|  |  |
| --- | --- |
| Ex no: 8  Date: 22/04/2019 | **MINI PROJECT** |
| Title | **Library Management System** |
| Team Lead | **185001146 SG Shivanirudh** |
| Team Members | **185001137 Satish Prasad VT**  **185001143 Sharvan Ram Kumaran** |

* **Abstract of the project:**

This program is designed to simulate the working of a Library Management System, in real time. The system works on different modes for Administrative, Member, and Book handling purposes, along with duty lists for staff members.

* **Description of the project:**

The system is accessible to Administrators, Members/Customers and Staff.

This is facilitated by the use of a few structures and several methods for each mode of working.

1. ***STRUCTURES:***

* The **Admin** structure consists of details for the admin’s profile, including the admin’s name, username, password, and adminID. This structure is called upon for any operations involving the admin.
* The **Member** structure consists of details for the member’s profile, including the member’s name, username, memberID, number of books borrowed, names of books borrowed, and wallet. This structure is used for any operations pertaining to the member.
* The **Book** structure consists of details about the books in the library, such as name of the book, author, number of copies available, edition of the book, bookID, price, rating and number of people who have borrowed the book.

1. ***FUNCTIONS:***

* ***Admin:*** The admin function serves to handle the profiles of the members of the library and also update the edition of the books available. The details are stored with the help of a structure containing the name, password etc.
* ***Member:*** The member function allows the user to check and recharge his wallet if necessary, a system which is useful for members who return books beyond the due date. It also allows the user to choose the book to borrow, as well as track the return of borrowed books. The details of a member are stored with the help of a structure containing the name, password, wallet, Number of books borrowed, etc.
* ***Book:*** The book function allows the management to keep a record of all the books in the library, along with the details of the books. The system allows for changes in the list of books, allowing for additions, update of editions, etc. The details of a book are stored with the help of a structure containing the title, name of author, number of copies available, which edition etc.

* **Salient Features:**

1. The system is designed to act as a management module for a library.
2. Once logged in, with a unique username and password, Admins are given the options to manage the profiles of other members, and track the borrowing and returning of books.
3. Admins are also given a checklist of chores to ensure the smooth running of the library, with duties such as dusting and categorizing books. This list is to be updated on a regular basis.
4. Admins are also responsible for making updates to the list of books when a new edition arrives for any book.
5. Once logged in, with a unique username and password, Members are given the options to choose a book to borrow (up to 5 books), return the borrowed books, and look at the list of books borrowed by them.
6. Members can also check the available balance their “*wallet*”, in which money is deposited, to be used in case of late returns.

* **Pseudocode:**

The following is the outline of the driver method of the system, which calls every other function as and when required, to run the system.

* ***Begin***
* ***Clear the screen, and enter fullscreen***
* ***Read the mode to be used by the user***
* ***If admin mode is chosen***

1. ***Accept the username and password***
2. ***Replace password characters with \* characters***
3. ***Check if the login details are correct***
4. ***Report error if details are incorrect***
5. ***If correct, proceed to Menu of options for admins tasks***

* ***Add/Delete Member***
* ***Edition Management (Books)***
* ***Management of checklist of chores (Cleaning, Arranging, etc.)***
* ***Return to Login Screen***
* ***If member mode is chosen***

1. ***Accept the username and password***
2. ***Replace password characters with \* characters***
3. ***Check if the login details are correct***
4. ***Report error if details are incorrect***
5. ***If correct, proceed to Menu of options for Member tasks***

