**Sprint 2 Plan**

|  |  |  |  |
| --- | --- | --- | --- |
| **US ID** | **Title** | **Priority** | **Size(Story Points)** |
| 7 | Add Book Listing | 4 | 5 |
| 27 | Search for books | 5 | 8 |
| 28 | Become a Member | 6 | 3 |
| 12 | Scans Book/Library Card | 7 | 3 |
| 15 | Check Out Book | 8 | 2 |
| 17 | Return Books | 9 | 2 |
| 14 | Clock In/Clock Out | 10 | 8 |
| 22 | Display due dates | 11 | 2 |
| 1 | Run Overdue Program | 12 | 8 |
| 23 | Place book on hold | 13 | 2 |
| 19 | Pay a Fine | 14 | 1 |
| 26 | Display Fines | 15 | 1 |
| 21 | Display Checked-out Books | 16 | 3 |
| 24 | Check Hold | 17 | 2 |
| 25 | Display Holds | 18 | 2 |
| 2 | Suspend Member | 19 | 3 |
| 6 | Delete Accounts | 20 | 5 |
| 18 | Renew Books | 21 | 3 |
| 20 | Report Lost Book | 22 | 5 |
| 13 | Display Member's Fine | 23 | 2 |
| 8 | Edit Book Listing | 24 | 2 |
| 9 | Delete Book Listing | 25 | 3 |
| 10 | Edit Employee Clock In/Clock Out | 26 | 5 |
| 11 | Edit Person's Info | 27 | 2 |
| 3 | Reactivate Accounts | 28 | 8 |

**User Stories**

US 1 - *Run Overdue Program*

Description: As a manager I want to run overdue program to display fine information.

**Functional/System Requirements**

1. We need a program to run and assess overdue books
2. Program will display name of member, book, money due, and overdue time

**Non-Functional Requirements**

None

**Notes**

Description

US 2 - *Suspend Member Accounts*

Description: As a Manager I want to suspend member accounts because a member is delinquent in overdue payments or has not returned a book after a given period of time.

**Functional/System Requirements**

1. Member username needed.
2. Place flag on username that indicates suspended account.
3. Suspends ability to checkout books.
4. Suspends ability to place books on hold.
5. Disables/Limits accounts functionality.

**Non-Functional Requirements**

System needs to determine if a suspension is due because of a certain parameter such as delinquent overdue payments or non-returned book.

**Notes**

A member can be suspended if they have overdue fines or if they lost a book. If a member’s fines exceed $25, the account is suspended. They have no functionality.

US 3 - *Reactivate Accounts*

Description: As a Manager I want to Reactivate Accounts because a member’s account may be suspended, and the suspension period may be up.

**Functional/System Requirements**

1. The reason for account suspension should be displayed
2. After reactivation, the member’s account should be totally functional again
3. Enable account if disabled
4. Sends a notification of account reactivation to member

**Non-Functional Requirements**

None

**Notes**

Reactivation can occur if a payment of at least $5 is made and balance is $25 or less

US 6 - *Delete Accounts*

Description: As a Manager I want to delete accounts to remove accounts that are not needed anymore in the system.

**Functional/System Requirements**

1. Manager must login to allow member deletion
2. Disable - Do not delete if account has overdue books, books on hold, or fines due

**Non-Functional Requirements**

None

**Notes**

Description

US 7 - *Add Book Listing*

Description: As a Manager I want to add a book listing because there is a new book that needs to be entered into the system.

**Functional/System Requirements**

1. Title, Author(s), ISBN, Publication year, Keywords, Availability, Quantity are required
2. ISBN must be 10 digits if published before 2007
3. ISBN must be 13 digits if published in or after 2007
4. Checks if book is already in the library; if so, increase inventory amount

**Non-Functional Requirements**

None

**Notes**

Description

US 8 - *Edit Book Listing*

Description: As a Manager I want to edit a book listing to update any information that may have changed, such as versions, etc.

**Functional/System Requirements**

1. Edit title field.
2. Edit ISBN Number field.
3. Edit description field.
4. Edit Author field.
5. Edit Version field.

**Non-Functional Requirements**

Description

**Notes**

Description

US 9 - *Delete Book Listing*

Description: As a Manager I want to delete book listings to remove a book from the system that is no longer available.

