**CSE 308**

**Final Requirements Document**

**Team Name:** Good Looks Books

**Team Members:** Paul Mannarino

Kevin Young

Kevin Setayesh

Hamza Hussaini

1. **User Roles**

|  |  |
| --- | --- |
| **Role** | **Description** |
| Guest | Guests are any site users who are not yet registered nor logged into the site. Guests are able to browse and search available items. They are also able to access user registration. |
| Registered Customer | Registered Customers have all access and functionality available to Guests with added  customer related actions. These include additional options when viewing a item such as “Borrow”, “Add to wishlist”, etc., viewing items currently owned or rented by the user, and editing account settings and details. Registered Customer role is for accessing and interacting with library items. |
| Site Librarian | A Site Librarian will have elevated permissions on the site such as looking at checked out items and being able to “check in” the items for the customer who has the items checked out. |
| Publisher | Publishers will be able to view trends and data related to their items. If they see discrepancies with their item information or would like to edit the information, they may edit the information.Their role is to be able to verify and monitor their item’s information. |
| Site Admin | The site admin will have full access to the website with the exception of sensitive Librarian, Publisher, and Customer data. The Site Admin's role is monitor the site's integrity. An admin has roles such as adding a book, generating a bestseller list, and resolving tickets. |

**2. List of Use Cases**

1. Title: UserLogIn
   1. Description: A user (Librarian, Customer, SiteAdmin, or Publisher) with a preexisting account logs in to the website.
   2. Note: We don’t need a use case for each kind of user because the homepage that the user is brought to, no matter what kind of user it is, is brought to a homepage with static data.
2. Title: CustomerSignUp
   1. Description: If a customer’s name isn’t in the database, it will create a new customer with all the information required and add it to the “Users” table in the database.
3. Title: ItemSearch
   1. Description: A user searches the database for an item with the same name.
4. Title: ItemTakeOut
   1. Description: Clicking on an item to take it out, will send the item and user information to the back end and update the respective table.
5. Title: UserLogOut
   1. Description: User logs out of the system from the website.
6. Title: PublisherSignup
   1. Description: When a publisher's account isn’t the database, create a new publisher user to add to database.
7. FavoriteItem
   1. Customer favorites an item
8. Title: ItemSocialMediaShare
   1. Description: The user clicks the “Share” button on an item in order to share it on a social media site.
9. Title: ItemSample
   1. Description: The user wishes to sample an item.
10. Title: WishListAdd
    1. Description: User picks an item they are interested in getting onto a list that is called “wish list”.
11. Title: ItemRate
    1. Description: User rates an item based on a 5 star scale
12. Title: PublisherUpdatesPublishedItem
    1. Description: A publisher edits one of their item’s details
13. Title: ItemReserve
    1. Description: A user reserves an item from the website.
14. Title: DownloadRequest
    1. Description: A user decides on a book and now wishes to download it. The library site makes a request to the server for the desired book, seeking its availability.
15. ViewItem
    1. Description: A User or Guest clicks a link for a book and is sent to a page that requests the information from the database on that particular book. The information is loaded into the page.
16. Title: ViewWishList
    1. Description: A registered User of the site requests to see a list of books that are currently in their wishlist. The user is sent to a page where the server will load the page with images and summaries of the books on said user’s wishlist.
17. ItemRatingRemove
18. Title: WishListItemRemove
    1. Description: A User views their wishlist and selects the option to remove one. This request is sent to the server where it is confirmed the book was actually on their wishlist, and then the system removes it. The book is no longer on their wishlist.
19. Title: ItemReturn
    1. Description: A User has taken out an item from the e-library and decides they are done with it and requests to return it before it is due back. This request is sent to the server, where the system will then set the return date to the time of the request and mark the item returned on time. The system will also remove the user’s access to this item.
20. Title: UserItemList
    1. Description: The user will be able to view all the items they currently own or have rented. The server will be contacted to confirm A) the user still owns this item, B) the item still exists on the server and is accessible.
21. Title: UserAccountRemoval
    1. Description: A User wishes to delete their account. The request is sent to the server and what follows is a dissociation of the user with all items they owned, favorited, or added to their wishlist; this data is not removed completely, but saved in case a user wishes to reopen an account. Reviews remain unchanged.
22. Title: LibrarianSignUp
    1. Description: A librarian signs up to make an account.
23. Title: UserEditInformation
    1. Description: A user changes information about their account on the change account settings page.
24. Title: AdminGetsSiteStatistics
    1. Description: The logged in Admin wants to see all the statistics of the website, such as number of unique visitors and item statistics
25. Title: ItemReviewAdd
    1. Description: The user adds a review to an item.
26. Title: ItemReviewEdit
    1. Description: The user edits a review they previously made on an item.
27. Title: ItemReviewDelete
    1. Description: The user deletes a review they previously made on an item.
28. Title: ItemEmailShare
    1. Description: The user clicks the “Share” button on an item in order to share it in an email.
29. Title: AdminAddItem
    1. Description: The admin adds a book by entering the book information and requesting a license.
30. AdminBestSellerListGeneration
    1. Description: Admin generates a bestseller list.
31. CustomerHomePageView
    1. Description: A customer visits their home page.
32. GuestHomePageView
    1. Description: A guest visits the general site home page.
33. PublisherHomePageView
    1. Description: A publisher visits their home page.
34. LibrarianHomePageView
    1. Description: A librarian visits their home page.
35. AdminHomePageView
    1. Description: Admin visits their home page.
36. ItemBuyItNow
    1. Description: Customer clicks “Buy it Now” button on an item’s page
37. PublisherEditItemsView
    1. Description: A publisher clicks on “Edit Item Data” on the publisher’s home page, which brings the publisher to a page with all of the items this publisher has published with each item having an option to edit that item.

