|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC01 | |
| **Name:** | Add Game | |
| **Actor(s):** | Employee | |
| **Goal:** | To add a game that to the inventory. | |
| **Pre-condition:** | 1. The Actor must be signed into an Employee account.  2. The Actor must be on the “Games” page.  3. The game must not exist already. | |
| **Post-condition:** | 1. The game data is saved to the database. | |
| **Constraints/Issues/Risks:** | 1. The employee may not be authorized to create a certain game, or add a game that does not exist. | |
| **Trigger Event(s):** | 1. The employee clicks on the Add Game button. | |
| **Primary Path:** | Actor: | System: |
| 1. Clicks on the “Add Game” button.  4. Enters all game information, and clicks “Submit” button. | 2. Returns to the “Add Game” page.  5. Saves data to database, and displays a notification of the result to the employee.  6. Returns to the “Games” page. |
| **Alternative Path(s):** | 4. Clicks the “Cancel” button. | 5. Returns to “Games” page. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC02 | |
| **Name:** | Edit Game | |
| **Actor(s):** | Employee | |
| **Goal:** | To edit a game that already exists, and update the changes to the database. | |
| **Pre-condition:** | 1. The Actor must be signed into an Employee account.  2. The Actor must be on the “Games” page.  3. The game must exist. | |
| **Post-condition:** | 1. The game is edited and the newly updated game replaces the original version. | |
| **Constraints/Issues/Risks:** | 1. The employee may not be authorized to edit a certain game, or replaces valid information with false information. | |
| **Trigger Event(s):** | 1. The employee views the game and clicks the “Edit Game” button. | |
| **Primary Path:** | Actor: | System: |
| 1. Selects a game and Clicks the “Edit Game” button.  3. Modifies the desired game information and clicks the “Update Game” button. | 2. Displays the “Edit Game” page and populates the data for the requested game.  4. Updates the game data and displays a notification of the result to the employee.  5. Returns to the “Games” page. |
| **Alternative Path(s):** | 3. Click’s the “Cancel” button. | 4. Returns to the “Games” page. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC03 | |
| **Name:** | Delete Game | |
| **Actor(s):** | Employee | |
| **Goal:** | To delete a game from the inventory. | |
| **Pre-condition:** | 1. The Actor must be signed into an Employee account.  2. The Actor must be on the “Games” page.  3. The Game must exist. | |
| **Post-condition:** | 1. The game is removed from inventory. | |
| **Constraints/Issues/Risks:** | 1. The employee may not be authorized to delete a certain game. | |
| **Trigger Event(s):** | 1. The employee clicks on the “Delete Game” button. | |
| **Primary Path:** | Actor: | System: |
| 1. Selects a game and clicks on the “Delete Game” button.  3. Clicks “Delete” button. | 2. Displays a message box, requesting the user to confirm deletion.  4. Deletes the game in the database and notifies the employee of the result.  5. Returns to the “Games” page. |
| **Alternative Path(s):** | 3. Clicks “Cancel” button. | 4. Returns to the “Games” page. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC04 | |
| **Name:** | Create Event | |
| **Actor(s):** | Employee | |
| **Goal:** | To create an event. | |
| **Pre-condition:** | 1. The Actor must be signed into an Employee account.  2. The Actor must be on the “Events” page. | |
| **Post-condition:** | 1. The event is created and made available to users. | |
| **Constraints/Issues/Risks:** | 1. The employee may not be authorized to create an event, or mistakenly add incorrect information. | |
| **Trigger Event(s):** | 1. The employee clicks on the “Create Event” button. | |
| **Primary Path:** | Actor: | System: |
| 1. Clicks “Create Event” button.  3. Enters details and clicks “Create” button. | 2. Displays the “Create Event” page.  4. Saves data to the database and notifies the employee of the result.  5. Returns to the “Events” page. |
| **Alternative Path(s):** | 3. Clicks the “Cancel” button. | 4. Returns to the “Events” page. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC05 | |
| **Name:** | Edit Event | |
| **Actor(s):** | Employee | |
| **Goal:** | To edit an event that already exists, and update the changes to the database. | |
| **Pre-condition:** | 1. The Actor must be signed into an Employee account.  2. The Actor must be on the “Events” page.  3. The event must exist. | |
| **Post-condition:** | 1. The event is edited and the newly updated event replaces the original version. | |
| **Constraints/Issues/Risks:** | 1. The employee may not be authorized to update a particular event or mistakenly enters incorrect information. | |
| **Trigger Event(s):** | 1. The employee clicks on the “Create Event” button. | |
| **Primary Path:** | Actor: | System: |
| 1. Clicks on the “Create Event” button.  3. Updates the required information, and clicks the “Update” button. | 2. Displays the “Edit Event” page with the requested event’s data.  4. Updates the event data in the database and notifies the employee of the result.  5. Returns to the “Events” page. |
| **Alternative Path(s):** | 3. Clicks the “Cancel” button. | 4. Returns to the “Events” page. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC06 | |