* ***If book option is chosen***

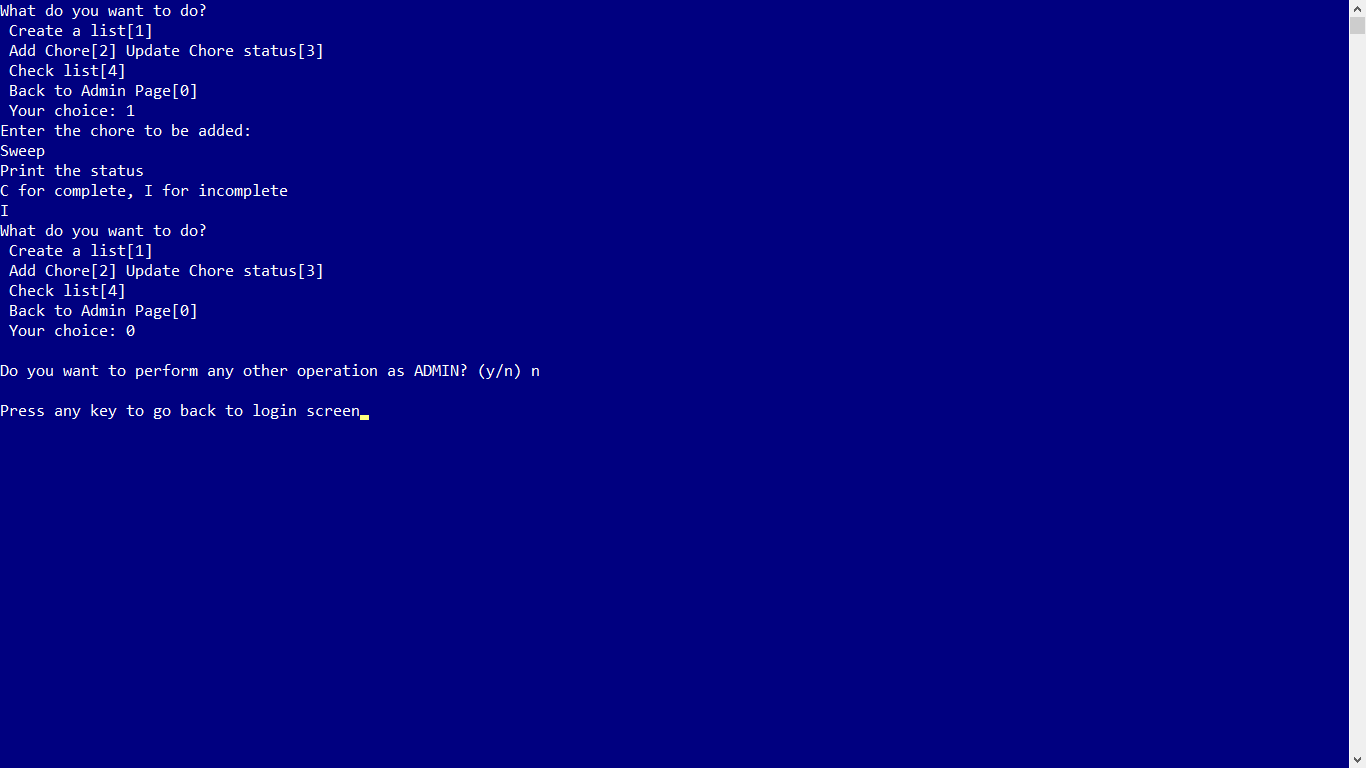
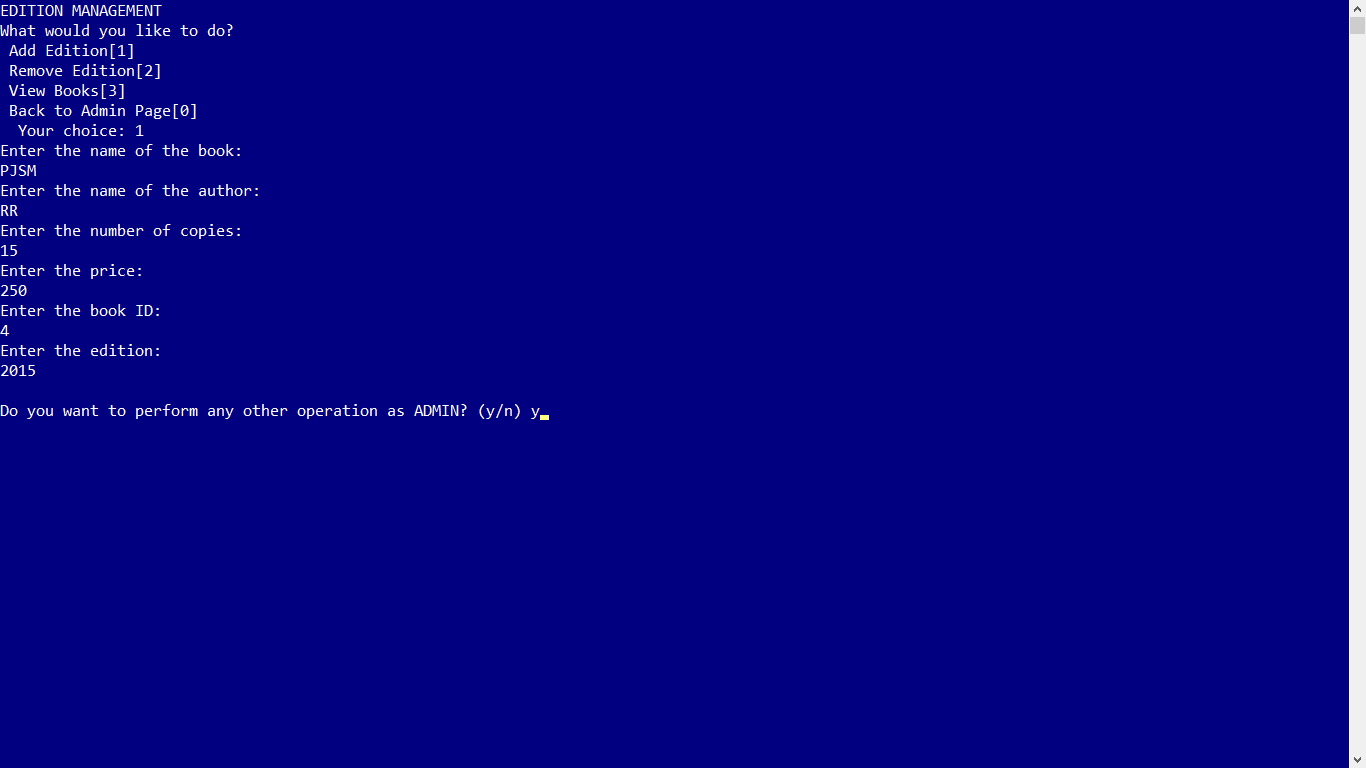
1. ***Choose book to borrow***
2. ***Return borrowed book***
3. ***View checked out books***
4. ***Return to Member options***