**3. Detailed Use Cases**

|  |  |
| --- | --- |
| Use-Case: | UserLogOut |
| Primary Actor: | User |
| Goal in Context: | User logs out of their session |
| Preconditions: | The Useris logged in. |
| Trigger: | The User clicks on the “Sign Out” button |
| Scenario: | 1. User clicks “Sign Out” button  2. User is logged out  3. The GUI is brought to the guest index page |
| Exceptions: | 1. The User has multiple tabs open when they are still logged in and logs out on one tab, and then logs out on another tab which has not been refreshed since logging off. |
| Priority: | Essential, must be implemented. |
| When available: | First increment. |
| Frequency of use: | Often |
| Channel to actor: | Via web browser interface. |
| Secondary actors: | Server, Admin |
| Channels to secondary actors: | Admin: web browser interface, program modification.  Server: network and local interface |
| Open Issues: | 1. If the User is signed on multiple computers and signs out on one computer, should the page be automatically refreshed on the other computer(s) to show that the user is logged out. |

|  |  |
| --- | --- |
| Use-Case: | ItemBuyItNow |
| Primary Actor: | Customer |
| Goal in Context: | Bring customer to server that acts like a 3rd party website where the item will be downloaded. |
| Preconditions: | The customer is logged in and viewing an item on its page |
| Trigger: | The customer clicks on the “Buy It Now” button. |
| Scenario: | 1. Customer goes to the page of the item  2. Customer clicks “Buy It Now” button  3. Customer is brought to server that acts like a 3rd party website  4. Customer downloads book from this server |
| Exceptions: | 1. The other server has an error preventing the customer from successfully buying the book. |
| Priority: | Essential, must be implemented. |
| When available: | First increment. |
| Frequency of use: | Occasionally |
| Channel to actor: | Via web browser interface. |
| Secondary actors: | Server, Admin |
| Channels to secondary actors: | Admin: web browser interface, program modification.  Server: network and local interface |
| Open Issues: | 1. How to successfully communicate between our server and the 3rd party website server  2. What information will we send to the 3rd party website server so that it knows what item to go to. |

