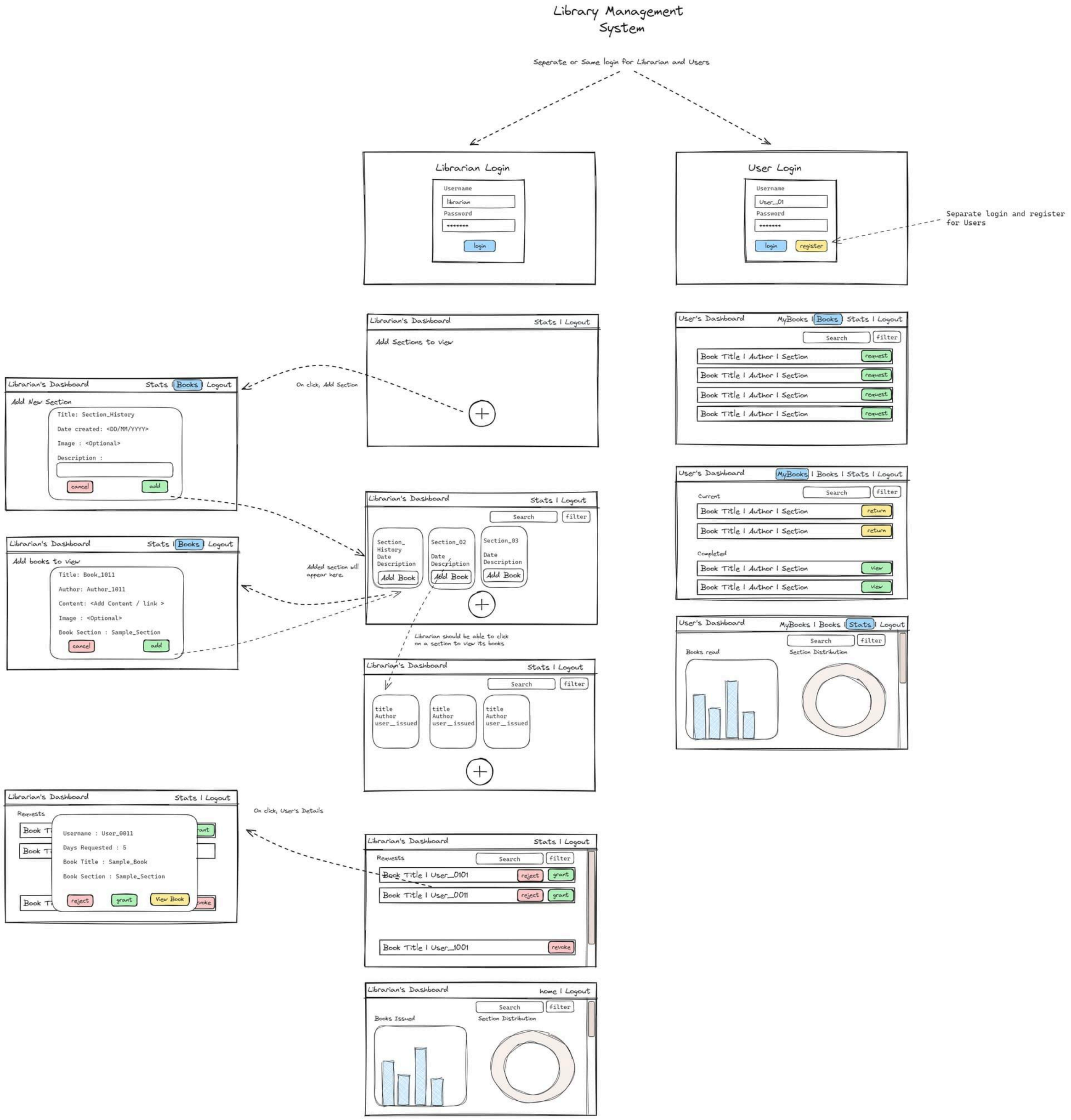
Library Management System V2 Report

Problem Statement:

The Library Management System is an essential tool for properly managing the process of distributing ebooks to users. This multi-user application caters to both regular users/students and librarians. While e-books can be requested, read, and returned by users, librarians also have the ability to add new sections and e-books, as well as grant or revoke access to books. Every area and book in the system has unique characteristics, and the system ought to have features like sorting by predetermined criteria and showcasing recently added sections and books. It's a multi-user app that is used to issue, grant/revoke and maintain e-books across various sections like an online library.

Wireframe provided:



Technologies Used:

Frontend:

- HTML
- CSS
- Bootstrap
- VueJs

Backend:

- Flask

Data Storage:

- SQLite

Data Storage:

- Redis
- Celery
- Pdfkit
- SMTPlib

Libraries Used:

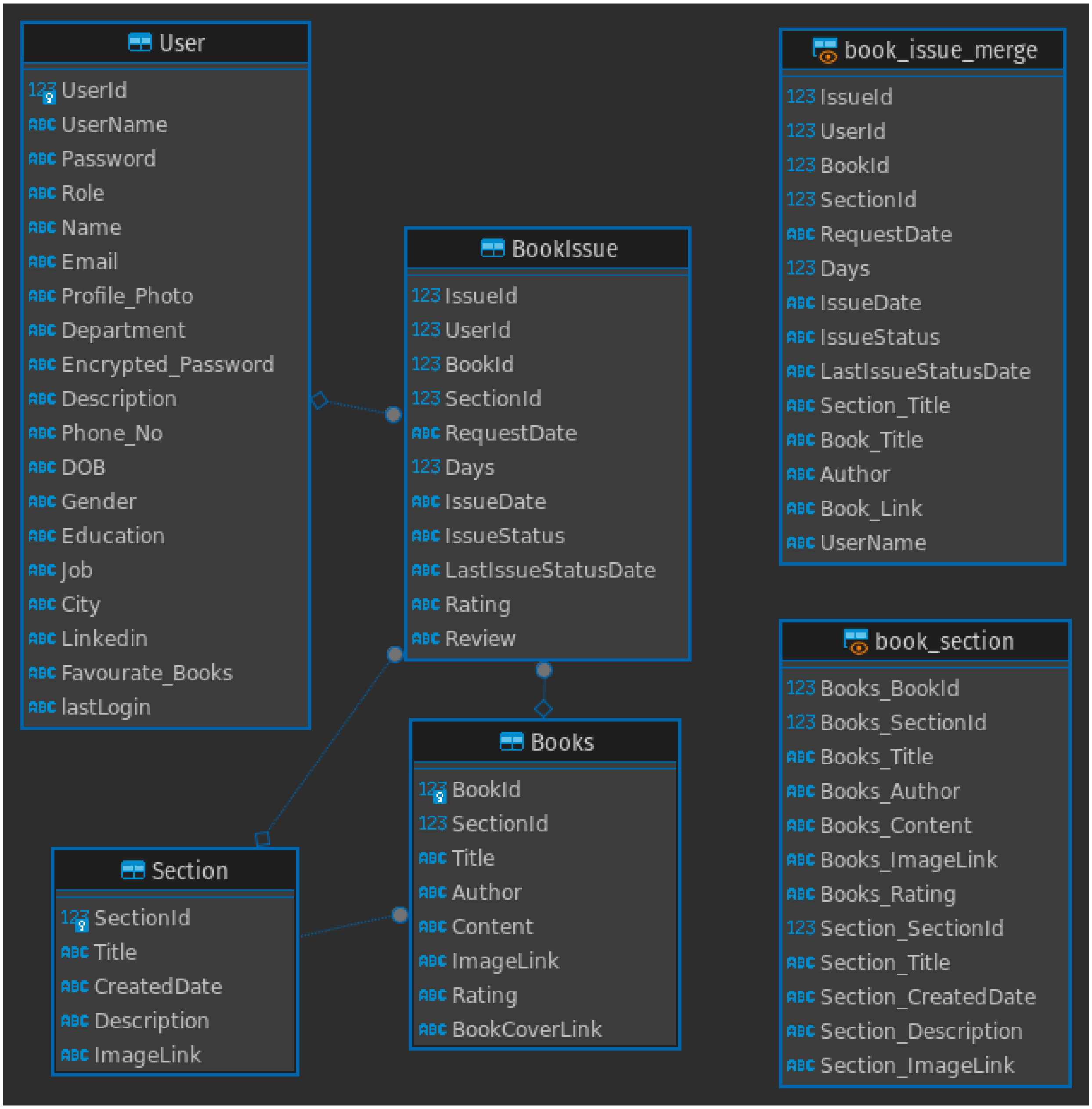
- **Flask:** A micro web framework for Python.
- **SQLAlchemy:** A Python SQL toolkit and Object-Relational Mapping (ORM) library.
- **Flask-SQLAlchemy:** An extension for Flask that adds support for SQL Alchemy.
- **Flask-RESTful:** An extension for Flask that adds support for quickly building REST APIs.
- **Matplotlib:** A comprehensive library for creating static, animated, and interactive visualizations in Python.
- **Seaborn:** A data visualization library based on Matplotlib that provides a high-level interface for drawing attractive and informative statistical graphics.
- **Pdfkit:** It helps to create PDF reports of websites which will be sent to users.
- **Smtplib:** It helps to send email to users with the help of mime and user credentials.

IDE Used:

- Used VS Code for the development of the project.
- DB browser for Table, View and Trigger creation.

ER Diagram:

Below is the ER Diagram of Schema of SQLite where all the data is being stored that consist of 4 table 2 views and 2 triggers.



Feature Implementation:

Apart from Version 1, which has features like section management, book management, book issuing, book return, and data visualization. In this 2nd version, I have added more features, like:

1. **User authentication:** Users and librarians can log in to access the system.
2. **Search Options:** It also serves as a search option for the user to find books, sections, and authors.
3. **5 books request limit:** Set the limit so that now the user can request up to a maximum of 5 books.
4. **Task Scheduling:** Send Monthly report or PDF and daily reminder if user haven't logged in with celery task.
5. **Caching:** Implemented caching with Redis for the stats page so that it won't make unnecessary requests to DB.
6. **Book Rating:** Now the user can rate a book once it has been returned after completion.
7. **CSV Download:** The user can now download CSV and can see all their books issued, requested, and completed book user-wise.
8. **One Click Download:** The user can now download a book in one click after access has been granted.

Conclusion:

The Library Management System V2 enhances the capabilities of the initial version by introducing significant features that improve user experience and system efficiency. The addition of user authentication, search options, and a book request limit ensures better control and accessibility for users. The implementation of Redis caching optimizes performance, especially for frequently accessed data. The integration of Celery for task management allows automated email reports and reminders, increasing user engagement. New features such as book rating, CSV downloads, and one-click book downloads further enrich the system, making it a comprehensive solution for managing e-books in a library environment. This upgrade demonstrates a successful blend of front-end and back-end technologies to create a robust, user-friendly application.

Video Link:

https://youtu.be/4h0BED7WHLE