

Northeast Valley Bookstore

¹Arundhati Pathrikar, ²Sourabh Kumar,

³Santosha Nagaratnakar Chakkapalli,

⁴Sathwik Reddy Chelemela

¹Information Systems, Northeastern University

²Information Systems, Northeastern University

³Information Systems, Northeastern University

⁴Information Systems, Northeastern University

{pathrikar.ar, kumar.sou, chakkapalli.s, chelemela.s}@northeastern.edu

Abstract - Promoting the habit of reading books is an invaluable endeavor that commences during our formative years. This begins with the enchanting tales narrated by our parents in early childhood, instilling in us an appreciation for stories. As we venture into the realm of education, the fundamental practices of reading and writing further solidify the book's significance in our lives. These foundational experiences endeavor to weave the fabric of literature into our daily existence.

Despite the increased accessibility to books in contemporary times, numerous individuals encounter hurdles in obtaining them. Financial constraints and their environment act as barriers preventing access to literature. Moreover, there exists a segment of people devoid of these access or financial impediments, yet they wholeheartedly embrace the culture of libraries, cherishing the serene atmosphere these sanctuaries of knowledge offer.

Our project embodies a visionary initiative – the design and implementation of a library system meticulously crafted to cater to the reading habits and thirst for knowledge of passionate readers. The objective is to curate an extensive array of book choices, catering to diverse interests and preferences, thereby enticing readers into a literary adventure tailored to their inclinations.

However, the crux of our endeavor extends beyond mere collection curation. It aspires to be a conduit for inclusivity, providing a bridge for those grappling with financial limitations or facing accessibility challenges. This innovative library system stands as a beacon, offering a means for individuals to borrow books that would otherwise be out of reach due to financial constraints or limited access.

By fostering a culture of reading and knowledge-sharing, our project seeks to break down barriers and kindle the flames of curiosity and intellectual growth. Through this endeavor, we aim to create a haven where the love for books transcends obstacles, empowering individuals to enrich their lives through the written word.

Keywords *JavaFX, MySQL, Data Structures, UI, Database, Book, Person, User, Admin*

I. PROBLEM DESCRIPTION

Our project is a system aimed at managing books, transactions, and members in a public bookstore setting. Detailing all the actions such as user and admin login, returns and issues books, issuing the fines that come with late books, deleting bookings, etc. The purpose of the project is to provide a streamlined and efficient way for book management from those issuing the books and people receiving the books. The output of this system would be a better user experience and higher efficiency. The scope only covers the bookstore, but you can use this information in any kind of inventory-based operation in which you want a better way to manage. This project helps Northeastern students who struggle with book management with delays and hindrances.

The project's scope includes creating and designing an intuitive and user-friendly user interface, integrating several features like searching books, issuing books, returning books, issuing fines, and maintaining a database to hold data about books, students, and transactions.

The Northeast Valley Bookstore project aims to alleviate the difficulties that students and staff encounter when handling books and transactions through manual means, which can be laborious and prone to mistakes. The project will boost the user experience and operational efficiency of the bookstore by offering an automated and simplified platform.

II. ANALYSIS (RELATED WORK)

In our research of this project and its apparent solution, we found many different peer-reviewed journals that detailed the overall problems that come with book and library management. Many proposed solutions on various levels of intensity. For our projects, we want to focus on the features of issuing books and fines, searching, and returning books. In that search, we also had to research many different books to enhance our skills within Java FX and what we should do to make our system.

During the development phase of our Northeast Valley Bookstore JavaFX application, we performed a comprehensive literature review to grasp the latest advancements in JavaFX

application development. Our findings revealed that JavaFX stands out as a favored and adaptable technology, widely utilized for crafting intricate and dynamic user interfaces across diverse applications, encompassing both desktop and mobile platforms.

