Library Management System

Project Documetation

Faculty 2: Computer Science
Course: Java – Object Oriented Programming
Lecturer: Prof. Dr. Doina Logofătu

Prepared by Group 1:
Nassim Laaraj
Mohamad Moaqaly
Yousef Ghanem
Javiel Figuereo Schade

Semester: 5

Submitted: 12.02.2023

Abstract

The Library Management System is a comprehensive software solution for managing the operations and resources of a library. It provides an efficient way to store, organize, and access information about books, students, and administrators. The system automates many of the manual tasks associated with managing a library, such as registering students and keeping track of borrowed and returned books.

One of the key features of the Library Management System is its centralized database, which serves as a single source of information about the library's collection of books and resources. This database allows administrators to easily manage and update information about the library's collection, including adding new books, removing books that are no longer needed, and updating information about existing books. The database also provides students with access to information about the books available in the library, including the titles, authors, and descriptions of books.

The Library Management System also provides students with the ability to borrow and return books by interacting with an administrator. When a student borrows a book, the system automatically updates the database to reflect the change in status of the book, ensuring that the information is always accurate and up to date. The system also provides administrators with a list of students who have borrowed books and when they are due to return them, allowing them to monitor and manage the borrowing process effectively.

Author’s declaration

We, the undersigned, hereby declare that this submission is entirely our own work, in our own words, and that all sources used in researching it are fully acknowledged and all quotations properly identified. It has not been submitted, in whole or in part, by us or another person, for the purpose of obtaining any other credit / grade. We understand the ethical implications of our research, and this work meets the requirements of the Faculty of Applied of Sciences, Faculty Computer Science and Engineering, Project Java „Library Management System“, Professor Logofatu.

Student Name: Nassim Laaraj
Matriculation number: 1366099

Student Name: Yousef Ghanem
Matriculation number: 1310760

Student Name: Mohamad Moaqaly
Matriculation number: 1310757

Student Name: Javiel Figuereo Schade
Matriculation number: 1342563

Contents

List of Figures

1 Introduction

1.1 Description

The goal of this project is to develop a Graphical User Interface called Library Management System. The System is made for students who need to borrow a book from the university library, before that they must be registered by an administrator so the student can login with his Surname and his studentID number.

There are two scenarios for login which are for Admin and student.

1.1.1 Login as Admin

Once you logged in as an admin, you will have access to a range of functions that allow you to manage the library's collection and users. These functions may include:

Adding books: As an administrator, you will be able to add new books to the library's collection by entering the relevant information, such as the title, author, and subject matter.

Adding users: You can create new accounts for users who want to borrow books from the library. This may include students, faculty, and other members of the community.

Viewing users: You can view a list of all registered users with the library, as well as their borrowing history and other relevant information.

Viewing books: You can view a list of all books in the library's collection, including information about their availability and more.

Viewing issued books: You can view a list of books that are currently checked out by users.

Viewing returned books: You can view a list of books that have been returned by users.

Issuing books for users: As an administrator, you will be able to check out books on behalf of users, either in person or remotely.

Making entries of returned books: You can update the library's records to reflect when a book has been returned by a user.

As an administrator, it is important to keep the library's records accurate and up-to-date. This will ensure that users are able to borrow and return books smoothly, and that the library has an accurate record of its holdings and circulation.

1.1.2 Login as Student

To login as a student to a library management system, you will need to have an account with the library. This account is usually created when you first register with the library as a student by an administrator.

Once you have an account, you can access the library using a dedicated login portal (login as student) . You will be prompted to enter your login credentials, which consist of your last name and student ID.

Once you are logged in, you will be able to borrow books, return books, view books that are available in the library's collection and view the issue book.

1.2 Motivation

A university library management system is a vital tool for any university library, as it helps to efficiently organize and manage the library's collection, as well as assist students in finding and borrowing books. A good library management system can greatly improve the overall functioning and efficiency of a library, making it a valuable investment for any university.

There are several benefits to implementing a university library management system. First and foremost, it allows for better organization and management of the library's resources. With a library management system, librarians can easily track the availability of books, as well as keep track of who has borrowed which books and when they are due to be returned. This helps to ensure that the library's collection is being used efficiently and effectively.

1.3 Requirements

The requirements are very important for the success of the software project, they provide a clear picture of the work that needs to be done. When the Requirements are not respected, the project can be in the end very expensive and can lead to fail. Our project, called "Library Management System", fulfils all requirements that were expected. The GUI allows the login as administrator with the required information and also the registration of many students by the admins

In a library management system where students are required to be registered by an administrator before they can login, the registration process is typically the first step in using the system. As an administrator, you will be responsible for creating student accounts and providing the necessary login credentials, such as a student ID and the student's last name.

Once a student has been registered, they will be able to login to the system using their student ID and last name. From there, they will be able to borrow and return books, and view the library's collection. It is important to ensure that only students who are eligible to borrow books from the library are registered in the system. This may involve verifying their enrollment status or other qualifications. It is also important to keep student records up to date and accurate, so that the system can function smoothly and efficiently. If you have any questions or issues with the registration process, you should contact the library for assistance.

1.4 Software and Tools

In this Java project, the team used a laptop with the following specifications:
Device name: LAPTOP-A26M7RAF
Processor: AMD Ryzen 9 5900HS with Radeon Graphics, with a clock speed of 3.30 GHz
Installed RAM: 16 GB, with 15.4 GB being usable.
System type: 64-bit operating system, with a x64-based processor

The team used several software and tools to collaborate and complete the project efficiently. GitHub: GitHub is a web-based platform for version control and collaboration. It is widely used by developers to manage their projects and keep track of changes made to the codebase. GitHub allows multiple team members to work on the same project and merge their changes into one codebase. The platform also provides a history of changes made to the codebase, which makes it easy to revert to a previous version if necessary. Additionally, GitHub makes it easy to share code with others and collaborate on projects with remote team members. (Source: https://en.wikipedia.org/wiki/GitHub)

Microsoft Word: Microsoft Word is one of the most widely used word processing software. It is a powerful tool for creating and editing text documents and is capable of handling a wide range of formatting and layout options. The team used Microsoft Word to write the documentation for the project. This allowed team members to collaborate on the document in real-time, making it easy to edit and update the document as needed. (Source: https://en.wikipedia.org/wiki/Microsoft_Word)

Microsoft Teams: Microsoft Teams is a collaboration platform that integrates with other Microsoft products such as Word, Excel and PowerPoint. This platform allows team

members to communicate, collaborate and share files in real-time. The team used Microsoft Teams to share the word file for the documentation, making it easy for everyone to work on the document together. This platform also provided an easy way for team members to discuss and make changes to the document in real-time, making collaboration much more efficient. (Source: https://en.wikipedia.org/wiki/Microsoft_Teams)

LaTeX: LaTeX is a typesetting software that is widely used in the creation of academic and scientific documents. It is a powerful tool for producing well-formatted documents, with a range of formatting options that make it easy to create professional-looking documents. The team used LaTeX to create the project documentation, ensuring that the document was well-formatted and easy to read. (Source: https://en.wikipedia.org/wiki/LaTeX)

IntelliJ IDEA: IntelliJ IDEA is a Java integrated development environment (IDE) that is widely used by Java developers. However, the team found it not to be suitable for their project, so they switched to Apache NetBeans IDE. (Source: https://en.wikipedia.org/wiki/IntelliJ_IDEA)

Apache NetBeans IDE: Apache NetBeans IDE is a free, open-source, cross-platform Java IDE. This platform provides a range of features that make it easy to develop Java applications, including a visual debugger, code completion and refactoring tools. The team used version 15.8.2 of Apache NetBeans IDE for this project, ensuring that the development process was efficient and streamlined. (Source: https://en.wikipedia.org/wiki/NetBeans)

Figma: Figma is a cloud-based interface design tool that enables designers to collaborate in real-time. This platform provides a range of features that make it easy to create professional-looking designs, including a visual editor, prototyping tools and collaboration features. The team used Figma to create a poster that gave an overview of the project and a disruption table. This allowed team members to collaborate on the design in real-time, making it easy to make changes and updates as needed. (Source: https://en.wikipedia.org/wiki/Figma_(software))
JCalendar-1.4.jar: JCalendar is a Java date chooser bean that allows for graphically picking a date. It consists of several Java beans. (Source: http://www.java2s.com/example/jar/j/download-jcalendar14jar-file.html)

Absolute Layout: AbsoluteLayout is a LayoutManager that acts as a replacement for the ”null” layout. It allows for the placement of components in absolute positions. (Source: https://www.cs.brandeis.edu/~hosang/BiVoSite/API/org/netbeans/lib/awtextra/AbsoluteLayout.html)

MySQL Connector/J: Connector/J is a library that provides connectivity for client applications developed in Java programming language with MySQL. It implements the Java Database Connectivity (JDBC) API, as well as several value-adding extensions. It also supports the X DevAPI. (Source: https://dev.mysql.com/doc/connector-j/8.0/en/connector-j-overview.html)

rs2xml.jar: rs2xml is a jar library that allows for making the result set of a query an input for a table model. It is a useful tool for working with JTables. (Source: https://stackoverflow.com/questions/27679867/jtable-how-to-use-rs2xml)

Java Development Kit (JDK): The JDK is a software development environment used for developing Java applications and applets. It includes the Java Runtime Environment (JRE), a compiler, an interpreter/loader, an archiver, a documentation generator, and other tools required for Java development. (Source: https://www.techopedia.com/definition/5594/java-development-kit-jdk)

In conclusion, the team used a combination of powerful tools and software to complete their project efficiently. From GitHub for version control and collaboration.□

1.5 Team Management

Given the rapid advancement of technology, we have established a regular meeting schedule to keep track of our progress, provide support, and plan for upcoming work. Our project has been divided into multiple parts, with each team member being assigned specific responsibilities for different aspects of the application. To efficiently manage our teamwork, we have created a team management table 1.

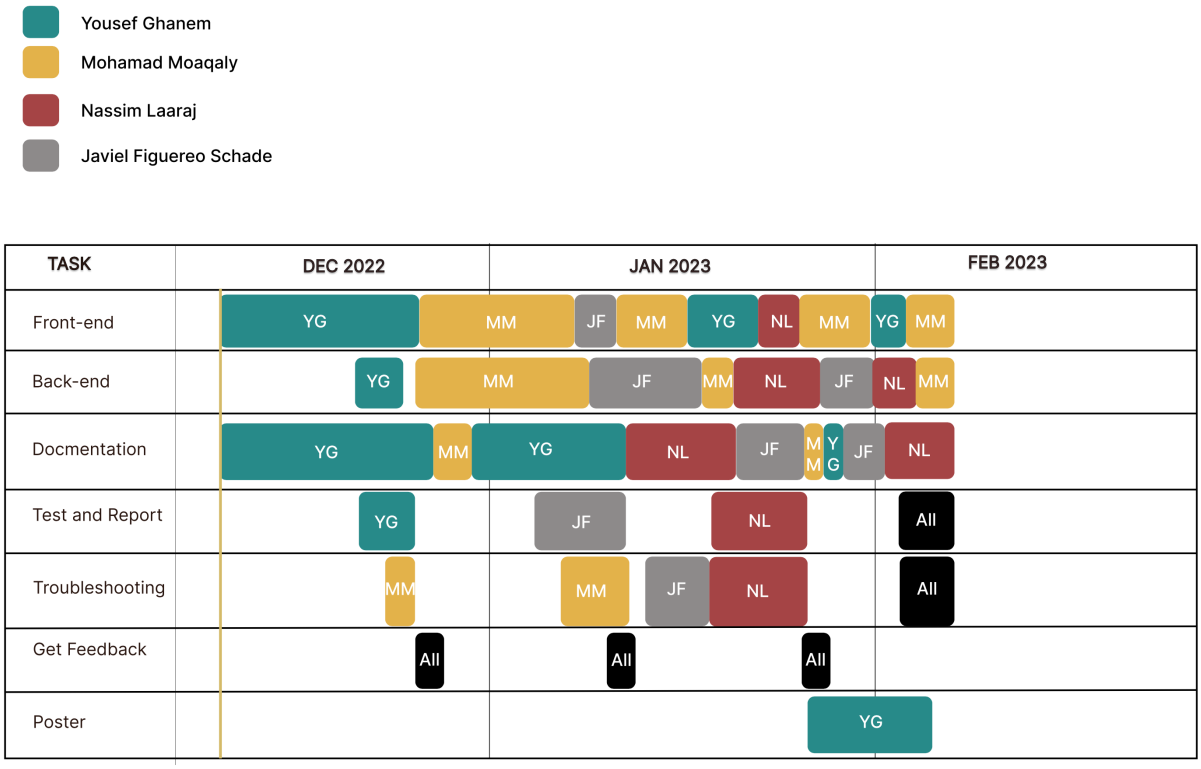

Figure 1: Task Distribution Tabel

2 Starting the project

2.1 Software Process Model:

For the Analysis of this study, we used the Agile Method.

- **Agile Method:**

The Agile Model delivers a rapidly functioning product which is used as a realistic development steppingstone. The product is divided into different cycles, which enables the before mentioned properties. The model constantly allows new versions. These have each time minimal, differential changes compared to their predecessor. These are executed recurrently test. Sometimes the difference is so small that it is negligible.

This model focuses on interaction as the developers, customer, and testers work in concert, with each other, throughout the project. However, because this model depends a lot on interaction with the customer, the project can head in the wrong direction if the customer doesn't know what direction they would like to go in.
(See: https://en.wikipedia.org/wiki/Agile_software_development)

2.2 Main development activities

Phases of the implementation of our agile concept:

1. Planning the goals for the object. Sequence diagram, activity diagram, requirements, class diagram, etc.
2. Predict potential risks associated with system implementation and use, e.g., system crash due too high number of registrations for the students, book registrations issues, incorrectly executed, exception error etc.
3. Engineering and testing: Programming of the system according to our project plan using the documents in this part, the product will be tested.
4. Evaluation and release: Each new iteration requires an oval period and the approval of the software. If the software needs improvement, the development team returns to the first point and readjusts the plan and risks.

(See: https://www.tutorialspoint.com/sdlc/sdlc_agile_model.htm)

2.3 Scrum Framework and Roles

The Scrum framework has three management functions: Product Owner, Scrum Master, and the Developer. The totality of these responses is called a Scrum team. A Scrum team enters contact with the stakeholders. Progress and interim results are transparent to all stakeholders. Stakeholders are allowed to listen in on most meetings. (Source: https://www.scrum.org/resources/what-is-scrum)

2.4 Scrum Steps

The Scrum steps can be divided into 5 small groups:

1. **Introduction**
   - Generate the project idea
   - Set up the Backlog
   - Forming Scrum Team
   - Picking Stakeholder and Manager
   - Execution of release planning
2. **Planning and estimation**
   This phase consists of tasks planning and scheduling estimating processes such as creating user stories, approving, estimating, and committing user stories, creating challenges, estimating commitments, and creating the dash backlog.
3. **Implement**
   This stage is about performing the tasks and activities to generate a project product. These activities consist of creating the multiple deliverables, holding daily stand-up sessions, and maintenance of the product backlog on a regular basis.
4. **Review and outlook**
   This step is about checking the results and the work performed and determining ways to enhance the practices and procedures used for the project's work.
5. **Release**
   This phase focuses on handing over Accepted Jobs to the customer and identifying, then documenting and assimilating any lessons learned during the exercise. (See: https://www.scrum.org/resources/what-is-scrum

2.5 Scrum product Backlog

In Scrum, the Product Backlog consists of a list of elements that need to be completed during the engineering of the product. Each backlog item contains a raw estimate to support the planning process. The Product Backlog lists changes and additions to be made to the product. The product owner prioritizes the most important items first. This means that these tasks are performed first.
(See: https://www.youtube.com/watch?v=8dGdIcyDk1w&ab_channel=edureka%21)

2.6 Scrum sprint Backlogs

The Sprint Backlog is a task list of key items from the Product Backlog to be addressed in the next step, and the tasks are allocated to individual members by the Development Team.
(See: https://www.youtube.com/watch?v=8dGdIcyDk1w&ab_channel=edureka%21)

2.7 Scrum Team-Meetings

The following steps are events by the sprint, it is one of their main jobs.

