Project Description: Home Library Management App

Objective:

The goal of this project is to design and develop a Home Library Management App, allowing users to effectively manage their personal book collections. This app will help users keep track of the books they own, their reading progress, and organize them in categories such as genre, author, or series. The application should be user-friendly, intuitive, and provide easy ways for users to add, update, and view their books.

This project provides an opportunity for you to showcase your skills in software development, user interface design, and documentation. Take your time to plan your app, break it down into manageable steps, and don't forget to document your progress along the way!

Key Features:

1. **Book Management:**
    - Add new books with details such as title, author, genre, publication year, ISBN, and number of pages.
    - Edit or remove book entries.
    - Search and filter books by title, author, genre, or year.
2. **Reading Progress:**
    - Track reading progress for each book (e.g., unread, reading, completed).
    - Option to add and update notes for each book.
3. **Categorization:**
    - Organize books by genre, author, or series.
    - Option to create custom categories or shelves.
4. **Library Overview:**
    - Display total number of books, categories, and reading status breakdown (e.g., number of unread, reading, or completed books).
    - Display a summary page with recommended books based on user’s library.
5. **User Interaction:**
    - Simple and intuitive user interface.
    - Allow for user authentication (optional), so each user has a personal library.
6. **Export & Import Data:**
    - Ability to export the library data in formats like CSV or JSON.
    - Option to import books from external sources (e.g., CSV file, online databases).

7. **Business rules:**
    - Each book can have many author but it doesn't have to have one
    - Each author can write many books (at least one)
    - Each publisher can publish many books (at least one)
    - Each book is published by exactly one publisher
    - Each series can have many books (at least one)
    - Each book belongs to at most one series
    - Each series can be written by many authors (at least one)
    - Each author can have many books' series
    - Each book can represent many genres but it doesn't have to represent any
    - Each genre can have many books but it doesn't have to have one
    - Each book can have many topics but it doesn't have to have one
    - Each topic can have many books but it doesn't have to have one
    - Each book can belong to one category
    - Each category can have many books but it doesn't have to have one

Project Requirements:

1. **Programming Language:**
    - The app can be written in any programming language you are comfortable with. Suggested languages include Python, JavaScript (React/Node.js), Java (Android), Swift (iOS), or C# (.NET).
2. **Database (Optional):**
    - If you choose to implement a database, you can use SQL (e.g., MySQL, SQLite) or NoSQL (e.g., MongoDB) to store the books and user data.
3. **Documentation:**
    - Comprehensive **user documentation** explaining how to install, use, and navigate the app.
    - **Technical documentation** describing the system architecture, logic, and code structure.

- **UML Diagrams** for the following:
    - **Use Case Diagram**: Showing the interactions between users and the system.
    - **Class Diagram**: Representing the structure of the app's code, including classes and relationships.
    - **Activity Diagram**: Illustrating the workflow of main actions, such as adding a new book or updating reading progress.
    - **Sequence Diagram**: Showing the interaction between different objects in the system during key operations (e.g., user adding a book or updating progress).

Deliverables:

1. **Source Code:**
    - Fully functional app, including all relevant features and functionality.
2. **Documentation:**
    - A user manual that guides users on how to set up and use the app.
    - A technical document with explanations of the code structure, UML diagrams, and database schema (if applicable).
3. **UML Diagrams:**
    - Create UML diagrams to visually represent the system design and user interactions.

Evaluation Criteria:

- **Functionality:** Does the app perform the required tasks (book management, reading progress tracking, etc.) smoothly and efficiently?
- **Code Quality:** Is the code clean, organized, and well-commented? Does it follow good coding practices?
- **Documentation:** Are the user and technical documentation clear, complete, and well-structured? Are the UML diagrams accurate and useful?
- **User Interface:** Is the app intuitive and user-friendly? Does it have an aesthetically pleasing design?
- **Creativity:** Have you added any additional features beyond the core requirements?

Technical details:

1. Project can be prepared in 1-5 -students groups
2. Via Moodle sot upload:
    - Link to public git repository with project
    - ZIP archive with main branch of the project
    - PDF file with technical documentation and diagrams
    - PDF file with user manual