We looked at the article “Designing Web-based Library Management System” by Tsega, Weldu, and Araya which gave us the rundown of a library, management system and gave us insight on how we can design our system. This article emphasizes design and further making a great user experience by specifically saying “User Interface (UI) Design emphasizes expecting what users want to do and confirming that the interface has features that are easy to access, understand, and uses to smooth those actions” (Tsega, Weldu, Array). Bookstores are already very drab and outdated and innately comes because physical books feel outdated innately, and often get left behind, As the technical world catches up books always end up left behind so to have a system up to date, we have a User Experience that can rival the experience someone would have with a modern device fast and intuitive.

“Research on “Book Shop Management System” by Sneha Sakharkar, Snehal Doifode, and Shubhangi Karnuke was extremely helpful in establishing the basis of what it means to be what we are solving and making in our way. This journal does not give us the answer but brings us close to the solution using the automated bookshop. The proposed bookshop is said by Sakhaekar, Doifoide, and Kanrnuke to be able to provide “an easy way of searching reserving and purchasing of books. It’s worth analyzing and identifying the benefits as it would directly influence the productivity of the shop.” Furthermore, we can use this as a bridge to what our solution would be, what we strive for it to do, overall benefits, and the end goals. However, the proposed solution in this journal is a web solution means it uses more of a HTML based application and does not use Java FX nor does it say anything about issuing fines and making user and admin accounts, so this system is mostly used only with book managing and is limiting in that scope. We decided to open the scope more and follow more of what people can do within the account of a regular user and an admin and make the actions revolve around those. This literature helped us to a solution that was more expanding and had a niche.

III. SYSTEM DESIGN

Use case diagrams are representations illustrating the general interactions between individuals and a system. Within our Bookstore System project, two primary actors engage with the system: the Admin and the User. The admin assumes responsibility for the fundamental operations within the library, engaging more extensively with the system compared to the User. Conversely, the User possesses the ability to borrow books, view a history of past purchases, conduct searches within the library, and has the privilege to list books. This

access is granted upon logging into the system using a designated username and password determined by the User.

Person: Each individual possesses unique attributes such as their name, surname, password, and a record of purchased books. A person object is equipped with a constructor containing these properties, along with public getter and private setter methods for accessing this information. Both Admin and User entities are categorized as persons.

Admin: Admin objects have dedicated constructors and functions including deleteBook (to remove a book), giveBook (to provide a book to a user), addBook (to include a book in the library), searchBook (to explore among all available books), and listBook (to display a list of books).

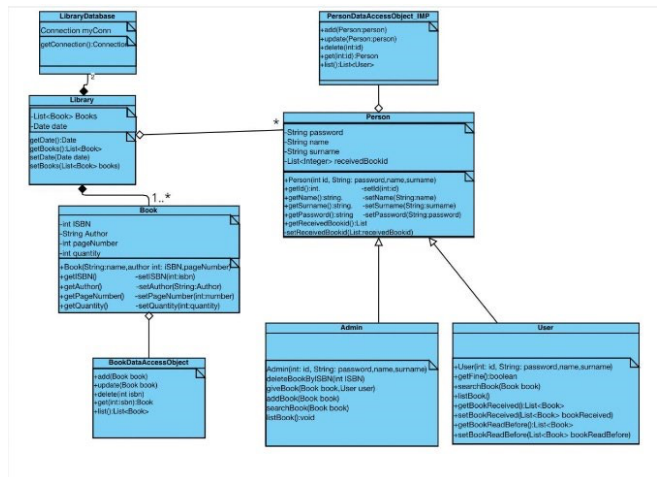


Figure 1: Class UML Diagram

User: Similarly, the User object possesses its constructor and functions such as searching for penalties (getting), searching among owned books (searchBook), displaying received books (getBookReceived), and viewing previously read books (getBookReadBefore).

Library: A private list within the library object manages the collection of books available. Additionally, a private variable called Date denotes the date. Functions with getters and setters are available for accessing and modifying these variables.

Book: Books reside within the library and are characterized by attributes like ISBNs, author names, page counts, quantities, and titles. Functions equipped with getters and setters facilitate access to these book attributes.

LibraryDataBase: The library is associated with a single database, comprising two tables, which necessitated

representation in the diagram. The Database object incorporates a local connection (myConnection) and a function getConnection that returns a Connection object.