- Plan
  The Questions “what” and “who” are the main questions to clarify, in this step. It will be developed in the actual sprint and how the agreed Backlog will be realized.
- Review
  This is a meeting, at the end of the sprint, to check and authorize the done work.
- Looking back
  This is a fest meeting of the team, which is used for Enhancement. Also the team checks their last Sprint, which is the reason for Enhancement.

(See: https://www.youtube.com/watch?v=XU0llRltyFM&ab_channel=Axosoft)

3 UML Diagrams

This part describes the UML diagrams, which are a big substance of this study.

3.1 What are the so-called UML-diagrams?

UML stands for “Unified Modelling Language. It is a good tool to model the design, implementation, and the construction of difficult software systems. By writing programs, it is normal to have hundreds of lines in your program. It is too hard for the engineers to see all connections and relationships, because of that UML diagrams are a solution, which can represent the system visually.

3.2 Why should UML diagrams be deployed?

UML diagrams do not belong to a programming language and can therefore be used flexibly. Because of that software engineers can understand it. Thus they will be able to use it to their work.

3.3 What type of diagrams are the substance of this project?

The following 3 different types of UML diagrams are content of this project:

- Activity diagram.
- Class diagram.
- System architecture

(See: https://www.w3computing.com/systemsanalysis/importance-using-uml-modeling/)

4 Technical Description and Solutions

4.1 System Architect

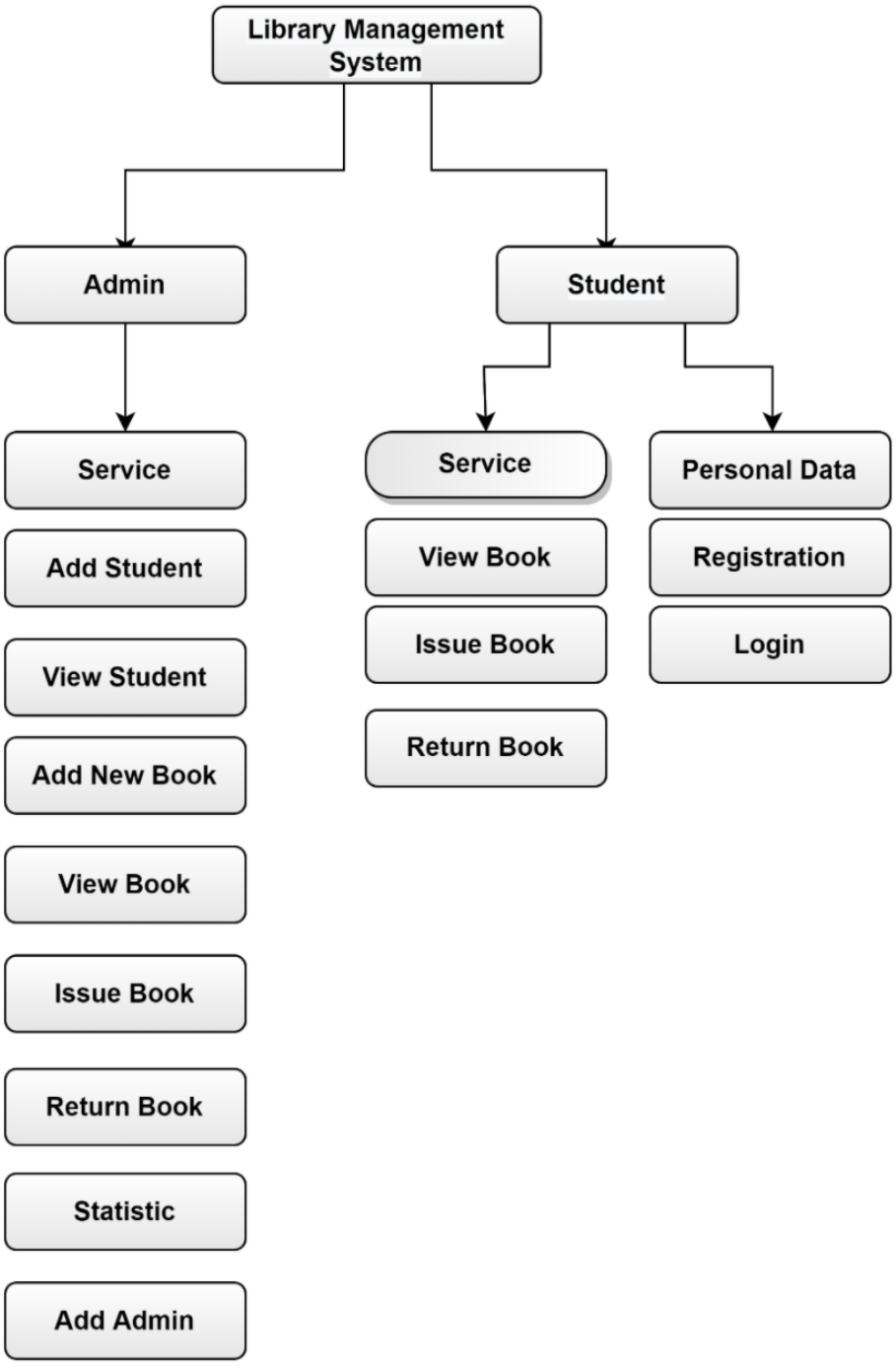

Figure 2: System Architekt

This diagram illustrates the overall structure of the “Library Management System”. It is designed to facilitate the borrowing and returning of books for students. The System is distributed in two actors, which have different specializations.

Starting off with the “User/Student” which itself is divided into “personal data” and “Service“. On the one Hand, in order to use this system, a student is required to be registered by an administrator, so he can use his information to login into the library system as a student. Moving on to the service for the student, which provides to view the book catalogue, borrow a book, and return the book.

Then there Are “Administrators” have a number of responsibilities within the system. In addition to registering and deleting students, they also have the ability to issue and return books. They also have access to all student information, which allows them to view the list of registered students and manage their borrowing and returning activities. Overall, the Library Management System aims to streamline the process of borrowing and returning books, making it easier for students to access the resources they need.

4.2 Activity Diagram

4.2.1 Login as Admin

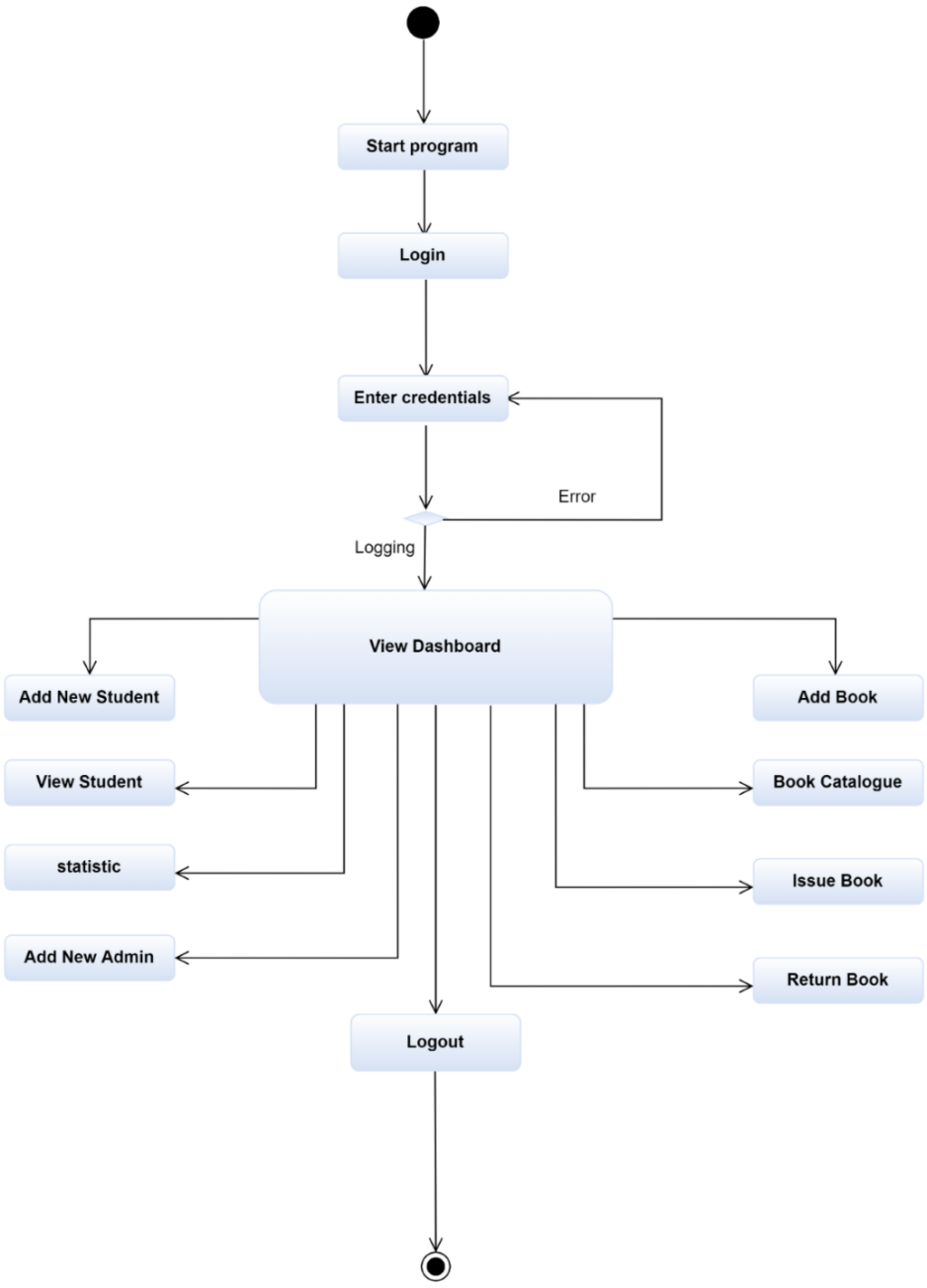

Figure 3: Activity Diagram

The activity diagram shows the flow of all functionalities which the “Library Management System” provides. In the beginning the admin starts the program and needs to enter his/her credentials to login into the system. After Login screen follows the dashboard where the admin has a list of services, which are divided into services for students and services for the library books. For the students the admin can register new students and is able to view the list of the registered students. Also searching and deleting are functions included in the view student screen, then the admin can go back to the dashboard screen. Books services are the main topic for our library, the admin can add new books by writing the required information such as ISBN, Title...etc. and also view the book catalogue. He/she is also able to issue and return the books for only registered students in the system, the last service is to get an overview of the issued book and returned books by going to the statistics, the admin can view 2 lists of the borrowed books and returned books. As the diagram shows, the admin does not need to go back to the dashboard screen after any new step, he can switch the windows while he works on any service without going back to the dashboard.

4.2.2 Login as Student

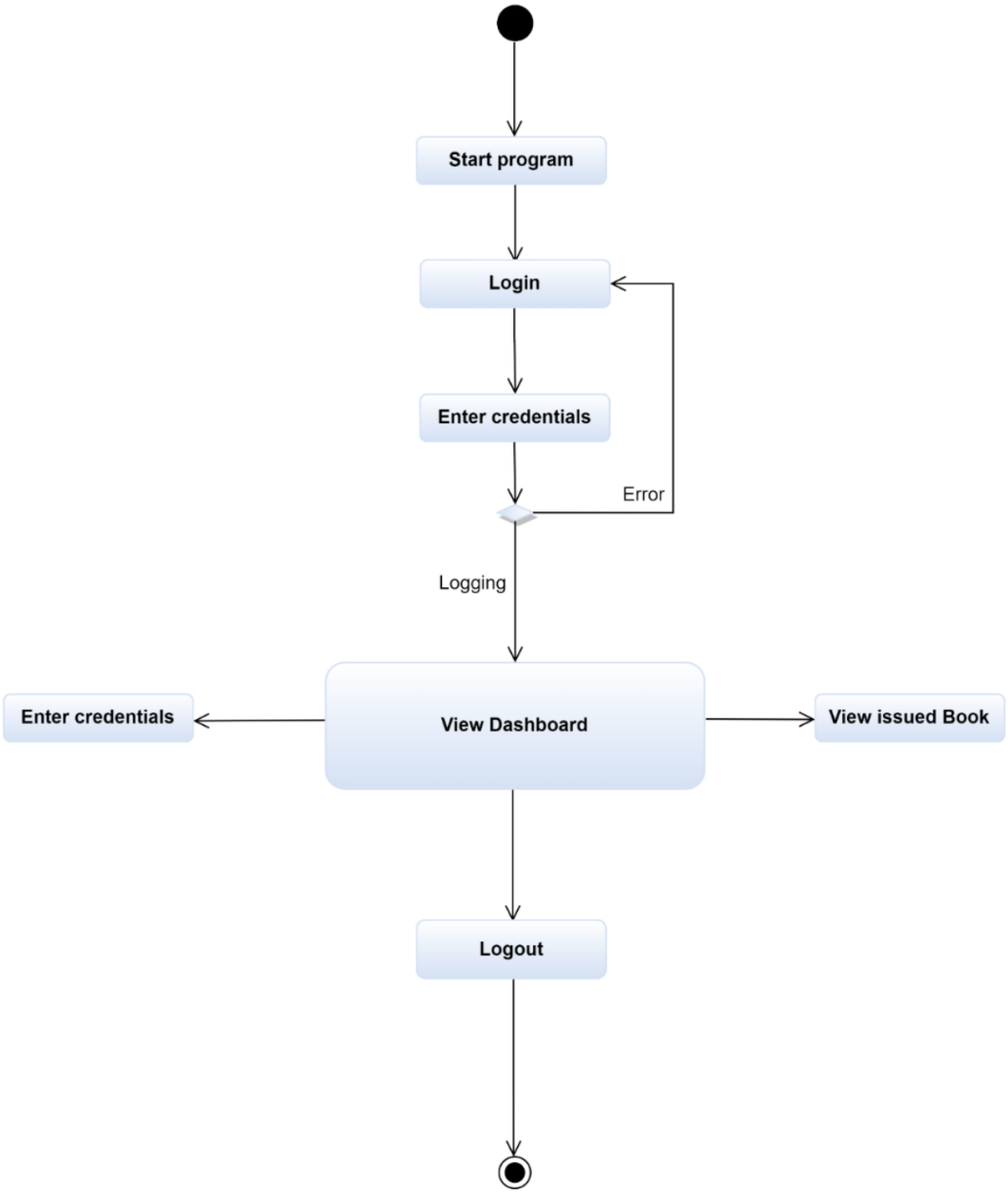

Figure 4: Activity Diagram

The activity diagram shows the flow of all functionalities which the “Library Management System” provides. In the beginning the student starts the program and needs to enter his/her credentials to login into the system. The student should be registered by an administrator otherwise he cant login.

After login the student will view the dashboard with two services which are books catalogue and view issued books. In the books catalogue the student is able to see the list of Books added by the administrators, so he can decide for borrowing books after searching for any book he needs if it is available or not.

View issued book/s provide a list of borrowed books also shows the important information such as the date and the status of the book in case it is already returned or not.