**Functional/System Requirements**

1. Manager should login to delete book
2. A notification of deletion of the book
3. Book should be removed entirely from system

**Non-Functional Requirements**

None

**Notes**

Description

US 10 - *Edit Employee Clock In/Clock Out*

Description: As a Manager I want to edit an employee's time sheet because the employee could not properly clock in or clock out.

**Functional/System Requirements**

1. Manager login should be required to change any time
2. A reason must be provided

**Non-Functional Requirements**

None

**Notes**

Description

US 11 - *Edit Person’s Info*

Description: As an Associate I want to enter a person’s info because the person is becoming an associate member of the library and information for the associate is needed.

**Functional/System Requirements**

1. Edit First Name field.
2. Edit Last Name field.
3. Edit Address Field.
4. Edit Telephone Number field.
5. Enter login username.

**Non-Functional Requirements**

New associate needs to receive a PIN number.

**Notes**

Description

US 12 - *Scans Book/Library Card*

Description: As a Associate I want to scan a book or library card because I want to be able to check information on a book or member, or allow a member to check out a book . The associate also can scan the books in the drop-box so they can release a member from a loan and mark the book “available”.

**Functional/System Requirements**

1. Member must have 4 digit code for Associate to “scan” in
2. System records date and time of the loan
3. System displays information on member, and alerts if they have fines, etc.
4. System displays if book is on hold
5. Mark books as available to be checked out if returned

**Non-Functional Requirements**

None

**Notes**

Description

US 13 - *Display Member’s Fine*

Description: As a Associate I want to See a member’s Fine because they may need to know if the member has a fine/no fine .

**Functional/System Requirements**

1. System needs to assess overdue fines based on date.
2. Display how many days overdue.
3. Display fine amount.
4. Display next date that an additional fine will be assessed.
5. Display possible additional fine amount.

**Non-Functional Requirements**

Description

**Notes**

Description

US 14 - *Clock In/Clock Out*

Description: As a Associate I want to Clock-In/Clock-Out because I need keep a log of how many hours I worked on my shift .

**Functional/System Requirements**

1. An associate or manager clocks in and out using their login

**Non-Functional Requirements**

Description

**Notes**

Description

US 15 - *Check Out Book*

Description: As a Member I want to check out a book to take the book home and use it at my leisure.

**Functional/System Requirements**

1. Member’s four digit code is required to check out a book
2. Associate must scan both the book and library card to allow check out
3. Cannot have more than 10 books checked out at once
4. Check out period is 2 weeks
5. When a book is checked out, it should not be available for check out again until returned
6. Can be checked out for two weeks and can be renewed 2 times
7. System records date and time of loan

**Non-Functional Requirements**

Description

**Notes**

Description

US 17 - *Return Books*

Description: As a Member I want to return books so no late fees are charged.

**Functional/System Requirements**

1. Book can be placed in a “dropbox”
2. When associate scans the book in, the system will release the loan
3. Mark the book is available unless there is a hold

**Non-Functional Requirements**

Description

**Notes**

If two weeks have passed and book has not been returned, overdue fine of $.10 per day is charged until book is returned.

US 18 - *Renew Books*

Description: As a Member I want to be able to renew any book I checked-out because My books are about to be due, but I am not finished with them .

**Functional/System Requirements**

1. Member must log into system or by phone
2. Supply ISBN to associate
3. Certain new release books cannot be renewed
4. Allows the book to be checked out for a longer period of time.
5. Keeps count of renewal amount
6. CAN BE DONE OVER THE PHONE OR IN PERSON
7. If done over the phone, isbn needs to be supplied
8. Renewal may be denied if the book has been placed on hold

**Non-Functional Requirements**

Description

**Notes**

Description

US 19 - *Pay a Fine*

Description: As a Member I want to pay a fine so there are no penalties being charged.

**Functional/System Requirements**

1. Must be in person
2. Login
3. Display amount of current fine.
4. Enter credit card #.
5. Process payment.
6. Display successful or unsuccessful.
7. If successful, remove current balance owed.
8. Reactivate account if disabled.

**Non-Functional Requirements**

Description

**Notes**

Description

US 20 - *Report Lost Book*

Description: As a Member I want to report a book as lost because I no longer have/ Know of the books location .

