**Project Initiation – Library Management API**

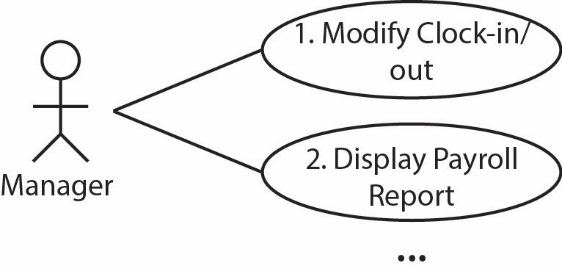
This document provides an overview of the high-level requirements, their priorities, and estimates of their size.

1. **Actors**

Provide a numbered list that names and briefly describes each actor in the system.

1. Guest – guests can search for books and apply for memberships.
2. Members – can checkout books, return, hold, renew, lose books, and have fines.
3. Associates – employees that can make the members, scan books, notify members of holds, and display member information.
4. Managers – suspend and activate accounts, asses overdue fines, make employee accounts, and add, edit, and delete books.
5. **Use Case Diagram**

Provide a use case diagram where the use cases are uniquely numbered.



1. **User Stories**

Provide a user story for each use case in the format shown below. Remove sub-sections (Functional…, Non-…, Notes) if not needed. The numbers (*e.g.* US 1) should correspond to the use cases above.

US 1 – *Applying for Memebership*

*Description… As a guest, I want to apply for membership, so I can check out books.*

**Functional/System Requirements**

*Must have an application for membership, so guests can give it to associates to apply them in.*

**Non-Functional Requirements**

*Description… usernames msut be unique*

**Notes**

*Description…*

US 2 – *Searching for Books*

*Description… As a guest, I want to search for books, so I can determine if I need to become a member.*

**Functional/System Requirements**

*Need to allow non-members to also search.*

US 3 – Checkout Books

*Description… As a member, I want to check out books, so I can read them at home.*

**Functional/System Requirements**

System will record date and time of loan, after associate enters it in.

**Non-Functional Requirements**

**Notes**

Cannot check out more than 10 books at a time.

US 4 – Return books

*Description… As a member, I want to return my books, so I can checkout more.*

*Description… As an associate, I want to return books, so they’ll be available.*

**Functional/System Requirements**

Release member from loan and mark the book as available.

US 5 – Renew book loans

*Description… As a member, I want to renew my book loan, so I can keep the books without overdue fines.*

**Functional/System Requirements**

Can be renewed through the system, or by phone with help of an associate.

US 6 – Pay fines

*Description… As a member, I want pay my fines, so I can check out more books.*

**Functional/System Requirements**

Release member from loan and mark the book as available.

US 7 – Report lost books

*Description… As a member, I want to report my lost book, so I can stop accruing fees.*

**Functional/System Requirements**

1. Suspends account for lost books
2. Stops accruing overdue fees

US 8 – Display member information

*Description… As a member, I want to display the books I have checkout out, and my fines, so I can know.*

**Functional/System Requirements**

Display checked out books and fines associated with account.

US 9 – Place Holds

*Description… As a member, I want to place a book on hold, so I can check it out as soon as it returns.*

**Functional/System Requirements**

Can be done on the system, by member or associate.

Notify member once the book is available.

US 10– Create members

*Description… As an associate, I want to create an account, so I can add more members.*

**Functional/System Requirements**

Username must be unique, and system will reply with a four-digit code.

US 11– Notify members of holds

*Description… As an associate, I want to notify members of holds, so they will come get their book once available.*

**Functional/System Requirements**

Associate sends notification through the system.

US 12– Scans books and library cards

*Description… As an associate, I want to scan books and cards, so I can add keep track of who has what books.*

**Functional/System Requirements**

System records date and time of loan if it’s a check out, if it’s a return it will release the member of their loan.

US 13– Display members’ information

*Description… As an associate, I want to check a members’ account, so I can let them know what they have checked out and any fines.*

**Functional/System Requirements**

Associates must be able to display any member’s information.

US 14– Edit books

*Description… As a manager, I want to edit a book listing, so I can add, edit, or delete a listing.*

**Functional/System Requirements**

Must only be accessible by managers.

US 15– Suspend/Reactivate accounts

*Description… As a manager, I want to suspend or reactivate an account, so I can prevent members from checking out any more books, or reinstate these privileges.*

**Functional/System Requirements**

Must only be available to managers.

US 16– Assess overdue fees

*Description… As a manager, I want to assess overdue fees.*

**Functional/System Requirements**

Allow managers to assess overdue fees.

US 17– Create employee accounts

*Description… As a manager, I want to create an employee account, so my associates can work.*

**Functional/System Requirements**

1. **User Story Prioritization & Size Estimates**

You will do two independent tasks here. First you will prioritize the user stories then you will estimate their size with *story points*. Neither task should influence the other, thus the order you do them is not important. However, the prioritization will probably be quicker.

|  |  |  |  |
| --- | --- | --- | --- |
| US ID | Title | Priority | Size (Story Points) |
| 17 | Create Employee Accounts | 1 | 13 |
| 10 | Create Members | 2 | 8 |
| 2 | Search for Books | 3 | 13 |
| 3 | Check-out books | 4 | 3 |
| 4 | Return Books | 5 | 2 |
| 12 | Scan Books and Library Cards | 6 | 1 |
| 14 | Edit Booklist | 7 | 8 |
| 1 | Apply For Membership | 8 | 2 |
| 8 | Display User Information | 9 | 5 |
| 13 | Display Member Information | 10 | 5 |
| 5 | Renew Book loans | 11 | 2 |
| 6 | Pay Fines | 12 | 3 |
| 7 | Report Lost Books | 13 | 8 |
| 9 | Place Holds | 14 | 3 |
| 11 | Notify Members of Holds | 15 | 2 |
| 15 | Suspend/Reactivate Accounts | 16 | 5 |
| 16 | Assess Overdue Fees | 17 | 3 |

Prioritization:

We placed Create Employee Accounts first because they can do everything, including make user accounts which is why that is so high up. Searching, checking-out, and returning books comes next because these are the main functions of a library. Scanning books and editing the booklist help to run the library smoothly. We added display user and member information next in order to help the next few user stories function, which include the ability to place holds on books and place and pay fines placed on accounts.

Story Point Estimates:

To assign story points we just went down the list of User Stories and verbally discussed story point values using the Fibonacci sequence. When we encountered disagreements, we discussed until a consensus was reached. We assigned different story point values between creating an employee account and creating a member account because employees have more privileges in the system than standard members. Returning books should be easier than checking out books since it will just undo the checkout process. Making a robust search function could be very challenging and take a lot of work so we assigned a very high story point value to that function of the system.