Requirements Engineering

Library System

Submitted By: Hinza Noor Amin Mughal (T00203592)

Computing with Software Development

Date Submitted: 12/04/2019

**Table of Contents**

[1. Introduction/overview 4](#_Toc536002599)

[2. Functional Components 5](#_Toc536002600)

[3. User Requirements 6](#_Toc536002601)

[3.1. LibrarySYS will manage Members 6](#_Toc536002602)

[3.2. LibrarySYS will manage Books 6](#_Toc536002603)

[3.3. LibrarySYS will process Transactions 6](#_Toc536002604)

[3.4. LibrarySYS will perform Administration reporting 6](#_Toc536002605)

[4. System Requirements 7](#_Toc536002606)

[4.1. System Level Use Case Diagram 8](#_Toc536002607)

[4.2. Manage Members 9](#_Toc536002608)

[4.2.1. Add Member 9](#_Toc536002609)

[4.2.2. Update Member 12](#_Toc536002610)

[4.2.3. View Member 15](#_Toc536002611)

[4.3. Manage Books 17](#_Toc536002612)

[4.3.1. Add Book 17](#_Toc536002613)

[4.3.2. Update Book 19](#_Toc536002614)

[4.3.3. View Book 22](#_Toc536002615)

[4.3.4. Delete Book 24](#_Toc536002616)

[4.4. Process Loans 26](#_Toc536002617)

[4.4.1. Loan Book 26](#_Toc536002618)

[4.4.2. Return Book 31](#_Toc536002619)

[4.4.3. Pay Fine 33](#_Toc536002620)

[4.5. Perform Admin 35](#_Toc536002621)

[4.5.1. List Overdue Books 35](#_Toc536002622)

[4.5.2. Perform Book Category Analysis 37](#_Toc536002623)

[5. System Model 39](#_Toc536002624)

[5.1. Level-0 DFD 39](#_Toc536002625)

[5.2. Level-1 DFD 40](#_Toc536002626)

[5.3. Level-2 DFD (Process P1: Manage Members) 41](#_Toc536002627)

[5.4. Level-2 DFD (Process P2: Manage Books) 42](#_Toc536002628)

[5.5. Level-2 DFD (Process P3: Process Loan) 43](#_Toc536002629)

[5.6. Level-2 DFD (Process P3: Perform Admin) 45](#_Toc536002630)

[6. Data Model (Class Diagram) 46](#_Toc536002631)

[6.1. Class Diagram 47](#_Toc536002632)

[6.2. Relational Schema 48](#_Toc536002633)

[6.3. Database Schema 49](#_Toc536002634)

[7. Conclusion 51](#_Toc536002635)

# Introduction/overview

Library consists of thousands of books, and with the increase in the number of readers it needs to be well organized. This system provides the facility of organization of books, monitoring them and controlling the transactions. It maintains the details of books and members. It provides the ease of searching the available books in the library. It provides the functionality of adding and updating the books.

The Library Management System will have two end users, Member and Admin. Admin will be able to check member’s details, check-in or check out books from the library, manage the books. Admin will also be able to view member details such as list of borrowed books, view the due date and view the payment history.

The main objective of this system is to simplify day to day process of library. It will be able to provide quick and efficient service in a quick manner. This system will be able to allow to remove the drawbacks of large member information physical files will are tough to manage. Secure transaction, quick retrieval of information and ease of use are some of the benefits of using this system.

The project aims and objectives that will be achieved after completion of this project are discussed below. The aims and objectives are as follows:

* Addition of new book and members
* Update books and members
* Book issue and return
* Fine payment
* A view feature to search books and members
* A report/analysis of category and overdue books

Library Management System is an application which refers to library systems which are generally small or medium in size. It is used by librarian to manage the library using a computerized system where he/she can record various transactions like issue of books, return of books, addition of new books, addition of new members etc.

Books and members maintenance modules are also included in this system which would keep track of the members using the library and also a detailed description about the books a library contains.

In addition, analysis module is also included in Library Management System. The admin is able to generate different kinds of analysis like lists overdue books and category analysis.

All these modules are able to help librarian to manage the library with more convenience and in a more efficient way as compared to library systems which are not computerized.

