Subject System’s Introduction

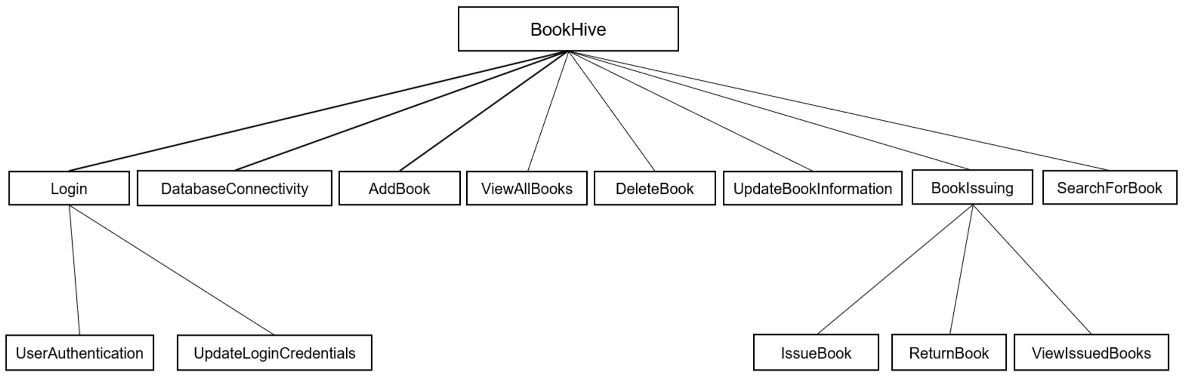
Subject System: BookHive (a Library Management System)

BookHive is a Library Management System (LMS), a Java GUI application designed specifically for small to medium sized libraries. It features a user-friendly interface that meets the needs of librarians. Librarians use this powerful tool to simplify the management of book collections, user records, and transactions. As a facilitator, the system creates a friendly environment for librarians and increases their efficiency in organizing collections. At the same time, users benefit from a seamless and intuitive experience when interacting with the library.

In the past, libraries required a lot of paperwork: card cataloging, inventory management, and book checks and returns, resulting in late tracking and limited information available to librarians. Telephone calls and written letters were the common forms of communication between libraries. The game in the present day has changed. Modern library management systems have shattered these traditional methods. Now tasks are automated, efficiency is through the roof, information is at reach for everyone, and analysis and reporting have reached new levels.

BookHive caters for a range of users, including administrators, staff members, students and librarians. In every category of users, the BookHive offers and facilitates different tasks, such as collection management, simplifying login/logout procedures, updating login credentials, carrying out book searches, borrowing different books, ensuring the visibility of all books, including visibility of borrowed books, and of course, returning books, adding books to the system, and removing books from the system. BookHive has been designed to ensure the needs of various users, reducing the library's manual operations, and contributing to enhancing the experience for both staff members and library users.

Feature Model for BookHive



Login: Is the entry point to the BookHive system. Users can enter their credentials, such as a username and password, to access into the system. Users need then to validate their identity here before they can explore the BookHive. This ensures that only authorized users can enter the system. Location: LoginDAO.java.

UserAuthentication: Is responsible for overseeing the user login process. Users initiate the login process by entering their ID and password. This information is then checked against the database records to confirm its validity. If the provided credentials match the records, the system authorizes access, ensuring a secure entry into the BookHive system. User Authentication is therefore responsible for monitoring the management of credentials and verifying the identity of users throughout the login process. **Location:** Loginform.java/actionPerformed(); Loginform.java/getAuthorization() ;LoginForm.java/setLogin();

UpdateLoginCredentials: Allows users to update or modify their login information, such as username and password, the authentication details that ensure the security of user accounts in order to log in to an online account. **Location:** UpdateLogin.java; UpdateLogin.From; IssueDAO.java LoginDAO.java/setUser(); LoginDAO.java/setPass().

DatabaseConnectivity: Is the backbone of the BookHive system and is responsible for connecting the system to the database where all the important information is stored. It's like a bridge between the BookHive and its storage information, allowing seamless communication between them, such as storing, retrieving and managing data. It is therefore important to keep everything organised and running efficiently within the BookHive. **Location:** ConfigDB.java; DBCredentials.java.

ViewAllBooks: Allows all users to view a list of all the books available in the BookHive system. They can also search the catalog by keyword and explore the entire information, i.e., the collection of books. Users can filter books by book ID, book author, book name, and quantity. **Location:** ViewBooks.java; ViewBooks.form; ViewBooks.java/line264-269.

AddBook: Allows all registered users of BookHive system to add new books to the system, providing relevant details such as book ID, name, author, and quantity. **Location:** AddBooks.java/ line 89-97 actionPerformed().

DeleteBook: Allows all registered users to remove unpopular books from the BookHive system using this feature, then the quantity of available books will be updated. **Location:** ViewBooks.java/ line 189-287 AddActionListener().

UpdateBookInformation: Facilitates the modification of book details, allowing all registered users to update relevant information such as book ID, name, author, and quantity. **Location:** UpdateDialog.java; UpdateDialog.form; ViewBooks.java/printTable(); IssueBook.Java/line126-130.

BookIssuing: Abstract feature that manages the process of issuing books to users, returning issued books, and viewing issued books.

IssueBook: Allows all registered users to borrow specific books from the Hive library system. Users should provide information such as reader ID, reader name, book name and the date of issuing, otherwise the date for today will be added automatically. **Location:** IssueBook.java; IssueBook.form.

ReturnBook: Allows all registered users to return a specific borrowed book, updating the system status accordingly once the return has been confirmed. **Location:** ViewIssuedBooks.java/ line 150-205 actionPerformed().

ViewIssuedBooks: Provides a view of books by keyword search for specific details about the books. It is also responsible for checking whether the book has been returned or whether it is still in the 'pending' status using the date of the book in the Issued Books database system. **Location:** ViewIssuedBooks.java; ViewIssuedBooks.from.

SearchForBook: Users can search for specific books in the book database system, allowing them to find items based on various keywords such as reader ID, reader name, book name, date of issue, date of return, and even by issue status. **Location:** ViewIssuedBooks.java/search()