**Part A - Capturing requirements with a USE - Case Diagram**

**USE-Case: Checking out a book for loan**

Start of Primary Scenario

1. Include *Identify User*
2. If identification fails then <<Scenario 1>>
3. The user scans the book’s barcode
4. Include *Verify With* *Book Database*
5. If verification fails <<Scenario 2>>
6. The system updates the book’s status to mark it as ‘on-loan’. The user’s information is updated to note that they have taken out the book; it also marks a due date for the book.
7. If user chooses to have a receipt
8. Extends Print Receipt.

End of Primary Scenario

Scenario 1: Notifies user of failure to identify. End of Scenario

Scenario 2: The book is rejected. End of Scenario

**USE-Case: Pay Fines**

Start of Primary Scenario

1. Include *Identify User*
2. If identification fails then <<Scenario 1>>
3. The system prompts the user for their method of payment (cash, credit, or debit).
4. The user deposits the money.
5. The system verifies the amount of money. If the funds are insufficient then <<Scenario 2>>
6. The system updates the user database to note that all funds have been paid off.
7. If user chooses to have a receipt
8. Extends Print Receipt.

End of Primary Scenario

Scenario 1: Notifies user of failure to identify. End of Scenario

Scenario 2: The system notifies the user and requests for further payment

**USE-Case: Check if a book is available**

Start of Primary Scenario

1. The user enters the book’s information into the system.
2. Include *Verify With* *Book Database*
3. If verification fails <<Scenario 1>>
4. Display status of the book for the user

End of Primary Scenario

Scenario 1: The request is rejected. End of Scenario

**USE-Case: Check in a returned book**

Start of Primary Scenario

1. The user presents the book to the system.
2. Include *Verify With* *Book Database*
3. If verification fails <<Scenario 1>>
4. The system checks to see if the book is overdue.
5. If the book is overdue then <<Scenario 2>>
6. The system updates the book’s status to mark it as ‘available’. The user’s information is updated to note that they have returned the book.

End of Primary Scenario

Scenario 1: The book is rejected. End of Scenario

Scenario 2: The system calculates a fine and allocates it to the proper user. End of Scenario

**USE-Case: Reserve a book that is currently on loan**

Start of Primary Scenario

1. Include *Identify User*
2. If identification fails then <<Scenario 1>>
3. The user enters the book’s information into the system.
4. Include *Verify With* *Book Database*
5. If verification fails <<Scenario 2>> (check for book in the system as well as on loan status)
6. The system updates the book’s information to ensure it is on hold after it is returned.

End of Primary Scenario

Scenario 1: Notifies user of failure to identify. End of Scenario

Scenario 2: The request is rejected. End of Scenario

**USE-Case: Request Statement of Fines**

Start of Primary Scenario

1. Include *Identify User*
2. If identification fails then <<Scenario 1>>
3. The user requests statement of fines.
4. The system checks for any fines and displays the results for the user.

End of Primary Scenario

Scenario 1: Notifies user of failure to identify. End of Scenario

**USE-Case: Deal with Changes of Member Details**

Start of Primary Scenario

1. Include *Identify User*
2. If identification fails then <<Scenario 1>>
3. The user request to change their member details.
4. The system prompts the user to verify their identity again as a security measure.
5. Include *Identify User*
6. If identification fails then <<Scenario 2>>
7. The system grants the user permission to change their current information.
8. The system confirms that the user is satisfied with their changes.
9. If the user is not satisfied then <<Scenario 3>>
10. The system updates the changes and saves it to the user database

End of Primary Scenario

Scenario 1: Notifies user of failure to identify. End of Scenario

Scenario 2: Notifies user of failure to identify. End of Scenario

Scenario 3: The system does not save the new changes and prompts user for additional changes.

**USE-Case: Delete Old Members from Library**

Start of Primary Scenario

1. Include *Identify User*
2. If identification fails then <<Scenario 1>>
3. The user request to delete their account.
4. If the user has any fines or books to return <<Scenario 2>>
5. The system prompts the user to verify their identity again as a security measure.
6. Include *Identify User*
7. If identification fails then <<Scenario 3>>
8. The system prompts the user to ensure that they would like to continue to delete their account.
9. The user approves and the system deletes their account.
10. The system updates the user database.

End of Primary Scenario

Scenario 1: Notifies user of failure to identify. End of Scenario

Scenario 2: The system prevents the user from deleting their account if they have any fines or books to return.

Scenario 3: Notifies user of failure to identify. End of Scenario

**USE-Case: Add New Members to the Library**

Start of Primary Scenario

1. The system requests for the user to enter their personal information (name, address, date of birth).
2. The system automatically generates an appropriate user name (that is not already used) based on their name.
3. The system then requests for the user to enter a password for their account.
4. The system accepts the addition and updates the user database.

End of Primary Scenario

**Mini USE-Case: Identify User**

Start of Primary Scenario

1. Request that the user inserts their library card or enter their username password.
2. Verifies the library card information or username and password with the user database.
3. Notifies system whether verification was or was not successful

End of Primary Scenario

**Part B – Capturing Use-Case Behaviour Using a Sequence Diagram**

When the user presents a book, this would be represented by a book class. The book class contains the following:

Book

-title: string

-ISBN: int

-author: string

-Check-in status: bool

User

-name: string

-pin: int

-Card Number: int

-books: map<book, due date>

The book database contains books stored as instances of the class Book. Similarly, the user database contains user information stored as instances of the class User.

For steps 10-11, the user is only prompted for a payment if the due date was found to be earlier than when the book is returned. This is checked in step 8.