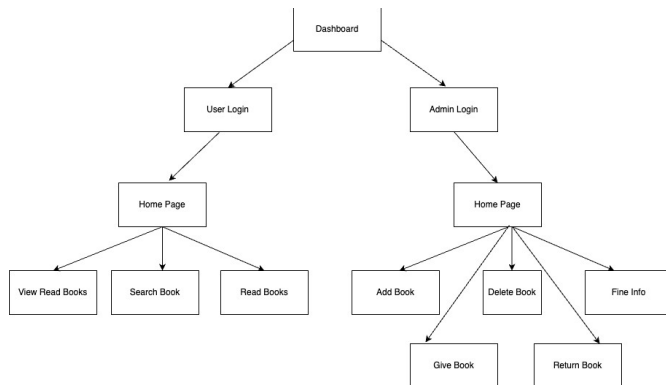


Figure 2: System Architecture

IV. IMPLEMENTATION

When delving into the finer details of our project's scope, the admin assumes the role of a user with comprehensive access privileges within the system. The admin possesses the authority to access all information stored within the system. This includes the capacity to add books and members to the library, conduct searches within the book and member databases, and review details such as purchased books by members, due dates for book returns, and any associated penalties.

Conversely, Members have access to a subset of information within the system. They can view the books they have borrowed, along with details regarding the date of borrowing and the deadline for returning the book. Both Members and the Admin are required to log in to the library system using their unique username-ID and password credentials.

Application Interface functionality:

User login functionality:

- Display the login form.
- Create an account option.

Admin login capability

- Verifies access permission through Java to connect with the database.
- Display login form for admins.

Support and contact

- Redirects to designated sites when clicked.

Add Book feature.

- Input book properties (ISBN, Name, Author, etc.) into the SQL for books
- Lists books based on their attributes.

Book Search

- Users can search for books using specific features via a search box.
- Matching books retrieved from the book database and displayed to users through the interface.

Book lending process

- User eligibility check and ensure no outstanding books.
- Lends books.
- Updates marked books database and adjusted book quantities if multiple copies exist.
- User status changes to "books received" upon successful transaction.

Book Return Process

- Users can return the book once they have completed reading.
- Admin can accept the book.

Fine Information

- Admin can check if the user has a fine or not.
- Users with fine can see their fines on the profile page.

Login Validation

- The system won't allow users and admin to log in if the credentials don't.

Performance Requirements:

- Book-related functions operate more swiftly compared to alternative storage systems due to SQL. For instance, a book query takes less than a second.
- Immediate redirection to essential sites upon clicking the contact and support buttons.

Security Requirements (Token):

- Physically stored databases offer faster and more secure solutions than cloud-based alternatives.
- Enhanced security for member logins achieved through token utilization.

Quality Attributes:

- The interface showcases superior display quality, facilitated by JavaFX, surpassing other interfaces.
- Consistently ensuring content remains easily readable for users.