| **Name:** | Delete Event | |
| **Actor(s):** | Employee | |
| **Goal:** | To delete an event. | |
| **Pre-condition:** | 1. The Actor must be signed into an Employee account.  2. The Actor must be on the “Events” page.  3. The event must exist. | |
| **Post-condition:** | 1. The event is deleted from the database and no longer available. | |
| **Constraints/Issues/Risks:** | 1. The employee may not be authorized to delete an event or may delete the wrong event. | |
| **Trigger Event(s):** | 1. The employee clicks on the “Delete Event” button. | |
| **Primary Path:** | Actor: | System: |
| 1. Selects an event and clicks on the “Delete Event” button.  3. Clicks the “Delete” button. | 2. Displays a message box, requesting the user to confirm deletion.  4. Deletes the event in the database and notifies the employee of the result.  5. Returns to the “Events” page. |
| **Alternative Path(s):** | 3. Clicks the “Cancel” button. | 4. Returns the “Events” page. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC07 | |
| **Name:** | View Report | |
| **Actor(s):** | Employee | |
| **Goal:** | To view a report. | |
| **Pre-condition:** | 1. The Actor must be signed into an Employee account.  2. The Actor must be on the “Reports” page. | |
| **Post-condition:** | 1. The details of the report are displayed. | |
| **Constraints/Issues/Risks:** | 1. The employee may not be authorized to view the report. | |
| **Trigger Event(s):** | 1. The employee clicks on the “View Reports” button. | |
| **Primary Path:** | Actor: | System: |
| 1. Clicks the “View Reports” button.  3. Selects desired report and clicks “View Report.” | 2. Displays the available reports.  4. Retrieves report data from database and displays for employee. |
| **Alternative Path(s):** | 1. Clicks the “Cancel” button. | 2. Returns to “Reports” page. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC08 | |
| **Name:** | Print Report | |
| **Actor(s):** | Employee | |
| **Goal:** | To print a report for review. | |
| **Pre-condition:** | 1. The Actor must be signed into an Employee account.  2. The Actor must be on the “Reports” page. | |
| **Post-condition:** | 1. The report is printed. | |
| **Constraints/Issues/Risks:** | 1. The report may not be printed correctly or sent to the wrong printer. | |
| **Trigger Event(s):** | 1. The employee clicks on the “Print Reports” button. | |
| **Primary Path:** | Actor: | System: |
| 1. Clicks the “Print Reports” button.  3. Selects desired report(s) and clicks “Print Report” button.  5. Selects desired printer. | 2. Displays the available reports.  4. Displays available printers for user to select destination printer.  6. Creates print job and sends data to selected printer. Displays confirmation to employee.  7. Returns to “Reports” page. |
| **Alternative Path(s):** | 3. Clicks “Cancel” button. | 4. Returns user to “Reports” page. |
| 5. Click’s “Cancel” button. | 6. Returns user to “Reports” page. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC09 | |
| **Name:** | Approve User Review | |
| **Actor(s):** | Employee | |
| **Goal:** | To approve a user’s submitted game review. | |
| **Pre-condition:** | 1. The Actor must be signed into an Employee account.  2. The Actor must be on the “Reviews” page.  3. A review exists to approve. | |
| **Post-condition:** | 1. The review is added to the list of reviews. | |
| **Constraints/Issues/Risks:** | 1. The review may be accidentally denied or not seen by the employee. | |
| **Trigger Event(s):** | 1. The employee clicks on the “Pending Reviews” button. | |
| **Primary Path:** | Actor: | System: |
| 1. Clicks on the “Pending Reviews” button.  3. Inspects the review and clicks the “Approve” button. | 2. Displays the list of reviews pending approval.  4. Adds the review to the list of available reviews, and removed from the list of “Pending Reviews.”  5. Returns to “Pending Reviews” list. |
| **Alternative Path(s):** | 3. Clicks the “Reject” button. | 4. Sends the review back to the author for revision or deletion.  5. Returns to “Reviews” page. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC10 | |
| **Name:** | Rejects User Review | |
| **Actor(s):** | Employee | |
| **Goal:** | To reject a user review. | |
| **Pre-condition:** | 1. The Actor must be signed into an Employee account.  2. The Actor must be on the “Reviews” page.  3. A review exists to deny. | |
| **Post-condition:** | 1. The review is removed from the list of pending reviews and send back to the author for revision or deletion. | |
| **Constraints/Issues/Risks:** | 1. The review may be accidentally approved or not seen by the employee. | |
| **Trigger Event(s):** | 1. The employee clicks on the “Pending Reviews” button. | |
| **Primary Path:** | Actor: | System: |
| 1. Clicks on the “Pending Reviews” button.  3. Inspects the review and clicks the “Reject” button. | 2. Displays the list of reviews pending approval.  4. Sends the review back to the author for revision or deletion.  5. Returns to “Reviews” page. |
| **Alternative Path(s):** | 3. Inspects the review and clicks the “Approve” button. | 4. Adds the review to the list of available reviews, and removed from the list of “Pending Reviews.”  5. Returns to “Pending Reviews” list. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC11 | |