|  |  |
| --- | --- |
| Use-Case: | CustomerLogIn |
| Primary Actor: | Customer |
| Goal in Context: | Logs the specified customer in to the website |
| Preconditions: | The actor has been authenticated and identified as a library card holder |
| Trigger: | The user clicks on the “Log In” button. |
| Scenario: | 1. Customer types in username or barcode.  2. Customer types in password.  3. Customer clicks the “Log In” button. |
| Exceptions: | 1. If user types in an incorrect username/barcode or password, prompts the user that the incorrect information will be displayed underneath the login button.  2. If user types in an incorrect format for the username/barcode or password, such as an empty field, than the user will be prompted underneath the login button that the information is in an incorrect format. |
| Priority: | Essential, must be implemented. |
| When available: | First increment. |
| Frequency of use: | Many times a day. |
| Channel to actor: | Via web browser interface. |
| Secondary actors: | Server, Admin |
| Channels to secondary actors: | Admin: web browser interface, program modification.  server: network and local interface |
| Open Issues: | 1. Where on the webpage to have the user log in their information along with the log in button.  2. How to prompt the user for exceptions such as incorrect username/password.  3. Should we have a “show” button to show the password being typed instead of ellipses?  4. If the customer is already logged in on one system, such as on desktop, should they be able to log in from another system, such as on a different desktop? |

|  |  |
| --- | --- |
| Use-Case: | ItemSearch |
| Primary Actor: | Customer |
| Goal in Context: | To search for an item via search bar |
| Preconditions: | The user must be logged in already to have access to the page. |
| Trigger: | The user clicks the “search” button |
| Scenario: | 1.User types in item they are searching for  2.User clicks search  3.The item the user searched for appears on the webpage |
| Exceptions: | 1. There are no items that match the given search terms and criteria (the message “No matching items could be found” will be displayed below the search fields)  2. The user pressed the search button without typing anything into the search bar. |
| Priority: | Essential, must be implemented. |
| When available: | First increment. |
| Frequency of use: | Many times per day. |
| Channel to actor: | Via web browser interface. |
| Secondary actors: | server, Admin |
| Channels to secondary actors: | Admin: web browser interface, program modification.  server: network and local interface |
| Open Issues: | 1. Where on the web interface will the search fields and buttons be displayed?  2. What other criteria will the Customer want to search by?  3. Should we have a “Clear Fields” button that clears all entered text in the search fields? |

|  |  |
| --- | --- |
| Use-Case: | ItemRate |
| Primary Actor: | Customer |
| Goal in Context: | Rate the item based on preference. There will be a system consisting of 5 stars. |
| Preconditions: | The user must be logged in already to have access to the page and has selected an item. |
| Trigger: | The user clicks on the amount of stars they believe the item should be rated by |
| Scenario: | 1. Customer: observes an item  2. Customer: selects a 1-5 star rating on the observed item |
| Exceptions: | 1.The user has already rated the item, in that case the action in updating the database will be different than if the item hasn’t already been rated by that specific user. |
| Priority: | Not necessarily essential |
| When available: | First increment. |
| Frequency of use: | Many times per day |
| Channel to actor: | Via web browser interface. |
| Secondary actors: | Admin, server |
| Channels to secondary actors: | Admin: web browser interface, program modification.  server: network and local interface |
| Open Issues: | 1.Should the user be able to rate any item available on the website, or only the items they have checked out / bought.  2.Should we dynamically change the average rating of the item when a user inputs a rating  3.If an item has 0 ratings, what should it’s default rating be (0 or 5). |

|  |  |
| --- | --- |
| Use-Case: | UserLogOut |
| Primary Actor: | Customer, Librarian, Publisher, Admin |
| Goal in Context: | Log user out of account. |
| Preconditions: | The user must already be logged in |
| Trigger: | The user selects the “Log Out” button on the website |
| Scenario: | 1.User is logged in, and is on any of the pages on the site.  2.Clicks on logout button  3.The user is then sent to the sign in page, where they no longer have access to the website unless they log back in. |
| Exceptions: | Will be no exceptions because the user would have to be logged in to see the “Log Out” button |
| Priority: | Essential, must be implemented. |
| When available: | First increment. |
| Frequency of use: | Many times per day. |
| Channel to actor: | Via web browser interface. |
| Secondary actors: | Admin, server |
| Channels to secondary actors: | Admin: web browser interface, program modification.  server: network and local interface |
| Open Issues: | 1. If a customer is logged in on the website on 2 or more different systems, if one of the systems logs out, should the other system logout as well or should they remain signed in, and if they are logged out when should take place. |