V. EVALUATION

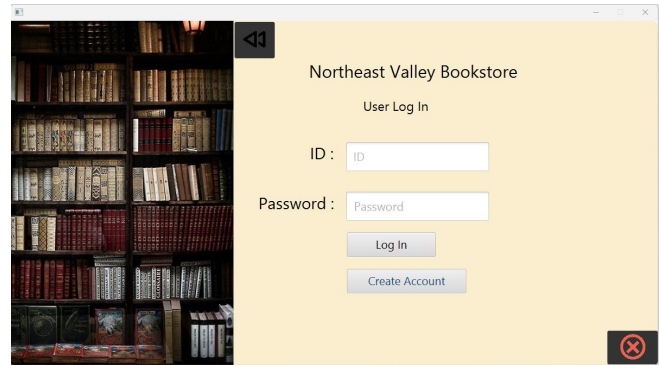


Figure 3: User login page

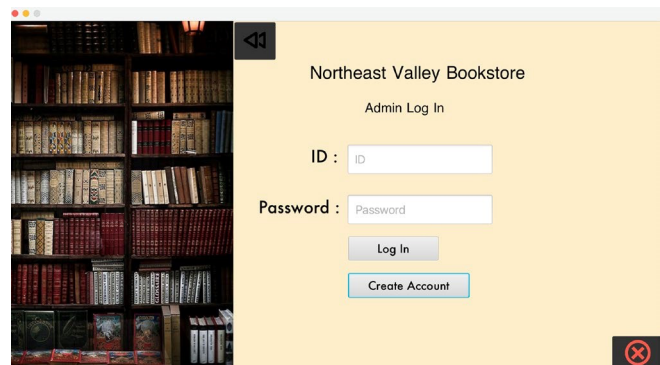


Figure 4: Admin login page

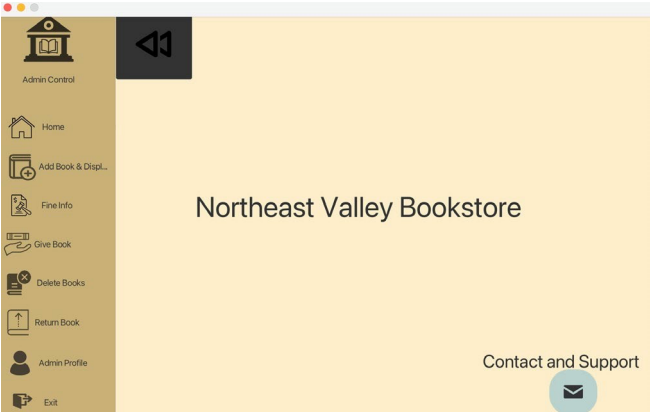


Figure 8: Give book

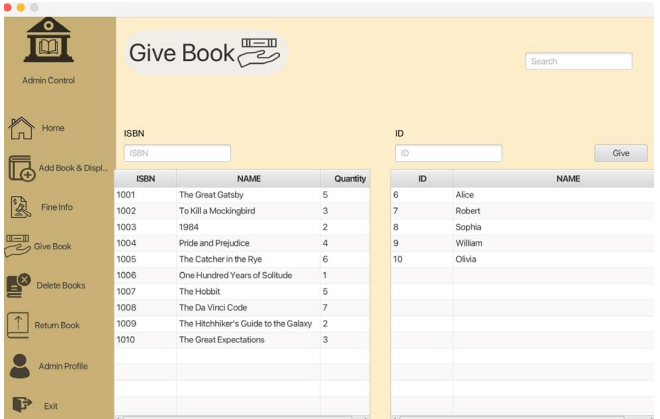
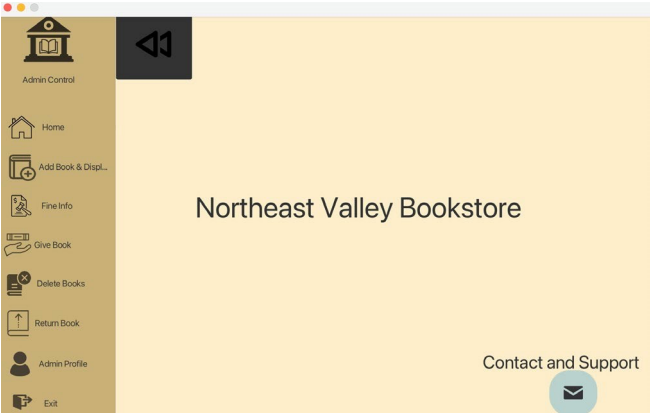


