

**Transaction Processing System**

**Sketching use cases,**

**Data Modelling**

**&**

**interface modelling**

**Nushura Islam**

**Contents**

|  |  |
| --- | --- |
| **Graphical Use Cases**  **Textual Use Cases**  **Data Modelling**  **Interface Modelling**  **References** | **03**  **05**  **10**  **12**  **17** |

**Requirement Analysis**

**Graphical Use Cases**

High-level use case diagram of the whole system:

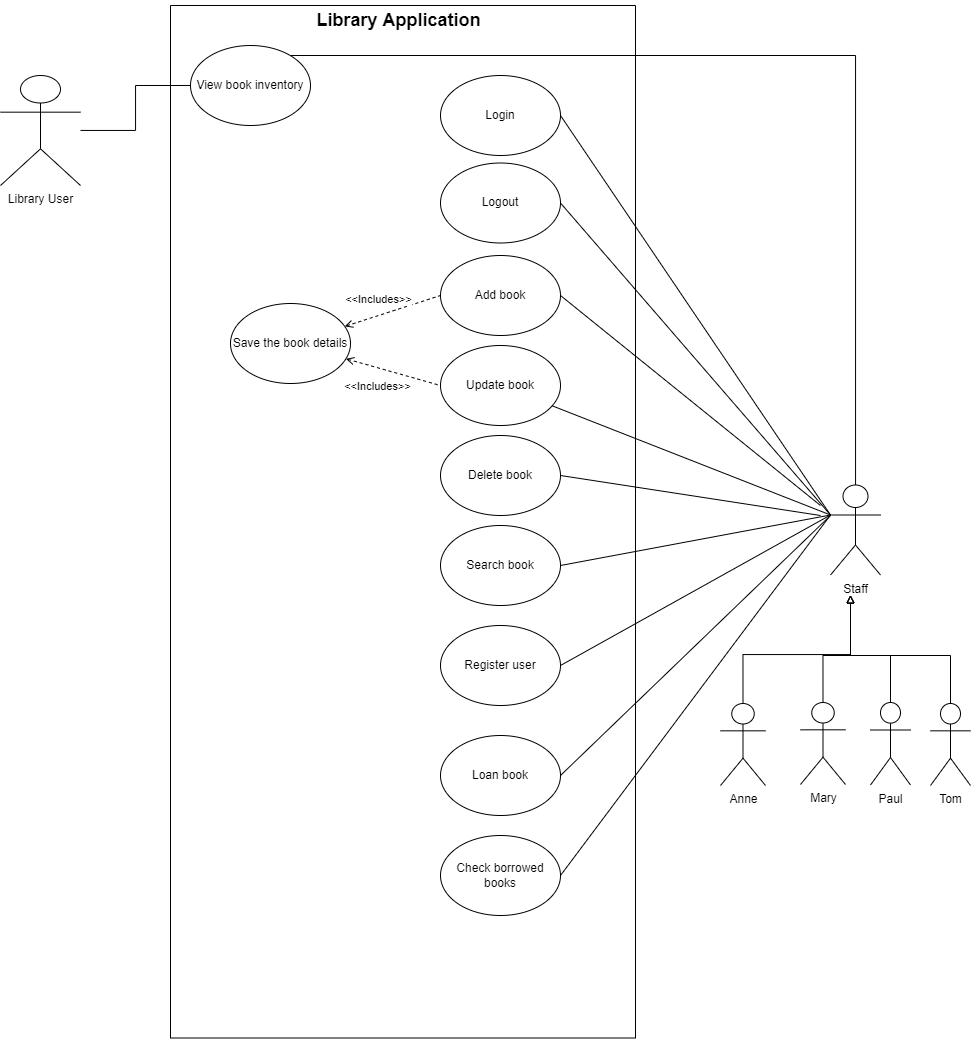
Diagram

Description automatically generated

This use case highlights the main users of the library application which are the volunteer staff members of EFL including Anne. At a high level the staff members need to maintain library users such as register them or update their records. They can also use the application to add more staff members or edit the existing members information. Lastly, they can manage actions regarding books.