|  |  |
| --- | --- |
| Use-Case: | WishListAdd |
| Primary Actor: | Customer |
| Goal in Context: | Add a selected item to a wishlist. |
| Preconditions: | User must have clicked on an item they want to add to their wish list |
| Trigger: | User selects on the “Add to wish list” button that is on the page of the item that the user clicked on |
| Scenario: | 1. User observes the list of items..  2. User selects the item.  3. User clicks on the add to wishlist button on the selected item’s page. |
| Exceptions: | The item has already been added to that customer's wish list. |
| Priority: | Not necessarily essential |
| When available: | First increment. |
| Frequency of use: | A few times per day. |
| Channel to actor: | Via web browser interface. |
| Secondary actors: | Admin, server |
| Channels to secondary actors: | Admin: web browser interface, program modification.  server: network and local interface |
| Open Issues: | 1. Should there be a limit to how many items can be on a single wishlist?  2. If a customer takes out or purchases an item that is on their wishlist, should it be automatically removed from the wishlist?  3. Which users, if any at all, should be able to view another user’s wishlist? |

**4. Issues Statement**

**Risks:**

1. Having several users simultaneously logged on to the system.

2. Redirecting to third-party websites when the user wants to do certain actions such as share an item on a social media site.

3. Having more than one user trying to check out the same item at the same time when there is limited stock.

4. Where will items be hosted for when a user wants to download that item.

5. How to deal with too much traffic on the website.

6. What should we do with items that are no longer available.

7. What to do with users items if they decide to deactivate their account.

8. Making sure our application will run similarly on all browsers.

9. Making sure each user has the proper permissions so they are only allowed to access parts of the system based on their role.

**Ways to reduce risks:**

1. The issue of having several users logging simultaneously into the system can be solve by making a normalized database with efficient queries that way communication with the backend will be quick.

2. We need to first gain permission to use the servers of these third-party websites. We will also have to find a way to properly transfer the item information to this third-party website in a way that the website can understand. Our information may be stored differently than how these third-party websites can recognize information. Therefore, we need to make sure the transfer of information is fluid so the third-party website can use the information properly.

3. When users check out the same item at the same time this can lead to issues such as lag

in the website, a way this can be solved is by having a queue that processes each query

one at a time and will not move onto the next process till the current one is completed,

than it will be popped out of the queue and move onto the next one.

4. This can be a risk because in order for a user to download an item, that item first needs to be hosted somewhere. For this risk we will need a dedicated server with some decent bandwidth capabilities in order to handle the load of multiple users possibly downloading simultaneously.

5. This can be a possible risk because the site may crash if too many users are on the site. If the site doesn’t crash it may become very slow and lag a bit which would also basically make the site unusable for the user. Some ways to solve this problem include server-side caching and possibly compressing our web page content. Server side caching would help in that a page could be pre-built so the webpage wouldn't need to be reloaded every time. Compressing our content would reduce load on our site and deliver content quicker.

6. An item being no longer available can pose a problem because a user cannot be allowed to select it. Otherwise there will be errors in the database involving this item, and the user may be promised something they will surely not receive. To solve this risk we can simply gray out unavailable items or let the user know that those specific items are not available at the time if the user selects it.

7. This can be a risk because it will create problems once again in our database involving item numbers. To solve this risk, when a user deactivates their account we can simply send all items they have checked out to be made available. This will ensure the number of items being correct and aligned with what is actually available.

8. This is a risk that is seen often in sites. On one browser the site may run fine while in another some part of the site will be buggy or not work altogether. To solve this risk we will use the jquery framework in our frontend. This framework is cross compatible among all major web browsers.

9. This can be a risk because all users cannot be allowed equal access to different parts of the site. Clearly a publisher and a regular member will not have the same permissions. To solve this, we can simply make a different account page for each user. On top of that, if either user is allowed special permissions into different parts of the site on pages other than the account page, we can simply make it so that those links only show up for the user who is permitted to access that section of the site.

**5. MIS Requirements Remarks**

Our team believes that the MIS requirements ask for enough in order to make a functional and operative website. However, we decided to add the feature to not only rate a book, but to favorite a book. Also, we decided that the Admin should be able to see the site’s integrity by viewing such things as user statistics to return information such as the number of unique visitors to the site. We will also add the ability to write a review on an item.