| **Name:** | Create Order | |
| **Actor(s):** | Employee | |
| **Goal:** | To create an order for a user. | |
| **Pre-condition:** | 1. The Actor must be signed into an Employee account.  2. The Actor must be on the “Orders” page. | |
| **Post-condition:** | 1. The order is created and processed. | |
| **Constraints/Issues/Risks:** | 1. The order that the employee creates may contain the incorrect items.  2. The order may be created for the wrong customer. | |
| **Trigger Event(s):** | 1. The employee clicks on the “Create Order” button. | |
| **Primary Path:** | Actor: | System: |
| 1. Clicks the “Create Order” button.  3. Enters the order information, and clicks the “Create” button. | 2. Displays the “Create Order” page.  4. Saves the order to the database, displays the result for the employee, and emails an order confirmation to customer.  5. Returns to the “Orders” page. |
| **Alternative Path(s):** | 3. Clicks the “Cancel” button. | 4. Returns to the “Orders” page. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC12 | |
| **Name:** | Edit Order | |
| **Actor(s):** | Employee | |
| **Goal:** | To modify an existing order. | |
| **Pre-condition:** | 1. The Actor must be signed into an Employee account.  2. The Actor must be on the “Orders” page.  3. The order must exist. | |
| **Post-condition:** | 1. The order is updated and recorded. | |
| **Constraints/Issues/Risks:** | 1. The employee may incorrectly change the order. | |
| **Trigger Event(s):** | 1. The employee clicks on the “Edit Order” button. | |
| **Primary Path:** | Actor: | System: |
| 1. Selects an order and clicks the “Edit Order” button.  3. Enters the updated information and clicks the “Update” button. | 2. Displays the “Edit Order” page with the requested order’s information.  4. Saves the updated information, and notifies the employee of the result.  5. Returns to “Orders” page. |
| **Alternative Path(s):** | 3. Clicks “Cancel” button. | 4. Returns to “Orders” page. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC13 | |
| **Name:** | Delete Order | |
| **Actor(s):** | Employee | |
| **Goal:** | To delete an existing order. | |
| **Pre-condition:** | 1. The Actor must be signed into an Employee account.  2. The Actor must be on the “Orders” page.  3. The order exists. | |
| **Post-condition:** | 1. The order is deleted from the system. | |
| **Constraints/Issues/Risks:** | 1. The employee may accidentally delete the wrong order. | |
| **Trigger Event(s):** | 1. The employee clicks on the “Delete Order” button. | |
| **Primary Path:** | Actor: | System: |
| 1. Selects an order and Clicks on the “Delete Order” button.  3. Clicks “Confirm” button. | 2. Displays a message box requiring the employee to confirm deletion.  4. Deletes the order in the database and notifies the employee of the result.  5. Returns to “Orders” page. |
| **Alternative Path(s):** | 3. Clicks “Cancel” button. | 4. Returns to “Orders” page. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC14 | |
| **Name:** | Close Order | |
| **Actor(s):** | Employee | |
| **Goal:** | To complete and close an open order. | |
| **Pre-condition:** | 1. The Actor must be signed into an Employee account.  2. The Actor must be on the “Orders” page.  3. An open order exists. | |
| **Post-condition:** | 1. The order is marked closed and removed from the open orders list. | |
| **Constraints/Issues/Risks:** | 1. An order may be erroneously closed by an employee. | |
| **Trigger Event(s):** | 1. The employee clicks on the “Pending Orders” button. | |
| **Primary Path:** | Actor: | System: |
| 1. Clicks on the “Open Orders” button.  3. Reviews order and clicks “Close Order” button. | 2. Retrieves the order data from the database and displays for the employee.  4. Marks the order as closed, notifies the employee of the result, and removes it from the open orders list.  5. Returns to “Open Orders” page. |
| **Alternative Path(s):** | 3. Clicks “Cancel” button. | 4. Returns to “Open Orders” page. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC15 | |
| **Name:** | Create Account | |
| **Actor(s):** | User | |
| **Goal:** | To create an account for the system. | |
| **Pre-condition:** | 1. The user must not be logged in.  2. The user cannot already have an account registered to their email address. | |
| **Post-condition:** | 1. A new account is created for the user. | |
| **Constraints/Issues/Risks:** | 1. An account under a different email may already exist.  2. False information could be used by the user. | |
| **Trigger Event(s):** | 1. The user clicks on “Create Account” button. | |
| **Primary Path:** | Actor: | System: |
| 1. Clicks “Create Account” button.  3. Enters required information and clicks “Create” button. | 2. Displays “Create Account” page.  4. Queries database to ensure account doesn’t exist.  5. Saves information and notifies user of result.  6. Returns to “Main” page. |
| **Alternative Path(s):** | 3. Clicks “Cancel” button. | 4. Returns to “Main” page. |
|  | 5. Notifies user of pre-existing account, and displays “Login” page. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC16 | |
| **Name:** | Search Games | |
| **Actor(s):** | User, Customer | |