Figure 5: Support and contact



Book

Figure 6: Add Book

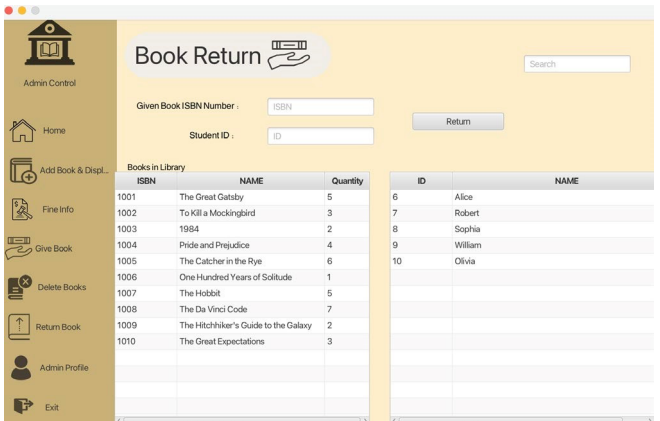


Figure 9: Return

NAME	ISBN	AUTHOR	PAGES
The Great Gats...	1001	F. Scott Fitzgerald	300
To Kill a Mocki...	1002	Harper Lee	250
1984	1003	George Orwell	150
Pride and Preju...	1004	Jane Austen	500
The Catcher in...	1005	J.D. Salinger	200
One Hundred ...	1006	Gabriel García Márquez	350
The Hobbit	1007	J.R.R. Tolkien	280
The Da Vinci C...	1008	Dan Brown	400
The Hitchhiker...	1009	Douglas Adams	320
The Great Exp...	1010	Charles Dickens	450

Book

ID	NAME	SURNAME
6	Alice	Clark
8	Sophia	Anderson
10	Olivia	Thomas

Figure 10: Admin Fine Info Page Figure 7: Search

PROFILE

You have a fine
You must return the book: The Great Gatsby

Name : Alice Clark

Current Books

Book Name
The Great Gatsby
To Kill a Mockingbird

Deadline

2023-12-01

Change Password

Old password

New Password

Figure 11: User Profile where fine is visible

Northeast Valley Bookstore

Admin Log In

ID :

Password :

Wrong Username/Password, Try Again.

Figure 12: id/Password Checksum

VI. DISCUSSION (REFLECTION)

Our project goal was to create a JavaFX application for a Northeast Valley Bookstore that allows users to search, issue, and return books. We linked the UI and functionality with the backend by storing book and student information in a MySQL database. Overall, our project was a success in terms of meeting its goals and generating a usable application.

In addition to the core functionalities of book searching, issuing, and returns, we implemented a fine management system to enhance user accountability. Users can now check if they have any outstanding fines, and the system prevents the issuance of

books to users with unpaid fines. This feature not only promotes responsible borrowing but also streamlines the administrative process, ensuring a fair and efficient operation for both users and bookstore staff.

Our project also illustrated the usage of many JavaFX components to create a user-friendly and visually appealing interface, including Anchor pane, Button, Textfield, ImageIcons, Font Awesome, GridColumns, and Image View. We also employed data structures like Stack, Array List, and Queue to improve the efficiency of our programs.

Furthermore, we introduced an email integration feature to facilitate communication and support. Users can now easily reach out for assistance or inquiries directly through the application. This not only adds a layer of convenience for users but also establishes a more direct and immediate channel for addressing concerns. Email integration serves as a valuable tool for fostering a responsive and user-centric environment within the Northeast Valley Bookstore application.

Reflecting on our project, there are areas where we could have further improved the application. One avenue for enhancement could be the incorporation of user feedback mechanisms within the application, allowing users to provide suggestions or report issues seamlessly. Additionally, refining the application's errorhandling mechanisms and incorporating more robust security measures would contribute to a more resilient and reliable user experience. Continuous user testing and feedback loops could have been integrated during the development process to identify and address potential usability issues earlier on. Overall, while we have achieved our primary objectives, there is always room for iterative improvement to ensure the application remains adaptive and user centric.

VII. CONCLUSIONS AND FUTURE WORK

In conclusion, the Northeast Valley Bookstore project has successfully addressed the need for an efficient and userfriendly system to manage books, members, and transactions in a public bookstore setting. The implementation of the project using Eclipse with the MVC architecture has provided a structured

and modular approach, allowing for easier maintenance and future expansions. The incorporation of multiple user logins, including Admin and User roles, enhances the flexibility and security of the system.

The advantages and benefits of our solution are numerous. The system significantly streamlines bookstore operations, providing a seamless experience for both members and staff. The ability to search for books, issue and return them, manage fines, and facilitate transactions has enhanced overall efficiency and customer experience. The support and contact feature, along with the email integration, further contributes to improved communication and user engagement.

