Books Management Systems
Python Project Report

Course: INT213 (Python Programming)

Project Number - 18

Submitted To

Dr Ramandeep Sandhu

“School of Computer Science & Engineering”
Lovely Professional University

Submitted by

Anand Ubarhande
(K21MR**51) (12102327)**
Suraj Singh
(K21MR**52) (12102882)**
Vasudeva Kilaru
(K21MR**53) (12102356)**

Introduction

The project "Design a Books Management System in LPU using python" (BMS using python) aims to develop a compatible, extensible and general book management system software that can be integrated with any existing library management software to serve the key responsibilities such as in BMS issuing books, returns, record keeping, etc. Book management is an essential component of library management in any institution. The software we develop will adhere to the compatibility preferences without compromising on performance and security. The book management system accounts for how robust a library management system is and considers the importance of transaction management in the design of a book management system.

Project Objective

The book management system, since a vital part of any library management system, needs to be consistent about the storage and the activity such as transactions of books, their statuses, user accounts etc.

- The project is ensured to satisfy the basic needs of the book management system.
- The fundamental functionalities such as issuing books, returning books, viewing books on a stack, renewing a book, etc along with the requirements of any user account-based software.
- It also supports the creation of individual user accounts personalized to their transactions.
- Additionally, the interaction with the database i.e., MySQL connector in our software is made asynchronous.

The project support functionalities through independent, highly cohesive modules which are divided based on the operations they are responsible for.

Design

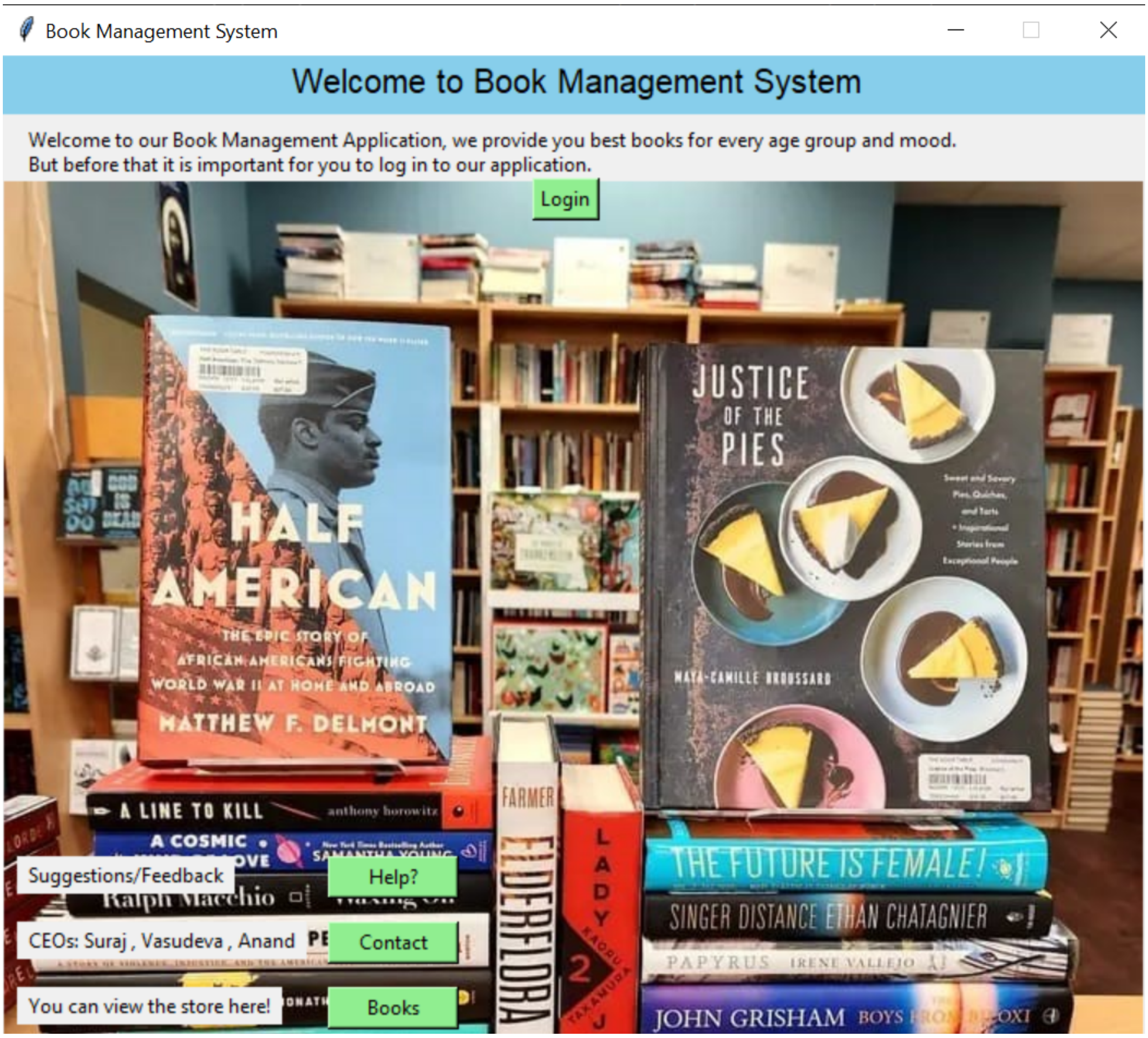
Design is the very crucial phase in designing any software. We have spent most part of the project duration brainstorming various designs to use for integrating the modules and establishing a secure database connection.