# Functional Components

This section of the document presents the functional components of the proposed software system.

There are 4 main components in the proposed system.

# User Requirements

This section presents the user requirements as a set of high level abstract statements including the scope of the system, key objectives for the project, and the applicable regulatory concerns.

## LibrarySYS will manage Members

* + 1. LibrarySys will add a member.
    2. LibrarySys will update a member.
    3. LibrarySys will view a member.

## LibrarySYS will manage Books

* + 1. LibrarySys will add a book.
    2. LibrarySys will update a book.
    3. LibrarySys will view a book.
    4. LibrarySys will delete a book.

## LibrarySYS will process Transactions

* + 1. LibrarySys will process loan for book.
    2. LibrarySys will manage the book return.
    3. LibrarySys will record payment of a fine.

## LibrarySYS will perform Administration reporting

* + 1. LibrarySys will list overdue books.
    2. LibrarySys will perform book category analysis.

# System Requirements

The system is divided into 4 main modules.

* The members’ module provide functions to add members, update members and view members.
* The books module provides functions to add books, update books, view books and delete books.
* The transaction module provides functions to loan books, return books and pay fine.
* The admin module provides functions to list overdue books and perform book category analysis.

These requirements, in brief, shows the type of information that a LibrarySys needs to be able to process.

## System Level Use Case Diagram

Admin

Member

## Manage Members

This module provides functions to add a member, view a member, and update a member.

### Add Member

In order to access the library resources a person should be enrolled as the member of the library. This function allows to add new members to the system.

<<includes>>

Admin

<<extends>>

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Add Member** | |
| **Use Case Id** | 1 | |
| **Priority** | High | |
| **Source** | Admin | |
| **Primary Business Actor** | Admin | |
| **Other Participating Actors** |  | |
| **Description** | This function collects the details of a new member, fills up the details in the designated fields and adds the person in the library system. | |
| **Preconditions** | The admin must collect required information for a member to be registered. | |
| **Trigger** | A member requests admin to register for the library. | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The admin invokes the add member function.  **Step 4:** Admin enters member details.   * Surname * Forename * Street * Town * County * DOB * Gender * Phone   **Step 6:** Admin confirms the member is to be added. | **Step 2:** The system assigns the member ID.  **Step 3:** The system displays the UI.  **Step 5:** System validates the member details entered:   * All fields must be entered * Member ID (auto increment) numeric(6) * Surname and forename must be unique and char(15) each * Street must be char(30) * Town must be char(20) * County must be char(15) * DOB must show that the member is at least 18 * Gender must be char(1) * Phone number must be numeric(15) and unique   **Step 7:** The system sets the member status to ‘Active’ and Fine to 0.00 in Members File.    **Step 8:** The system saves the member details in the Member File.   * Member Id * Surname * Forename * Street * Town * County * DOB * Gender * Phone * Fine * Status   **Step 9:** The system displays a confirmation message.    **Step 10:** The system clears the UI. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| Invalid data is entered |  | **Step 4:** Invalid data is entered or a field is empty.    **Step 5:** The system displays an appropriate error message. |
| **Conclusions** | A member has been added to the system. | |
| **Post conditions** | This member can now borrow books. | |
| **Business Rules** | The member must be over 18. | |
| **Implementation Constraints** |  | |

### Update Member

The admin must be able to alter the member data in case of some changes to member details. This function allows to update members.

