CHITTAGONG UNIVERSITY OF ENGINEERING & TECHNOLOGY

Project Report on

Library Management System

Course Title: **Operating Systems (Sessional)**
Course Code: **CSE-336**

Prepared For:
Md. Shafiul Alam Forhad
Assistant Professor
Dept. of CSE

Hasan Murad
Assistant Professor
Dept. of CSE

Prepared By:
ID: 2004129,2004130,2004131,2004132
Session: 2020-21

Date: 30 June 2024

Department of
Computer Science and Engineering

Proposed Project:

A Library Management System with functionalities for both administrators and users.

Introduction:

The Library Management System is a simple yet functional project developed using Bash scripting. This system is designed to facilitate the management of a library's book inventory and streamline the process of borrowing and returning books. It serves two primary roles: the administrator and the user. Administrators can add, edit, view, and delete books, as well as manage their login credentials. Users can view available books, borrow books, return borrowed books, and check their borrowed books list. The project utilizes text files to store information, ensuring data persistence across sessions. This system is particularly useful for small libraries or personal collections where a lightweight and straightforward solution is needed.

Objective:

The main objectives of the Library Management System project are:

- To develop a basic library management system that allows for easy and efficient management of books within a library setting.
- To provide a functional interface for administrators to perform operations such as adding, editing, viewing, and deleting book records, and updating login credentials.
- To enable users to interact with the library system by viewing available books, borrowing books, returning borrowed books, and checking the status of their borrowed books.
- To utilize Bash scripting for creating a command-line interface (CLI) application that is simple, easy to use, and effective in managing library operations.
- To implement file-based data storage to ensure that book and user information is stored persistently and can be accessed and modified across different sessions.
- To demonstrate the practical use of Bash scripting in real-world applications, highlighting its strengths in automating and simplifying administrative tasks.

Tools and Technologies Used:

1. Bash Scripting

- **Description**: A Unix shell and command language used for writing scripts to automate tasks.
- **Usage**: The entire system is implemented using Bash scripting for handling operations and user interactions.

2. Text Files

- **Description**: Simple, human-readable files for storing data.
- **Usage**:
    - login.txt: Stores admin credentials.
    - books.txt: Stores book details and availability.
    - borrowed_books.txt: Logs borrowed books.

3. Command-Line Interface (CLI)

- **Description**: A text-based user interface for interacting with the system by typing commands.
- **Usage**: Both administrators and users navigate the system through CLI menus.

4. Unix-Based Operating System

- **Description**: Operating systems like Linux or macOS that support Bash.
- **Usage**: The system is designed to run on Unix-based OS for native support of Bash scripting.

5. Text Processing Tools

- **Description**: Unix tools like grep and sed for searching and editing text.
- **Usage**: Used within the script to manipulate text files, search for entries, and update records.

System Requirements:

Software

- **Unix-based Operating System**: Linux or macOS
- **Bash Shell**: Installed by default on Unix-based systems

Hardware

- **Basic hardware**: No special requirements; a standard computer running a Unix-based OS will suffice.

Working Principle:

This script implements a simple library management system with functionalities for both administrators and users. The system uses text files to store login credentials, available books, and borrowed books. Below is a detailed explanation of the working principles:

1.Initialization

- **initialize_system()**
    - **Purpose**: Ensures that necessary files (login.txt, books.txt, borrowed_books.txt) are present.
    - **Functionality**:
        - If login.txt does not exist, it creates the file with default admin credentials (admin:admin).
        - If books.txt does not exist, it creates an empty file for storing book details.
        - If borrowed_books.txt does not exist, it creates an empty file for storing borrowed book details.

2.Admin Functions

- **admin_login()**
    - **Purpose**: Authenticates admin users.
    - **Functionality**:
        - Prompts the user for a username and password.
        - Checks the credentials against the login.txt file.
        - If credentials match, grants access to the admin menu; otherwise, denies access.
- **admin_menu()**
    - **Purpose**: Provides various options for the admin to manage the library.
    - **Functionality**:
        - Displays a menu with options to add, view, edit, and delete books, change admin credentials, and exit.
        - Invokes corresponding functions based on user selection.
- **change_admin_credentials()**
    - **Purpose**: Allows the admin to update the login credentials.
    - **Functionality**:

        - Prompts for a new username and password.
        - Updates the login.txt file with new credentials.

- **add_book()**
    - **Purpose**: Adds a new book to the library.
    - **Functionality**:
        - Prompts for the book name and author name.
        - Adds the book details to books.txt with a status of "Available".
- **view_books()**
    - **Purpose**: Displays the list of books available in the library.
    - **Functionality**:
        - Checks if books.txt has any content.
        - Displays the book list if available, otherwise informs that no books are available.
- **edit_book()**
    - **Purpose**: Edits the details of an existing book.
    - **Functionality**:
        - Prompts for the exact name of the book to edit.
        - If the book exists, prompts for new details and updates the books.txt file.
- **delete_book()**
    - **Purpose**: Deletes a book from the library.
    - **Functionality**:
        - Prompts for the book name to delete.
        - If the book exists, removes it from books.txt.