| **Goal:** | To provide users/customers with the ability to search the store’s inventory for a particular game. | |
| **Pre-condition:** | 1. The user/customer is on the “Games” page. | |
| **Post-condition:** | 1. The search results are displayed for the user. | |
| **Constraints/Issues/Risks:** | 1. Typographical errors, short forms, or slang used in search parameters provided by the user/customer. Will result in false negative search results (e.g. searching for “Super Mario Brothers” when “Super Mario Bros.” is used in the database).  2. Alternate spellings, or symbols used in the game data stored in database (e.g. Q\*Bert). | |
| **Trigger Event(s):** | User or Customer wants to search for a specific game. | |
| **Primary Path:** | Actor: | System: |
| 1. Clicks the “Search” menu button.  3. Enters search parameters, and clicks “Search.”  6. Clicks “View Game Details. | 2. Displays “Search” page.  4. Performs a database query.  5. Displays results for search on “View Games” page. Includes a “New Search”, “View Game Details” (if the game is found), and “Cancel” buttons.  7. Retrieves game data, and displays for user. |
| **Alternative Path(s):** | 6. Clicks “New Search”. | 7. Returns user to “Search” page. |
| 6. Clicks “Cancel”. | 7. Returns user to “Games” page. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC17 | |
| **Name:** | View Game Details | |
| **Actor(s):** | User, Customer | |
| **Goal:** | To provide the user/customer with detailed information on a specific game. | |
| **Pre-condition:** | 1. The user/customer has selected a game in the store’s inventory.  2. The user/customer is on the “View Games” page. | |
| **Post-condition:** | 1. Additional information for the specified game is displayed for the user/customer. | |
| **Constraints/Issues/Risks:** | 1. No additional information is stored for the game. | |
| **Trigger Event(s):** | The user is interested in more details about a game. | |
| **Primary Path:** | Actor: | System: |
| 1. Selects a game and clicks “View Game Details” button. | 2. Retrieves game data and displays for user/customer. |
| **Alternative Path(s):** | 1. Clicks “Cancel” button. | 2. Returns to “Games” page. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC18 | |
| **Name:** | View Inventory | |
| **Actor(s):** | User, Customer, Employee | |
| **Goal:** | To display the available inventory of the store. | |
| **Pre-condition:** | 1. The user/customer/employee is on the “Games” page. | |
| **Post-condition:** | 1. The inventory is displayed. | |
| **Constraints/Issues/Risks:** | N/A | |
| **Trigger Event(s):** | 1. The user/customer/employee desires to see the available inventory. | |
| **Primary Path:** | Actor: | System: |
| 1. Click’s “View Inventory” button. | 2. Queries database and displays available inventory on “View Games” page. |
| **Alternative Path(s):** | N/A | |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC19 | |
| **Name:** | Add to Cart | |
| **Actor(s):** | User, Customer | |
| **Goal:** | To add an item to the shopping cart. | |
| **Pre-condition:** | 1. User/Customer is on the “View Games” page.  2. An item is selected. | |
| **Post-condition:** | 1. The item is added to the user’s/customer’s cart. | |
| **Constraints/Issues/Risks:** | 1. The required quantity must be available in inventory. | |
| **Trigger Event(s):** | 1. The user/customer desires to add an item to their cart. | |
| **Primary Path:** | Actor: | System: |
| 1. Clicks “Add to Cart” button. | 2. Queries database to ensure stock is available.  3. Adds item(s) to user’s/customer’s cart. |
| **Alternative Path(s):** | 4. Clicks “Cancel” button. | 3. Notifies customer/user of stock deficit, and amount available if applicable.  5. Returns to “View Games” page. |
| 4. Clicks “Add Available” button. | 5. Adds remaining available stock to cart, notifies user, and displays “View Cart” page. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC20 | |
| **Name:** | Remove from Cart | |
| **Actor(s):** | User, Customer | |
| **Goal:** | To remove an item from the user’s/customer’s cart. | |
| **Pre-condition:** | 1. The user/customer is on the “View Cart” page.  2. An item is selected. | |
| **Post-condition:** | 1. The item is removed from the cart. | |
| **Constraints/Issues/Risks:** | N/A | |
| **Trigger Event(s):** | The user/customer desires to remove an item from their cart. | |
| **Primary Path:** | Actor: | System: |
| 1. Clicks “Remove from Cart” button.  3. Clicks “Confirm” button. | 2. Displays message box requesting confirmation of removal.  4. Removes item from cart. |
| **Alternative Path(s):** | 3. Clicks “Cancel” button. | 4. Returns to “View Cart” page. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC21 | |
| **Name:** | Check Out | |
| **Actor(s):** | User, Customer | |
| **Goal:** | To place an order for items in cart. | |
| **Pre-condition:** | 1. User/Customer is on the “View Cart” page  2. Items are in the cart. | |
| **Post-condition:** | 1. The order is placed. | |
| **Constraints/Issues/Risks:** | 1. Login in required for processing. | |
| **Trigger Event(s):** | 1. User/Customer desires to process their order. | |
| **Primary Path:** | Actor: | System: |
| 1. Clicks on “Check Out” button. | 2. Checks if submitter is logged in.  3. Saves cart items and creates an order. It adds the new order to open orders list, notifies customer of result, and emails confirmation to customer.  4. Returns to “Main” page. |
| **Alternative Path(s):** |  | 3. Saves cart items, and directs user to “Login” page and requests that user login or create an account to process order. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC22 | |
| **Name:** | Login | |
| **Actor(s):** | Customer, Employee | |
| **Goal:** | To provide access to restricted features. | |
| **Pre-condition:** | 1. The individual has an account.  2. The account is active.  3. The individual knows the correct login and password. | |
| **Post-condition:** | 1. Access is granted. | |
| **Constraints/Issues/Risks:** | 1. The account is inactive.  2. The account is locked.  3. An individual misrepresents themselves as another. | |
| **Trigger Event(s):** | 1. The customer/employee desires access to the site. | |
| **Primary Path:** | Actor: | System: |
| 1. Provides login name and password and clicks “Login” button. | 2. Verifies credentials.  3. Provides access to site.  4. Returns to “Main” page. |
| **Alternative Path(s):** |  | 3. Notifies customer/employee of problem.  4. Returns to “Login” page. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC23 | |
| **Name:** | Edit Profile | |
| **Actor(s):** | Customer | |
| **Goal:** | To allow customers to enter personal preferences and details to their profile. | |
| **Pre-condition:** | 1. The customer is logged in.  2. The customer is on the “Profiles” page. | |
| **Post-condition:** | 1. The profile is updated. | |
| **Constraints/Issues/Risks:** | 1. Inappropriate information is entered. | |
| **Trigger Event(s):** | 1. The customer desires to provide personal information about their interests and personality. | |
| **Primary Path:** | Actor: | System: |
| 1. The customer clicks “Edit Profile” button.  3. Enters information and clicks “Save” button. | 2. Displays existing user data.  4. Updates user data in database and notifies user of result.  5. Returns to “Profiles” page. |
| **Alternative Path(s):** | 3. Clicks “Cancel” button. | 4. Returns to “Profiles” page. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC24 | |
| **Name:** | Change Password | |
| **Actor(s):** | Customer, Employee | |
| **Goal:** | To change the customer’s/employee’s password. | |
| **Pre-condition:** | 1. The customer/employee is logged in.  2. The customer/employee is on the “Profiles” page.  3. The customer/employee knows their current password. | |
| **Post-condition:** | 1. The password is updated. | |
| **Constraints/Issues/Risks:** | 1. An individual could be misrepresenting the actual customer/employee and effectively locking them out of their own account. | |
| **Trigger Event(s):** | 1. The customer/employee desires to change their password. | |
| **Primary Path:** | Actor: | System: |
| 1. Clicks on “Change Password” button.  3. Enters correct password.  5. Enters and confirms new password. | 2. Requests that user confirm current password.  4. Requests new password and confirmation.  6. Password is updated. Notification of result is displayed to customer/employee.  7. Returns to “Profiles” page. |
| **Alternative Path(s):** | 3. Enters incorrect password. | 4. Notifies customer/employee of mismatch. Allows 3 attempts before locking account. |
| 3. Clicks “Cancel” | 4. Returns to “Profiles.” |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC25 | |
| **Name:** | Reset Password | |
| **Actor(s):** | Customer, Employee | |
| **Goal:** | To reset a customer’s/employee’s forgotten password. | |
| **Pre-condition:** | 1. The email address of the individual is in the database. | |
| **Post-condition:** | 1. A new temporary password is sent to the email address on file. | |
| **Constraints/Issues/Risks:** | 1. The customer/employee gives the wrong email address.  2. The customer/employee no longer has access to that address. | |
| **Trigger Event(s):** | 1. The customer is unable to login and cannot recall their password. | |
| **Primary Path:** | Actor: | System: |
| 1. Clicks “Reset Password” button.  3. Enters address and clicks “Submit” button. | 2. Requests email address associated with account.  4. Searches for address.  5. Resets password, sends temporary password to address, notifies customer/employee of result.  6. Returns to “Login” page. |
| **Alternative Path(s):** |  | 5. Notifies customer/employee that email address is not found.  6. Returns to “Login” page. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC26 | |
| **Name:** | Download | |
| **Actor(s):** | Customer | |
| **Goal:** | To download an available game. | |
| **Pre-condition:** | 1. The customer is logged in.  2. The selected game has downloadable content.  3. The customer is on the “View Game Details” page. | |
| **Post-condition:** | 1. The customer downloads the game. | |
| **Constraints/Issues/Risks:** | 1. The system is able to locate the file.  2. The file could be moved or deleted.  3. The customer selects the appropriate version for their operating system. | |
| **Trigger Event(s):** | 1. The customer desires to download the file. | |
| **Primary Path:** | Actor: | System: |
