**National University of Computer & Emerging Sciences**

**Karachi Campus**



**The Library Management System**

**Project Report**

**Programming Fundamentals**

**Section: BCS1-K**

**Group Members**

**Muhammad Saad khan 24K-0680**

**Muhammad Shaheer Ahsan 24K-0915**

**Hamza Malik 24K-0799**

**Introduction:**

The **Library Management System** is a software application designed to streamline and automate the management of library operations. This system provides a user-friendly interface for efficiently handling tasks such as book cataloging, issuing, returning, and maintaining member records. Built using the C programming language, the project leverages fundamental programming concepts such as file handling, functions, and structures to deliver a practical solution.

The primary goal of this project is to simplify library management, reduce manual workload, and minimize errors in record-keeping. It serves as an excellent opportunity to apply theoretical knowledge of programming fundamentals to a real-world scenario, fostering both technical and problem-solving skills.

**Background:**

Libraries traditionally relied on manual processes for cataloging, issuing books, and managing records, which were time-consuming and prone to errors. To address these challenges, Library Management Systems were introduced, automating these tasks for greater efficiency and accuracy. This project aims to implement such a system using the C programming language, applying fundamental programming concepts to solve real-world problems.

**Project Specification:**

1. **Programming Language:**
   * Implemented in C language, leveraging concepts like structures, file handling, and conditional statements.
2. **Functional Modules:**
   * **Authentication System:**
     + Users can sign up and log in with unique credentials.
     + Admin access is secured with a predefined password ("admin123").
   * **Admin Features:**
     + Add new books to the library, including title, author, ISBN, genre, and availability status.
     + Remove books from the library database.
     + View all books in the library (display directory).
     + View the user directory containing registered users.
   * **User Features:**
     + Search for books by title.
     + Issue books if available (updates the book's availability status).
     + Return books (restores availability status).
     + Change their password.
3. **Data Management:**
   * Book records are stored in a binary file (books.dat), ensuring persistence across sessions.
   * User credentials are stored in a separate binary file (Users.dat).
4. **Error Handling:**
   * Validates user input and provides appropriate feedback for incorrect inputs or invalid operations (e.g., non-existent books, mismatched passwords).
   * Handles file access errors gracefully with error messages.
5. **Key Features:**
   * Serial Number Generation: Automatically assigns serial numbers to books based on existing records.
   * Availability Check: Ensures books can only be issued when available.
   * File Updates: Updates binary files in real-time for added, removed, or modified records.
6. **User Interface:**
   * Provides a simple text-based menu-driven interface.
   * Separates functionality for admin and regular users.
7. **Other Features:**
   * Supports secure and modular operations with clear differentiation between admin and user privileges.
   * Ensures consistent file updates with the use of temporary files during delete operations.
   * Automatic screen clearing for better user navigation (using system("cls") on Windows).

**Problem Analysis:**

The project addresses inefficiencies in manual library management, such as tracking books and users. It provides functionalities for:

1. Admins: Add/remove books, view records, and manage users.
2. Users: Search, issue, return books, and update passwords.

Core Features include authentication, file-based storage for persistence, book availability tracking, and user data management.

**Solution Design:**

**Key Components**

1. **Data Structures:**
   * struct Book: Stores book details like title, author, ISBN, genre, and availability.
   * struct User: Stores user details such as username and password.
2. **Modules and Functions:**
   * **Authentication:**
     + signup(): Allows new users to register with a username and password.
     + login(): Validates user credentials for access.
     + admin\_login(): Provides admin-specific authentication with a predefined password.
   * **Admin Features:**
     + addBook(): Adds new book records to the system.
     + removeBook(): Deletes existing books from the records.
     + displayDirectory(): Displays all book records in a tabular format.
     + view\_users(): Lists all registered users and their credentials.
   * **User Features:**
     + **i**ssue book(): Allows users to borrow a book if it is available.
     + return\_book(): Enables users to return a borrowed book, updating its status.
     + search\_book(): Searches for books by title.
     + change\_password(): Lets users update their password securely.
   * **Utility Functions:**
     + get\_next\_serial\_number(): Generates sequential serial numbers for new books.
     + clearScreen(): Clears the console screen for better readability.
3. **Storage Mechanism:**
   * **Files:**
     + books.dat: Stores all book records.
     + Users.dat: Stores user account details.
   * Ensures persistence of data between program executions using file I/O.
4. **Menu System:**
   * Admin Menu: Offers options to manage books and view user records.
   * User Menu: Provides options to search, issue, return books, and manage account details.