High-level use case diagram for maintaining particular entity:

* Representing the system as a rectangle
* As actors are external to the application, they are placed outside the rectangle



**Textual Use Cases**

|  |  |
| --- | --- |
| **Use Case** | View book inventory |
| **User** | Library user and staff |
| **Pre-conditions** | User is inside the library platform |
| **Basic path** | |  |  | | --- | --- | | 1 | User selects “Book” from the menu | | 2 | System displays the list | | 3 | User selects a book by selecting the books ID from the list box. | | 4 | System displays book details in form format., | |

|  |  |
| --- | --- |
| **Use Case** | Login |
| **Inputs** | Staff’s ID and password |
| **User** | Staff |
| **Pre-conditions** | Staff is inside the library application. |
| **Basic path** | |  |  | | --- | --- | | 1 | Staff click “Login” from menu | | 2 | System displays the login form for staff to input data | | 3 | Staff enters data and click on “Login” button | | 4 | System validates data entered by staff and re-direct to homepage while providing admin access. | |
| **Alternative path** | |  |  | | --- | --- | | 4.1 | System displays error message above the “Login” button, stating that either the ID or password is invalid and to try again | |

|  |  |
| --- | --- |
| **Use Case** | Logout |
| **User** | Staff |
| **Pre-conditions** | 1. Staff is logged into the library application. 2. Staff wants to logout |
| **Basic path** | |  |  | | --- | --- | | 1 | Staff selects “Logout” from the menu | | 2 | System logs out the staff member and displays a success message below the logout link. | |

|  |  |
| --- | --- |
| **Use Case** | Add book |
| **Inputs** | A list of books with details such as book title, author name, genre, names of the people who donated the book, donation date and staff ID. |
| **User** | Staff |
| **Pre-conditions** | 1. Staff is logged into the library application. 2. New book has been donated. |
| **Basic path** | |  |  | | --- | --- | | 1 | Staff selects “Book” from the menu | | 2 | System displays a form for staff to enter data | | 3 | Staff click’s on ‘Add’ button and enters data | | 4 | Staff click on “Save” button | | 5 | System validates the inputted data | | 6 | System saves data in the database and displays success message that states that the book has been saved. | |
| **Alternative path** | |  |  | | --- | --- | | 5.1 | System displays error message below the “Save” button, stating that invalid data type entered. | |

|  |  |
| --- | --- |
| **Use Case** | Update book |
| **Inputs** | A list of book details that needs to be updated. |
| **User** | Staff |
| **Pre-conditions** | 1. Staff is logged inside the library application. 2. A book’s author has changed their email. |
| **Basic path** | |  |  | | --- | --- | | 1 | Staff selects “Book” from the menu | | 2 | System displays a form for staff to choose book records from the list-box and to edit data. | | 3 | Staff chooses and enters data | | 4 | Staff click on “Save” button | | 5 | System validates the inputted data | | 6 | System saves data in the database and displays success message that states that the book has been saved. | |
| **Alternative path** | |  |  | | --- | --- | | 5.1 | System displays error message below the “Save” button, stating that invalid data type entered. | |

|  |  |  |
| --- | --- | --- |
| **Use Case** | | Delete book |
| **Inputs** | | Book ID |
| **User** | | Staff |
| **Pre-conditions** | | 1. Staff is logged into the system 2. Staff needs to delete a book from the library inventory |
| **Basic path** | | |  |  | | --- | --- | | 1 | Staff selects “Book” from the menu | | 2 | System displays a form for staff to choose book records and to edit data. | | 3 | Staff chooses the book ID from the list-box | | 4 | System displayed the book details containing that ID | | 5 | Staff click on “Delete” button | | 6 | System validates the action and deletes the respective data from the database. | | 7 | System displays success message that states that the book has been deleted. | |
| **Use Case** | Search book | |
| **Inputs** | Book ID and title | |
| **User** | Staff | |
| **Pre-conditions** | 1. Staff is logged into the system 2. Staff wants to view book | |
| **Basic path** | |  |  | | --- | --- | | 1 | Staff selects “Book” from the menu | | 2 | System displays a form with a list-box containing all the book ids for staff to choose from. | | 3 | Staff chooses the book ID from the list-box | | 4 | System displays the book details containing that ID | | 5 | Staff validates the book by checking the book title from the form. | | |
| **Alternative path** | |  |  | | --- | --- | | 5.1 | Staff did not get the correct book that was being searched | | |