| 1. Clicks “Download Game” button.  3. Selects location to save the file to, and clicks “Download”. | 2. Retrieves the file and presents a “Save As” dialog box.  4. File is transferred to customer, and the customer is notified once complete.  5. Returns to “View Games” page. |
| **Alternative Path(s):** | 3. Clicks “Cancel” button. | 4. Returns to “View Games” page. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC27 | |
| **Name:** | Rate Game | |
| **Actor(s):** | Customer | |
| **Goal:** | To provide a rating for a selected game. | |
| **Pre-condition:** | 1. Customer is logged in.  2. Customer is on the “View Game Details” page. | |
| **Post-condition:** | 1. Rating is saved for the game. | |
| **Constraints/Issues/Risks:** | 1. The rating could already exist. | |
| **Trigger Event(s):** | 1. Customer desires to rate a game. | |
| **Primary Path:** | Actor: | System: |
| 1. Clicks “Rate Game” button.  4. Enters rating and clicks “Save Rating” button. | 2. Checks if customer has already rated game.  3. Displays rating entry field for customer.  5. Saves rating.  6. Returns to “View Game Details” page. |
| **Alternative Path(s):** |  | 3. Advises user a rating already exists.  4. Returns to “Edit Rating” page. |
| 4. Clicks “Cancel” button. | 5. Returns to “View Game Details” page |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC28 | |
| **Name:** | Edit Rating | |
| **Actor(s):** | Customer | |
| **Goal:** | To update a pre-existing game rating created by the customer. | |
| **Pre-condition:** | 1. Customer is logged in.  2. The customer is on the “View Game Details” page of a specific game.  3. The customer has already rated the selected game. | |
| **Post-condition:** | 1. The rating is updated. | |
| **Constraints/Issues/Risks:** | N/A | |
| **Trigger Event(s):** | 1. The customer has decided to alter their rating of a particular game. | |
| **Primary Path:** | Actor: | System: |
| 1. Clicks “Edit Rating” button.  3. Modifies rating and clicks “Update” button. | 2. Displays rating entry field for customer populated with prior rating.  4. Saves rating.  5. Returns to “View Game Details” page. |
| **Alternative Path(s):** | 3. Clicks “Cancel” button. | 4. Returns to “View Game Details” page. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC29 | |
| **Name:** | Create Game Review | |
| **Actor(s):** | Customer | |
| **Goal:** | To create a game review for a specific game. | |
| **Pre-condition:** | 1. Customer is logged in.  2. Customer is on “Reviews” page. | |
| **Post-condition:** | 1. The customer’s review is saved. | |
| **Constraints/Issues/Risks:** | 1. The customer may have never played or purchased the game.  2. The content may be inappropriate. | |
| **Trigger Event(s):** | 1. The customer desires to express his/her opinions and experiences related to a specific game. | |
| **Primary Path:** | Actor: | System: |
| 1. Clicks “Create Review” button.  3. Enters details and clicks “Create” button. | 2. Displays input fields for customer.  4. Saves review and adds to “Pending Reviews” list.  5. Returns to “Reviews” page. |
| **Alternative Path(s):** | 3. Clicks “Cancel” button. | 4. Returns to “Reviews” page. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC30 | |
| **Name:** | Edit Game Review | |
| **Actor(s):** | Customer | |
| **Goal:** | To update an existing review written by the customer. | |
| **Pre-condition:** | 1. The customer is logged in.  2. The customer is on the “Reviews” page. | |
| **Post-condition:** | 1. The review is updated. | |
| **Constraints/Issues/Risks:** | 1. The customer must be the author of the review.  2. Reviews are subject to employee review. | |
| **Trigger Event(s):** | 1. The customer wishes to modify their review of a game. | |
| **Primary Path:** | Actor: | System: |
| 1. Clicks on “Edit Game Review” button.  3. Selects review and clicks “Edit”.  5. Modifies review and clicks “Update” button. | 2. Displays reviews written by the customer.  4. Displays input fields for customer with review data populated.  6. Saves review and adds to “Pending Reviews” list.  7. Returns to “Reviews” page. |
| **Alternative Path(s):** | 5. Clicks “Cancel” button. | 6. Returns to “Reviews” page. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC31 | |
| **Name:** | Delete Game Review | |
| **Actor(s):** | Customer | |
| **Goal:** | To delete a review. | |
| **Pre-condition:** | 1. The customer is logged in.  2. The customer is on the “Reviews” page. | |
| **Post-condition:** | 1. The review is deleted from the database. | |
| **Constraints/Issues/Risks:** | 1. The customer must be the author of the review. | |
| **Trigger Event(s):** | 1. The customer wishes to delete their review of a game. | |
| **Primary Path:** | Actor: | System: |
| 1. Clicks “Delete Game Review” button.  3. Clicks “Confirm” button. | 2. Displays a message box requiring the customer to confirm deletion.  4. Deletes the review in the database and notifies the customer of the result.  5. Returns to “Reviews” page. |
| **Alternative Path(s):** | 3. Clicks “Cancel” button. | 4. Returns to “Reviews” page. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC32 | |
| **Name:** | Add Friend/Family | |
| **Actor(s):** | Customer | |
| **Goal:** | To add another customer who is a friend or family member to their list. | |
| **Pre-condition:** | 1. The customer is logged in.  2. The customer is on the “View Profile” page for another customer. | |