**Flow of Execution**

1. Program starts with a login/signup prompt.
2. Based on user type:
   * Admin: Navigates to admin-specific operations.
   * User: Accesses features to manage personal library interactions.
3. Operations update the respective files to ensure accurate and persistent data.

**Extra Features:**

The Library Management System includes additional functionalities to enhance user experience and administrative control:

1. **Change User Password**:
   * **Description**: Users can securely update their account password.
   * **Benefit**: Increases security by allowing users to reset passwords in case of suspicion or forgetfulness.
2. **View User Directory** (Admin Only):
   * **Description**: Admins can view a list of all registered users along with their credentials.
   * **Benefit**: Enables admins to monitor user activity and manage accounts effectively.
3. **Remove Existing Book** (Admin Only):
   * **Description**: Admins can delete a book's record from the system.
   * **Benefit**: Maintains the library's collection by removing outdated, damaged, or unnecessary entries.

These features improve the overall functionality of the system, making it more comprehensive and user-friendly for both admins and users.

**Project Workload Distribution:**

Firstly, we all laid out an outline for our program, deciding on what functions to include and which would be prioritized the most. Hamza was the one who eventually compiled the base outline of our code that we started to work on.

I did the base functions and menus for the admin and user. The directory of books was also arranged for by me. I also created login and logout functions and set them to loop.

Saad took over and did half of the functions, while the other half was split between me and Hamza. Most of the technical functions, like check\_fine, issue\_book, return\_book, etc. were done by him.

Hamza did the AddBook, RemoveBook, clearScreen and get\_next\_serial\_number functions. He also helped out with general debugging as well.

**Result (Output Screenshot):**

1.Signup:

A screen shot of a computer

Description automatically generated

2.Login:

A screen shot of a computer

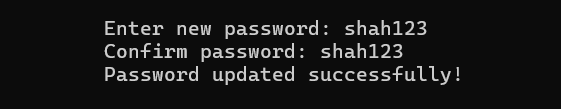
Description automatically generated

3.User Menu:

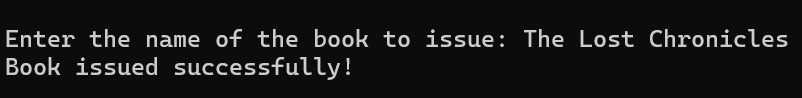
A screen shot of a computer

Description automatically generated

5.Change my pass:



6.Issue a book:

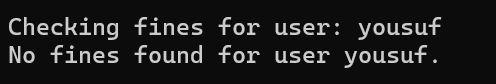


7.Return a book:

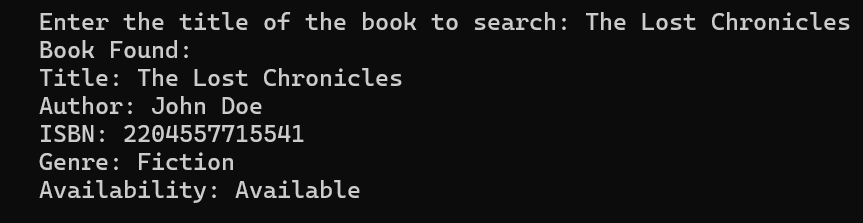
A black background with white text

Description automatically generated

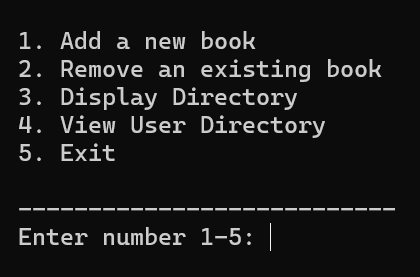
8.See my Fines:



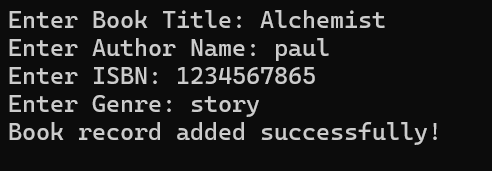
9.Search for a book:



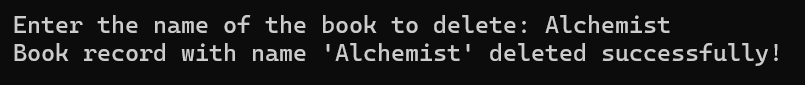
10.Admin Menu:



11.Add a new book:



12.Remove an existing book:



13.Display directory:

A screenshot of a computer screen

Description automatically generated

14.View user directory:

