SOFTWARE ENGINEERING

YOUR LIBRARY

TEAM MEMBERS:
CEREN AKTAŞ 181805027
DAMLA GÖNÜLSÜZ 181805037
ERSU KARPUZ 171805043
REFİYE ŞEHNAZ YILDIRIM 171805012

1. **Begin with a narrative to give a general overview of the system's or functional requirements. Provide one big use case diagram illustrating the overall functionality of the system. Describe each use case in an easy to understand natural language.**
   - **General overview of the system's functional requirements**
     - The Your Library should allow Admin to delete Users.
     - The Your Library should allow user to store books on their private library and delete them.
     - The Your Library should allow user to search books by their titles.
     - The Your Library should store books and categories in same database with separate tables.
     - The Your Library should store books and categories in same database as default with separate tables.
     - The Your Library is going to take payments by PayPal.
     - The Your Library should enable to search, request, and renew platform from the any computer around the world.
     - The Your Library allow to everyone the registration and during that it checks emails if its already exist.
   - **Use-Case diagram**
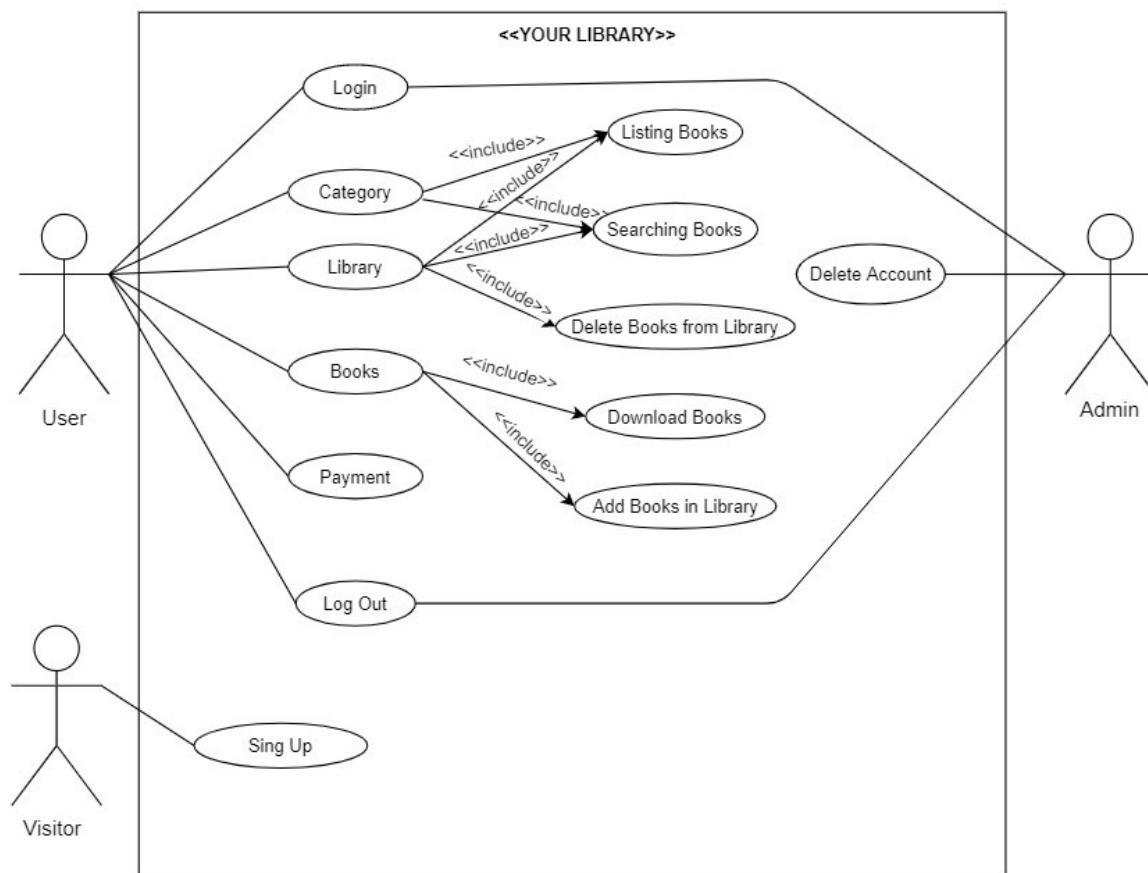
User:

- Login: Allows to user to enter their accounts.
- Category: Listing books and searching books.
- Library: It is also search books and delete books from library.
- Books: Allow to download books and add books to library.
- Payment: It takes payment from the users by PayPal.
- Logout: Allows to leave from the system.

Admin:

- Login: Allows to user to enter their accounts.
- Delete Account: Allows to delete accounts which are no longer pay or requested to be deleted.
- Logout: Allows to leave from the system.

Visitor:

- Sign Up: Allows to visitor to register.

2. & 3. Using the use case diagram as a starting point, convert each use case into a user story. Organize your user stories as a numbered list. For each user story, provide a set of pre- and post-conditions (refer to each corresponding user story by their numbers).