**Functional/System Requirements**

1. Login or reach an associate by phone or in person
2. Display current list of books checked out.
3. Displa\*\*Have field available on each book to select whether book is lost.
4. \*\*Have option to notify manager of lost books with list of books to add to lost book notification.
5. \*\*Add cost of book to Member’s current balance.

**Non-Functional Requirements**

Description

**Notes**

System stops accruing overdue fees for a lost book and member is charged for the replacement cost of the book (added to overdue fees). Entire replacement cost must be received before the suspension can be lifted.

US 21 - *Display Check-out books*

Description: As a Member I want to see what book’s I currently have checked out because I may need to know what books I have. .

**Functional/System Requirements**

1. Member logins in
2. Can check account info to see books that are currently loaned out.

**Non-Functional Requirements**

Description

**Notes**

Description

US 22 - *Display Due Dates*

Description: As a Member I want to see the due dates on my book’s because I do not remember their return time .

**Functional/System Requirements**

1. Login.
2. Display books currently checked out.
3. System needs to have a timestamp of when book was check out.
4. Based on previous timestamp and current day, system needs to display how many days are left before book is due.
5. If book is overdue, list original due date and how many days overdue book is based on previous timestamp.
6. List any fines that have been accrued

**Non-Functional Requirements**

Description

**Notes**

Description

If the book has a hold, renewal option is stated as unavailable

US 23 - *Place book on hold*

Description: As a Member I want to place a book on hold because I am not able to be in person to check-out a book currently, but I know I will need the book soon and I do not want it to be taken .

**Functional/System Requirements**

1. Member logs in
2. Checks if hold is available
3. Can call to place books on hold
4. Member can see if hold is available
5. Associate finds the book and notifies the user through the system
6. Member has 4 days to check out the book, other wise the hold is released.
7. When a book is returned, a person with the book on hold is alerted through the system.

**Non-Functional Requirements**

Description

**Notes**

Many users can place holds on the same book and put into a queue. A book cannot be renewed if it has a hold on it.

US 24 - *Check hold*

Description: As a Member I want to check holds on books to see if there is a queue.

**Functional/System Requirements**

1. Need to be able to search for a particular book.
2. Display book after search.
3. Display whether a queue exists for the book
4. Display how many people are in the queue.

**Non-Functional Requirements**

Description

**Notes**

Description

US 25 - *Display Holds*

Description: As a Member I want to display holds information to see personal holds.

**Functional/System Requirements**

Description

1. Displays the book(s) on hold and the possible time left to retrieve the book before hold removal.

**Non-Functional Requirements**

Description

**Notes**

Description

US 26 - *Display Fines*

Description: As a Member I want to display my current fines because I need to know if fines have been incurred on my account and the balance of the account.

**Functional/System Requirements**

1. Display if a fine has been incurred.
2. Display current balance on account.
3. Display which book is overdue and has incurred a fine.
4. Display amount of days book is overdue.
5. Display when additional fines will be incurred.

**Non-Functional Requirements**

Description

**Notes**

Broken down to show books, due dates, return dates, amounts of fine for each book.

US 27 - *Search for books*

Description: As a Person I want to search for a book because I need to know if the book is available for check-out or unavailable .

**Functional/System Requirements**

1. No login required to simply search for books
2. Checks if the book is available or not.
3. Allow member to place hold on book from search

**Non-Functional Requirements**

Description

**Notes**

Description

US 28 - *Become a Member*

Description: As a Person I want to become a member so that I can check out books.

**Functional/System Requirements**

1. Name, address, phone number, username required
2. Username must be unique
3. Password is provided
4. Four digit code is provided

**Non-Functional Requirements**

Description

**Notes**

Description

**US 7 - *Add Book Listing***

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Time Est** | **Assigned To** | **Description** |
| 1 | 2 | Group | * Create method in Manager class to add new books   + Asks user for ISBN |
| 2 | 1 | Brandon | * Increase inventory if book already exists |
| 3 | 1 | Brandon | * New book added if ISBN doesn’t exist   + Title, Author(s), ISBN, Publication year, Keywords, Availability, Inventory are required |

**US 27 - *Search for books***

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Time Est** | **Assigned To** | **Description** |
| 1 | 2 | Amanda | * Create method for searching outside of any Role   + Person can search for books |
| 2 | 1 | Amanda | * Show if the book is available, how many copies |
| 3 | 2 | Amanda | * Allow Members to place hold on a book from search |