Admin

<<includes>>

<<extends>>

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Update Member** | |
| **Use Case Id** | 2 | |
| **Priority** | High | |
| **Source** | Admin | |
| **Primary Business Actor** | Admin | |
| **Other Participating Actors** |  | |
| **Description** | When a member details are changed, the member details in the database has to be updated. | |
| **Preconditions** |  | |
| **Trigger** | Member details has changed. | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** Admin enters into update member portal.  **Step 3:** Admin enters a valid member Id.  **Step 6:** Admin updates the member details:   * Surname * Forename * Street * Town * County * DOB * Gender * Phone * Status   **Step 8:** The admin confirms the member is to be updated. | **Step 2:** The system displays the UI.  **Step 4:** System validates Member Id is not blank.  **Step 5:** System retrieves full details of the member from the Members File and displays on UI for updating.  **Step 7:** The system validates the member details entered:   * All fields must be entered * Surname and forename must be unique and char(15) each * Street must be char(30) * Town must be char(20) * County must be char(15) * DOB must check that the member in at least 18 * Phone number must be numeric(15) and unique   **Step 9:** The system updates the member details in the Members File.  **Step 10:** The system displays a confirmation message.    **Step 11:** The system clears the UI. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| No Name Entered. |  | **Step 4:** Member Id is blank    **Step 5:** The system displays an appropriate error message. |
| No Member Found |  | **Step 4:** No members found  **Step 5:** The system displays an appropriate error message. |
| Invalid data is entered. |  | **Step 7:** Invalid data is entered or a field is empty.    **Step 8:** The system displays an appropriate error message. |
| **Conclusions** | The member details have been updated. | |
| **Post conditions** |  | |
| **Business Rules** | The member must be above 18 | |
| **Implementation Constraints** |  | |

### View Member

The admin might need view details of member. This function allows to view members.

Admin

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **View Member** | |
| **Use Case Id** | 3 | |
| **Priority** | High | |
| **Source** | Admin | |
| **Primary Business Actor** | Admin | |
| **Other Participating Actors** |  | |
| **Description** | Admin views the detail of member. | |
| **Preconditions** | Admin gets the member Surname. | |
| **Trigger** | An authorized person (or the member itself) requests the information on member. | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** Admin enters into view member portal.  **Step 3:** Admin enters a valid surname or part of surname. | **Step 2:** The system displays the UI.  **Step 4:** System validates surname is not blank.  **Step 5:** The system retrieves full details of ALL members with matching surname (in order of MemId) from the Members file and displays following information:   * Member Id * Surname * Forename * Address * DOB * Phone * Status * Fine |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| No Name Entered |  | **Step 4:** Name is blank    **Step 5:** The system displays an appropriate error message. |
| No Member Found |  | **Step 4:** No members found  **Step 5:** The system displays an appropriate error message. |
| **Conclusions** | I can conclude that member’s information is viewed. | |
| **Post conditions** | The details for the requested member are displayed by the library system. | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

## Manage Books

### Add Book

This function records the details of a book, its description and its availability on the system. In short this function allows to add new books to the system.

<<includes>>

Admin

<<extends>>

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Add Book** | |
| **Use Case Id** | 4 | |
| **Priority** | High | |
| **Source** | Admin | |
| **Primary Business Actor** | Admin | |
| **Other Participating Actors** |  | |
| **Description** | This function collects available information of a book, fills up the details in the designated fields and adds the book to the library system. | |
| **Preconditions** | The admin must collect required information for a book. | |
| **Trigger** | The book is newly available. | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The admin invokes the add book function.  **Step 5:** Admin enters book details.   * Title * Author * Category * Isbn   **Step 7:** The admin confirms the book is to be added. | **Step 2:** System retrieves category details from category file and load on UI  **Step 3:** The system assigns book a Book Id.  **Step 4:** The system displays the UI.  **Step 6:** The system validates the book details entered:   * All fields must be entered * Title must be unique char(50) * Author must be char(25) * Category must be char(10) * ISBN String(13)   **Step 8:** System sets the book status to ‘Available’ in Books File.  **Step 9:** The system saves the book details in the Book File.   * BookId * Title * Author * Category * ISBN * Status   **Step 10:** The system displays a confirmation message.    **Step 11:** The system clears the UI. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| Invalid data is entered |  | **Step 6:** Invalid data is entered or a field is empty.    **Step 7:** The system displays an appropriate error message. |
| **Conclusions** | A book is added to the system. | |
| **Post conditions** | A new book is now available in the library. | |
| **Business Rules** |  | |

### Update Book

The system must be able to alter the book details. This function allows the admin to update a book.