4.3 Class diagram

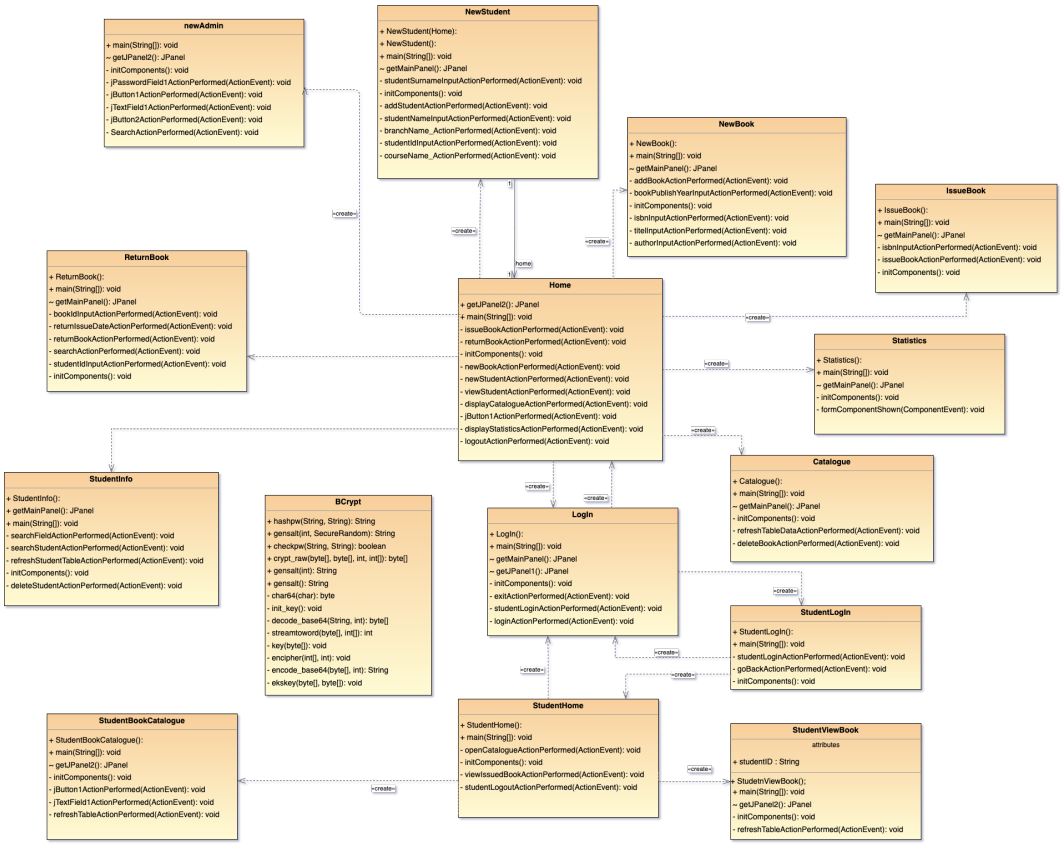

Figure 5: Class Diagram

4.3.1 Home Class

The Home class is a graphical user interface that displays the main functionality of the system to the admin. The Home class includes the following methods:

- `issueBookActionPerformed(ActionEvent e)` : Handles the event triggered by the user when they click the ”Issue Books” button.
- `returnBookActionPerformed(ActionEvent e)` : Handles the event triggered by the user when they click the ”Return Books” button.
- `newBookActionPerformed(ActionEvent e)` : Handles the event triggered by the user when they click the ”Add New Books” button.
- `newStudentActionPerformed(ActionEvent e)` : Handles the event triggered by the user when they click the ”Add New Student” button.
- `viewStudentActionPerformed(ActionEvent e)` : Handles the event triggered by the user when they click the ”View Students” button.

- displayCatalogueActionPerformed(ActionEvent e) : Handles the event triggered by the user when they click the ”Display Catalogue” button.
- displayStatisticsActionPerformed(ActionEvent e) : Handles the event triggered by the user when they click the ”Display Statistics” button.
- logoutActionPerformed(ActionEvent e) : Handles the event triggered by the user when they click the ”Logout” button.

The Home class provides a graphical interface for the admin to access and manage the various functionalities of the system, such as registering new students, viewing existing students, adding new books, viewing the book catalogue, issuing books, returning books, displaying statistics, and logging out of the system.

4.3.2 Login Class

The Login class is the first screen that users see when they access the system. It provides a graphical interface for the user to log in to the system. The Login class includes the following methods:

- exitActionPerformed(ActionEvent e) : Handles the event triggered by the user when they click the ”Exit” button.
- studentLoginActionPerformed(ActionEvent e) : Handles the event triggered by the user when they click the ”Student Login” button.
- loginActionPerformed(ActionEvent e) : Handles the event triggered by the user when they click the ”Login” button.

The Login class provides the user with the ability to log in to the system as either an admin or a student. When the user logs in successfully, they are taken to the appropriate screen for their user type (i.e., the Home screen for the admin or the Student screen for the student).

4.3.3 StudentLogIn Class

The StudentLogIn class provides a graphical interface for students to log into the system. It includes the following methods:

- studentLoginActionPerformed(ActionEvent e) : Handles the event triggered by the user when they click the ”Student Login” button.

- goBackActionPerformed(ActionEvent e) : Handles the event triggered by the user when they click the ”Go Back” button.

The StudentLogIn class provides a way for students to log into the system, so they can access the functionalities that are available to them. The class implements the necessary methods to handle the events triggered by the user and to initialize the components of the screen.

4.3.4 NewStudent Class

The NewStudent class is a graphical user interface that allows the admin to register a new student in the system. The NewStudent class includes the following methods:

- studentSurnameInputActionPerformed(ActionEvent e) : Handles the event triggered by the user when they enter input into the ”Student Surname” text field.
- addStudentActionPerformed(ActionEvent e) : Handles the event triggered by the user when they click the ”Add Student” button.
- studentNameInputActionPerformed(ActionEvent e) : Handles the event triggered by the user when they enter input into the ”Student Name” text field.
- branchName_ActionPerformed(ActionEvent e) : Handles the event triggered by the user when they select an option from the ”Branch Name” drop-down list.
- studentIdInputActionPerformed(ActionEvent e) : Handles the event triggered by the user when they enter input into the ”Student ID” text field.
- courseName_ActionPerformed(ActionEvent e) : Handles the event triggered by the user when they select an option from the ”Course Name” drop-down list.

The NewStudent class provides a graphical interface for the admin to register a new student in the system by entering the student’s name, surname, ID, branch, and course.

4.3.5 StudentInfo Class

The StudentInfo class provides a graphical interface for displaying information about students registered in the system. The class includes the following methods:

- searchFieldActionPerformed(ActionEvent) : Handles the event triggered by the user when they enter a value in the search field.code

- searchStudentActionPerformed(ActionEvent) : Handles the event triggered by the user when they click the ”Search” button.
- refreshStudentTableActionPerformed(ActionEvent) : Handles the event triggered by the user when they click the ”Refresh” button.
- deleteStudentActionPerformed(ActionEvent) : Handles the event triggered by the user when they click the ”Delete” button.

4.3.6 NewBook Class

The NewBook class provides a graphical interface for the admin to add new books to the system, by filling in the details of the book such as ISBN, title, author, and publish year. The class includes the following methods:

- addBookActionPerformed(ActionEvent): void Handles the event triggered by the user when they click the ”Add Book” button.
- bookPublishYearInputActionPerformed(ActionEvent): void Handles the event triggered by the user when they make changes to the ”Publish Year” field.
- isbnInputActionPerformed(ActionEvent): void Handles the event triggered by the user when they make changes to the ”ISBN” field.
- titelInputActionPerformed(ActionEvent): void Handles the event triggered by the user when they make changes to the ”Title” field.
- authorInputActionPerformed(ActionEvent): void Handles the event triggered by the user when they make changes to the ”Author” field.

4.3.7 Catalogue Class

The Catalogue class is a graphical user interface that displays the current book catalogue of the system to the admin. The class includes the following methods:

- refreshTableDataActionPerformed(ActionEvent): void Handles the event triggered by the user when they click the ”Refresh Table Data” button.
- deleteBookActionPerformed(ActionEvent): void Handles the event triggered by the user when they click the ”Delete Book” button.

The Catalogue class provides a graphical interface for the admin to view and manage the current book catalogue of the system, such as refreshing the data displayed in the table and deleting books from the catalogue.

4.3.8 StudentBookCatalogue

The StudentBookCatalogue class is a graphical user interface that displays the books available in the library's catalogue to the students. The class includes the following method:

- $\mathrm{refreshTableActionPerformed(ActionEvent)}$: Handles the event triggered by the user when they click the "Refresh Table" button.

4.3.9 IssueBook Class

The IssueBook class is a graphical user interface that allows the admin to issue books to students. The IssueBook class includes the following methods:

- $\mathrm{isbnInputActionPerformed(ActionEvent)}$: Handles the event triggered by the user when they enter the ISBN of the book they want to issue.
- $\mathrm{issueBookActionPerformed(ActionEvent)}$: Handles the event triggered by the user when they click the "Issue Book" button.

4.3.10 ReturnBook

The ReturnBook class is used to return a previously borrowed book. The user inputs the book's ID and the student's ID, and the system searches for the corresponding book and student records. If the records are found, the user can return the book by providing the return date. The ReturnBook class includes the following methods:

- $\mathrm{bookIdInputActionPerformed(ActionEvent)}$: Handles the action performed when the user inputs the book ID.
- $\mathrm{returnIssueDateActionPerformed(ActionEvent)}$: Handles the action performed when the user inputs the return date.
- $\mathrm{returnBookActionPerformed(ActionEvent)}$: Handles the action performed when the user initiates the book return process.
- $\mathrm{searchActionPerformed(ActionEvent)}$: Handles the action performed when the user searches for a book and student record.
- $\mathrm{studentIdInputActionPerformed(ActionEvent)}$: Handles the action performed when the user inputs the student ID.

4.3.11 Statistics Class

The Statistics class is a graphical user interface that provides an overview for the admin the number of books issued and books returned

4.3.12 NewAdmin Class

The NewAdmin implements the GUI for the new administrator account creation feature. It provides the user interface for entering the required information to create a new administrator account, such as username and password.

4.3.13 BCrypt

The BCrypt class provides methods for encoding a password and verifying a password encoded with the BCrypt algorithm.

- $\mathrm{hashpw(String,\ String)}$: Returns the hashed password in the form of a string.
- $\mathrm{gensalt(int,\ SecureRandom)}$: Generates a salt for the hash function. The salt is used to make the hash more secure.
- $\mathrm{checkpw(String,\ String)}$: Verifies that the plain text password matches the hashed password.
- $\mathrm{crypt\_raw(byte[],\ byte[],\ int,\ int[])}$: Encodes the password using the BCrypt algorithm.
- $\mathrm{gensalt(int)}$: Generates a salt for the hash function. The salt is used to make the hash more secure.
- $\mathrm{gensalt()}$: Generates a salt for the hash function. The salt is used to make the hash more secure.
- $\mathrm{char64(char)}$: Converts a character to a byte representation.
- $\mathrm{init\_key()}$: Initializes the key used by the BCrypt algorithm.
- $\mathrm{decode\_base64(String,\ int)}$: Decodes a base64 encoded string.
- $\mathrm{streamtoword(byte[],\ int[])}$: Converts a stream of bytes to an array of integers.
- $\mathrm{key(byte[])}$: Initializes the key used by the BCrypt algorithm.
- $\mathrm{encipher(int[],\ int)}$: Enciphers the password using the BCrypt algorithm.
- $\mathrm{encode\_base64(byte[],\ int)}$: Encodes a string as base64.

- $\text{ekskey}(\text{byte}[], \text{byte}[])$: Expands the key used by the BCrypt algorithm.

4.4 Connection to a database

4.4.1 Why is there a need of a database?

In order to effectively manage all the accounts of librarians and students, keep track of all the books, and perform tasks such as issuing books and tracking which student has borrowed which book and when, it is necessary to store all the relevant information in a database. The database is a crucial component of the Library Management System and enables the efficient management of all the data required to run the library. With a properly configured database, the library can keep track of all the books, librarian and student accounts, and provide the necessary functions and tasks, such as issuing books, to ensure a well-organized and smooth-running library system. To connect the Library Managemenet System with a MYSQL we used JDBC

4.4.2 Database connection with JDBC

Java Database Connectivity (JDBC) is an application programming interface (API) in Java that provides a standard approach for accessing databases. This API enables the retrieval and storage of data in and from a database through the establishment of a connection, the execution of SQL statements, and the processing of the results obtained.[1]

The JDBC library incorporates APIs for various tasks commonly associated with database utilization, including:

- making a connection to a database
- creating SQL or MySQL statements
- executing SQL or MySQL queries in the database
- viewing and modifying the resulting records[2]

JDBC utilizes a driver manager and database-specific drivers to offer transparent connectivity to heterogeneous databases. The JDBC driver manager guarantees that the appropriate driver is employed to access each data source. This driver manager is

[1] Vgl. tutorialspoint contributors, 2021a
[2] Vgl. tutorialspoint contributors, 2021a

capable of accommodating multiple concurrent drivers connected to multiple heterogeneous databases.[3]

Figure 6: JDBC Architecture

The JDBC API includes the following interfaces and classes:

DriverManager: This class is responsible for managing a list of database drivers and matching connection requests from the Java application with the appropriate database driver through the use of communication subprotocols. The first driver that recognizes a certain subprotocol under JDBC will be utilized to establish a database connection.[4]

Driver: This interface manages the communications with the database server. Direct interaction with Driver objects is rare, as they are mostly managed by DriverManager objects, which abstract the details associated with working with Driver objects.[5]

Connection: This interface provides all methods required for contacting a database.