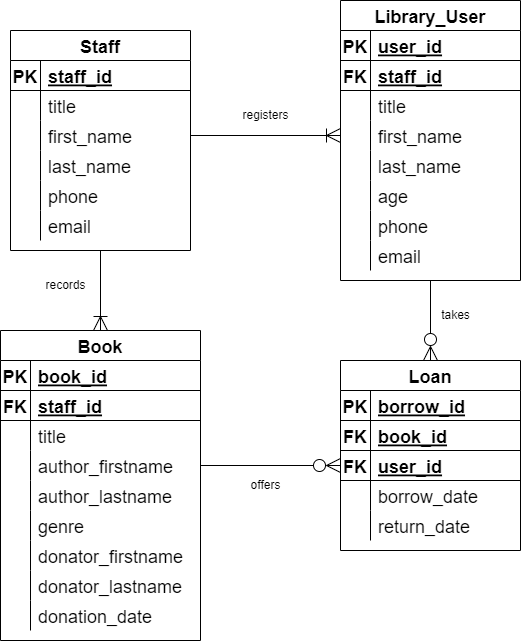
|  |  |
| --- | --- |
| **Use Case** | Register user |
| **Inputs** | User title, first name, last name, age, phone number, email and staff ID. |
| **User** | Staff |
| **Pre-conditions** | 1. Staff is logged into the system 2. A user is new to the EFL and wants to borrow a book or few |
| **Basic path** | |  |  | | --- | --- | | 1 | Staff selects “Member” from the menu | | 2 | System displays a form for staff to register user | | 3 | Staff clicks on ‘Add’ button and enters user details | | 4 | Staff clicks on “Save” button. | | 5 | System validates the data types of user details | | 6 | System saves data in the database | | 7 | System displays success message that states that the user has been registered. | |
| **Alternative path** | |  |  | | --- | --- | | 5.1 | System displays error message below the “Save” button, stating that invalid data type entered. | |

|  |  |
| --- | --- |
| **Use Case** | Loan book |
| **Inputs** | Book Id, Library user ID, borrow date, return date |
| **User** | Staff |
| **Pre-conditions** | 1. Staff is logged into the system 2. A library user wants to borrow a book |
| **Basic path** | |  |  | | --- | --- | | 1 | Staff selects “Loan” from the menu | | 2 | System displays loan form | | 3 | Staff enters inputs | | 4 | Staff clicks on “Save” button | | 5 | System validates the data types of user details | | 6 | System saves data in the database | | 7 | System displays success message that states that the book status has been changed to borrowed. | |
| **Alternative path** | |  |  | | --- | --- | | 5.1 | System displays error message below the “Save” button, stating that invalid data type entered. | |

**System Design**

**Data Modelling**

The following entity-relationship diagram represents how the entities within the library application are related to each other.



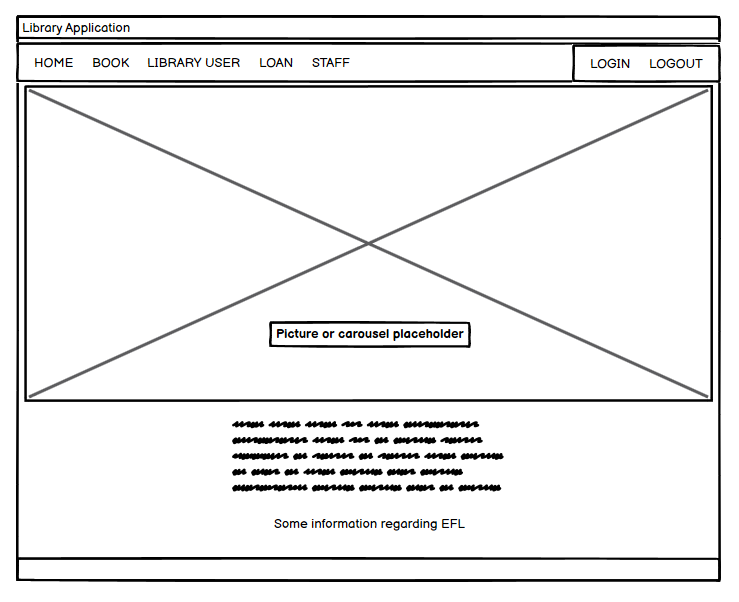
There are four main entities identified, which are staff, library user, book and loan. The staff entity comprises of title, first and last name, phone number and email attribute so that EFL can keep track of their volunteers. The library user entity contains the title, first and last name, age, phone and email attributes required when a staff registers a new user. Moreover, it contains staff id as a foreign key which can assist to determine which staff registered the user at a later stage.

The book entity has the title, author, genre, donator name and date attributes. It also contains the staff id as a foreign key which can assist to determine which staff recorded the book into the system. Lastly, the loan entity has borrowed and return date attributes with two foreign keys coming from the book and library user table. Those foreign keys are the primary keys that can uniquely identify a particular book and user. Hence, the foreign keys in the loan entity will help to identify the books which are borrowed along with the user who borrowed them.

In terms of the relations between attributes, a particular staff can register one to many library users but a user can only be registered by individual staff. Staff can also record one to many donated books, but a particular book can only be recorded by a distinct staff. Then both book and library users have zero to many relationships with the loan entity because a particular book can either be in a state of borrow or it can be available in the EFL. A distinct library user can be recorded as either they borrowed books or not.