<<includes>>

Admin

<<extends>>

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Update Book** | |
| **Use Case Id** | 5 | |
| **Priority** | High | |
| **Source** | Admin | |
| **Primary Business Actor** | Admin | |
| **Other Participating Actors** |  | |
| **Description** | When the book details are changed, the details in the book file has to be updated. | |
| **Preconditions** | Admin collects the book information that is to be updated. | |
| **Trigger** | Book details has changed. | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** Admin enters into update book portal.  **Step 3:** Admin enters a valid BookId.  **Step 6:** Admin updates the book details:   * Isbn * Title * Author * Category * Status   **Step 8:** The admin confirms the book is to be updated. | **Step 2:** The system displays the UI.  **Step 4:** System validates BookId is not blank or that the BookId entered exists in the database.  **Step 5:** System retrieves full details of the book from the Books File and Category File and displays a UI for updating.  **Step 7:** The system validates the book details entered:   * All fields must be entered * Title must be unique and char(50) * Author must be char(25) * Category must be char(10) * Status must be char(1)   **Step 9:** The system updates the book details in the Book File.  **Step 10:** The system displays a confirmation message.    **Step 11:** The system clears the UI. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| No Author Entered. |  | **Step 4:** BookId is blank.    **Step 5:** The system displays an appropriate error message. |
| No Book Found. |  | **Step 4:** No books found.  **Step 5:** The system displays an appropriate error message. |
| Invalid data is entered. |  | **Step 9:** Invalid data is entered or a field is empty.    **Step 10:** The system displays an appropriate error message. |
| **Conclusions** | I can conclude that the book details is updated. | |
| **Post conditions** | The book information is changed. | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### View Book

Every book in the library is assigned with an ISBN. With the input of correct ISBN this function allows to view the book.

Admin

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **View Book** | |
| **Use Case Id** | 6 | |
| **Priority** | High | |
| **Source** | Admin | |
| **Primary Business Actor** | Admin | |
| **Other Participating Actors** |  | |
| **Description** | Admin view the details of a book. | |
| **Preconditions** |  | |
| **Trigger** | An authorized person (or the member itself) requests the information on member. | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** Admin enters into view member portal.  **Step 3:** Admin enters a valid author or part of author. | **Step 2:** The system displays the UI.  **Step 4:** System validates author is not blank.  **Step 5:** System retrieves full details of the selected book from the Books File and Category File  And displays data including the following:   * ISBN * Title * Author * Category * No. of Copies * Status |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| No Author Entered. |  | **Step 4:** Author is blank.    **Step 5:** The system displays an appropriate error message. |
| No Book Found. |  | **Step 4:** No books found.  **Step 5:** The system displays an appropriate error message. |
| **Conclusions** | I can conclude that book information is displayed. | |
| **Post conditions** | The details for the requested book are displayed out by the library system. | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Delete Book

When a book in the library is no more available then it needs to be deleted from the system. This function allow to delete books.

Admin

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Delete Book** | |
| **Use Case Id** | 7 | |
| **Priority** | High | |
| **Source** | Admin | |
| **Primary Business Actor** | Admin | |
| **Other Participating Actors** |  | |
| **Description** | Admin deletes book from the system. | |
| **Preconditions** | This function deletes a book by taking in the correct details. | |
| **Trigger** | Book is no more available. | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** Admin enters into delete member portal.  **Step 3:** Admin search for book by author.  **Step 6:** Admin selects the book.  **Step 8:** Admin clicks delete icon.  **Step 10:** Admin confirms the operation. | **Step 2:** The system displays the UI.  **Step 4:** System validates author is not blank or the author exists in Books File.  **Step 5:** The system retrieves a summary of ALL available books with matching author from the Books file and displays on the UI.  **Step 7:** System retrieves full details of the selected book from the Books File and Category File displays a delete-icon for deleting. The details includes:   * ISBN * Title * Author * Category * No. of Copies     **Step 9:** System asks for confirmation.  **Step 11:** System sets the book status to ‘Removed’ in the Book File. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| No Author Entered. |  | **Step 4:** Author is blank.    **Step 5:** The system displays an appropriate error message. |
| No Book Found. |  | **Step 4:** No books found.  **Step 5:** The system displays an appropriate error message. |
| **Conclusions** | I can conclude that the book is deleted. | |
| **Post conditions** | The requested book is deleted from the library system. | |
| **Business Rules** | Only ‘Available’ books may be deleted. | |
| **Implementation Constraints** |  | |

## Process Loans

### Loan Book

When a member picks books from the library to rent, then admin need to process loan for the books. This function allows to rent books. More than one book, but less than 3, may be borrowed at a time.