| **Post-condition:** | 1. The friend/family member is added to the customer’s list. | |
| **Constraints/Issues/Risks:** | 1. The friend/family member must be a different customer.  2. The friend/family member is not already on the customer’s list. | |
| **Trigger Event(s):** | 1. The customer desires to add a friend or family member to their list. | |
| **Primary Path:** | Actor: | System: |
| 1. Clicks on “Add Friend” button.  3. Clicks “Confirm Friend Request” button. | 2. Requests confirmation from customer.  4. Sends confirmation request to customer that was added.  5. Notifies current customer that request was sent.  6. Returns to “View Profile” page. |
| **Alternative Path(s):** | 3. Clicks “Cancel” button. | 6. Returns to “View Profile” page. |
| **Exception Path(s):** |  | 4. Notifies customer that the friend/family member is already on their list.  5. Returns to “View Profile” page. |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC33 | |
| **Name:** | Delete Friend/Family | |
| **Actor(s):** | Customer | |
| **Goal:** | To remove a friend/family member from the customer’s list. | |
| **Pre-condition:** | 1. The customer is logged in.  2. The customer is on the “View Friends” page. | |
| **Post-condition:** | 1. The friend/family member is removed from the list. | |
| **Constraints/Issues/Risks:** | 1. There must be a friend/family member on the list. | |
| **Trigger Event(s):** | 1. The customer desires to remove a friend/family member from their list. | |
| **Primary Path:** | Actor: | System: |
| 1. Clicks “Delete” button.  3. Clicks “Confirm” button. | 2. Requests confirmation for deletion.  4. Deletes friend/family member from the database and removes from customer’s list.    5. Returns to “View Friends” page. |
| **Alternative Path(s):** | 3. Clicks “Cancel” button. | 4. Returns to “View Friends” page. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC34 | |
| **Name:** | Add Credit Card | |
| **Actor(s):** | Customer | |
| **Goal:** | To add a credit card to the customer’s account. | |
| **Pre-condition:** | 1. The customer is logged in.  2. The customer is on the “Profiles” page. | |
| **Post-condition:** | 1. The credit card is added to the customer’s account. | |
| **Constraints/Issues/Risks:** | 1. The customer may not be the legitimate owner of the card.  2. The card isn’t already on the customer’s list. | |
| **Trigger Event(s):** | 1. The customer desires to add a credit card to their account for purchases. | |
| **Primary Path:** | Actor: | System: |
| 1. Clicks “Credit Cards” button.  3. Clicks “Add Credit Card” button.  5. Enters data, and clicks “Save” button. | 2. Displays “Credit Cards” page.  4. Displays “Add Credit Card” page, providing input fields for data entry.  6. Verifies the payment information.  7. Saves data in database, and notifies customer of the result.  8. Returns to “Credit Cards” page. |
| **Alternative Path(s):** | 5. Clicks “Cancel” button. | 6. Returns to “Credit Cards” page. |
| **Exception Path(s):** |  | 7. Notifies customer that information was entered incorrectly, or other issue was present.  8. Requests that customer revise the data entered, and redisplays the form populated with entered data and error messages. |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC35 | |
| **Name:** | Delete Credit Card | |
| **Actor(s):** | Customer | |
| **Goal:** | To remove a credit card from the customer’s account. | |
| **Pre-condition:** | 1. The customer is logged in.  2. The customer is on the “Profiles” page.  3. A card is saved on the customer’s account. | |
| **Post-condition:** | 1. The customer’s credit card is removed from their list. | |
| **Constraints/Issues/Risks:** | 1. A credit card must already exist on file. | |
| **Trigger Event(s):** | 1. The customer desires to remove a card from their account. | |
| **Primary Path:** | Actor: | System: |
| 1. Clicks “Credit Cards” button.  3. Selects a card from the list, and clicks “Delete Credit Card” button.  5. Clicks “Confirm” button. | 2. Displays the “Credit Cards” page.  4. Requests confirmation of deletion.  6. Removes card from list and deletes from the database.  7. Returns to “Credit Cards” page. |
| **Alternative Path(s):** | 5. Clicks “Cancel’ button. | 6. Returns to “Credit Cards” page. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC36 | |
| **Name:** | Add Address | |
| **Actor(s):** | Customer | |
| **Goal:** | To add a billing or shipping address to the customer’s account. | |
| **Pre-condition:** | 1. The customer is logged in.  2. The customer is on the “Profiles” page. | |
| **Post-condition:** | 1. The address is added to the customer’s account. | |
| **Constraints/Issues/Risks:** | 1. The address could be entered incorrectly.  2. The address isn’t already on file. | |
| **Trigger Event(s):** | 1. The customer desires to add an address to their account. | |
| **Primary Path:** | Actor: | System: |
| 1. Clicks “Addresses” button.  3. Clicks “Add Address” button.  5. Enters data, and clicks “Save” button. | 2. Displays “Addresses” page.  4. Displays “Add Address” page with input fields.  6. Saves data and displays notification of result.  7. Returns to “Addresses” page. |
