confirm” button has been pressed, if there is already an account with this phone number linked to it the user will be sent back to the “link a phone number” screen with a feedback message saying “there is already an account linked with this phone number”
- **The password entered does not match the saved password:** Once something has been entered into the text-field and the “confirm” button has been pressed, if the password entered does not match the saved password they will be sent back to the “link a phone number” screen with a feedback message saying “the password entered is incorrect”

**Priority:** This should be a moderate priority as some features require a phone number to become available (e.g 2FA). However, this will be a lower priority than the e-mail as the e-mail can access more features.

**When available:** This feature becomes available when the profile settings are opened.

**Frequency of use:** This will occur whenever the “link a phone number” option is selected in the profile settings of an account

**Channel to actor:** The screen in the multiplayer platform window will change if the “link a phone number” button is clicked. There will be various prompts as well as feedback messages (including a text message) to denote to the user whether something is happening correctly or not. A feedback message will be displayed to the user once a phone number has been linked to the account.

**Secondary actors:** External database storing all the account information.

**Channel to secondary actors:** An API built into the platform that allows the link a phone number method to communicate with the external database allowing it to request and update the required information.

**Open issues:**

- Once a new phone number has been updated, should the previous phone number be notified via text message that the account in question is no longer linked to that phone number
- Once a phone number has been successfully linked, should there be a cooldown period where for a certain period the phone number cannot be changed?
- Should there be other methods implemented to verify the phone number (e.g via phone call)?
- When verifying the phone number, should the six digit code require a limited duration before the code becomes invalid and a new code needs to be requested?

### Use Case 9: Setting Up 2FA on Your Account

#### **Iteration: 1**

**Primary Actor:** An existing user who wants to set up two-factor authentication for their already registered profile on the multiplayer game platform

**Goal in context:** The user should be able to go to their profile settings after logging in and click a setting that says, “enable two-factor authentication”. If it is clicked the user will be given a drop-down menu that lists the available methods they can use to enable two-factor authentication (e-mail or phone number) as well as a text-field requesting their password. Once the method has been selected and the password is entered correctly, the user can click a “confirm” button. Upon clicking the “confirm” button, depending on the method chosen a text message or an e-mail will be sent containing a six-digit code that will need to be entered. Once the code has been entered successfully two-factor authentication will now be enabled on the account.

**Preconditions:** There must already be a registered account on the system and the account must be currently logged into the multiplayer platform. The user must have a valid phone number or e-mail already linked to the account they are trying to enable two-factor authentication for. Two-factor authentication must not already be enabled. The user must know their password so they can enter it when requested.

**Trigger:** The user enters their account settings and presses an “enable two-factor authentication” button. Once the method has been selected, the text-field has been filled with the required information and the user has progressed through each verification step the user can click a “confirm” button and two-factor authentication is now enabled. A feedback message will be shown to the user indicating to them that two-factor authentication is now enabled.

### **Scenario:**

- User clicks the “profile settings” and the “enable two-factor authentication” button
- The user is provided a drop down menu with possible methods and a text-field with a prompt that states “enter your password”
  - o If the password matches, then there is no error when the “confirm” button is pressed
  - o If the password does not match the saved password, then an error prompt will be shown to the user upon clicking the “confirm” button indicating the password was incorrect
- Once the password have been filled correctly and the method has been selected the user can click a “confirm” button that will proceed to the next step
- A text message or e-mail will be sent to the linked phone number/e-mail with a six digit code
- This code is then entered into the application once the text message/e-mail has been accessed by the user
- Once the code has been entered correctly, the user can click a “confirm” button that will enable two-factor authentication
- There will be a “two-factor authentication is now active for this account” message displayed to the user, and they will be sent back to the profile settings screen

### **Exceptions:**

- **The password entered does not match the saved password:** Once something has been entered into the text-field and the “confirm” button has been pressed, if the password entered does not match the saved password they will be sent back to the “link a phone number” screen with a feedback message saying “the password entered is incorrect”
- **There is no linked e-mail or phone number to the account:** If there is no linked e-mail or phone number for the account trying to activate two-factor authentication, then an error prompt will be shown to the user when they click the “enable two-factor authentication” button. This error prompt will state that “there

is no e-mail or phone number associated with this account, link one in order to be able to access this feature” and they will be sent back to the profile settings screen.