3.User Functions

- **user_menu()**
    - **Purpose**: Provides various options for users to interact with the library.
    - **Functionality**:
        - Displays a menu with options to view books, borrow books, return books, check borrowed books, and exit.
        - Invokes corresponding functions based on user selection.
- **borrow_book()**
    - **Purpose**: Allows a user to borrow a book.

    - **Functionality**:
        - Prompts for the exact name of the book to borrow.
        - If the book is available, updates its status to "Borrowed" in books.txt and logs the borrowing in borrowed_books.txt.
- **return_book()**
    - **Purpose**: Allows a user to return a borrowed book.
    - **Functionality**:
        - Prompts for the exact name of the book to return.
        - If the book was borrowed, updates its status to "Available" in books.txt and removes the entry from borrowed_books.txt.
- **check_borrowed_books()**
    - **Purpose**: Displays the list of books borrowed by the user.
    - **Functionality**:
        - Displays the content of borrowed_books.txt.

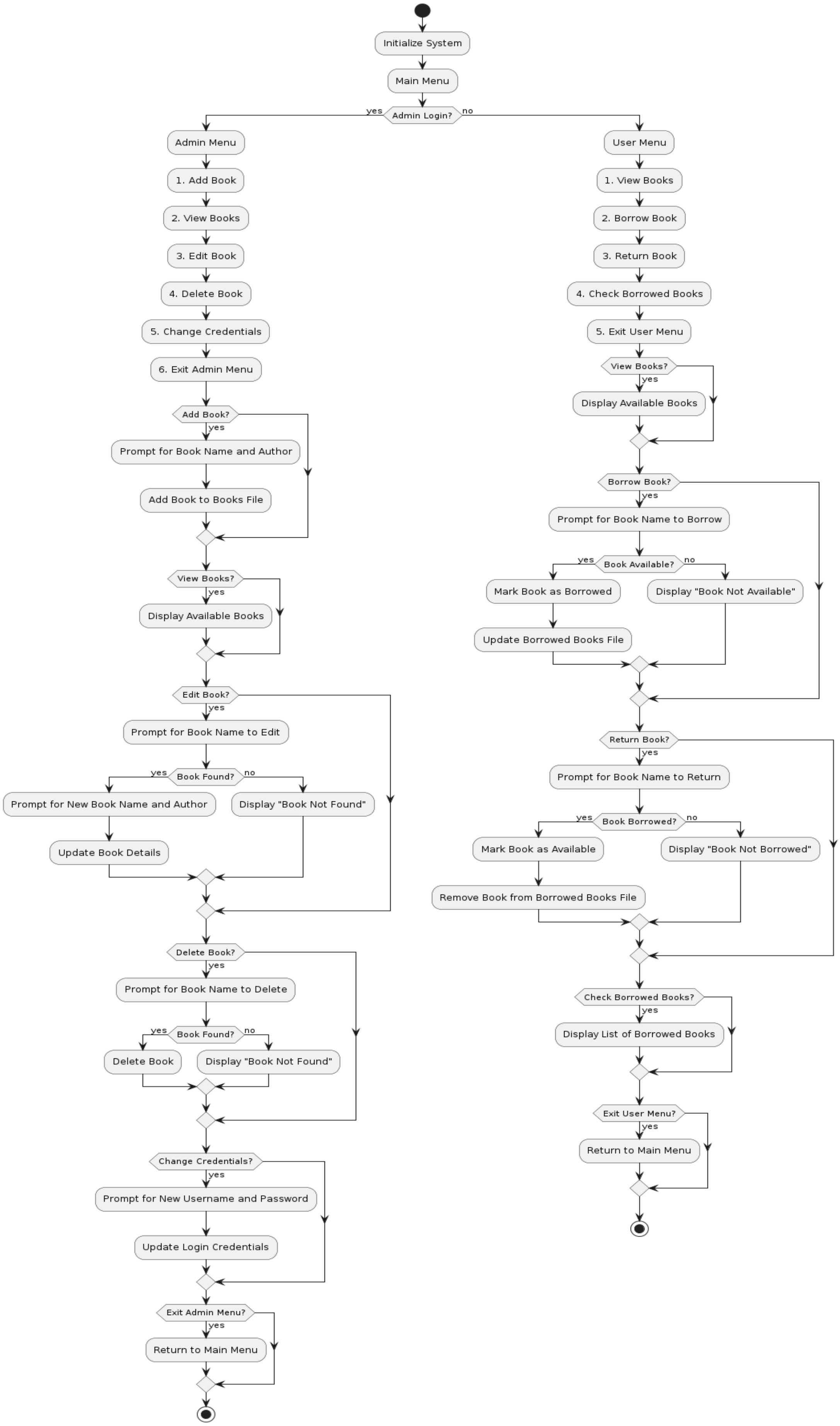
4.Main Menu

- **main_menu()**
    - **Purpose**: Entry point for the system.
    - **Functionality**:
        - Displays a menu with options for admin login, user access, and exit.
        - Invokes corresponding functions based on user selection.

