

- **Use Case Name:** PlayGame

Participating Actor(s): Player

Description: The player would use this case to enter into the actual gameplay of the game.

Flow of Events:

1. The player selects the option for playing the game from the main menu.
2. The main menu disappears and another menu is displayed, asking the player to choose between playing on his/her own (i.e., single-player) or having the ability to see and interact with characters played by other people (i.e., multiplayer).
3. If the player selects the multiplayer option, it would invoke InitiateMultiplayer, which would also give the player the ability to play alongside other players as a team by invoking JoinMultiplayer.
4. Once the player's choice between single-player and multiplayer is made, the menu disappears and GameController requests data from the Database via the getSaveData(userID) method.
5. After the Database is accessed, it searches for an account with an ID that matches userID.
6. When the correct account is found, the data that has been saved for it is retrieved and returned to GameController.
7. GameController, then, sends this data to PlayGame via the loadSaveData() method.
8. PlayGame, then, updates GameController's display and receives input generated by the player via enterInput().
9. While the game is played, the player's progress is continually updated and saved to his/her user account in the Database via updateSaveData().

Entry Conditions: The player has invoked LaunchGame.

Exit Conditions: The player either resumes the gameplay from where he/she left off, or begins a new adventure. He/She can perform the gameplay functions (i.e., movement, speech, etc.).

Alternative Flow: I could not think of one.

Exceptions Flow:

1. The player selects the option for playing the game from the main menu.
2. However, the player loses his/her Internet connection and is unable to load, update, and save his/her data.

3. A message is displayed, telling the player that he/she is no longer connected to the Internet and, therefore, is unable to play the game.
4. The player closes the message and gets returned to the previous menu.

- **Use Case Name:** LaunchGame

Participating Actor(s): Player

Description: The player would use this case to activate the game.

Flow of Events:

1. The player clicks on the launch icon.
2. Game opens
3. The player enters their username and password into text fields to login to their account
4. Click enter button/hit enter key to submit.
5. System executes checkInputData() to find the relevant user
6. Relevant data is sent to the user
7. Game goes to main menu via the mainMenu() method

Entry Conditions: The player has downloaded/installed the game.

Exit Conditions: The player can now perform the activities provided in the main menu.

Alternative Flow: I could not think of one.

Exceptions Flow:

1. The player enters his/her username and password to login to his/her account.
2. Something was typed wrong, so a message is displayed to let the player know what the issue is and in which textbox it occurred.
3. The player is asked to check the spelling and either correct it or re-type the information before attempting to submit it again.

- **Use Case Name:** CloseGame

Participating Actor(s): Player

Description: The player would use this case to end/exit the game.

Flow of Events:

1. There is a button on the screen during gameplay for leaving it and returning to the main menu.
2. The player clicks on the button.
3. The main menu also has an option for terminating the game entirely.
4. The player selects this option.
5. PlayGame is called via the startClose() method.
6. The Database's data for the player's account is updated and saved via updateSaveData().
7. PlayGame, then, calls GameControl via startClose().
8. GameControl initiates the logout process by calling LaunchGame via the logout() method.
9. Finally, LaunchGame confirms that the preceding steps have been executed successfully via confirmClose().