Throughout the development process, we encountered various challenges that were successfully addressed. However, if more time were available, there are areas for potential improvement. Enhancing the user interface and experience, incorporating additional features based on user feedback, and conducting further testing to identify and resolve any remaining issues would be valuable. Additionally, considering future developments in technology, exploring the integration of advanced security measures and additional functionalities could further elevate the system's capabilities.

In terms of future work, continuous refinement of the application based on user feedback and evolving bookstore requirements is essential. Expanding the system to accommodate new bookstore functionalities, such as inventory management or analytics reporting, could be explored to enhance the overall bookstore management capabilities. Regular updates and iterations will ensure the Northeast Valley Bookstore system remains adaptive and aligned with the dynamic needs of the bookstore environment.

VIII. JOB ASSIGNMENT

Arundhati Pathrikar primarily focused on optimizing the system for regular users. She dedicated her efforts to planning and crafting user-facing components like the Homepage, book search, user profiles, and reading sections. Arundhati meticulously ensured that features like adding, editing, and deleting information worked seamlessly for users. She proactively addressed potential user issues and implemented strong security measures for user logins. Moreover, Arundhati conducted thorough tests to guarantee that users would find the system intuitive and easy to navigate.

Sathwik Reddy Chelemela played a crucial role in shaping how the system functions for administrators. His responsibilities revolved around planning and ensuring the smooth operation of the parts controlling administrators' activities. He meticulously designed and developed various user interface elements specifically tailored for administrators, such as the Homepage and sections facilitating book addition and distribution. Additionally, Sathwik focused on fortifying the system against errors by implementing robust error-handling mechanisms. He also prioritized security by creating a secure login system for administrators. Furthermore, Sathwik integrated an email

feature, enabling seamless communication among administrators within the system.

Santosha Nagaratnakar Chakkapalli brought expertise to managing the system's data and implementing specialized functionalities for administrators. He designed the system's data storage and connections while spearheading the development of features facilitating administrators' tasks, such as book deletion, returns, and profile management. Santosha emphasized system security by implementing robust password protection measures. He also led comprehensive testing to ensure the system's reliability and glitch-free functionality.

Sourabh Kumar concentrated on ensuring the system's overall efficiency and integration. He worked closely on developing the core backend functionalities that link the system's front end with its database. Sourabh ensured that administrators and regular users experienced a seamless flow of operations. He contributed to the error-handling mechanisms and collaborated on designing secure login features. Additionally, Sourabh tested cross-functional components to confirm smooth operation and user satisfaction.

Collectively, Sathwik, Arundhati, Santosha, and Sourabh collaborated to create a user-friendly system, focusing on administrators' ease of use, robust data management, and seamless user experience. Their joint efforts ensured the system's security, smooth operation, and readiness for widespread user satisfaction.

REFERENCES

- [1] Oracle Scene Builder Documentation
- [2] Java Documentation - Get Started ([oracle.com](https://www.oracle.com)) [3] JavaTpoint. (n.d.). *JavaFX tutorial* - javatpoint. www.javatpoint.com.
<https://www.javatpoint.com/javafxtutorial>
- [4] Gluon (2023), *JavaFX Scene Builder*. 21.0, Computer Software, <https://gluonhq.com/products/scene-builder/> [5] MySQL. (n.d.). *MySQL Documentation*.
<https://dev.mysql.com/doc/>
- [6] Oracle Corporation. (n.d.). JavaFX. Retrieved from <https://www.oracle.com/java/technologies/javase/javafxoverview.html>
- [7] Sneha Sakharkar, Snehal Doifode, Shubhangi Karnuke, Research on "Book Shop Management System", *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, Volume 5 Issue III, March 2017, 490492 <https://doi.org/10.22214/ijraset.2017.3089>
- [8] Snehal Shedage, Maithili Kamble, Neha Kadam, Sanika Mali, Online Book Store Management, *International Journal of Research Publication and Reviews*, Vol 2, no 12, pp 11121115, December 2021, <https://www.ijrpr.com/ISSN/25827421>