[3] Vgl. tutorialspoint contributors, 2021a
[4] Vgl. tutorialspoint contributors, 2021a
[5] Vgl. tutorialspoint contributors, 2021a

The connection object symbolizes the communication context, meaning that all communication with the database is conducted through this object.[6]

Statement: Objects created from this interface are used to submit SQL statements to the database. Some derived interfaces also accept parameters in addition to executing stored procedures.[7]

ResultSet: These objects store data retrieved from a database after the execution of an SQL query using Statement objects. They serve as an iterator, allowing movement through the data.[8]

SQLException: This class handles any errors that occur in a database application. It is important to note that the architectural diagram of JDBC includes the driver manager in relation to JDBC drivers and the Java application.[9]

4.4.3 How to use JDBC for Queries

1.**Load the JDBC Drive**: The first step is to load the JDBC driver class. This is usually done using the Class.forName() method, which dynamically loads the class and creates an instance of it.[10] For example:

```
try {
  Class.forName("com.mysql.jdbc.Driver");
} catch (ClassNotFoundException e) {
  e.printStackTrace();
}
```

2.**Establish a Connection**: Once the JDBC driver is loaded, a connection to the database using the DriverManager.getConnection() method can be established. This method takes the URL of the database, a username and a password as arguments.[11] For example:

```
String url = "jdbc:mysql://localhost:3306/database_name";
String user = "username";
String password = "password";

try (Connection connection = DriverManager.getConnection(url, user,
    password)) {
```

[6] Vgl. tutorialspoint contributors, 2021a
[7] Vgl. tutorialspoint contributors, 2021a
[8] Vgl. tutorialspoint contributors, 2021a
[9] Vgl. tutorialspoint contributors, 2021a
[10] Vgl. tutorialspoint contributors, 2021b
[11] Vgl. tutorialspoint contributors, 2021b

```
12    // do something with the connection
13  } catch (SQLException e) {
14    e.printStackTrace();
15  }
```

3.**Execute SQL Statements**: With the connection established, you can execute SQL statements using a Statement or a PreparedStatement object. The Statement object is used for executing simple SQL statements, while the PreparedStatement object is used for executing parameterized SQL statements.[12]

4.**Process the Results**: Finally, you can process the results returned by the database using the ResultSet object. The ResultSet object provides methods for accessing the data, such as ResultSet.getInt(), ResultSet.getString(), and so on.[13]

[12]Vgl. tutorialspoint contributors, 2021c
[13]Vgl. tutorialspoint contributors, 2021d

Example for Execute a SQL Statement and Processing the Results:

```
18
19
20 try (Connection connection =     DriverManager.getConnection(url, user
       , password);
21 String query = "SELECT title, author,publicationYear FROM book WHERE
       isbn = xxxx";
22   PreparedStatement preparedStatement = connection.prepareStatement(
       query)) {
23
24   ResultSet resultSet = preparedStatement.executeQuery();
25
26   If (resultSet.next()) {
27
28     String title = resultSet.getString("title");
29     String author = resultSet.getString("author");
30     int publicationYear = resultSet.getInt("publicationYear");
31
32     System.out.println("Title: " + title);
33     System.out.println("Author: " + author );
34     System.out.println("Puböication Year: " + publicationYear);
35   }
36 } catch (SQLException e) {
37   e.printStackTrace();
38 }
```

5.**Close the Connection**: Once you're done working with the database, make sure to close the connection using the Connection.close() method. This is usually done using a try-with-resources block, which automatically closes the connection at the end of the block.[14]

```
41 try (Connection connection = DriverManager.getConnection(url, user,
       password)) {
42     // do something with the connection
43     connection.close();
44 } catch (SQLException e) {
45   e.printStackTrace();
46 }
```

[14]Vgl. tutorialspoint contributors, 2021b

4.5 Database Relation Schema

4.5.1 Database Relation Schema Definition

A database schema is a comprehensive and detailed representation of a database's structure, data, and relationships. It serves as a blueprint for a database, providing a comprehensive view of how data is organized, stored, and accessed. The schema defines the structure of the database and provides a clear understanding of how data is related to each other.[15]

A database schema includes a detailed description of the tables and columns in the database, the relationships between tables, and any constraints or rules that apply to the data stored in the database. It also includes information about data types, default values, and any indexes or keys that are used to optimize data retrieval.[16]

4.5.2 Library Management System Database Relation Schema

The MySQL database schema of the Library Management System represents the data stored in the database and the relationships between the different data entities. The database relation diagram consists of five tables:

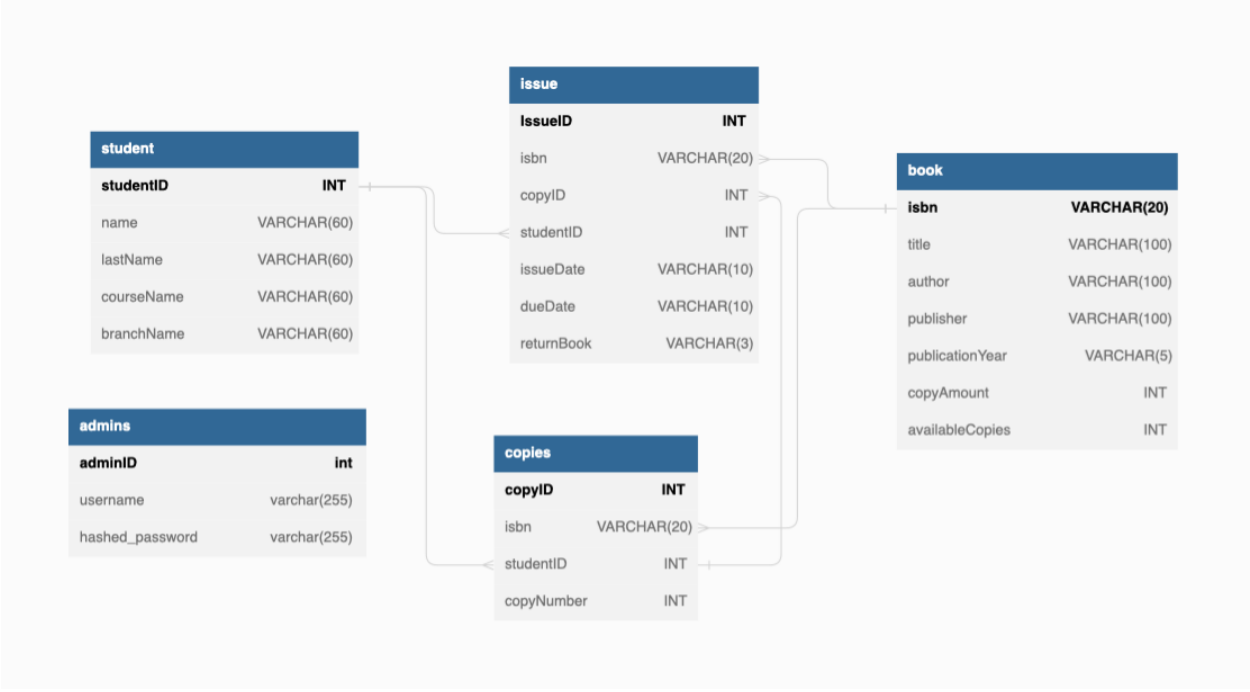

Figure 7: MySQL Database Relation Schema

1.**Admins table**: This table stores information about the administrators who manage

[15]Vgl. Christina Kopecky, 2020
[16]Vgl. Christina Kopecky, 2020

the library. The information stored in this table is the name and ID number of the administrators. The primary key for this table is the administrator's ID number.

2.**Student table**: This table stores information about the students who use the library. The information stored in this table could include the student's ID number, the student's full name, also the course and branch name. The primary key for this table is the student's ID number.

3.**Book table**: This table stores information about the books in the library's collection. The information stored in this table could include the book's title, author, ISBN, publisher, publication date, the total amount of the copies and current amount of available copies. The primary key for this table is the book's ISBN.

4.**Copies table**: This table stores information about the number of copies of each book in the library's collection. The information stored in this table could include the book's ISBN, the student's ID and the copy number. The primary key for this table is the copy's ID.

5.**Issue table**: This table stores information about the books that have been issued to students. The information stored in this table include the student's ID number, the book's ISBN, the date the book was issued, the due date and the status if the book has been returned or not. The primary key for this table is the issue‘s ID.

The relationships between the tables are as follows:

1. The relationship between the student and issue tables is a one-to-many relationship, where one student can have multiple books issued to them, but each book can only be issued to one student. This relationship is modeled by including a foreign key in the issue and copies table that references the primary key in the student table.

2. The relationship between the book and copies tables is also a one-to-many relationship, where one book can have multiple copies in the library's collection, but each copy of the book is associated with only one book. This relationship is modeled by including a foreign key in the copies table that references the primary key in the book table.

3. The relationship between the copies and issue tables, is a one-to-many relationship, where multiple copies can be issued, but a specific copy can be issued only once.

4. The relationship between the book and issue tables, is a many-to-many relationship as many as multiple copies of the same book with the same ISBN can be issued multiple times.

This detailed MySQL database relation schema provides a comprehensive representation of the data stored in the Library Management System database, the relationships between the different data entities, and the constraints that ensure the data in the database is valid and consistent. It is a valuable tool for designing and modeling the database, and for understanding the database structure and design.

5 Scenarios

5.1 Admin Scenarios

This section describes the various scenarios that an admin can perform using the library management system.

5.1.1 Admin Login

The first step in accessing the system is to log in as an admin. In this scenario, the admin is required to enter their username and password. If the credentials are correct, the system grants access to the admin and directs them to the admin homepage.

Figure 8: Admin Login Window

Figure 9: Login with incorrect credentials

5.1.2 Admin Homepage

The admin homepage displays an overview of all the features available to the admin. From here, the admin can access the different functionalities such as student registration, view students, add new books, view book catalogue, issue books, return books, view statistics, and create a new admin account.

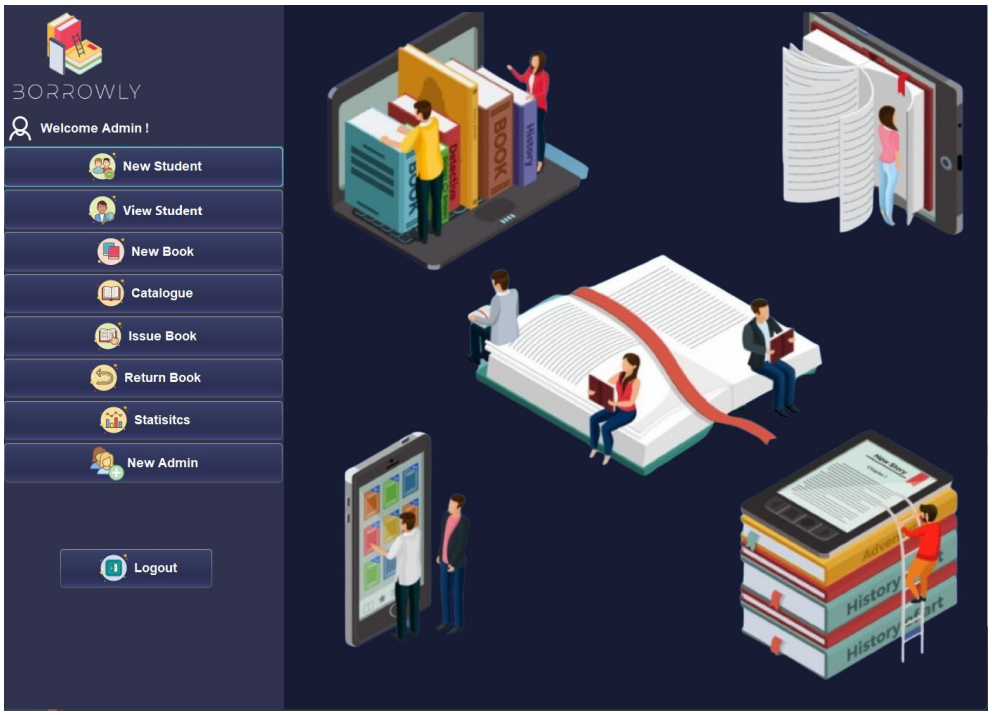

Figure 10: Admin Homepage Window

5.1.3 Student Registration

In this scenario, the admin can add new students to the system by providing their personal details and assigning them a unique identifier (such as a student ID). The information is stored in the database and can be retrieved later by the admin or other authorized users.

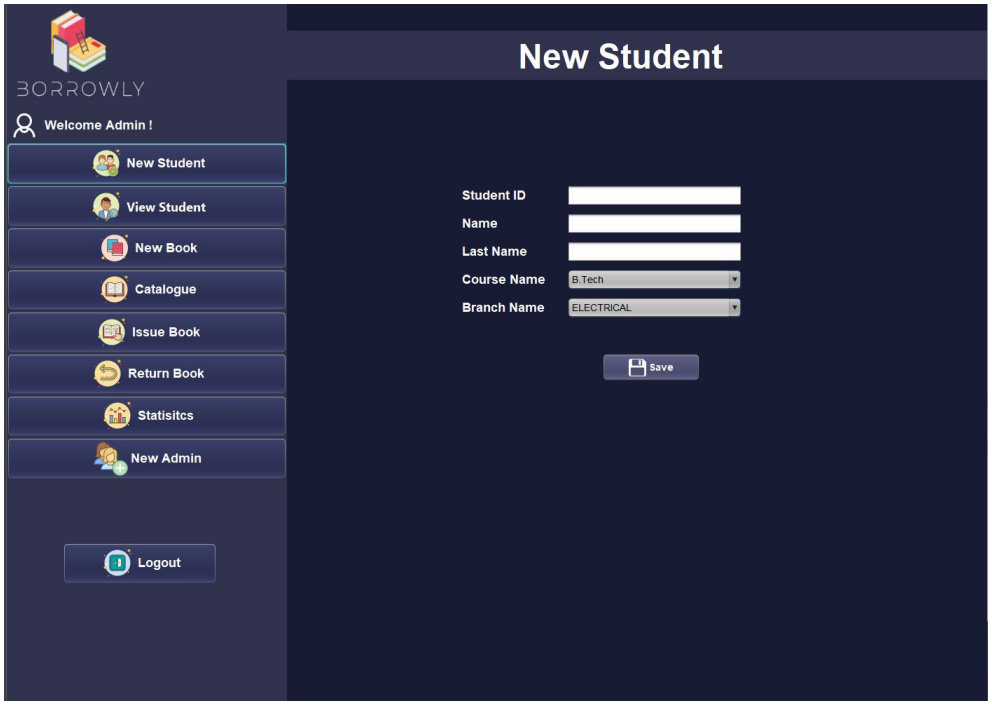

Figure 11: Adding A New Student

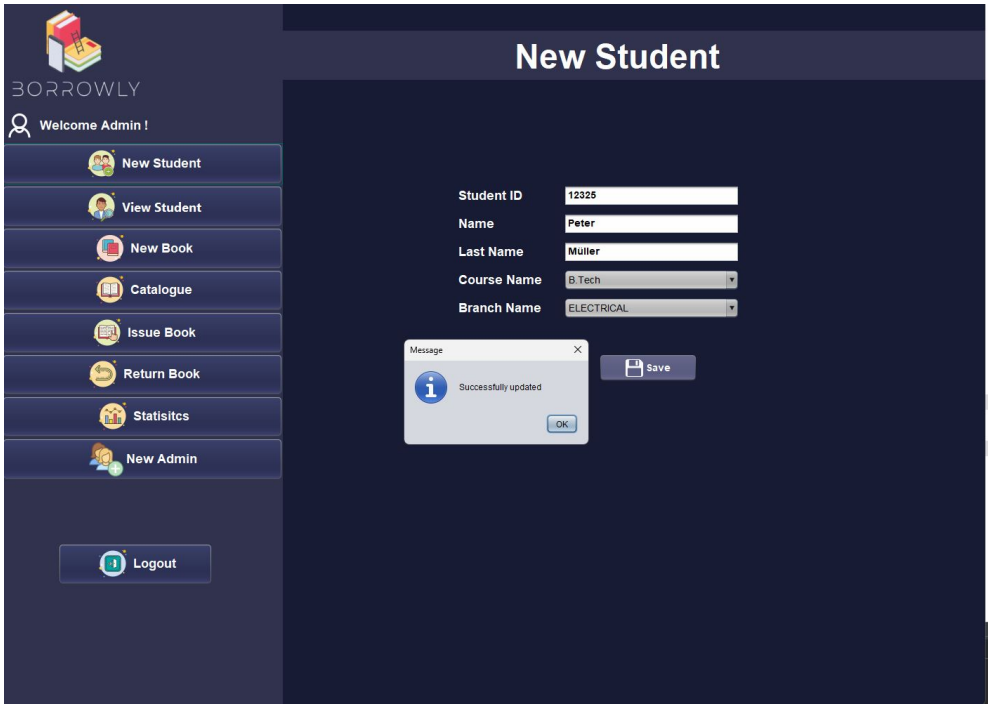

Figure 12: New Student Successfully Added

5.1.4 View Student

The admin can view the details of a student by searching for their record using their student ID. This functionality allows the admin to update or retrieve information about a student.