Admin

Member

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Loan Books** | |
| **Use Case Id** | 8 | |
| **Priority** | High | |
| **Source** | Admin | |
| **Primary Business Actor** | Admin | |
| **Other Participating Actors** | Member | |
| **Description** | Admin process rent for the books the member has selected. | |
| **Preconditions** |  | |
| **Trigger** | Member wants to rent book from the library. | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** Admin enters into rent book portal.  **Step 4:** Admin enters member ID.  **Step 8:** Admin enters the Book Id.  **Step 10:** Admin clicks ‘add book to cart’.  **Step 14:** Admin clicks Finish. | **Step 2:** The system assigns a loan id loan date and due date.  **Step 3:** The system displays the UI.  **Step 5:** System validates member ID.  **Step 6:** System retrieves member details from Members File.  **Step 7:** System displays UI.  **Step 9:** System retrieves book details from Books File.  **Step 11:** System adds book to cart.  **Step 12:** If no more books to be borrowed, go to step 14.  **Step 13:** If more books to be borrowed (no more than 3) books borrowed, go to step 7.  **Step 15:** System saves due date for the respective books in Loans File (Due date = loan Date + 6 days).  **Step 16:** System saves loan in Loan File.   * Loan Id * Member Id * Loan Date * Due Date   **Step 17:** System updates book status to ‘Unavailable’ in Books File.  **Step 18:** System save book details in Loan Items File.   * Loan Id * BookId * Return Date (null)   **Step 19:** System confirms loan.  **Step 20:** System clears UI. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| No Name Entered. |  | **Step 4:** Member ID is blank.    **Step 5:** The system displays an appropriate error message. |
| No Member Found. |  | **Step 4:** No members found.  **Step 5:** The system displays an appropriate error message. |
| Max quota exceeded. |  | **Step 10:** More than 3 books are added.  **Step 11:** The system display message “Book limit exceeded”. |
| **Conclusions** | I can conclude that the rent for the books has been processed. | |
| **Post conditions** | The member has successfully rented books. | |
| **Business Rules** | No more than 3 books to be rented at a time. | |
| **Implementation Constraints** |  | |

Y

Books > 3?

Y

Click ‘Add Book to Cart’

Enter Book ID

Save loan in Loans File

Save due date in Loans File

N

More Books?

N

Adds book to cart

Display Book Details

Display UI

Retrieve Member Details

Error Message

Valid?

Validate Member ID

Enter Member ID

Display UI

Enter into Rent Book portal

Assigns Loan ID

System

Admin

Y

N

.

System

Admin

Clear UI

Confirm Loan

Y

N

More Books?

Save book details in the Loan Items File

Update book status to ‘Unavailable’ in Book File

### Return Book

When a member drops up the books, then admin have to check in the books. This function allows to processes returned books.

Admin

Member

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Return Book** | |
| **Use Case Id** | 9 | |
| **Priority** | High | |
| **Source** | Admin | |
| **Primary Business Actor** | Admin | |
| **Other Participating Actors** | Member | |
| **Description** | Admin collects in the books returned by the member. | |
| **Preconditions** |  | |
| **Trigger** | Member brings the book back to the library. | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** Admin enters into return book portal.  **Step 3:** Admin enters the Member ID.  **Step 5:** Admin confirms return. | **Step 2:** The system displays the UI.  **Step 4:** The system retrieves book details from Books File, Members File, loan file and loan items file.  **Step 6:** System updates the status to ‘Available’ in Books File.  **Step 7:** System updates the return date in Loan Items File.  **Step 8:** System checks the return date (in loan items file) <= due date (in loan file).  **Step 9:** If the return date > due date, then system updates fine in Members file. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| Invalid data is entered. |  | **Step 4:** Invalid ISBN is entered.  **Step 5:** The system displays an appropriate error message. |
| Book is returned late. |  | **Step 8:** Return date is > due date.  **Step 9:** The system displays an appropriate error message and updates fine in Members file. |
| **Conclusions** | I can conclude that the books are returned to the library. | |
| **Post conditions** | The member has returned books. | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Pay Fine

When there is a delay is returning the books or the books have exceeded their due dates, then member needs to pay fine for the books. This function allows to record fine and payment of fine.