Actors: User, Admin, Visitor

- **User:**
  1. As a user role, I want to login the Your Library so that I can access books and my personal library in Your Library.
     - ➢ **Pre-condition**
       User must have an account.
     - ➢ **Post-condition**
       User successfully login to the system.
  2. As a user role, I want to search the books by their titles in a category or my own library so that I can reach a specific book.
     - ➢ **Pre-condition**
       The user should search for the book they want to read in the category they are interested in.
     - ➢ **Post-condition**
       The user successfully found the book she/he was looking for.
  3. As a user role, I want to delete books from my own library so that I can manage the books in my own library the way I want.
     - ➢ **Pre-condition**
       The user should click the delete book button in her/his library.
     - ➢ **Post-condition**
       User successfully deleted the book in her/his library.
  4. As a user role, I want to download books from the Your Library so that I can read book offline.
     - ➢ **Pre-condition**
       The user should press the download button next to the book they want to download.
     - ➢ **Post-condition**
       The user successfully downloaded the book they wanted to download.
  5. As a user role, I want to add books in my own library so that I can easily access the book that I am reading.
     - ➢ **Pre-condition**
       The user should press the start reading button next to the book they want to read.
     - ➢ **Post-condition**
       The library was created with the books the user reads.
  6. As a user role, I want to pay so that I can access the Your Library's book archive.
     - ➢ **Pre-condition**
       The user must be able to enter card information into the system.

> ➤ **Post-condition**
> User successfully paid the money.

7. As a user role, I want to logout the Your Library so that I can end my session.
   > ➤ **Pre-condition**
   > If the user wants to log out, she/he should be able to press the exit button.
   > ➤ **Post-condition**
   > User successfully logout to the system.

- **Admin:**
  8. As an admin role, I want to login The Your Library so that I can access books and my personal library in Your Library.
     > ➤ **Pre-condition**
     > Admin should be able to login to the site with the admin password.
     > ➤ **Post-condition**
     > Admin successfully login to the system.
  9. As an admin role, I want to delete non-paying users and users who request deletion of their accounts in order so that the platform to be reliable.
     > ➤ **Pre-condition**
     > Admin should be able to detect users who are not reliable and do not pay money.
     > ➤ **Post-condition**
     > Admin deleted this user from the system.
  10. As an admin role, I want to logout the Your Library so that I can end my session.
      > ➤ **Pre-condition**
      > If the admin wants to log out, she/he should be able to press the exit button.
      > ➤ **Post-condition**
      > Admin successfully logout to the system.

- **Visitor:**
  11. As a visitor, I want to register and join the platform. Also, I want to add books on my own library and downloads the book for offline reading. And if I do not like the book which I was reading then I want to remove from my library so that I can read books which I interested and see useful for myself online and offline.
      > ➤ **Pre-condition**
      > If the visitor likes the site when he / she reviews it, he / she must register by entering his / her information on the site.
      > ➤ **Post-condition**
      > The visitor has accessed all the books on the site.

4. **Are there any user stories that are too big or complex (these user stories are usually referred to as 'epics')? Can some of them be decomposed into smaller user stories? For each user story, mention whether or not it may need to be broken down.**
   - Yes, there is. It's Visitor User Story. It may decomposed into smaller user stories.

5. **Provide a separate list of any relevant nonfunctional requirements.**
   - **Product Requirements:**
     > ➤ **Performance Requirements:**
     >   1. Available 7/24.
     >   2. Run fast enough on any operating system for personal computer.
     > ➤ **Security Requirements:**
     >   1. The Your Library provides access to books' data only for users and access to users' data for admin.

2. The Your Library provides equal rights for all pay. If someone does not pay, that user is going be removed from the system.

➢ **Usability requirements:**
1- User-friendly interface.
2- The interface controls the system to not to do mistake.
3- The platform can be used by any people.

- **Organizational Requirements:**
  ➢ **Operational Requirements**:
  1. The Your Library can work smoothly in any Internet browser.
  2. Books download in pdf format.

- **External Requirements:**
  ➢ **Legislative Requirements:**
  1. Personal data of users will be protected.
  2. The Your Library allows users to pay securely.
  ➢ **Safety Requirements:**
  1. Data will be backed up to another environment against any danger.
  ➢ **Ethical Requirements:**
  1. The Your Library respects user's preferences. User can request deletion her/his account and all personal data at any time.

6. **Include a glossary that defines all relevant terms that may have a special meaning in the context of the system.**
- **User**: anyone who register and pays.
- **Library**: private library of books which are users read.
- **Admin:** the person who buys and operate the platform.
- **PayPal library from the nugget package manager:** The PayPal REST API uses HTTP verbs and a RESTful endpoint structure. OAuth2.0 is used as the API Authorization framework. Request and response payloads are formatted as JSON.
- **Migrations:** Entity Framework introduced a migration tool that automatically updates the database schema when your model changes without losing any existing data or other database objects.
- **Entity Framework:** Entity Framework is an open-source ORM framework for .NET applications supported by Microsoft. It enables developers to work with data using objects of domain specific classes without focusing on the underlying database tables and columns where this data is stored.