**US 28 - Become a Member**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Time Est** | **Assigned To** | **Description** |
| 1 | 1 | Brandon | * Allow an Associate or Manager to sign People up as Members |
| 2 | 2 | Zack | * Username is generated by first/last name, password generated randomly, pin generated randomly |
| 3 | 1 | Brandon | * Require First Name, Last Name, Street, City, Zip, Phone |

**US 12 - Scan Library Card/Pin**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Time Est** | **Assigned To** | **Description** |
| 1 | .5 | Zack | * Associate or Manager can “scan” a Member’s Pin |
| 2 | 2 | Zack | * Pin pulls in and displays all information on Member |
| 3 | 3 | Zack/Amanda | * Allow a member to check out a book   + This will require another column in the User database called “Books”, to keep track of books the Member has |
| 4 | .5 | Zack | * If a book gets checked out, record the date and time of the loan   + Should either be a new column in Users or Books “Date/Time” |
| 5 | .5 | Amanda | * Shows if book is on hold and by whom   + New column in Books database “On Hold” |
| 6 | .5 | Amanda | * Mark book as available once scanned back in   + New column in Books “Available” |

**US 15 - *Check Out a Book***

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Time Est** | **Assigned To** | **Description** |
| 1 | 1 | Thara | * Get book ISBN and Member’s PIN |
| 2 | 1 | Thara | * Check that member doesn’t have more than 10 books checked out at once |
| 3 | 2 | Group | * Book can be checked out for up to 2 weeks   + Have to do time passing |
| 4 | 1 | Thara | * Can be renewed only 2 times |
| 5 | 1 | Thara | * Record date and time of loan   + Same/similar to US 12, task 4 |

**US 6 - *Delete Accounts***

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Time Est** | **Assigned To** | **Description** |
| 1 | 1 | Thara | * Create a way to find user by Name/Pin |
| 2 | 1 | Zack | * Create method to Delete the user’s account and it must only be accessible for the manager (Manager class) |

**US 17 - *Return Books***

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Time Est** | **Assigned To** | **Description** |
| 1 | 1 | Zack | * Create a method for Associate to type in ISBN of the book being returned |
| 2 | 1 | Zack | * Create method to remove the due book from Member Account once the Associate types in its ISBN |

**US 14 - Clock In/*Clock Out***

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Time Est** | **Assigned To** | **Description** |
| 1 | 3 | Group | * Add a table in the database for “Employee hours” and then make the method to connect to that table based on what their username is and then timestamp the login. |
| 2 | 1 | Group | * Then have an option to timestamp the end of their shift |

**US 22 - *Display Due Dates***

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Time Est** | **Assigned To** | **Description** |
| 1 | 1 | brandon | * Ask member for their Pin(Library Card) |
| 2 | 3 | group | * Take the Pin, the book’s ISBN, and the date of check out.   + Also add on 2 weeks time from check out |
| 3 | 1 | brandon | * Display the information |

**US 23 - *Place Book on Hold***

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Time Est** | **Assigned To** | **Description** |
| 1 | 2 | brandon | * Ask the member for their Pin/Name and what book they want to hold   + Check to see if book is in inventory     - Book is available/not available   + Check to see if User has 10 book limit     - “Sorry you have too many books checked out”/ Proceed |
| 2 | 2 | brandon | * Create method for if book is available, Assign it to “In Holding” for the user |

**US 19 - Pay Fine**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Time Est** | **Assigned To** | **Description** |
| 1 | 1 | brandon | * Ask User for Pin/Name |
| 2 | 1 | brandon | * Create method to Calculate the fine amount   + ($0.10 per day overdue) |
| 3 | 2 | brandon | * Create method to allow User to pay the fine amount   + A portion or full amount (Minimum of $5) |
| 4 | 2 | brandon | * Create method that if $5 is paid and the fine is less than $25 account is in good standing.   + (Fine <25 || paid ==5) * Plus book(s) must be paid off (book replacement cost |

**US 26 -*Display Fines***

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Time Est** | **Assigned To** | **Description** |
| 1 | 1 | brandon | * Ask for Pin/Name |
| 2 | 2 | brandon | * Create Method that calls on Calculate Fine method |
| 3 | 1 | brandon | * Display Fine ( fine types) |