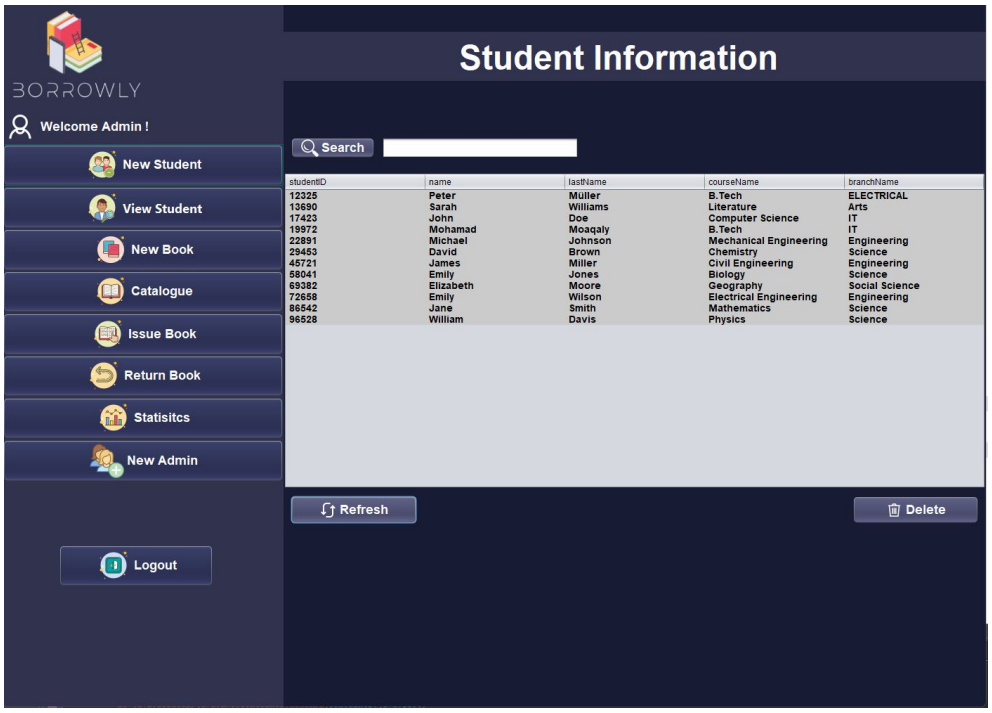

Figure 13: Overview Of All Student In The System

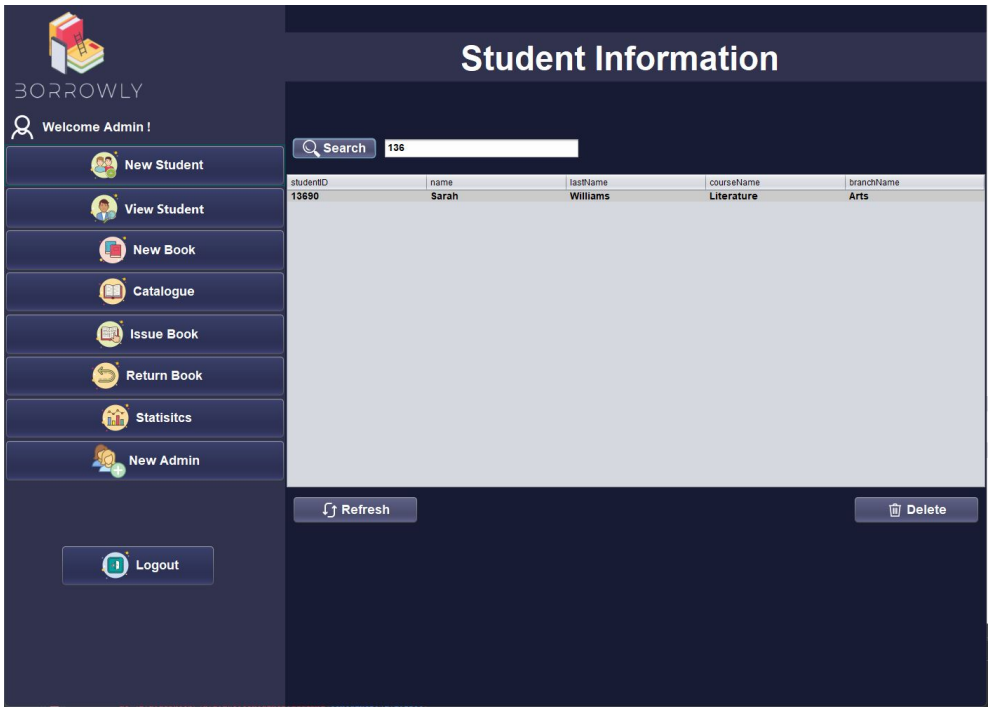

Figure 14: Searching For A Student With Their ID

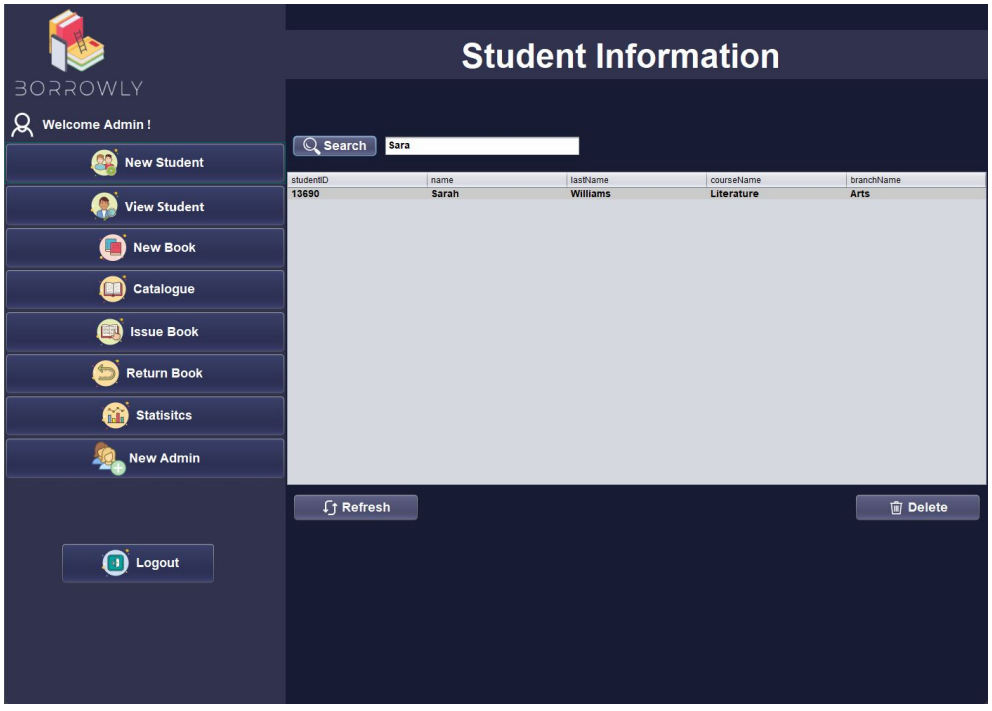

Figure 15: Searching For A Student With Their Name

5.1.5 Add New Books

The admin can add new books to the library by providing information such as the title, author, ISBN, publication date, and other relevant details.Every Book can have its own copies amount. The new books and its copies are stored in the database and become available for lending to students.

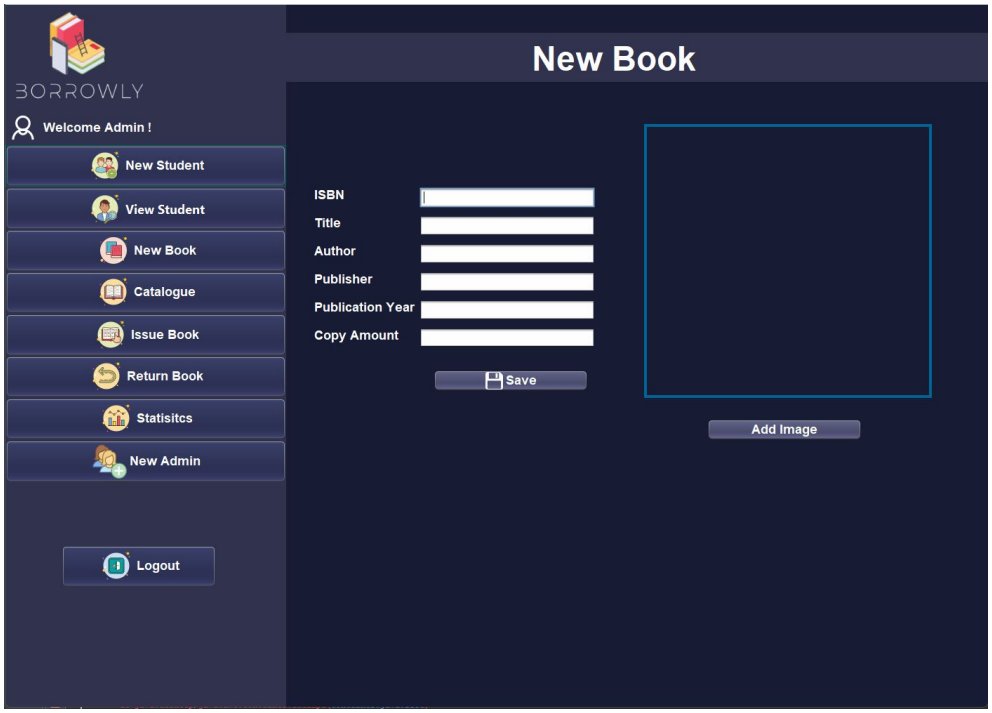

Figure 16: Login with incorrect credentials

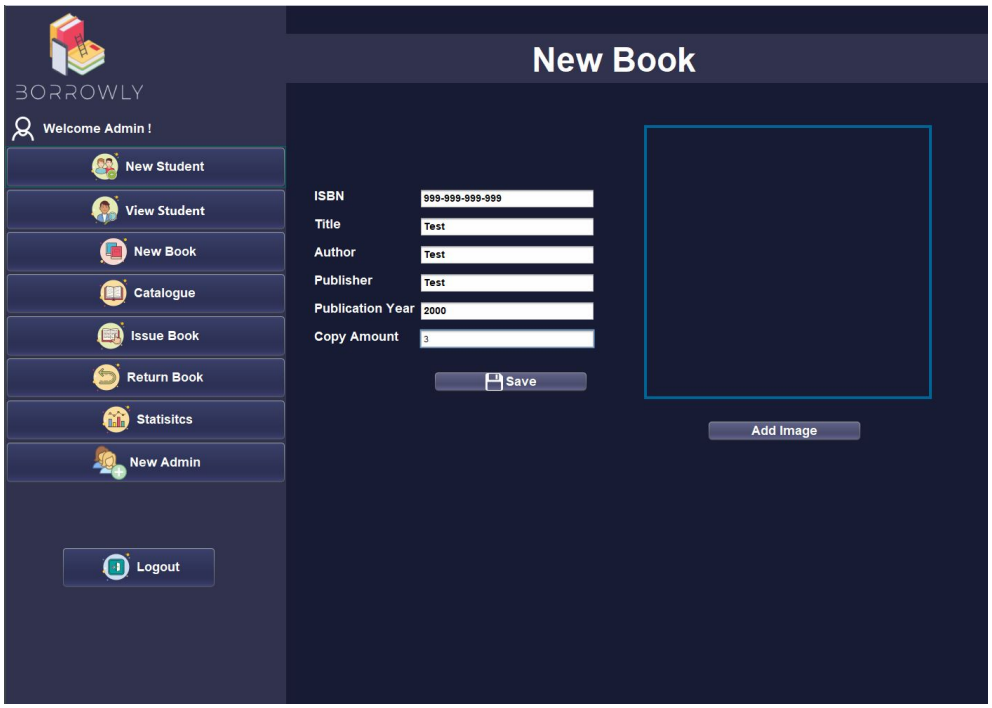

Figure 17: Providing All Information of The Book

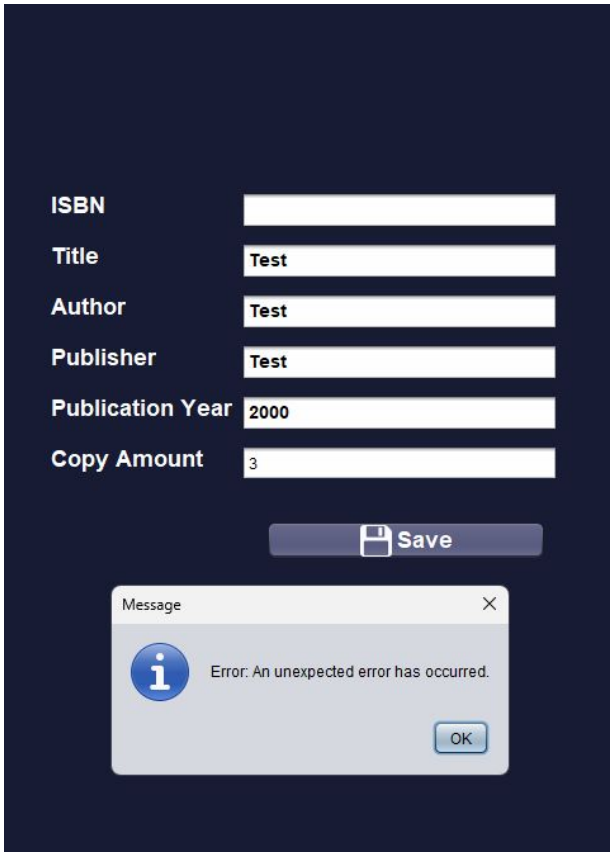

Figure 18: Error When Missing Information of The Book

5.1.6 Book Catalogue

The book catalogue provides a list of all the books available in the library. The admin can view the details of each book and also search for books based on specific criteria.