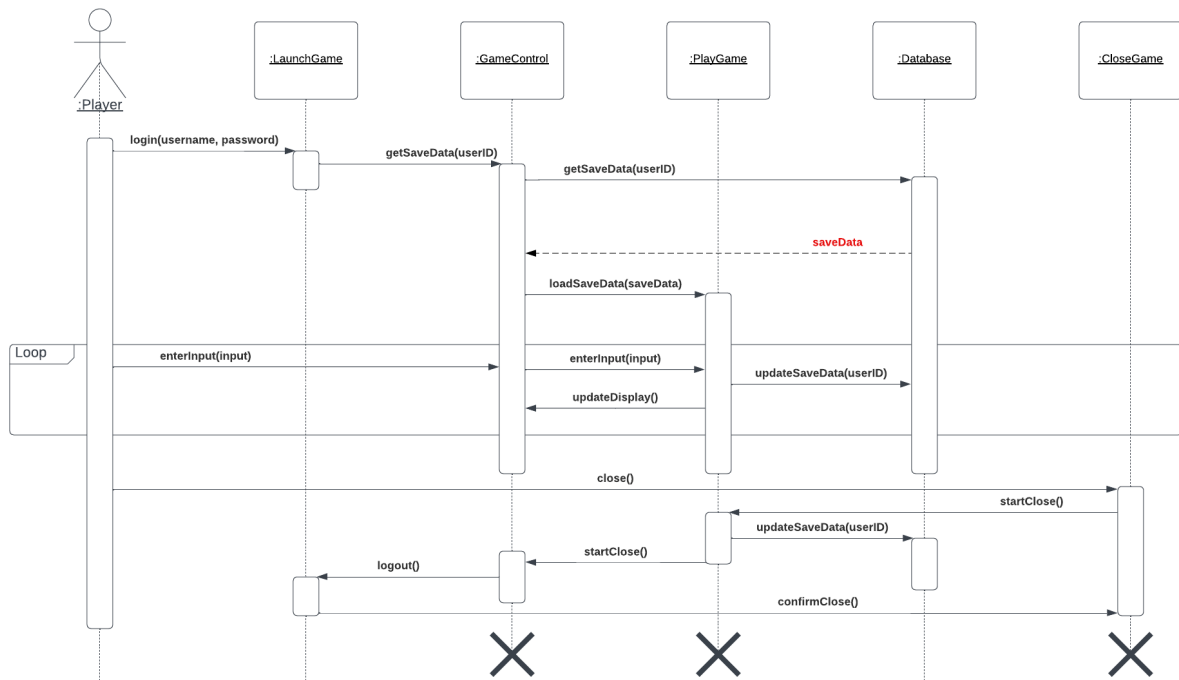
Entry Conditions: The player is done with the game.

Exit Conditions: The player's place in the game is saved and the screen disappears when he/she closes it.

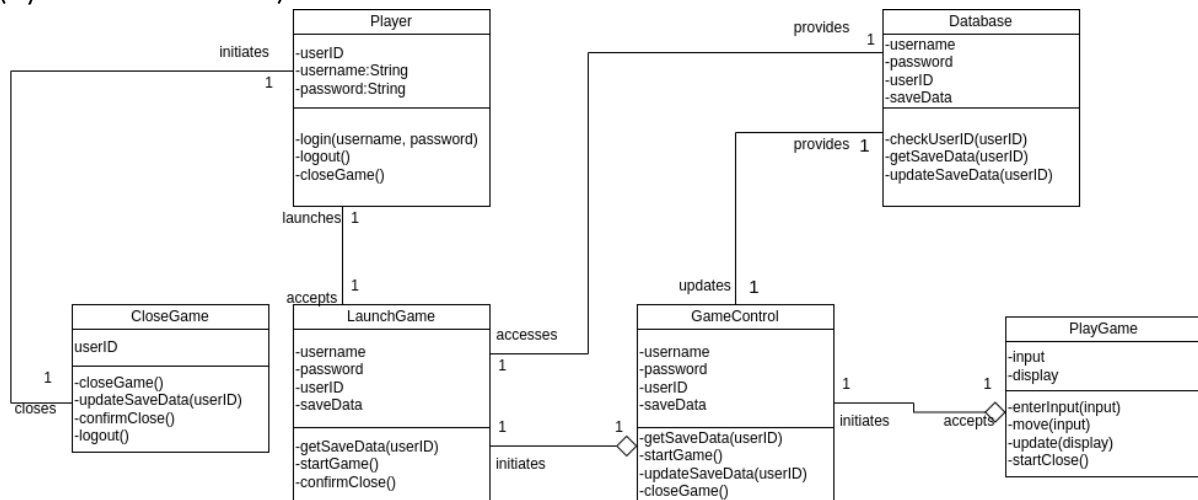
Alternative Flow: I could not think of one.

Exceptions Flow:

1. There is a button on the screen during gameplay for leaving it and returning to the main menu.
2. The player clicks on the button.
3. The main menu also has an option for terminating the game entirely.
4. The player selects this option.
5. An error occurs while the Database's data for the player's account is being updated and saved via updateSaveData().
6. A message tells the player what has happened and to try closing the game again.



(Dynamic Model Above)



(Analysis Object Model Above)