**sCCS 103 Final Project Documentation**

**Part I: Project Overview**

|  |  |
| --- | --- |
| Name of Application: | Liquor Ordering System |
| Group Members: | Subrio, Charryl Lomarda, Ismael Marlon Maestrado, Jazfer |
| Application Description: | This Liquor Ordering System application allows users to create an account, log in, browse a list of liquors, and place orders. Logged-in users can place an order by specifying the name of the liquor and quantity of order. The system checks stock availability and deducts the ordered amount if successful. It then displays an order confirmation with the total price and saves the order details to a text file. The application can load and save data of the user's credentials and list of liquors from and to text files. |
| String Functions used (include how it is used in the program): | **strlen** – used to get the length of username and password for validation purposes.  **strcmp** - used to compare usernames and passwords for login.  **strcpy** - used to copy the contents of the name array into the name field of a struct called liquor.  **strcspn -** used to remove the trailing newline character from the input string liquor\_name, making it easier to process and compare with other strings. |
| Array used (if there are any. Include how the array is used in the program): | **valid\_choice[3]:** Array of integers, used to store valid main menu choices.  **users[MAX\_USERS]**: Array of struct User, used to store user information such as usernames and passwords.  **liquors[MAX\_LIQUOR]**: Array of struct Liquor, used to store liquor information such as names, prices, milliliters, and stock quantities. |

**Part II: Functions used**

These is a listing of all the functions used in the program and a description of each.

|  |  |  |
| --- | --- | --- |
| **Function** | **Description** | **Member assigned** |
| *Ex: acceptStudents* | *This function accepts* ***N*** *names of students and stores these into an array* | *Juan dela Cruz* |
| *load\_data* |  | *charryl* |
| *getMenuChoice* |  | *ismael* |
| *create\_account* |  | *jazfer* |
| *is\_username\_valid* |  | *charryl* |
| *is\_password\_valid* |  | *ismael* |
| *login* |  | *jazfer* |
| *find\_user\_index* |  | *charryl* |
| *display\_liquors* |  | *ismael* |
| *order\_liquor* |  | *jazfer* |
| *find\_liquor\_index* |  | *charryl* |
| *strcmp\_nocase* |  | *Ismael* |
| *export\_receipt* |  | *jazfer* |
| *save\_data* |  | *charryl* |
| **getMenuChoice** |  |  |
| **strcmp\_nocase** | This function is used to compare two strings in a case-insensitive manner. It ensures that the comparison between user input and liquor names is not affected by case differences. | *Charryl* |
| is\_username\_valid | This function checks if a username is valid (alphanumeric characters only). It returns 1 if the username is valid, 0 otherwise.  This function checks whether a username is valid by ensuring it consists only of alphanumeric characters. |  |
| find\_user\_index | This function searches for a user based on username and returns the index of the user in the users array if found. It returns -1 if the user is not found.  This function searches for a given username in the users array and returns its index if found. | Jazfer |
| is\_password\_valid | This function checks if a password is valid (minimum 8 characters). It returns 1 if the password is valid, 0 otherwise. | Jazfer |
| create\_account() | This function allows users to create new accounts by entering a username and password. It performs basic validation and checks for existing usernames. | Jazfer |
| login() | This function allows users to login with their username and password. It provides a limited number of login attempts (optional). | Lon |
| load\_data() | This function reads user and liquor data from text files upon program startup. | Lon |
| save\_data() | This function saves user and liquor data back to text files before program exit. | Lon |
| display\_liquors() | This function displays a list of all available liquors with their names, prices, and stock. |  |
| order\_liquor() | This function allows logged-in users to order liquor. It checks for liquor availability, calculates the total amount, updates stock, and displays an order confirmation message. It also generates a receipt file. |  |
| export\_receipt | This function creates a receipt file containing order details (liquor name, quantity, total amount).  This function exports order details, including the username, liquor name, quantity, total amount, and date, to a receipt file with a unique filename. |  |
| find\_liquor\_index (char liquor\_name) | (Function implementation needed) This function should search the liquors array by name and return the index of the liquor if found. It can use similar logic to find\_user\_index. |  |
| find\_user\_index:  is\_password\_valid: This function validates the password by checking if it has a minimum length of 8 characters.  create\_account: This function allows users to create new accounts by providing a username and password, ensuring they meet the required criteria.  login: This function handles the user login process, verifying the entered username and password against the stored credentials.  load\_data: This function loads user and liquor data from external files into the program's arrays at startup.  save\_data: This function saves user and liquor data from the program's arrays back to external files before exiting.  display\_liquors: This function displays the list of available liquors, including their names, milliliters, prices, and stock levels.  order\_liquor: This function allows users to place orders for liquors by specifying the desired quantity, deducting the stock accordingly, and generating a receipt.  export\_receipt: This function exports order details, including the username, liquor name, quantity, total amount, and date, to a receipt file with a unique filename. |  |  |

Please follow this format:

*Note: Please remove the instructions (in red text color) and fill-in the table with appropriate entries.*