1. **Modules**

   The crucial modules, which might be simplified further, in building a book management system might include the following:

1. Main.py (Dashboard or Starter)
   Main dashboard window of the Book management system. There we will get options for login, Books, and Store.

The window of Main.py looks like



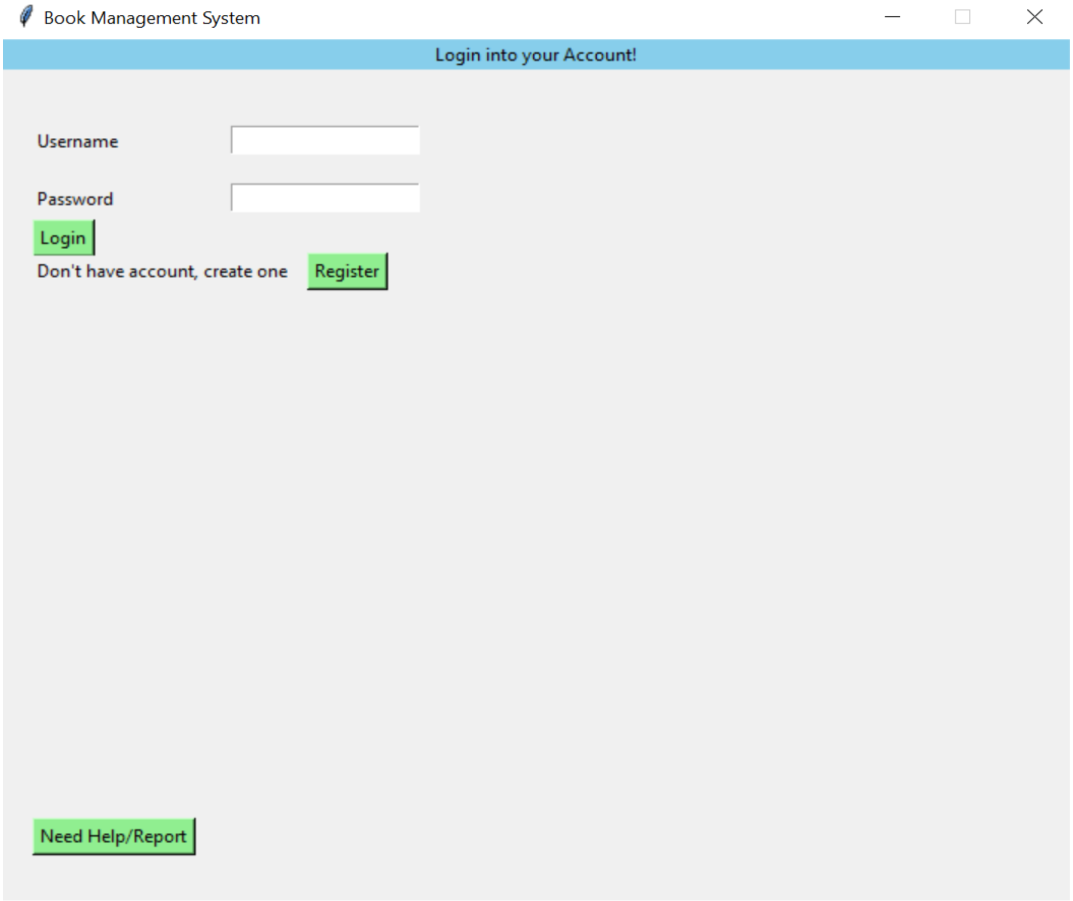
Book Management System

Welcome to Book Management System

Welcome to our Book Management Application, we provide you best books for every age group and mood.
But before that it is important for you to log in to our application.

2. Login.py (Login or Sign In)
   For entering the system for the book section or if you need help and to issue a book, return a book.
   New users are required to sign up on the system to get the books issued.

The window of Login.py looks like



Book Management System

Login into your Account!