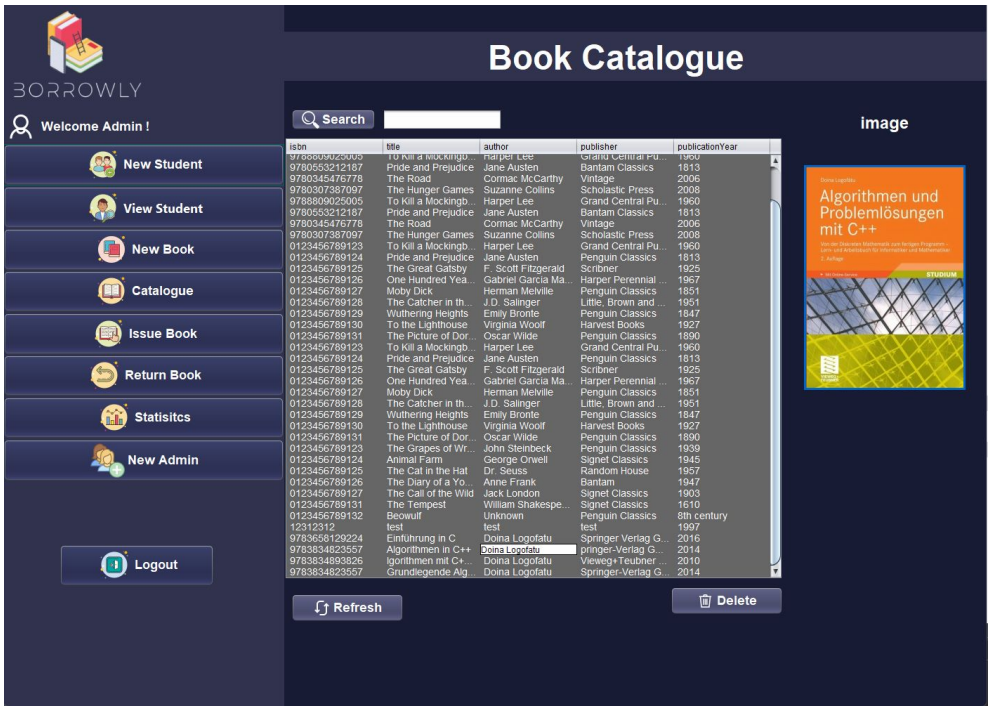

Figure 19: The Book Catalogue

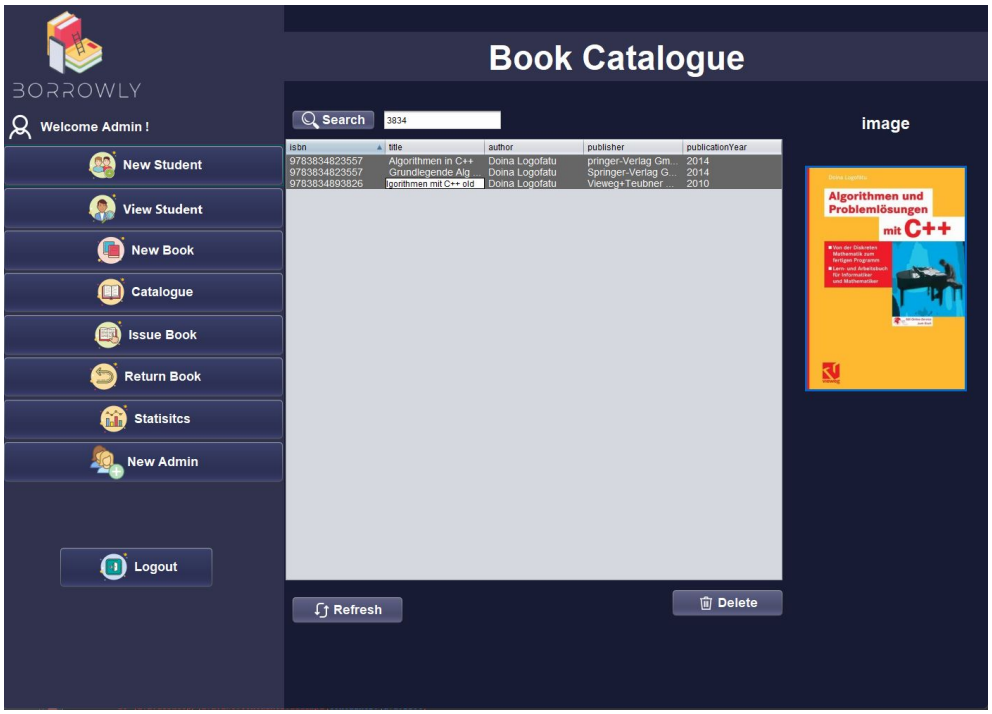

Figure 20: Search For Books

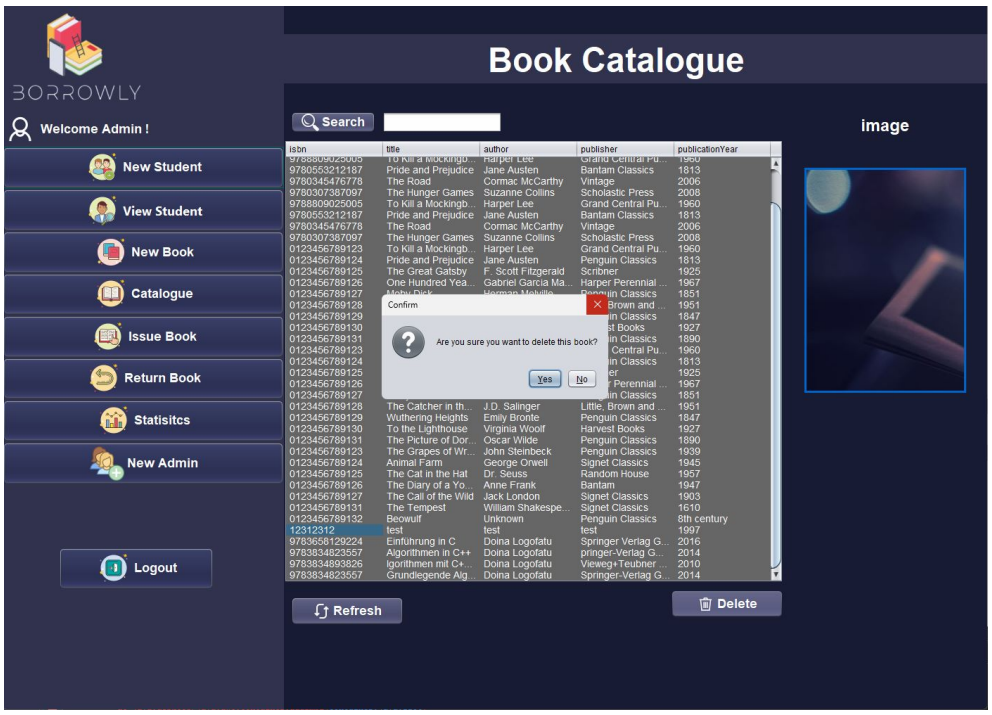

Figure 21: Delete A Book From The Catalogue

5.1.7 Issue Books

In this scenario, the admin can issue books to students. The admin can search for a student's record using their student ID and then select the books that the student wants to borrow. The system assigns to the student a specific copy of the book updates the availability of the books and keeps track of the due date for returning the books.

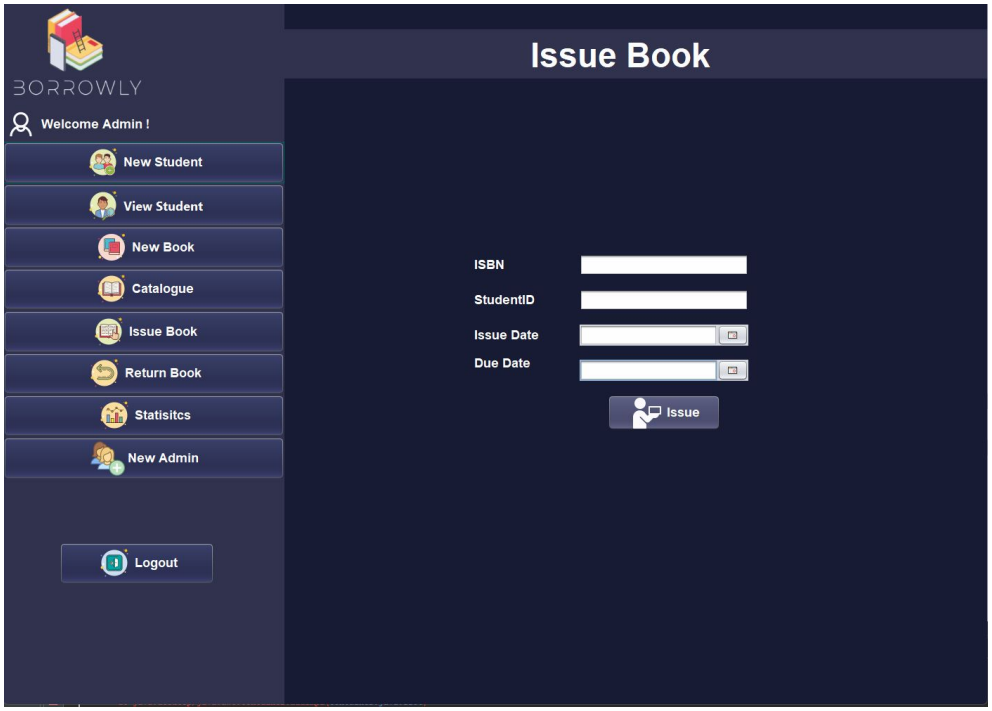

Figure 22: Issue Book Woindow

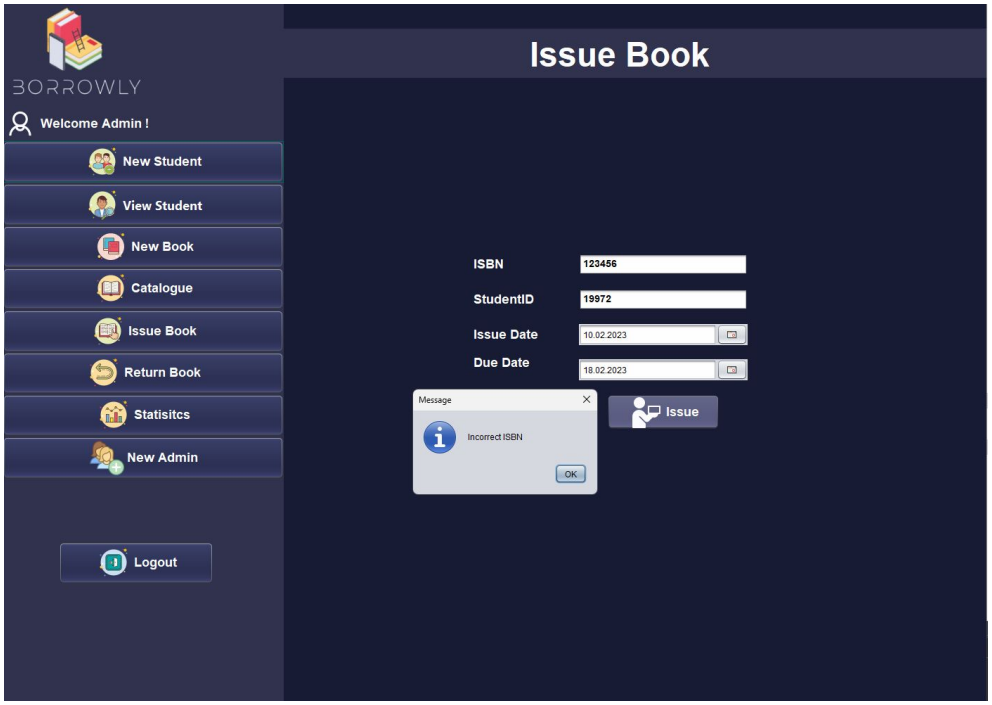

Figure 23: Incorrect Book Information

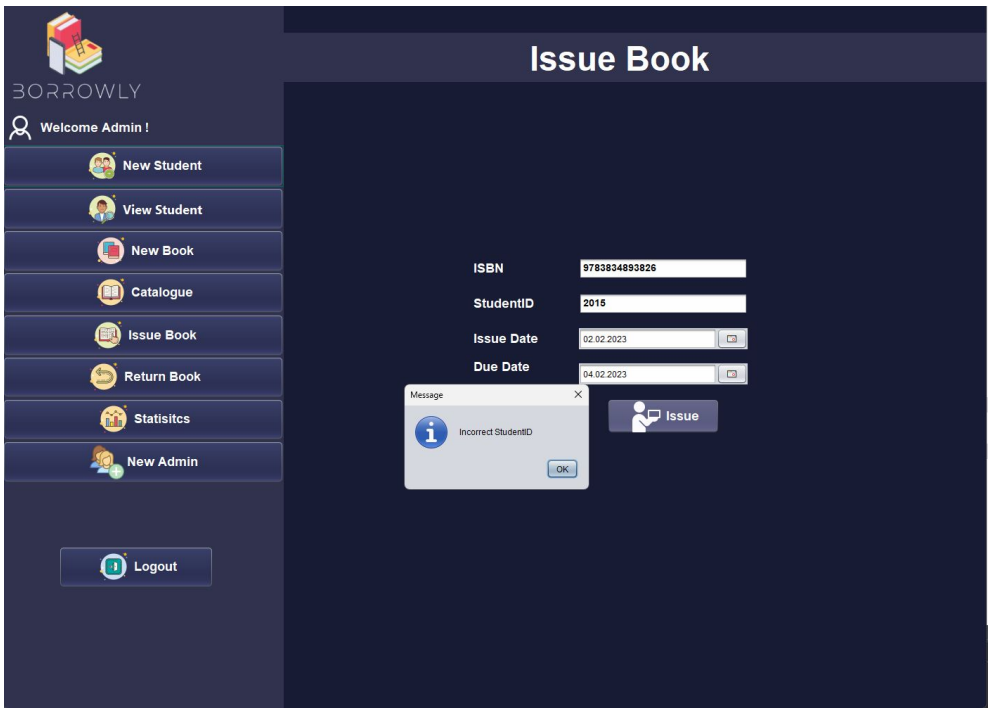

Figure 24: Incorrect Student Information

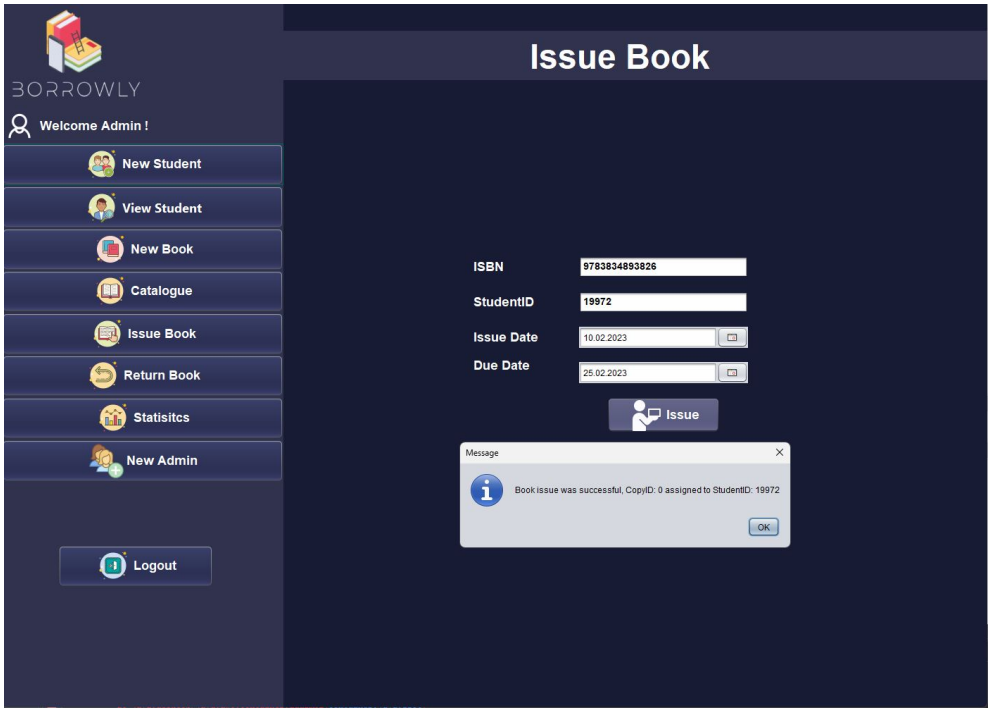

Figure 25: Successfully Borrowed The Book To Student

5.1.8 Return Books

The admin can process the return of books by searching for a student's record using their student ID. The system updates the availability of the books. On the background of the the specific copy which was assigned to the student is processed.

