Ankara Yıldırım Beyazıt University
Department of Computer Engineering

CENG 201 – Object Oriented Programming
Course Project

G11: Library Management System

Analysis Report

Mustafa Sefa Soysal,
Cüneyt Şahin,
Öykü Karaduman,
Sina Erdem Özdemir,
Enes Geldi

Instructor: Muhammed Abdullah Bülbül
Teaching Assistant: Elif Şanlıalp, Yusuf Şevki Günaydın
Date: 31/10/2024

Table of Contents

1. Introduction

In the mid-20th century, libraries really began to become hard to manage. Especially after World War II, there was a surge in demand for education and research materials, particularly in universities and public libraries. Those demands caused growth in book collections and book industry expanded significantly. The increase in the number of books made the operation of libraries difficult, and new solutions were needed. So in the 60s, the first library management systems occured such as MARC (Machine-Readable Cataloging). MARC is a standardised data format that had been used in computers.

In today's world, nearly all the libraries need a system because of runaway increase in the number of books and all these complexity. As we Group-11, we intended to program a library management system that can help both librarians and users. This system will facilitate efficient management of books for librarians, while also enabling users to easily browse and borrow books. Additionally, it will include features to track overdue items, ensuring timely returns and promoting accountability among users.

2. Requirements

2.1. Functional Requirements

1. Library:
   - All book data will be found here.
2. User panel:
   - Users can become members or log in with their account if they have an account.

3. Book Operations:
    - You can search for the book you want, borrow it and return it.
4. Notification Panel:
    - It displays the exact time left for the book and alerts you when the due date arrives.
5. Data of books:
    - It allows us to see the number of books in the library according to their types and the total number.

2.2. Non-Functional Requirements

1. Performance:
    - How fast is the system? The book you are looking for or the operations you have made will appear to you in seconds.
2. Usability:
    - How useful is the system? There are all the features that should be in a digital library.
3. Security:
    - How secure is the system? Even if the best hackers in the world come, they can't access the data you entered.
4. Portability:
    - How compatible is it with the other platforms? It has been tested on Windows systems.
5. Capacity:
    - How much is its capacity? The book which you are searching for but have not found has not yet been published.

3. System Models

3.1. Scenarios

Scenario-1:

A user had borrowed a book, but they haven't returned it in the time limit they had to brought back. Due to that, they received a notification which told them that their access to borrow a book is banned until the time they give the book back to library. So, when they gave the book to library, their access to borrow a book returned its default condition automatically (If it was late third time, only admin is able to remove the ban). The admin, also known as librarian may punish the user with specific penalties which include forcing the user buying a book for the library they were asked for by the other users.

Scenario-2:

An admin has received a package, which contains the books that were ordered by library boss themself, for the library's needs. Now, they have to add those books to the database on their own by hand. First, they must enter the information of the book, such as its name, author, ISBN number, genres etc. After adding all the books, they will confirm the books and will place them in the correct order on the shelves. From now on, these books will be shown to the users when they search for them in the application that is specified for the library itself.

Scenario-3:

A reader wants to borrow a book from the library system. So, reader searches the book and sees the book is available. Then reader clicks the button for borrowing. A random password created to show librarian who is admin. Admin matches the correct password of the book and the user's password. Admin clicks the accept button. The time becomes set and the time starts to count for deadline date. Reader reads the book before deadline and returns it to library. Librarian takes the book back and updates the system.

Scenario-4:

A person wants to borrow a book from a library. So person decides to use LibManager. He realizes he needs an account so he signs up to the system by entering his email, username, password, etc. He borrows a book he is alright with.

Scenario-5:

Librarian checks the library for all the books to update the system. She finds that some books are missing and when she checks the system, she also admits that the book is available in the system. She understands that the book is stolen by someone. She needs to delete the book from the database. By her admin privileges, she deletes the book successfully.