Username

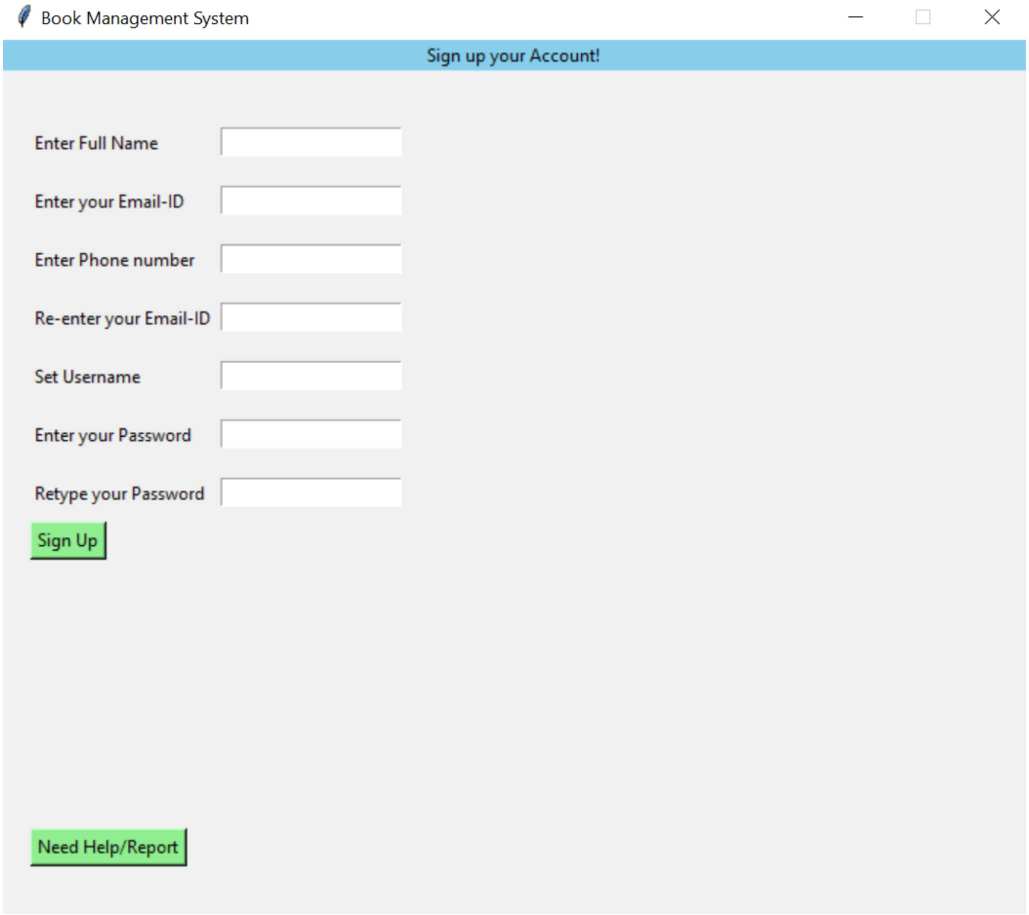
Password

Login

Don't have account, create one

Register

Need Help/Report



Book Management System

Sign up your Account!

Enter Full Name

Enter your Email-ID

Enter Phone number

Re-enter your Email-ID

Set Username

Enter your Password

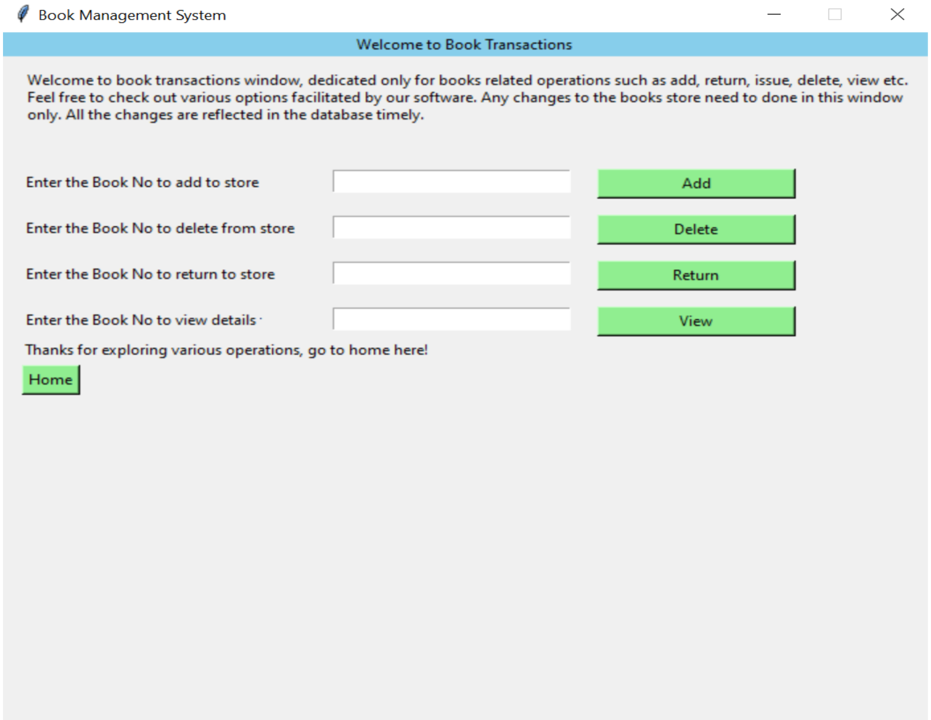
Retype your Password

Sign Up

Need Help/Report

3. Books.py(All books-related activities)
   An internal module connected to the main module, basically, for all books, contains related activities like

The window of Books.py looks like



Book Management System

Welcome to Book Transactions

Welcome to book transactions window, dedicated only for books related operations such as add, return, issue, delete, view etc. Feel free to check out various options facilitated by our software. Any changes to the books store need to done in this window only. All the changes are reflected in the database timely.