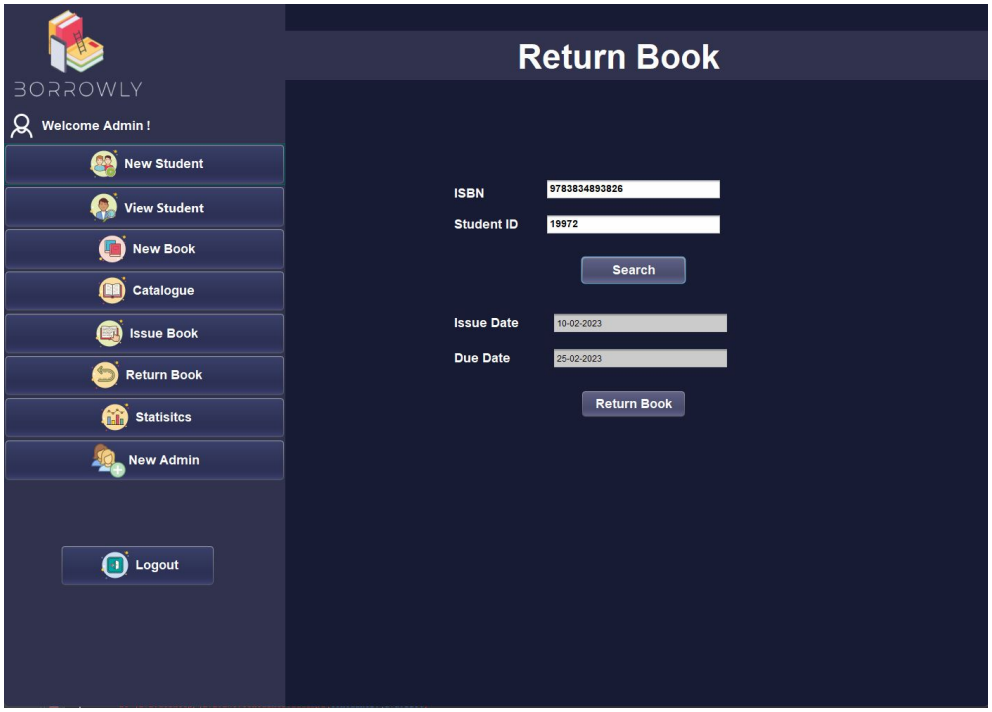

Figure 26: Processing The Return Of A Book

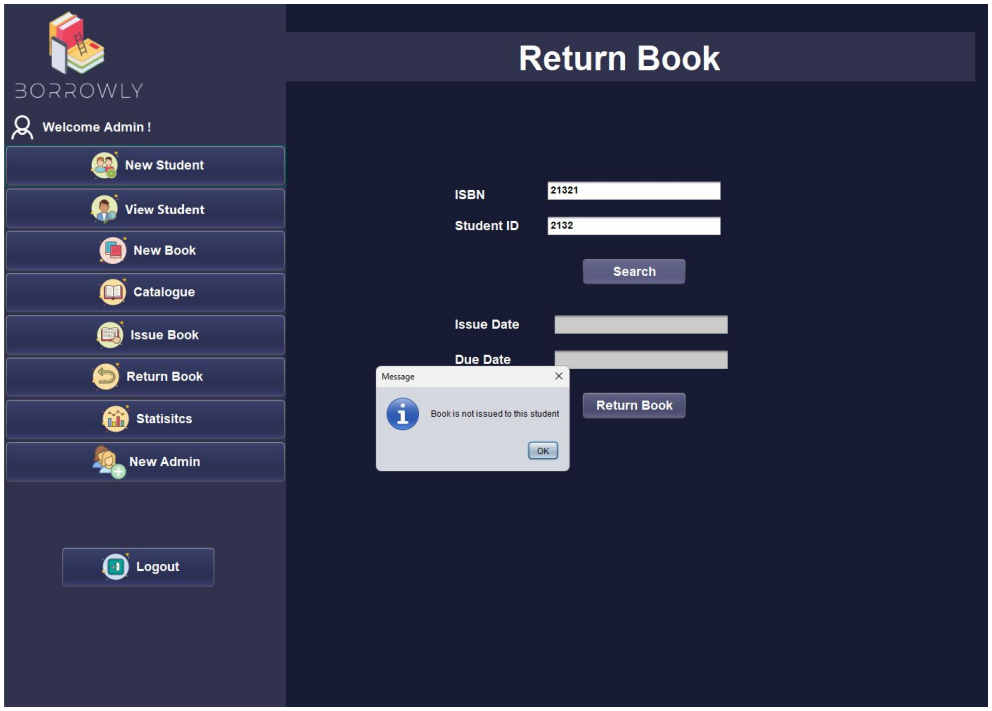

Figure 27: Error When Book Is Not Issued

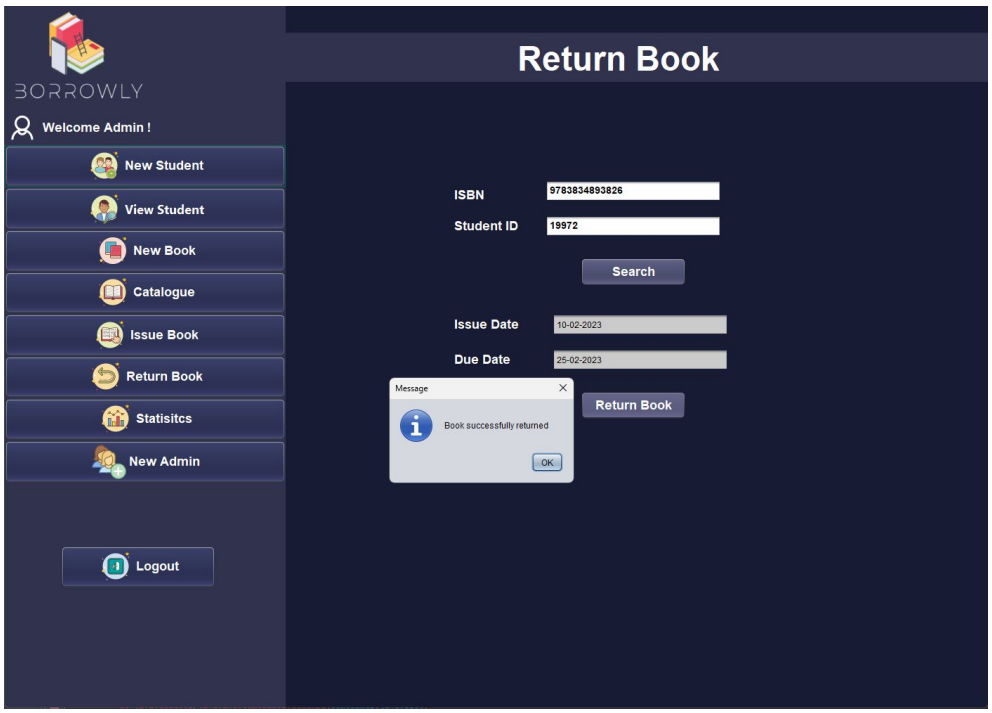

Figure 28: Successfully Returned The Book

5.1.9 Statistics

The statistics section provides an overview of the number of books issued and books returned. This information can help the admin make informed decisions about the functioning of the library.

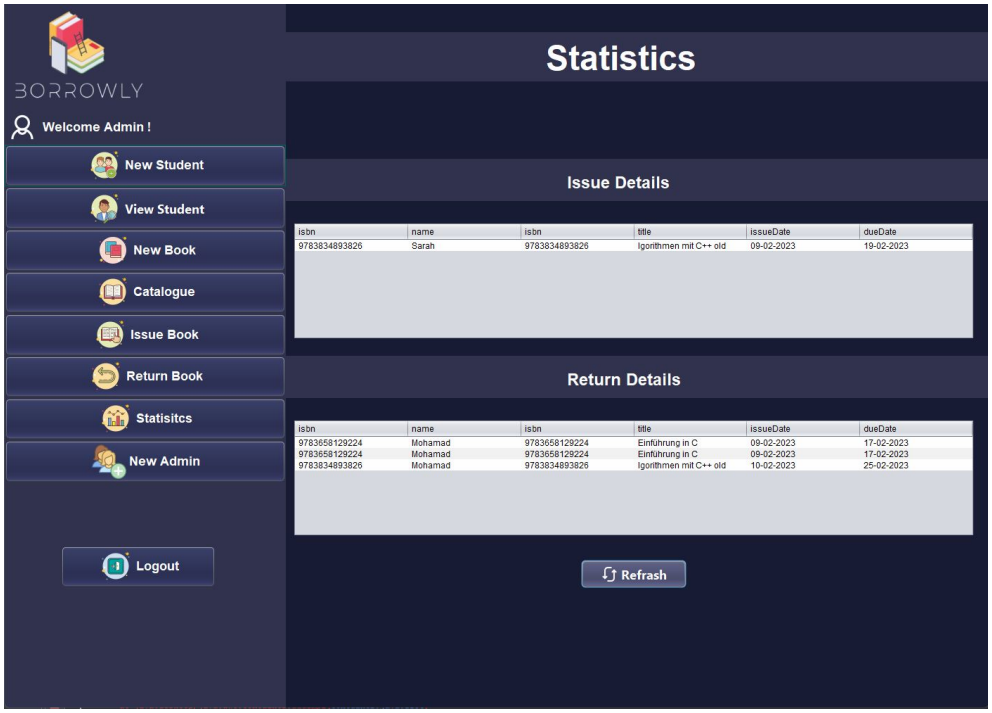

Figure 29: Overview Of The Books Issued/Returned

5.1.10 New Admin

The admin can create new admin accounts by providing the username and password for the new admin. This allows multiple admins to access the system and perform various tasks.

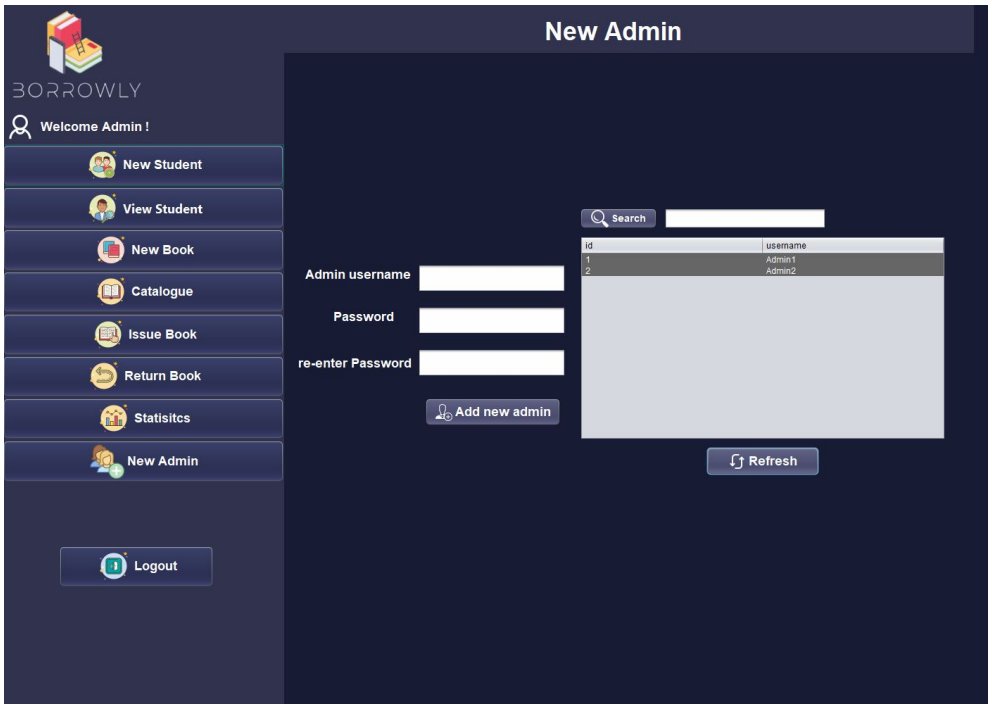

Figure 30: New Admin Window

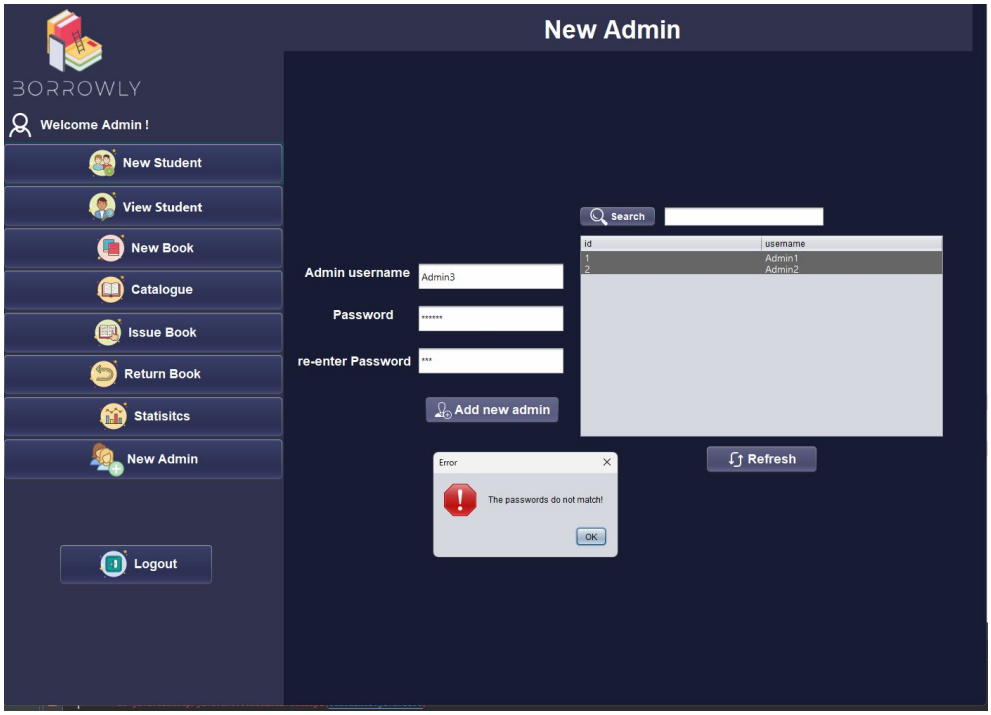

Figure 31: Error When The Re-Entered Password Doesn't Match

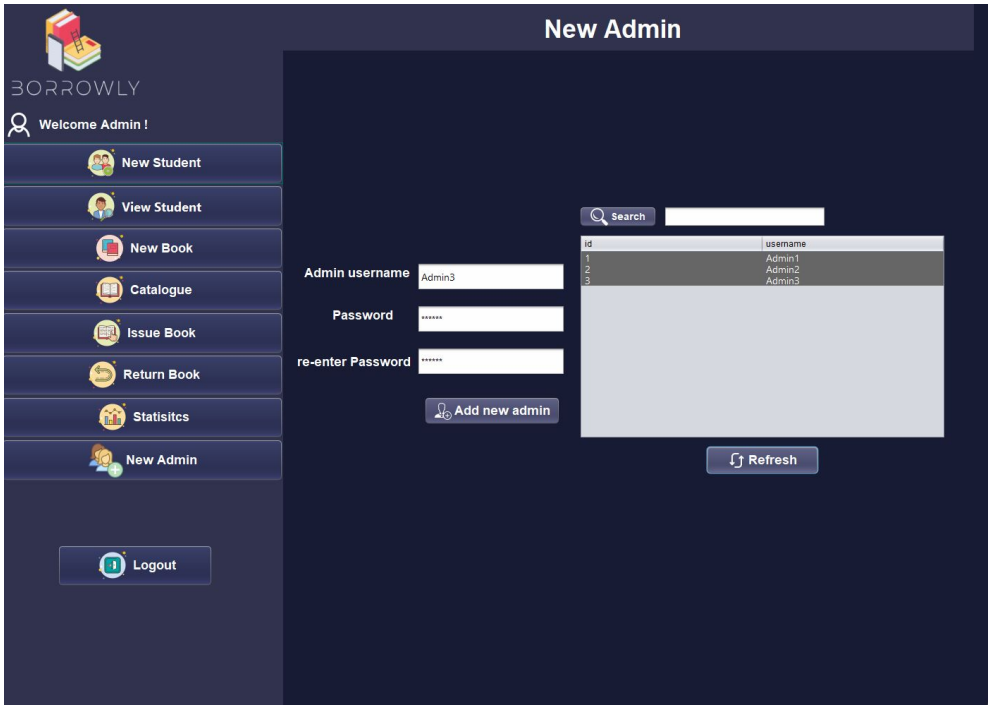

Figure 32: New Admin3 Successfully Registered

5.2 Student scenarios

The interface of the Library Management System is straightforward, making it easy for students to access the information they need quickly and efficiently. With this software, students have the ability to view all the available books of the library and to view their history of issued books, all in one place.