**Interface Modelling**

**Main Menu Frame:**

****

This is the mockup for the home page of the application. The page consists of the main menu located at the top of the window. The main menu contains links to the library application's other pages, which are ‘Book’, ‘Library User’, ‘Loan’, ‘Staff’, ‘Login’ and ‘Logout’. In addition, the main menu is located at the top of the window so that users of the application can easily access and navigate to the pages they want.

**Form designs for entities present in the ERD and use cases:**

Book entity form (With primary key and foreign key):

**Graphical user interface, diagram

Description automatically generated**

The form above contains a list box on the application screen's left-hand side. The list box contains the directory of all the book IDs in the EFL book database. It will be the most frequently used attribute; hence, it is placed on the left side of the screen for easy access.

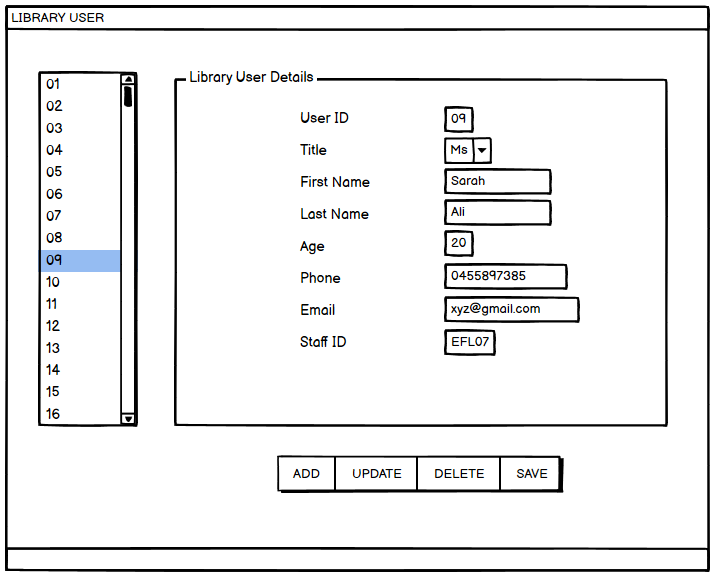
The form also contains labels indicating the attributes of the book entity. Most of the labels have an entry field beside them. The entry fields are selected as input boxes because those attributes will take inputs of the VARCHAR data type. However, there is a date field associated with a DATE data type.

Furthermore, the form contains the staff ID, which is a foreign key of the book entity. This links the staff table with the book table and will help determine which staff inserted particular book records within the database.

There are four kinds of buttons: ' Add’, ‘Update’, ‘Delete’ and ‘Save’. The ‘Add’ button will create a new form with empty fields to add new staff. The ‘Update’ button will assist in updating fields that need editing. The ‘Delete’ button will remove the selected record. Lastly, the ‘Save’ button saves any addition or update of a record.

Moreover, the placement of the buttons is done below the form to follow a universal standard because, typically, any button linked with a form is placed below the form. Therefore, as there is a presence of input fields along with buttons, it will allow users to carry out operations such as create, read, update and delete.

Library User entity form (With primary key and foreign key):



This form is structured similarly to the book form. The only widget used which was not previously used is the combo box. Users can choose from a few given options, which in the case of title would be Mr Ms. and Mrs.

Loan form (With primary key and two foreign keys):

**Graphical user interface, text

Description automatically generated**

The loan form is structured in a similar pattern to maintain consistency with the other pages of the application. However, it contains two foreign keys, bringing information and connecting two tables. In this case, it joins the book id and user id to determine in a single table which book has been borrowed by which user. Moreover, as it contains the borrow and returns dates, it can also be determined when the library user has borrowed the book and when they should return it. Thus, helping the library to keep track of their books.

Staff form (with only primary key):

**Diagram

Description automatically generated with medium confidence**

The staff form follows the pattern shown earlier but does not have a foreign key. Thus, it is not using any other table attributes. The ‘Add’ button will create a new form with empty fields to add new staff. The ‘Update’ button will assist in updating fields that need editing. The ‘Delete’ button will remove the selected record. Lastly, the ‘Save’ button saves any addition or update of a record.

Login form:

This is the form which the staff members can use to log into the library system.

**A picture containing diagram

Description automatically generated**

**References**

Cover picture: Harryarts (n.d) *Abstract blue geometric shapes background* [digital geometric art], Freepik, <https://www.freepik.com/free-vector/abstract-blue-geometric-shapes-background_6166980.htm#query=blue%20background&position=1&from_view=keyword>