5.Execution

- **Script Execution**
    - The script starts by calling initialize_system to ensure necessary files are in place.
    - It then enters the main_menu loop to handle user interactions.

Flowchart:


Initialize System
Main Menu
Admin Login?
yes
no
Admin Menu
1. Add Book
2. View Books
3. Edit Book
4. Delete Book
5. Change Credentials
6. Exit Admin Menu
Add Book?
yes
Prompt for Book Name and Author
Add Book to Books File
View Books?
yes
Display Available Books
Edit Book?
yes
Prompt for Book Name to Edit
Book Found?
yes
no
Prompt for New Book Name and Author
Display "Book Not Found"
Update Book Details
Delete Book?
yes
Prompt for Book Name to Delete
Book Found?
yes
no
Delete Book
Display "Book Not Found"
Change Credentials?
yes
Prompt for New Username and Password
Update Login Credentials
Exit Admin Menu?
yes
Return to Main Menu
User Menu
1. View Books
2. Borrow Book
3. Return Book
4. Check Borrowed Books
5. Exit User Menu
View Books?
yes
Display Available Books
Borrow Book?
yes
Prompt for Book Name to Borrow
Book Available?
yes
no
Mark Book as Borrowed
Display "Book Not Available"
Update Borrowed Books File
Return Book?
yes
Prompt for Book Name to Return
Book Borrowed?
yes
no
Mark Book as Available
Display "Book Not Borrowed"
Remove Book from Borrowed Books File
Check Borrowed Books?
yes
Display List of Borrowed Books
Exit User Menu?
yes
Return to Main Menu

Usage Instructions & Testing:

1.How to Run the Script

- Open a terminal on a Unix-based operating system (Linux or macOS).
- Navigate to the directory containing the script.
- Make the script executable by running “chmod +x Library_Management.sh”
- Execute the script by running “./Library_Management.sh”

Admin Operations

1. **Login as Admin**
   - Select "1. Admin Login" from the main menu.
   - Enter the username and password when prompted.
2. **Admin Menu Options**
   - **Add Book**: Choose option 1, enter book name and author.
   - **View Books**: Choose option 2 to see the list of available books.
   - **Edit Book**: Choose option 3, enter the exact book name to edit, then provide new details.
   - **Delete Book**: Choose option 4, enter the book name to delete.
   - **Change Credentials**: Choose option 5, enter new username and password.
   - **Exit**: Choose option 6 to log out from the admin menu.

User Operations

1. **Access User Menu**
   - Select "2. User Access" from the main menu.
2. **User Menu Options**
   - **View Books**: Choose option 1 to see available books.
   - **Borrow Book**: Choose option 2, enter the exact book name to borrow.
   - **Return Book**: Choose option 3, enter the exact book name to return.
   - **Check Borrowed Books**: Choose option 4 to view borrowed books.
   - **Exit**: Choose option 5 to exit the user menu.

Exiting the System

- Select "3. Exit" from the main menu to terminate the script.

Conclusion:

The Library Management System project demonstrates the effective use of Bash scripting to create a simple yet functional application for managing a library's book inventory. This system allows administrators to easily add, edit, view, and delete books, manage user borrowings, and update login credentials, all through a command-line interface. Users can interact with the system to view available books, borrow and return books, and check their borrowing history.

The project showcases the following key points:

- **Practical Application of Bash Scripting**: The system leverages the capabilities of Bash scripting to automate tasks and handle file operations efficiently.
- **File-Based Data Storage**: By using text files to store data, the system ensures data persistence across sessions without the need for complex database setups.
- **User-Friendly CLI**: The command-line interface provides an easy-to-use platform for both administrators and users to perform their respective tasks.

The testing process confirmed that the system works as intended, with all features performing correctly and data being managed accurately. This project provided valuable insights into the use of scripting for task automation and text file manipulation, demonstrating how these tools can be employed to create practical solutions for everyday problems.

Future improvements could include adding more robust error handling, enhancing the user interface for better usability, and possibly migrating to a database system for improved scalability and performance. Nonetheless, this project serves as a solid foundation for understanding the basics of library management through scripting and file handling.