

Name: Advertisement Step

BriefDescription: This use-case is strictly in order to control the advertisements that the user will view, based on whether or not they currently have the "full" version of the software or if they choose to view the advertisement in order to move onto the next level. User will be prompted to either view or purchase the ability to skip and remove all ads within the software.

Actors: User & Advertisement Database.

Preconditions: If the user already removed the advertisements by purchasing the "full" version of the game.

BasicFlow:

1. User is prompted with an advertisement or the option to purchase the "full" version of the software.
2. User can either watch the advertisement and continue to the next level, or they can purchase the "full" version to skip and remove all ads.
3. Prompt based on user selection
3.1. If the user selects to watch the advertisement, load an advertisement from the database.
3.2. If the user wants to purchase the "full" version, redirect them to the the payment section and allow them to enter their payment details and information to confirm purchase of the product.
4. Once either prompt has been selected, once complete, allow the user to continue playing onto the next level.

AlternateFlows:

AlternateFlows:

1. If the user tries to exit, prompt them with a message saying that they will have to watch a longer ad next time they clear a level.
1.1. If the user exits, this places them in a special category that is according to the amount of times the user tried to exit the ads.
1.2. Each category has longer ads increasing by 15s respectively.

ExceptionFlows:

1. If the user does not want to watch the ad or continue playing, they will have another prompt to allow them to exit the game and place them in another special category.
1.1. This special category changes the amount of ads or length of the ads they see, on the next attempt that they play the game.
2. If the user pauses the ad, the video will not continue until they hit play again. They must go through the full length of the ad in order to continue what they would like to do with the software.

PostConditions: Once the user selects to either view the advertisement to move onto the next level or if they choose to purchase the "full" version of the game to take out the advertisements completely.

Name: Hi-Score Check

BriefDescription: This use case entails the ability for the user to place their score from the level cleared into the hi-score database. This hi-score is viewable by the user whenever they want in their own leader boards section or publicly share it with other users or friends through the social media platforms available. This use case also connects to the level check use case, which is used to know which level the user is currently playing, and to store the respective hi-score into that section of the database.

Actors: Users & Score Database

Preconditions: User automatically allows the hi-score to be input into the database whenever the user clears the level and does not require them to answer a prompt.

BasicFlow:

- | |
|---|
| 1. Once the level is cleared, user is prompted with option to save the score into the hi-score section or skip to move onto the next level. |
| 1.1. If the user wants to save their score, prompt them with a view to see their scores for each level they cleared and the difficulties of these levels. |
| 1.2. If the user skips and wants to move to the next level, continue to main step 3. |
| 2. User can continue to next level if they have not already chosen to do so. |
| 3. If the user has accepted to store their score into the database, prompt them to continue to the next level. |

AlternateFlow:

- | |
|---|
| 1. If user does not want to continue or save their score onto the hi-score, prompt them to exit and go back to the main menu or retry the level if the user would like. |
| 2. If user selects to restart the level, prompt them with the Basic flow once they complete the level again. |

ExceptionFlows:

- | |
|--|
| 1. If the user closes the web-browser or client, auto-save the hi-score but prompt the user on the next sign-in if they would like to save it to the overall scoreboard or delete the score. |
| 2. If the user declines upon logging back in, continue to prompt the user to the instructions or selecting to play the game. |

PostConditions: Hi-score value is accepted and inserted into the database to display onto the overall leader boards.

Name: Login System

BriefDescription: This use-case handles the acceptance and verification of the login details which are inputted from the user. The system will check if the login details (username/email and password) are correct. If the details are not met, the error will be sent back to inform the user that their login credentials are invalid (e.g., incorrect password if it was a possible typo).

Actors: User & Login Database

Preconditions: Autosaved/autofill users already saved username and password. As long as the password has not been c

BasicFlow:

- | |
|---|
| 1. Enter the username/email alongside with the respective password for the account. |
| 2. Check if username & password combo are correct using the login database. |
| 3. If username & password combo not recognized, user can sign up or reset their password. |
| 3.1. User can select to sign up or reset their password. |
| 3.2. If user wants to sign up, provide fill-up form and ask user to create an account with an unused email & a password |
| 3.3. If user wants to reset their password, give them the prompt to enter the email they are trying to reset the password |
| 3.4. Reset link is sent to the email entered if it exists within the database. |
| 4. Prompt user accordingly based on previously selected option. |
| 5. Allow user to re-enter credentials. |
| 6. Verify the username & password combo and validate it. |

Alternate Flows:

1. If the device that the user is logging on is new, prompt the user with a security question or request to send a confirmation code to a registered email address or phone number.
2. If the verification is not completed, user will not be allowed to login.
3. If the user attempts to login multiple times on the new device, temporarily block login attempts from the device for 30 minutes.

ExceptionFlows:

1. If the username is recognized but the password has been entered incorrectly multiple times, time out the login attempt
2. User can continue to try to login or reset their password if they verify the email sent to them.
3. The link sent to the user is only valid for 15-20 minutes.
4. If user cannot access their email, prompt them to answer security questions if they have set it up.

PostConditions: User can now access the software/web-site and play the game.