3.2. Use Cases

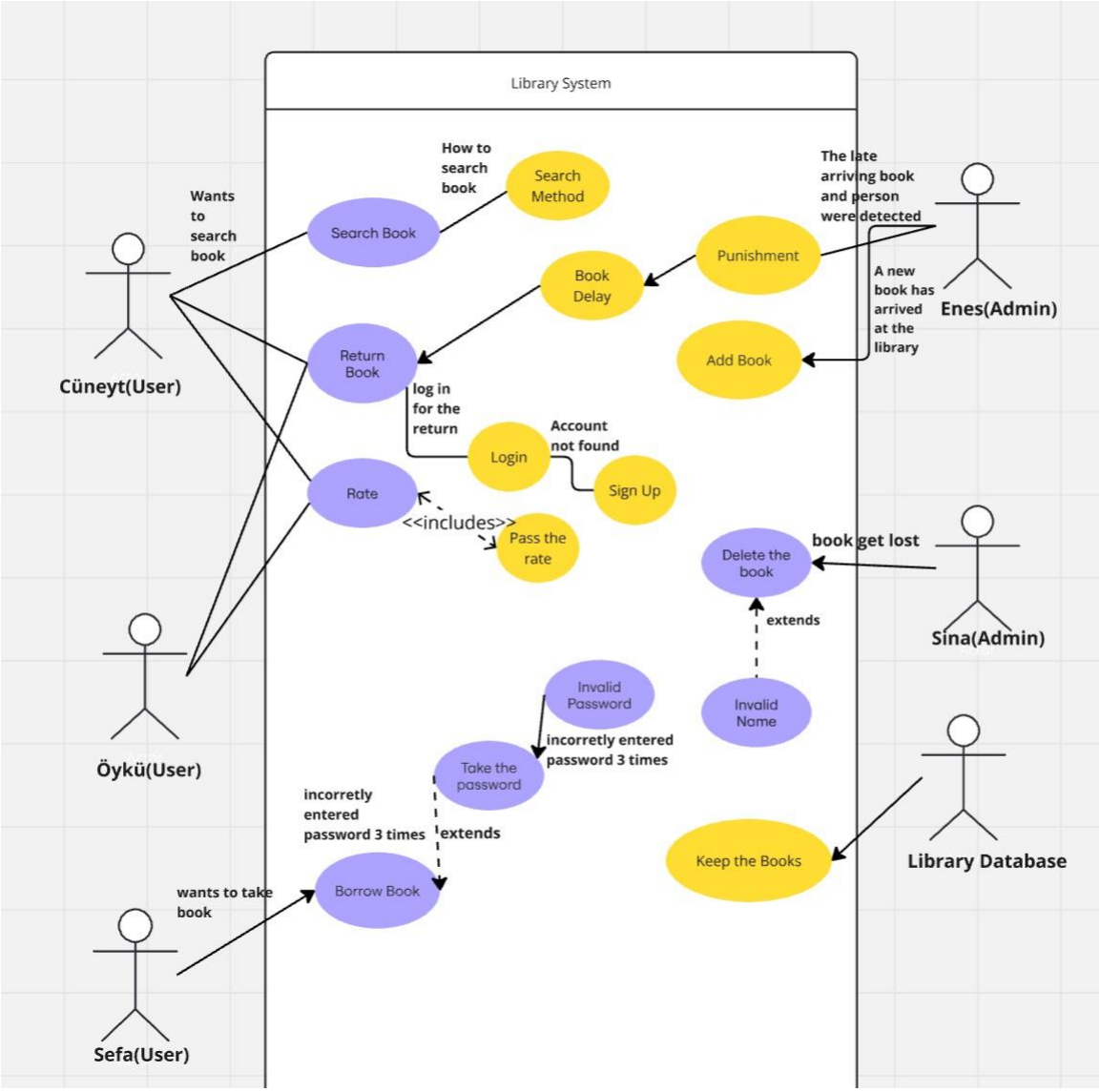

Figure 1: Use Cases Diagram

3.3. Object and Class Model

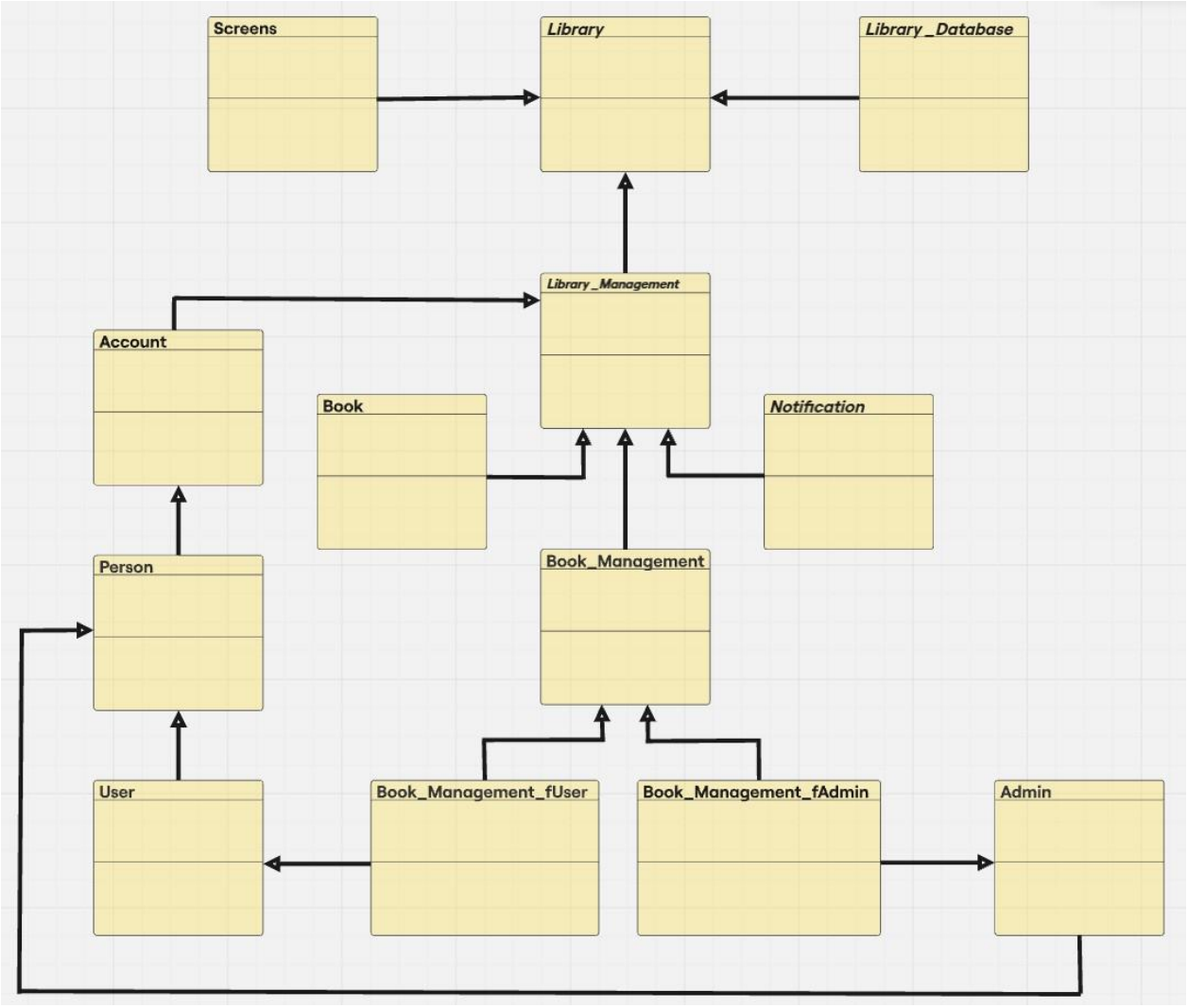

Figure 2: UML Class Diagrams of the System

3.4. User Interfaces

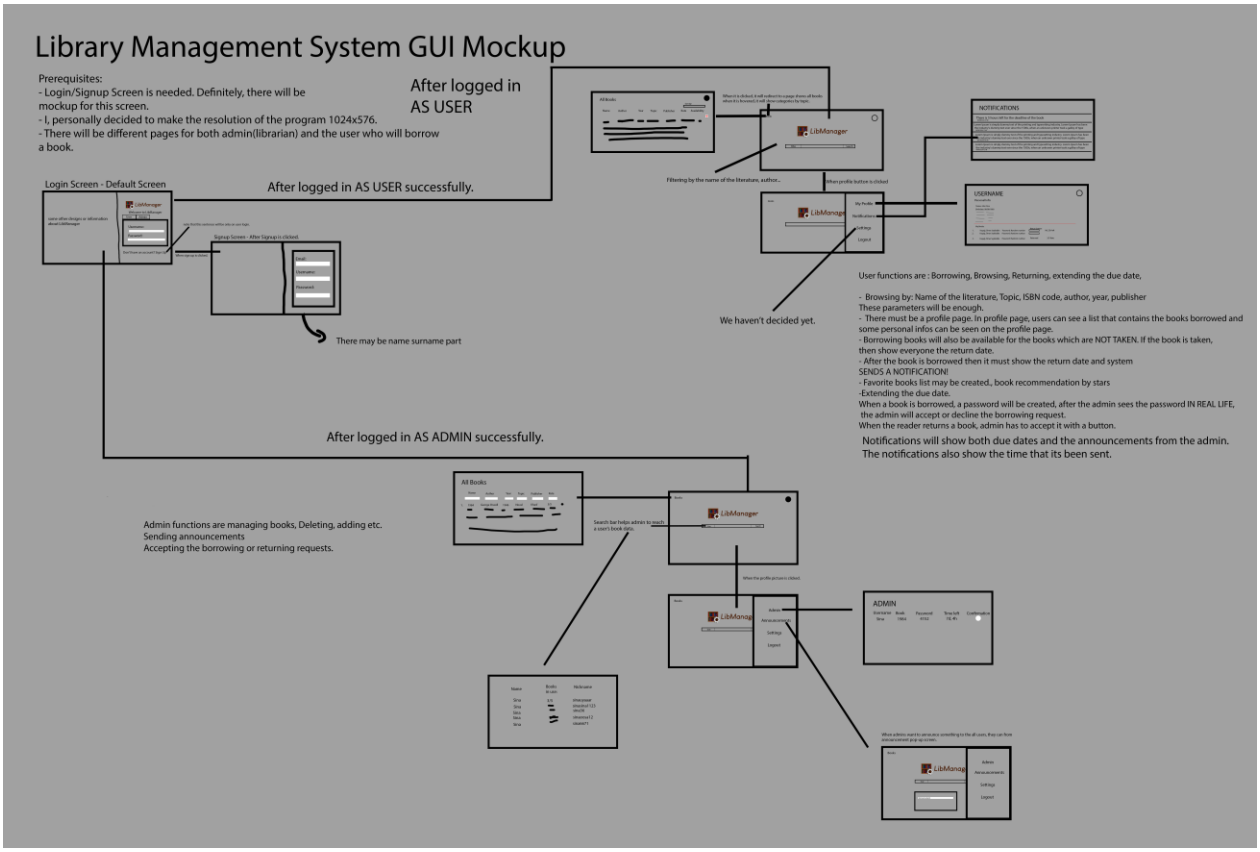

Figure 3: Mockup Designs of the Project

4. Conclusion

In our report, we created the first formation parts and draft of our project. We wrote the requirements and the introduction part. We also showed the behavior of the system against many scenarios.

First of all, we discussed the formation of the project with all group members. Everyone expressed their opinions. After discussing the draft of the project, Sina drew it in detail on paper. Sefa digitalized the draft drawn on paper and added extra details. Öykü, Enes and Cüneyt drew the uml diagram and uml case diagram on paper. Then Enes digitized the uml case diagram. Cüneyt digitized the uml diagram. Sefa and Öykü wrote scripts. And we combined all of them on Sefa's computer and uploaded them to Aybuzem. We can confirm that everybody gave their best on the Project stage one.