|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name of Use Case:** | Play Game | | | |
| **Created By:** | Sphynx | | **Last Updated By:** | L. Jones |
| **Date Created:** | 3/6/24 | | **Last Revision Date:** | 3/6/24 |
| **Association** | | Players can input keys that affect the flow of the game. | | |
| **Description:** | | Player of the game plays the rhythm game by inputting different button combinations as prompted by the screen. | | |
| **Actors:** | | Player, New Player, Returning Player | | |
| **Preconditions:** | | 1. Player has logged in. 2. Player has selected a song. 3. Player has selected a difficulty. | | |
| **Postconditions:** | | 1. Player gets a certain score based on the number of correct inputs. 2. Player’s score is checked against other highest scores. 3. If the score is higher than other scores, it is placed on the leaderboard. 4. The player is prompted if they would like to select another song or difficulty. | | |
| **Includes** | | 1. Select Song/Difficulty | | |
| **Flow:** | | 1. The player inputs the correct keys as the song progresses  2. The player completes the song | | |
| **Alternative Flows:** | | 1. In step 1, if the player inputs enough incorrect keys  a. The player is prompted with a game over screen.  b. The player can decide whether to try again, or select a new song/difficulty | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name of Use Case:** | Create User | | | |
| **Created By:** | Sphynx | | **Last Updated By:** | L. Jones |
| **Date Created:** | 3/6/24 | | **Last Revision Date:** | 3/6/24 |
| **Association** | | New Players can create their own login. | | |
| **Description:** | | New players can create a login to serve as their representation in the game and on the leaderboards. | | |
| **Actors:** | | New Player | | |
| **Preconditions:** | | 1. The webpage has successfully loaded | | |
| **Postconditions:** | | 1. The player is prompted with “successful login.”  2. The player is brought to the select a level screen. | | |
| **Flow:** | | 1. The player clicks create account  2. The player types in a username and password  3. The new player clicks “log me in.” | | |
| **Alternative Flows:** | | 1. In step 3, if the username is taken  a. The player is prompted with a popup declaring that the username has already been taken, and the account is not created. | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name of Use Case:** | Login User | | | |
| **Created By:** | Sphynx | | **Last Updated By:** | L. Jones |
| **Date Created:** | 3/6/24 | | **Last Revision Date:** | 3/6/24 |
| **Association** | | Returning players will have a login to enter the login screen. | | |
| **Description:** | | Returning players can login with an already created account. | | |
| **Actors:** | | Returning Player | | |
| **Preconditions:** | | 1. The webpage has successfully loaded.  2. The player has an account already. | | |
| **Postconditions:** | | 1. The player is prompted with “successful login.”  2. The player is brought to the select a level screen. | | |
| **Flow:** | | 1. The player types in their username and password.  2. The player clicks “log me in.” | | |
| **Alternative Flows:** | | 1. In step 2, if the player does not exist  a. The player is prompted that they must create an account, and the page does not log them in.  2. In step 2, if the player does not enter a correct password  a. The player is prompted to that the password was incorrect, and the page does not log them in. | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name of Use Case:** | Select Song/Difficulty | | | |
| **Created By:** | Sphynx | | **Last Updated By:** | L. Jones |
| **Date Created:** | 3/6/24 | | **Last Revision Date:** | 3/6/24 |
| **Association** | | The player needs to select a difficulty and song to play. | | |
| **Description:** | | Players can select between two difficulties and three different songs depending on what they have unlocked. | | |
| **Actors:** | | Player | | |
| **Preconditions:** | | 1. The player has successfully logged in. | | |
| **Postconditions:** | | 1. The webpage loads in the song.  2. The player can play the song. | | |
| **Flow:** | | 1. The player selects any of the available songs.  2. The player selects any of the available difficulties.  3. The player waits for the song to be loaded. | | |
| **Alternative Flows:** | | 1. In step 1, if the player selects a song that hasn’t been unlocked  a. The player is prompted with a dialogue that states what is required to unlock the song.  2. In step 2, if the player selects a difficulty that hasn’t been unlocked  a. The player is prompted with a dialogue that states what is required to unlock the song. | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name of Use Case:** | Get High Scores | | | |
| **Created By:** | Sphynx | | **Last Updated By:** | L. Jones |
| **Date Created:** | 3/6/24 | | **Last Revision Date:** | 3/6/24 |
| **Association** | | Anyone can request to get information about the high scores. This includes players and admins. | | |
| **Description:** | | The concerned party receives a JSON response with a list of usernames and the associated scores that in the top five highest scores. | | |
| **Actors:** | | Player, Admin | | |
| **Preconditions:** | | None | | |
| **Postconditions:** | | None | | |
| **Flow:** | | 1. The concerned party makes a request to a specified URL within the game’s webpage  2. The concerned party receives a JSON response that includes the top five highest scores and the associated usernames for these scores. | | |
| **Alternative Flows:** | | 1. In step 2, if there are not five scores catalogued yet  a. The concerned party will receive a response that includes the number of scores that have been entered (can be 0 if no one has played yet). | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name of Use Case:** | Reset High Scores | | | |
| **Created By:** | Sphynx | | **Last Updated By:** | L. Jones |
| **Date Created:** | 3/6/24 | | **Last Revision Date:** | 3/6/24 |
| **Association** | | An admin can make a request with a password to interact with the high score leaderboard. | | |
| **Description:** | | Using an HTTP request, an admin can reset the high score leaderboard. | | |
| **Actors:** | | Admin | | |
| **Preconditions:** | | None | | |
| **Postconditions:** | | None | | |
| **Flow:** | | 1. The admin makes a request to a specified URL within the game’s webpage using an admin password.  2. The database clears all entries for the current leaderboard.  3. The admin receives an HTTP response that confirms the changes have been made. | | |
| **Alternative Flows:** | | 1. In step 1, if the requester gives an invalid password.  a. A response will be sent that declares no changes have been made. | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name of Use Case:** | Modify Song | | | |
| **Created By:** | Sphynx | | **Last Updated By:** | L. Jones |
| **Date Created:** | 3/6/24 | | **Last Revision Date:** | 3/6/24 |
| **Association** | | An admin can make a request with a password to interact with the song database. | | |
| **Description:** | | Using an HTTP request, an admin can update information about a song such as the bpm or arrow mapping. | | |
| **Actors:** | | Admin | | |
| **Preconditions:** | | None | | |
| **Postconditions:** | | None | | |
| **Flow:** | | 1. The admin makes a request containing new information to a specified URL within the game’s webpage using an admin password.  2. The database updates a song entry with the new information.  3. The admin receives an HTTP response that confirms the changes have been made. | | |
| **Alternative Flows:** | | 1. In step 1, if the requester gives an invalid password  a. A response will be sent that declares no changes have been made.  2. In step 2 if the requester gives song information formatted incorrectly  a. A response will be send that declares no changes have been made. | | |