Enter the Book No to add to store

Enter the Book No to delete from store

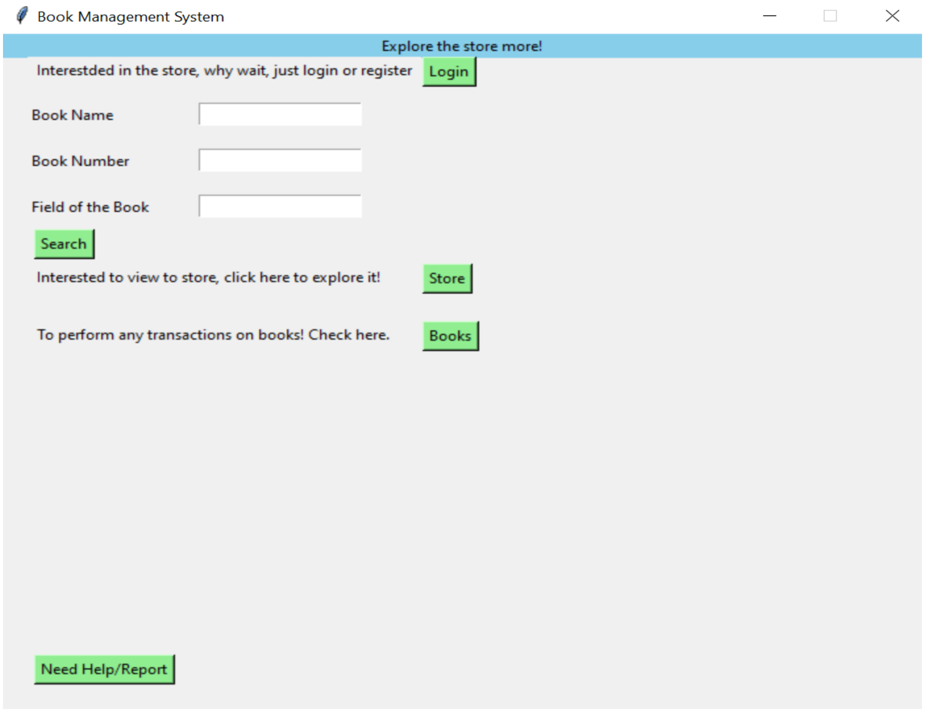
Enter the Book No to return to store

Enter the Book No to view details

Thanks for exploring various operations, go to home here!

4. Store.py(Book stocks and other activities)
   Updating the new arrival books in the database and setting up a book_id for that particular book.

The window of NeedHelp.py looks like



Book Management System

Explore the store more!

Interestded in the store, why wait, just login or register

Book Name

Book Number

Field of the Book

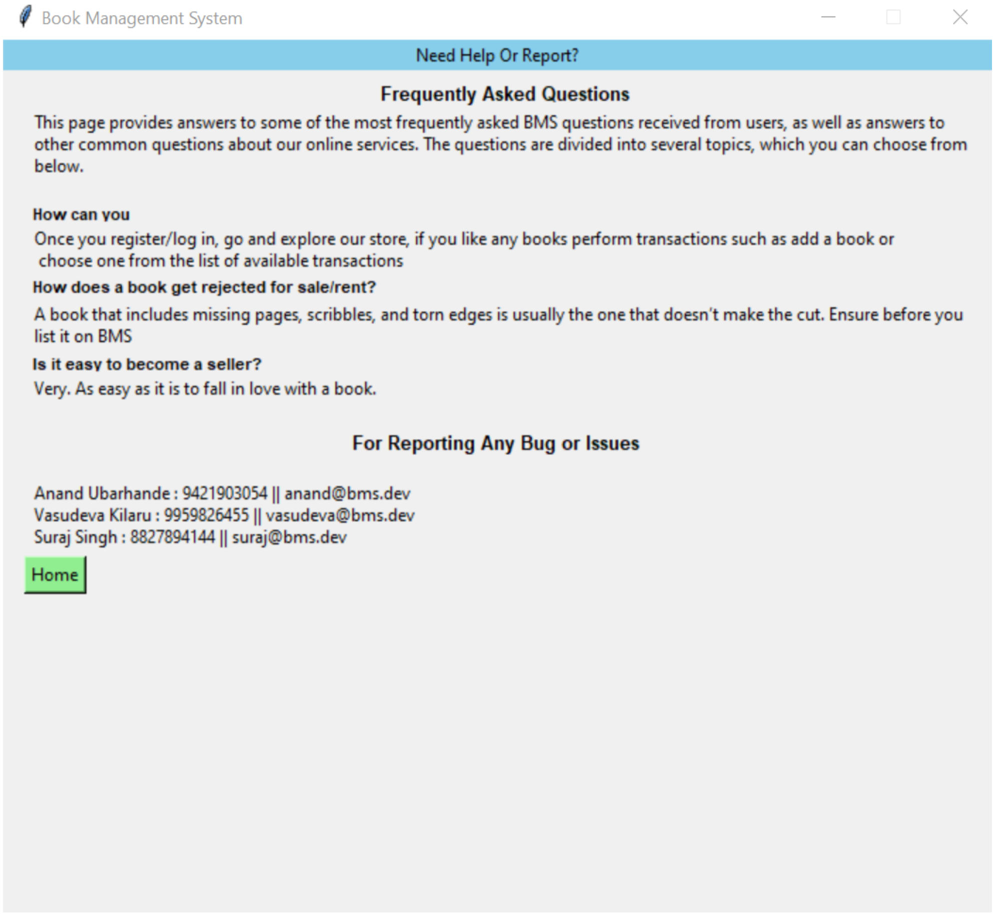
Interested to view to store, click here to explore it!