**Priority:** This should be the same priority as linking a phone number or e-mail as these features are linked.

**When available:** This feature becomes available when the profile settings are opened and only if there is a linked phone number or e-mail with the account currently logged in.

**Frequency of use:** This will occur whenever the “enable two-factor authentication” option is selected in the profile settings of an account and the conditions for the feature are satisfied.

**Channel to actor:** The screen in the multiplayer platform window will change if the “enable two-factor authentication” button is clicked. There will be a clear drop down menu shown to the user with two options as well as a text field asking for a password. There will be various prompts as well as feedback messages (including a text message or e-mail) to denote to the user whether something is happening correctly or not. A feedback message will be displayed to the user once two-factor authentication has been enabled

**Secondary actors:** External database storing all the account information.

**Channel to secondary actors:** An API built into the platform that allows the enable two-factor authentication method to communicate with the external database allowing it to request and update the required information.

**Open issues:**

- Should there be a “cooldown” setting implemented where once a two-factor authentication check is passed, there won’t be a two-factor authentication check for any feature requiring a password?

- Should there be a “trusted device” setting where one selected device will not require two-factor authentication for any feature that requires a password?
- Should one-time codes be implemented as a failsafe for when a user cannot or loses access to the account required for two-factor authentication to succeed?

### Use Case 10: Viewing Your Account Profile

#### **Iteration: 1**

**Primary Actor:** An existing user who wants to view their own profile when in a game which would include their play history, win/loss record, friend list and rank

**Goal in context:** The user should be able to click a “view profile” button when in a game which will allow them to view stats related to the game for their own profile. This would include their rank in the game, their play history, their win/loss record and friend list showing a list of friends that play the game.

**Preconditions:** There must already be a registered account on the system and the account must be currently logged into the multiplayer platform. The user must have an active session in a game of their choosing and the user should be in the main menu of the game and not currently in a match.

**Trigger:** The user must press a “view profile” button while their profile is selected. Once the “view profile” button is selected the user will be shown various stats and information related to their rank, play history, win/loss record and friends who play the game.

#### **Scenario:**

- The user logs into a platform and opens a game of their choice and is now at the main menu of the game
- User clicks the “view profile” button while their profile is selected in a game
- The user is provided various information pertaining to stats and information related to the game (rank, win/loss history, match history, and friend list)



- Once the player is done looking at their profile, they can press a “back” button which will redirect them back to the main menu of the game

**Exceptions:**

- **Server-side issues mean stats cannot be imported when the feature is accessed:** In the event of server-side issues, an error prompt will be shown to the user stating “this feature is not available right now, please try again later” when the “view profile” button is selected

**Priority:** This should be a relatively high priority to satisfy the requirements of the client, however it should not be higher relative to features related to account set up and management (creating an account, logging in etc.)

**When available:** This should be available whenever the user is in the main menu of a game

**Frequency of use:** This will occur whenever the “view profile” button is pressed while the user’s profile is highlighted

**Channel to actor:** The game menu interface will change and there will be various screens and information that will now be accessible to the user for them to view.

**Secondary actors:** Server database related to the game that stores all the play data of an account.

**Channel to secondary actors:** A call to the server to request the play data whenever the “view profile” button is pressed for the account that was selected.

**Open issues:** N/a

## Use Case 11: Sending a Friend Request to Another Account

**Iteration:** 1

**Primary Actor:** Player1

**Goal in Context:** Add a friend to the player's friend list

**Preconditions:** Player1 must have an account and be logged in

**Trigger:** Player1 searches for a username in the menu and clicks the "Add" button

**Scenario:**

1. Player1 searches for Player2.
2. Player2 exists and Player1 is not blocked by Player2.
3. Player1 presses the "+" button next to Player2's name.
4. The friend request is sent to an external server.
5. Player2 receives a friend request notification.

**Exceptions:**

- Player2 has blocked Player1.
- Player2 does not exist.
- The username was misspelled.

**Priority:** Medium

**Availability:** While servers are operational

**Frequency of Use:** Varies per user

**Channel to Actor:** Menu interface with friends sidebar

**Secondary Actors:** Player2

**Channel to Secondary Actors:** Same menu interface; notifications processed via external server

**Open Issues:**

- What happens if a player sends a friend request but is blocked before it is accepted?