Admin

Member

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Pay Fine** | |
| **Use Case Id** | 10 | |
| **Priority** | High | |
| **Source** | Admin | |
| **Primary Business Actor** | Admin | |
| **Other Participating Actors** | Member | |
| **Description** | Fines and description are displayed on the members account and needs to be paid. | |
| **Preconditions** |  | |
| **Trigger** | Member needs to pay fine for the book return delay. | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:**  Admin enters into pay fine portal.  **Step 3:**  Admin enters member ID.  **Step 6:**  Admin selects payment information.   * Full Payment/ Partial   **Step 10:**  Admin clicks on Pay button. | **Step 2:** The system displays the UI.  **Step 4:** System checks the fine is not zero.  **Step 4:** System retrieves details for the member ID entered from the Members file and displays on the UI for viewing only.  **Step 7:** If the payment is full, system will update the fine in Members File to 0.  **Step 8:** If the payment is partial, system will display a UI for amount.  **Step 9:** System will update the fine in Members File as fine = payment Amount - fine.  **Step 11:**  System process the payment.  **Step 12:**  System updates the payment information in Fine Payment File.   * Pay Id * Pay Date * MemId * Amount Paid   **Step 10:** System updates the fine in Members File.  **Step 11:** System provides the payment receipt. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| Invalid data is entered. |  | **Step 4:** Invalid member ID is entered.  **Step 5:** The system displays an appropriate error message. |
|  |  | **Step 7:** Fine amount is ‘0.00’.  **Step 8:** System displays an appropriate alert message. |
| **Conclusions** | I can conclude that the fine has been paid for the delayed books. | |
| **Post conditions** | The member has paid fine. | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

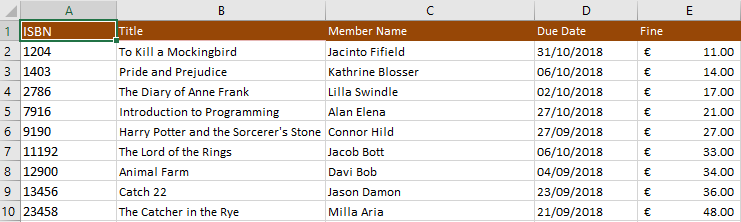
## Perform Admin

### List Overdue Books

At the end of the day admin might need to see the list for the overdue books. This function allows to list overdue books. A book is overdue if it has not been returned by its due date.

Admin

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **List Overdue Books** | |
| **Use Case Id** | 11 | |
| **Priority** | High | |
| **Source** | Admin | |
| **Primary Business Actor** | Admin | |
| **Other Participating Actors** |  | |
| **Description** | A list of overdue books is displayed. | |
| **Preconditions** |  | |
| **Trigger** | Admin carries out analysis to list overdue books. | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** Admin enters into list overdue books portal. | **Step 2:** The system retrieves details of all overdue books from Loan, loan items, books and member files.  **Step 3:** System displays the UI. |
| Print selected. | **Step 4:** Admin selects print. | **Step 5:** The system generates a print file.  **Step 6:** File is send to the printer.  **Step 7:** System displays a confirmation message. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | I can conclude that a list of overdue books is displayed. | |
| **Post conditions** | Admin gets the list from the system. | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |



### Perform Book Category Analysis

When admin needs to examine book categories (e.g. how many books are there for each category) then this function is used.