**US 21 - Display Checked-Out Books**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Time Est** | **Assigned To** | **Description** |
| 1 | 2 | Thara | * Ask member for their Pin(Library Card) |
| 2 | 1 | Thara | * Create method that keeps track of the time of book being checked out |
| 3 |  | Thara | * Show books the user has checked out |

**US 24 -Check Hold*s***

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Time Est** | **Assigned To** | **Description** |
| 1 | 1 | Amanda | * Ask user for Pin/Name |
| 2 | 1 | Amanda | * Go to database and grab their Book holds   + Display it for associate, if logged in as associate |

**US 25 - *Display Holds***

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Time Est** | **Assigned To** | **Description** |
| 1 | 1 | Amanda | * Create method to allow the user who logged in to see their holds (call on check holds) |

**US 2 - Suspend Member**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Time Est** | **Assigned To** | **Description** |
| 1 | 1 | Thara | * Create method to suspend user if they have a fine over $25 dollars(fine>25) |
| 2 | 1 | Thara | * Create method for when user logs in, they cannot check out, renew, or place holds (locked) |

**US 18 -Renew Books**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Time Est** | **Assigned To** | **Description** |
| 1 | 1 | Thara | * Ask member for Pin/Name   + or * Allow member to see their books checked out |
| 2 | 2 | Thara | * Create method to grab the books the member checked out |
| 3 | 1 | Thara | * Allow book renewal for 2 weeks if:   + Book has not been renewed twice   + Has less that $25 fine |

**US 20 - *Report Lost Book***

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Time Est** | **Assigned To** | **Description** |
| 1 | 2 | Zack | * When member logs-in allow them to report a book lost/stolen * Also allow Associate to report lost or stolen book for member |
| 2 | 1 | Zack/Thara | * Lock member account -   + may not check out, renew, or hold a book   + Stop adding overdue fees   + Add book cost to book fees   + Book(s) must be paid off before accounts unlocked |

**US 13 - Display Member Fines**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Time Est** | **Assigned To** | **Description** |
| 1 | 1 | brandon | * Add as an option in the console for Member |
| 2 | 1 | brandon | * Create new column in user table for fine amount |
| 3 | 2 | brandon | * Create method displayFines()   + If member, gets current username   + If associate or manager, asks for username input   + Queries database   + Returns fine amount   + Displays fine amount |

**US 8 - *Edit Book Listings***

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Time Est** | **Assigned To** | **Description** |
| 1 | 1 | brandon | * Check if manager, only manager can access |
| 2 | 1 | brandon | * Add option to manager console to edit book listing |
| 3 | 2 | brandon | * Run the search books method.   + Create options to search by (author, title, ISBN, etc.)   + Return results including ISBN number |
| 4 | 3 | Group, brandon | * Create method editBookListings()   + Gets ISBN number from manager input   + Displays field options that are editable   + Asks which field to edit   + Connects to database and edits the field in question |

**US 9 - *Delete Book Listings***

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Time Est** | **Assigned To** | **Description** |
| 1 | 1 | Thara | * Check if Manager, only Manager can access |
| 2 | 1 | Thara | * Manager can Delete Book listings (add to console) |
| 3 | 2 | Thara | * Allow manager to search for book(s)   + Allow removal of these books     - From inventory or entire listing |

**US 10- *Edit employee Clock In/Clock Out***

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Time Est** | **Assigned To** | **Description** |
| 1 | 3 | Group | * Add a table in the database for “Employee hours” and then make the method to connect to that table based on what their username is and then timestamp the login. |
| 2 | 1 | Group | * Then have an option to timestamp the end of their shift |

**US 11 - Edit Person’s Info**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Time Est** | **Assigned To** | **Description** |
| 1 | 1 | Group | * Check if manager, only manager can access |
| 2 | 1 | Group | * Manager may edit the account of a member   + (needs member name/Pin) |

**US 3 - Reactivate Accounts**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Time Est** | **Assigned To** | **Description** |
| 1 | 1 | brandon | * Check if manager, only manager can access |
| 2 | 1 | brandon | * Make an option in the choices to reactivate user. |
| 3 | 2 | Group, brandon | * Write a method called reactivateAccount()   + Checks for manager status   + If manager, then asks for user to check activation status   + If locked, gives option to reactivate.   + Logs into database and changes status from locked true to false for selected user |