* ***If wallet option is chosen***

1. ***Recharge wallet***
2. ***View Wallet balance***
3. ***Return to Member options***

* ***End***
* **Snapshots:**

A close up of a logo

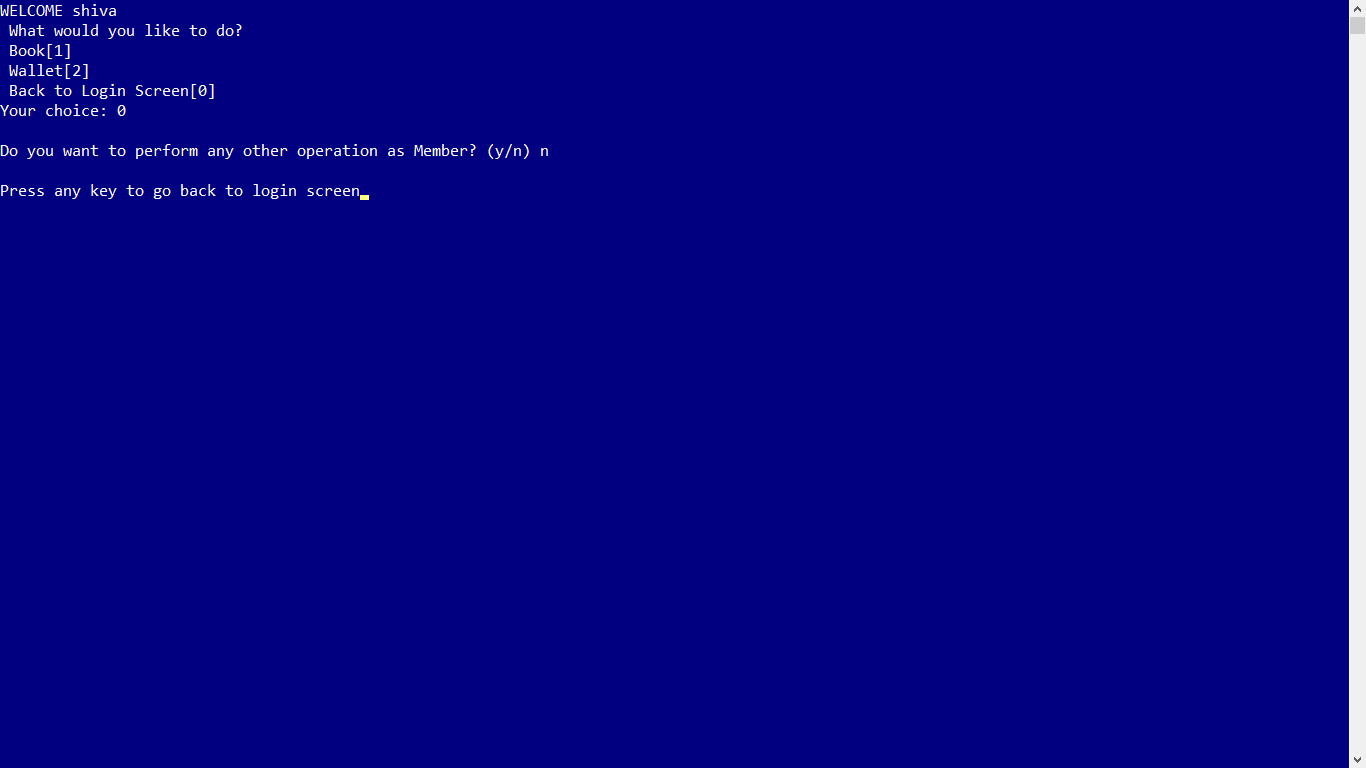
Description automatically generated

A close up of a logo

Description automatically generated

A close up of a logo

Description automatically generated



A close up of a screen

Description automatically generated

* **Learning experience:**

The aim of this project was to try and use all the concepts learned over the course of this semester:

1. Concept of Incremental Programming
2. Using the I/O statements, operators and expressions
3. Conditional and Looping constructs
4. Concept of modular programming
5. Call by Value, Call by Reference
6. Recursive Functions
7. Explicit Function Declaration
8. Handling multi-dimensional arrays
9. Passing arrays to functions
10. Handling strings
11. Passing strings to functions
12. Passing parameters to functions by call by reference by using pointers
13. Dynamic memory allocation
14. Pointers to functions
15. Creating and accessing elements of Structures
16. Passing structures to functions and array of structures to functions
17. Handling files
18. Fileoperations and manipulations
19. To write maintainable code

Having worked on quite a few parts of the code having distinct functions, I learned to use the concepts I knew in different ways, in addition to learning new concepts, some even outside the scope of syllabus. Though initially confused about how to put the pieces of the program together, I learned that in a team we all had to make our code such that it would work together in the main program smoothly, by avoiding similarities in method names, repetitive methods etc. Overall, the project has made me more experienced in coding.