Admin

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Book Category Analysis** | |
| **Use Case Id** | 12 | |
| **Priority** | High | |
| **Source** | Admin | |
| **Primary Business Actor** | Admin | |
| **Other Participating Actors** |  | |
| **Description** | When admin needs to examine book categories | |
| **Preconditions** |  | |
| **Trigger** | Admin carries out analysis of book category. | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** Admin enters into perform book category analysis portal. | **Step 2:** The system retrieves details from category file and book file.  **Step 3:** System a bar chart of the category with the following details.   * Category * No. Of Books |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | I can conclude that the analysis are displayed. | |
| **Post conditions** | Admin gets to view the book category analysis. | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

# System Model

The following dataflow diagrams have been produced for the system:

## Level-0 DFD

LibrarySys

Member

Loan request

Books

## Level-1 DFD



## Level-2 DFD (Process P1: Manage Members)



## Level-2 DFD (Process P2: Manage Books)



## Level-2 DFD (Process P3: Process Loan)



## Level-2 DFD (Process P3: Perform Admin)



# Data Model (Class Diagram)

The DFD represents how the data will flow in the system when it is build. In this modelling, the major functions in the software are identified. Diagram shows how data will flow through different processes or functions. From the context level diagram we examine high-level functional requirements of the system so main modules can be represented in level one of DFD using DFD notations. This whole unit is represented in the form of different levels (such as level 0, level 1, level 2 DFDs) capturing the detailed structure of the system as perceived by the user.

## Class Diagram

Class diagrams clearly map out the structure of a particular system by modelling its classes, attributes, operations, and relationships between objects.



## Relational Schema

Relational schema for the data requirements - Using ***bracket notation***

**FinePayments (**PayId, MemId, PayDate, Amount)

**Loans (**LoanId, MemId, LoanDate, DueDate)

**Genders (**GenderCode, Description)

**Members (**MemId, Surname, Forename, Address, Dob, GenderCode, Phone, Status, Fine**)**

**Categories (**CatCode, Description)

**LoanItems (**LoanId, BookId, ReturnDate**)**

**Books (**BookId, Isbn, CatCode, Title, Author, Status**)**

## Database Schema

A definition of the database to be implemented.

This includes primary key, foreign key and other constraints to be implemented.

**Schema:** LibrarySYS

1. **Relation Genders**

GenderCode varchar2(1)

Description varchar2(15) NOT NULL UNIQUE

**Primary Key:** GenderCode

1. **Relation Members**

MemId numeric(5)

Surname char(15) NOT NULL

Forename char(15) NOT NULL

Street char(30) NOT NULL

Town char(20) NOT NULL

County char(15) NOT NULL

Dob Date not null CHECK Dob >= 18 Years

GenderCode varchar2(1) NOT NULL

Phone varchar(15) UNIQUE

Status char(20) DEFAULT ‘A’

Fine decimal(5,2) DEFAULT NULL

**Primary Key:** Mem\_Id

**Foreign Key:** GenderCode References Genders

1. **Relation Categories**

Cat\_Code char(3)

Description char(15) NOT NULL UNIQUE

**Primary Key:** Cat\_Code

1. **Relation Books**

BookId numeric(5)

Isbn String(13)

Title char(50) NOT NULL

Author char(25) NOT NULL

Cat\_Code char(3) NOT NULL

Status char(20)

**Primary Key:** BookId

**Foreign Key:** Cat\_Code References Categories

1. **Relation Loans**

LoanId numeric(5) NOT NULL AUTO\_INCREMENT

MemId numeric(5) NOT NULL

LoanDate Date NOT NULL

DueDate Date NOT NULL

**Primary Key:** LoanId

**Foreign Key:** Mem\_Id References Member

1. **Relation LoanItems**

LoanId numeric(5) NOT NULL

BookId numeric(5) NOT NULL

Return\_Date Date DEFAULT NULL

**Primary Key:** LoanId,BookId

**Foreign Key:** LoanId References Loans

**Foreign Key:** Isbn References Book

1. **Relation FinePayments**

PayId numeric(5) NOT NULL AUTO\_INCREMENT

PayDate Date NOT NULL

MemId numeric(5) NOT NULL

Amount decimal(5,2) CHECK > 0

**Primary Key:** PayId

**Foreign Key:** MemId References Members

# Conclusion

This is a ‘Library Management System’. It handles tasks regarding books, members, loans and administrative analysis. The admin will be able to perform the following functions:

* Add, update and view members.
* Add, update, view and delete books.
* Loan book, return book and pay fine for the delayed books.
* Carry out category analysis and list all the due books.

This project is only a humble venture to satisfy the needs in a library. Several user friendly coding have also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the organization.