Use Case 12: Unfriending an Account

**Iteration:** 1

**Primary Actor:** Player1

**Goal in Context:** Remove a friend from the game interface

**Preconditions:** Player1 must be friends with Player2

**Trigger:** Player1 clicks on Player2's profile and selects the "Unfriend" button

**Scenario:**

1. Player1 is logged in and has Player2 added as a friend.
2. Player1 clicks on Player2's profile and selects "Remove Friend."
3. Data is sent to the server to remove the connection.
4. Player1 is removed from Player2's friend list.

**Exceptions:**

- Server-side issues prevent the friend list update.

**Priority:** Medium

**Availability:** Any time

**Frequency of Use:** Varies per user

**Channel to Actor:** In-game interface

**Secondary Actors:** Player2

**Channel to Secondary Actors:** Server-side data update

**Open Issues:** None

Use Case 13: Viewing a Friend's Profile

**Iteration:** 1

**Primary Actor:** Player1

**Goal in Context:** View a friend's play history, win/loss record, friends list, rank etc.

**Preconditions:** Player1 must be friends with Player2

**Trigger:** Player1 presses the "View Profile" button on Player2's profile

**Scenario:**

1. Player1 clicks on Player2's profile.
2. Player1 selects "View Profile."
3. A new menu opens, displaying Player2's statistics.

**Exceptions:**

- Player2 has a private profile, showing limited or no information.
- Player1 is blocked or not friends with Player2.

**Priority:** Medium

**Availability:** Anytime in the main menu

**Frequency of Use:** Varies per user

**Channel to Actor:** In-game menu interface

**Secondary Actors:** Player2

**Channel to Secondary Actors:** None

**Open Issues:**

- What if Player2 wants to hide stats from Player1 but not from other friends?

Use Case 14: Blocking an Account

**Iteration:** 1

**Primary Actor:** Player1

**Goal in Context:** Prevent Player2 from interacting with Player1's account

**Preconditions:** Both Player1 and Player2 must be valid users

**Trigger:** Player1 searches for or clicks on Player2's profile and selects the "Block" button

**Scenario:**

1. Player1 clicks on Player2's profile.
2. Player1 selects "Block."
3. A request is sent to the external server.
4. Player2 is now blocked from interacting with Player1.

**Exceptions:**

- Server-side issues prevent the block from being processed.

**Priority:** High

**Availability:** Any time

**Frequency of Use:** Varies per user

**Channel to Actor:** In-game menu interface

**Secondary Actors:** Player2

**Channel to Secondary Actors:** Server-side enforcement

**Open Issues:** None

Use Case 15: Logging Out of an Account

**Iteration:** 1

**Primary Actor:** Player1

**Goal in Context:** Log out from the game safely

**Preconditions:** Player1 must be logged in

**Trigger:** Player1 selects the "Log Out" button in the menu

**Scenario:**

1. Player1 accesses the game menu.
2. Player1 selects "Log Out."
3. A request is sent to the server.
4. Player1's session is terminated.
5. Player1 is returned to the login screen.

**Exceptions:**

- Server-side issues prevent logout, keeping Player1 logged in.

**Priority:** High



**Availability:** Any time

**Frequency of Use:** Frequent

**Channel to Actor:** In-game menu interface

**Secondary Actors:** None

**Channel to Secondary Actors:** None

**Open Issues:** None

Use Case 16: Check if it is a Valid Account

**Iteration:** 1

**Primary Actor:** External Database

**Goal in Context:** Verify that a newly created account meets all validity requirements

**Preconditions:** A user initiates account creation

**Trigger:** Account creation process is started

**Scenario:**

1. User enters required account details.
2. System checks if the username is unique.
3. System verifies password strength.
4. System ensures all required fields are filled.
5. Account creation is either approved or rejected based on validity checks.

**Priority:** High

**Availability:** Any time account creation is attempted

**Frequency of Use:** Every account creation

**Open Issues:** None

Use Case 17: Authenticating a User

**Iteration:** 1

**Primary Actor:** External Database

**Goal in Context:** Authenticate a user for any game interaction

**Preconditions:** User must have an account

**Trigger:** Any use case activation

**Scenario:**

1. User attempts an action.
2. System checks if the user is logged in.
3. If not, the user is prompted to log in.
4. Credentials are verified against the database.
5. If valid, the action proceeds; otherwise, access is denied.
6. For all other actions a hidden unique session token is provided to user upon logging in.
7. All user actions are checked before allowing the external database to change the data
8. If valid actions continue as normal

**Priority:** Critical

**Availability:** Every time a user initiates an action

**Frequency of Use:** Continuous

**Open Issues:** None