| **Alternative Path(s):** | 6. Clicks “Cancel” button. | 7. Returns to “Addresses” page. |
| **Exception Path(s):** |  | 6. Notifies customer of missing or invalid fields, or other issue.  7. Requests that customer revise the data entered, and redisplays the form populated with entered data with validation error messages displayed where applicable. |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC37 | |
| **Name:** | Delete Address | |
| **Actor(s):** | Customer | |
| **Goal:** | To Remove an address from the customer’s account. | |
| **Pre-condition:** | 1. The customer is logged in.  2. The customer is on the “Profiles” page.  3. An address exists on the customer’s account. | |
| **Post-condition:** | 1. The address is removed from the account. | |
| **Constraints/Issues/Risks:** | N/A | |
| **Trigger Event(s):** | 1. The customer desires to remove an address from their account. | |
| **Primary Path:** | Actor: | System: |
| 1. Clicks “Addresses” button.  3. Selects card, and clicks “Delete” button.  5. Clicks “Confirm” button. | 2. Displays the “Addresses” page with a listing of available cards.  4. Requests confirmation of deletion from customer.  6. Removes address from list and deletes from database and notifies customer of result.  7. Returns to “Addresses” page. |
| **Alternative Path(s):** | 5. Clicks “Cancel” button. | 6. Returns to “Addresses” page. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC38 | |
| **Name:** | Register for Event | |
| **Actor(s):** | Customer | |
| **Goal:** | To register for an upcoming event. | |
| **Pre-condition:** | 1. The customer is logged in.  2. The customer is on the “Events” page.  3. The customer isn’t already registered for selected event. | |
| **Post-condition:** | 1. The customer is registered to attend the event. | |
| **Constraints/Issues/Risks:** | 1. A customer can only register once for an event.  2. Space is available for the event.  3. A customer cannot register for an event that has passed or already started. | |
| **Trigger Event(s):** | 1. The customer desires to register for an event. | |
| **Primary Path:** | Actor: | System: |
| 1. Selects an event from “Upcoming Events” list, and clicks “Register” button.  3. Clicks “Confirm” button. | 2. Requests confirmation to register from customer.  4. Adds customer to event, and emails notification to customer and event administrator.  5. Returns to “Events” page. |
| **Alternative Path(s):** | 4. Clicks “Cancel” button. | 5. Returns to “Events” page. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC39 | |
| **Name:** | View Wish List | |
| **Actor(s):** | Customer | |
| **Goal:** | To view the customer’s wish list. | |
| **Pre-condition:** | 1. The customer is logged in.  2. The customer is on the “Main” page. | |
| **Post-condition:** | 1. The customer’s wish list items are displayed. | |
| **Constraints/Issues/Risks:** | 1. An item can only be added once, but quantity can be modified.  2. The item may no longer be in stock.  3. The item may have been discontinued since it was added to the list. | |
| **Trigger Event(s):** | 1. A customer desires to view the items on their wish list. | |
| **Primary Path:** | Actor: | System: |
| 1. Clicks “Wish List” button. | 2. Returns to the “Wish List” page including the list of items. |
| **Alternative Path(s):** | N/A | |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC40 | |
| **Name:** | Add to Wish List | |
| **Actor(s):** | Customer | |
| **Goal:** | To add an item to the customer’s wish list. | |
| **Pre-condition:** | 1. The customer is logged in.  2. The customer is on the “View Games” page.  3. The item isn’t already on the list. | |
| **Post-condition:** | 1. The item is added to the customer’s wish list. | |
| **Constraints/Issues/Risks:** | 1. An item can only be added once to the list. | |
| **Trigger Event(s):** | 1. The customer desires to add an item to their list to be purchased at a later date. | |
| **Primary Path:** | Actor: | System: |
| 1. Selects a game, and clicks “Add to Wish List” button.  3. Clicks “Confirm” button. | 2. Requests confirmation of addition to customer.  4. Adds item to customer’s wish list and notifies them of the result.  5. Returns to “View Games” page. |
| **Alternative Path(s):** | 3. Clicks “Cancel” button. | 4. Returns to “View Games” page. |
| **Exception Path(s):** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC41 | |
| **Name:** | Remove from Wish List | |
| **Actor(s):** | Customer | |
| **Goal:** | To remove an item from the wish list. | |
| **Pre-condition:** | 1. The customer is logged in.  2. The customer is on the “Main” page.  3. The customer has items on their wish list. | |
| **Post-condition:** | 1. The item is removed. | |
| **Constraints/Issues/Risks:** | N/A | |
| **Trigger Event(s):** | 1. The customer desires to remove an item from their wish list. | |
| **Primary Path:** | Actor: | System: |
| 1. Clicks “Wish List” button.  3. Selects item and clicks “Delete” button.  5. Clicks “Confirm” button. | 2. Displays “Wish List” page.  4. Requests confirmation of deletion from customer.  6. Deletes item from list and notifies customer of the result.  7. Returns to “Wish List” page. |
| **Alternative Path(s):** | 5. Clicks “Cancel” button. | 6. Returns to “Wish List” page. |
| **Exception Path(s):** | N/A | |