To perform any transactions on books! Check here.

5. NeedHelp.py (For newbies)

   For the consumers signing in on the system due to any glitch or technical problem users can seek help from the backend from the need help button

The window of NeedHelp.py looks like



Need Help Or Report?

Frequently Asked Questions

This page provides answers to some of the most frequently asked BMS questions received from users, as well as answers to other common questions about our online services. The questions are divided into several topics, which you can choose from below.

How can you

Once you register/log in, go and explore our store, if you like any books perform transactions such as add a book or choose one from the list of available transactions

How does a book get rejected for sale/rent?

A book that includes missing pages, scribbles, and torn edges is usually the one that doesn't make the cut. Ensure before you list it on BMS

Is it easy to become a seller?

Very. As easy as it is to fall in love with a book.

For Reporting Any Bug or Issues

Anand Ubarhande : 9421903054 || anand@bms.dev
Vasudeva Kilaru : 9959826455 || vasudeva@bms.dev
Suraj Singh : 8827894144 || suraj@bms.dev

2. Data Structures

Data Structures (DS) are the key components of designing and managing data interaction in any application. Multiple data structures are used to build the BMS. Above all the advanced data structures used, the important ideas and experiences gathered while working with a wide range of data structures and data management strategies have positively impacted the project in terms of performance, security and compatibility.

- *Data Management Strategy:*

  The data required to run a books management system is the user details to every registered user account and the data of books available in the store along with their statuses, expected arrival time etc. After a long study of SRS, we have decided to store the data of books in a pandas data frame, i.e., storing the data in a .csv file format, instead of using a dedicated table in the database. This helps to reduce the cost of maintenance while helping to make the software more efficient. Since the book's data is not a piece of confidential information i.e., the data is public and could be subject to any vulnerability. However, considering the importance of keeping the user details confidential we have set up a dedicated database to store and interact with the user accounts and information. We communicate with the database using a MySQL connector library that establishes a secure connection using a user id and password.

- *Data Structures & Designs:*
  The Data Structures are the key components of the BMS software. The data structures have enabled the connection between the modules and also interaction with the database. The primary data structures supporting the interaction between the database and application are lists, and tuples i.e., the database output is returned in a list of tuples where each row returned is a tuple. The **MySQL** database table is used to store the user account details and user preferences. An additional **CSV (comma-separated values)**. This CSV file is read in the program using the pandas and stores the data as a pandas data frame during the program's execution. The efficient data management mentioned in the earlier section is proven to be an effective design technique since the number of updates a books database statistically needs are far fewer than the number of updates a user profiles database. Every time a new user registers or new login happens, a request to the database has to be sent i.e., interaction with the database with a secure connection is required. This design enhances the performance of the application over an application built with books’ details stored in a dedicated database, i.e, this design increase the number of requests to the database decreasing the performance. These initiatives have facilitated the better design of our application.