5.2.1 Student Login

In the Library Management System, the user experience for a student begins with the login process. The student must provide their student ID and last name in order to access their account. If the student ID and last name provided do not match the information stored in the system, an error message will appear, indicating that the login has failed 34

Figure 33: Student Login Window

However, if the student ID and last name match, the student will be directed to the student homepage window.36

Figure 34: Student false Login

Figure 35: Student proper Login

5.2.2 Student Homepage

On the student homepage window, the student will be presented with a catalogue of all the available books in the library37

Figure 36: Student Homepage Window

5.2.3 View Catalogue

The student can browse through the catalogue, searching for books based on various criteria such as title, author, or publication date.38

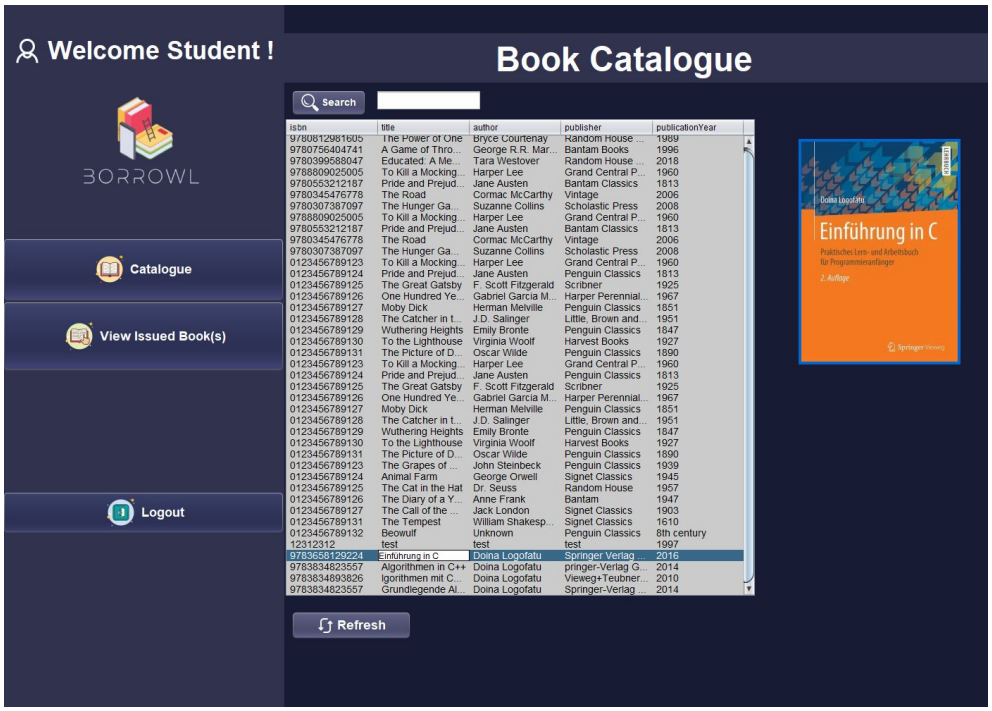

Figure 37: Student View Catalogue Window

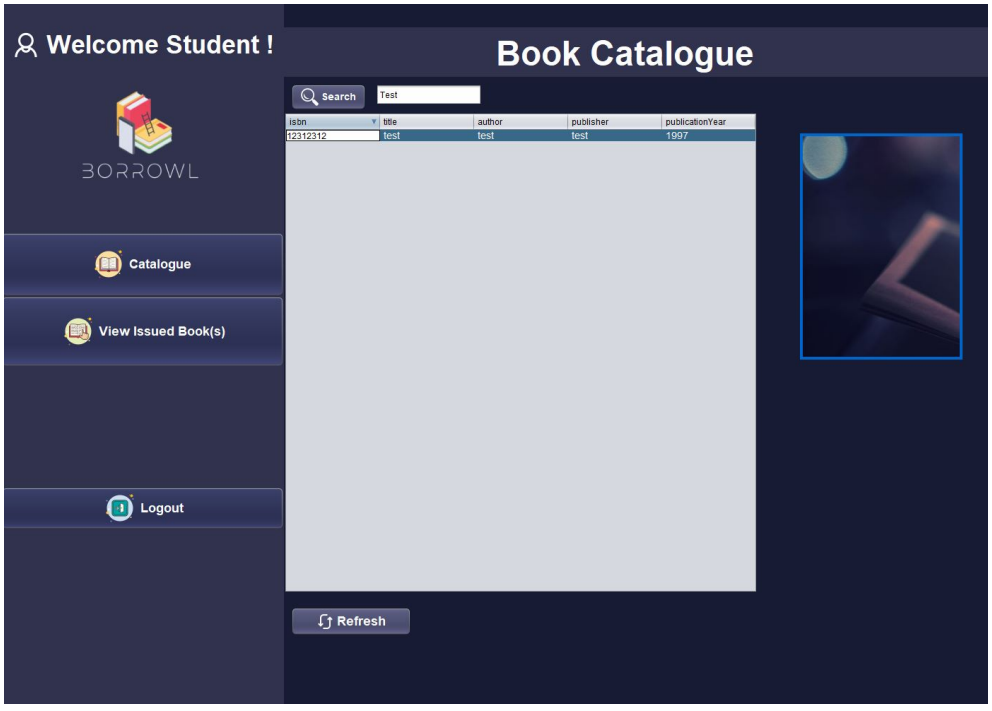

Figure 38: Student Search Book

5.2.4 View Issued Books

In addition to browsing the catalogue, the student can view a list of the books they have currently issued from the library. This list will include information such as the book title, the date it was issued, and the status of the book (i.e. whether it has been returned or not).39 The student will be able to view this information in real-time, ensuring that they are always up-to-date with their library issued books.

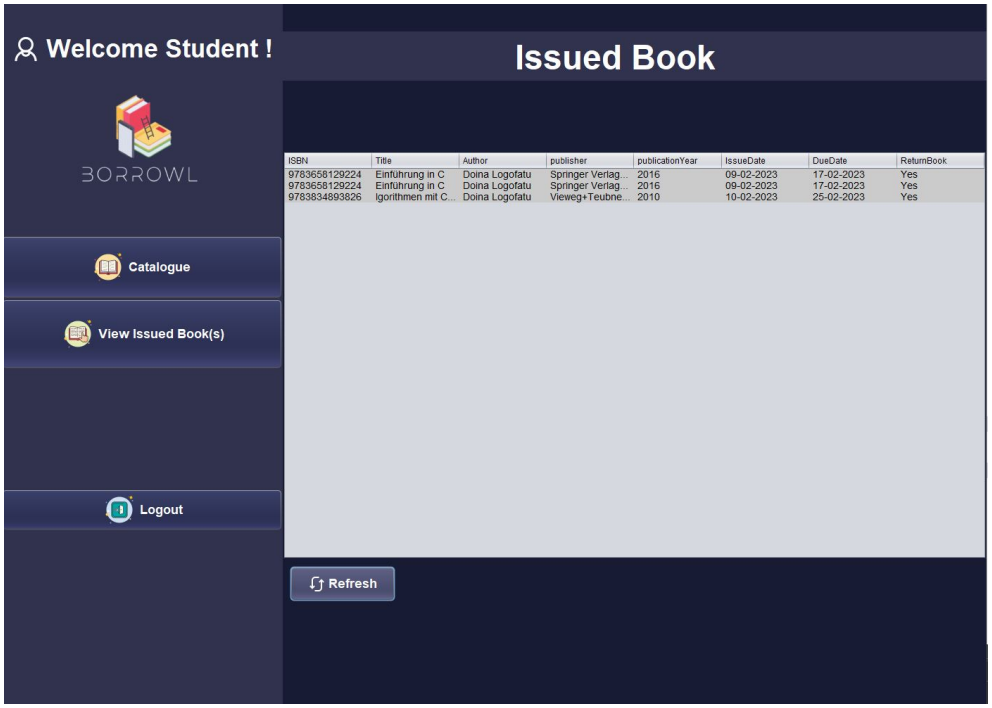

Figure 39: Student Issue Book Window

6 Experimental Results

6.1 Frontend experimentation

Our team experimented with various GUI elements such as colors, sizes, frames, pictures, icons, and buttons. We discovered that the use of specific color combinations, as well as the size of the frames and icons, had the greatest impact on user engagement and overall satisfaction with the interface over time. To improve visibility and accessibility, for example, we discovered that users preferred larger icons and buttons, as well as darker and contrasting color schemes. The results of these experiments enabled us to make more informed design decisions and improve the GUI's user experience.

The figure 40 shows how our GUI's used to look like and how they have improved to the final look.

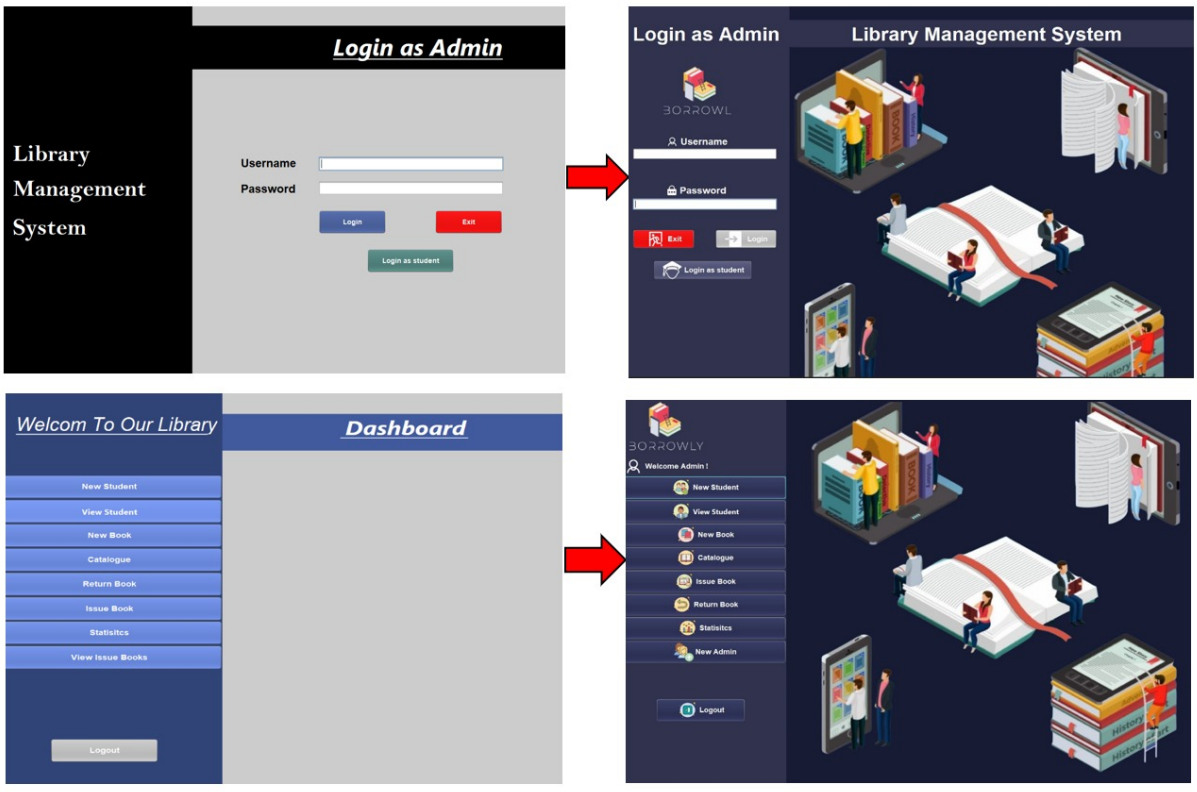

Figure 40: GUI Look Developement

6.2 Backend experimentation

To improve the security of the information or databases being stored, our team conducted an experiment in which we encrypted the login credentials of the actor admin. The results of this experiment demonstrated that the new encryption method was effective in securing login credentials and providing a higher level of security for the stored information. The encrypted password helped to prevent unauthorized access and ensured the security of sensitive information. Overall, the experiment provided us with the data we needed to confirm that the new encryption method was a safe and effective way to protect sensitive information in the app's backend. The BCrypt library uses a one-way encryption function called ”hashing” to encrypt the password. Hashing is a one-way function, meaning it is not possible to decrypt the hashed password back to its original form. This makes it more secure to store hashed passwords in a database. The class also uses a PreparedStatement instead of a regular Statement to interact with the database. A PreparedStatement is considered more secure against SQL injections compared to a Statement. SQL injections are a type of security vulnerability where an attacker can inject malicious code into a SQL statement and execute it on the database. A PreparedStatement helps prevent this by allowing you to specify placeholders for the values to be inserted into the SQL statement, rather than concatenating values into the statement as a string. (Source: https://www.securityjourney.com/post/how-to-prevent-sql-injection-vulnerabilities-how-prepared-statements-work)

id	username	hashed_password
1	admin	$2a$10$Xd.qzjOZMY.UOaH4NOzSyuqDEsM6CdUyAN2eqXiKz55w91hgK2qxu
2	admin1	$2a$10$V2gMKM4HAHnzIu/9LGKJgeNAoUEiGhrHz8m7PE7zujPezh3iy58oK
3	admin2	$2a$10$FV6YdVmXtds21prS4MggEO8XFfMihLwRKWUzTybu4XGgJAYNls85y
4	admin3	$2a$10$fMk7Q9kFtdaRn6k8uNKP/uWcA07TQ.JkZMXf/DvjLOUo0QSCoNqo6
NULL	NULL	NULL

Figure 41: Storing The Username And Hashed Password In The Database.

7 Conclusions and future work

7.1 How was our Teamwork?

The Borrowly Library Management System project was a resounding success, owing to our team's exceptional teamwork and collaboration. We made a concerted effort from the start to collaborate effectively and efficiently, and it paid off in the end. Our team members demonstrated a strong commitment to the success of Borrowly throughout the project, and we are proud of what we have accomplished. ”We sincerely thank our tutor Sheikh Sharfuddin Mim for his invaluable guidance and support throughout the project, which was critical to our success.”

7.2 What have we learned?

Borrowly's development taught us the value of effective project management, in addition to teamwork. We were able to keep the project on track and complete everything on time and to a high standard by breaking it down into smaller, manageable tasks. Finally, our time working on Borrowly has been a great learning experience for all of us. We learned new skills, gained experience in various areas of software development, and gained a better understanding of the importance of continuous testing and improvement.

7.3 Ideas for the future development of our application?

Going forward, we believe that the Library Management System has enormous potential for future development. We intend to prioritize improving search functionality, incorporating new features such as email notifications and resource recommendations, and making the system more accessible to users with disabilities. Overall, we have had an exciting and rewarding experience with this project, and we are proud of what we have accomplished together. We are confident that the Library Management System will remain a valuable resource for libraries and library users for many years to come.

Other Sources

Christina Kopecky (Oct. 8, 2020). *What are database schemas? 5 minute guide with examples*. URL: https://www.educative.io/blog/what-are-database-schemas-examples.

tutorialspoint contributors (July 28, 2021a). *JDBC - Introduction*. URL: https://www.tutorialspoint.com/jdbc/jdbc-introduction.htm.

tutorialspoint contributors (July 28, 2021b). *JDBC - Introduction*. URL: https://www.tutorialspoint.com/jdbc/jdbc-db-connections.htm.

tutorialspoint contributors (July 28, 2021c). *JDBC - Introduction*. URL: https://www.tutorialspoint.com/jdbc/jdbc-statements.htm.

tutorialspoint contributors (July 28, 2021d). *JDBC - Introduction*. URL: https://www.tutorialspoint.com/jdbc/jdbc-result-sets.htm.