3. Algorithms

Algorithms are the core of the performance of any real-world application. Similarly in our application as well, the algorithms played an essential role. On the whole multiple state-of-art algorithms are used. Some of them are the **Robin Karp** algorithm, a string-matching algorithm, used to get the search results for any search query to find a book in the store. Apart from that, multiple regular expressions are used to validate the data input from the user, which is underneath the hood using string matching algorithms. A **combination of insertion sort and merge sort** is used to sort the search results of any book.

4. **Design Patterns**
   Design techniques are the core software components that enhance the performance of the software. While working on the design and majorly during the initial days of the implementation phase, we brainstormed on almost every available design pattern among the 23 to understand the exact requirement of our BMS applications. However, we could conclude on two design patterns which are the **observer** pattern to notify the change in the status of any book if the status changes and the parent **factory** pattern to design the GUI for multiple windows in the application. The factory was developed by the name app.py in our project.

Results & Testing

The software testing is done by Anand. Since most of the coding and implementations are performed by Vasudeva & Suraj, we decided that it would be a good strategy to ask someone outside the development team to test the software, i.e. Anand. After a long testing phase is performed on each piece of software, the concerned person has approved the software with a few suggestions. The latest version available now is the version developed on the feedback given post the testing phase.

Conclusion

After thorough testing and multiple updated versions, the software available now works very closely to the one reported in the pre-evaluation report submitted. The code to the project has been open-sourced and is available for any over the internet to download with the dependency to play around. The link to download the application is here, where the installation guide is uploaded.

The skills acquired during the project are software engineering practices, best software design and implementation practices such as algorithms, data structures, and design patterns. One of the major milestones of the project is coming up with the data management scheme i.e., explained under the data management section. The one thing we innovated while working on the project is the effective method to manage data and its interactions in the application.

In summary, the project fulfils each and every requirement of a BMS application, with compatibility to extend with existing software upon tweaks. The highlights of the project are, extensively tested software, with robust algorithms such as Robin Karp, a blend of insertion sort and merge sort along with strong design patterns like a factory, and an observer along with a very innovative and efficient data management strategy. Besides these, a testable version is open-sourced for testing in the above-provided link.

Acknowledgement

The work was uniformly distributed among all the teammates. It is to acknowledge that all of us have put equal effort into building the project. Any further credits or critiques are subject to each contributor to the project.

The project's modules mentioned under modules are divided into equal ratios, whose GUIs are implemented by individuals. The integration and connection to the database using MySQL connector are done by Suraj and Vasudeva. The testing is performed by Anand. Each of us has equally contributed to the design discussions and report preparations.

References

1. Final project instructions provided for the course INT213 Python Programming are used to design the basic layout and prepare the initial level of SRS (Software requirements and specifications).
2. The database of books and their details are downloaded from online resources. The data was collected from different resources hence the credits due to whomsoever are concerned with the ownership of the data.
3. Documentation of Tkinter, GfG's tutorials, and other resources on using Tkinter to build system apps are heavily referred to implement the design.
4. Pandas documentation and MySQL documentation are referred to build databases and connect with the application.
5. The design implementations are due to open-source libraries such as python, pandas, MySQL connector, NumPy etc

6. Some of the relevant software implementations available in the market are referred to better understand and design the software to serve the needs of BMS (Books Management System).
7. A BMS software available online due to the GfGs is used as a base to come up with our own design and development of the project.
8. The pre-evaluation document submitted by our project is thoroughly referred to develop the project in alignment with the requirement of the BMS application.