PROBLEM STATEMENT

SECURE SOCKET IMPLEMTATION IN DATABASE MANAGEMENT

(ONLINE LIBRARY MANAGEMENT)

GROUP 1

DESIGN DOCUMENT

**1.INTRODUCTION:**

An eBook is a book in a digital format that can be read on a computer or mobile device. You can read classics, fiction, and non-fiction eBooks. So, in order to read books in digital format we need to download books from database. Database is created to store book details.

Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while the other socket reaches out to the other to form a connection. The server forms the listener socket while the client reaches out to the server

* 1. **PURPOSE:**

The purpose of this project is to provide access to user to download required books from the database which contains the list of book URLs which enables user to read books in digital format. This also enables Admin to modify database. Customer support is also provided to user based on user request to resolve queries.

**1.2 FUNCTIONALITIES OF THE SYSTEM:**

* Admin:

Admin should be able to modify database when logged in with valid login credentials.

Admin can either insert or retrieve or delete book details over database.

* User:

User should be able to download books digitally from database based on requirement.

User can utilize customer support whenever required to resolve any queries.

* Database:

In database, the URLs are uploaded/modified by the admin with certain login credentials and the list of URLs of books which are able to download by User.

* Signals:

Whenever ctrl+c signal is given then control will be transferred to main menu.

Ctrl+z signal is given to exit the program.

* Execv:

It is used to Execute the required function.

**1.3 OPERATING ENVIRONMENT:**

Operating environment are:

* Sockets
* Operating system: Linux
* Data Base (SQL)
* Signals
* Platform: Ubuntu/C++.

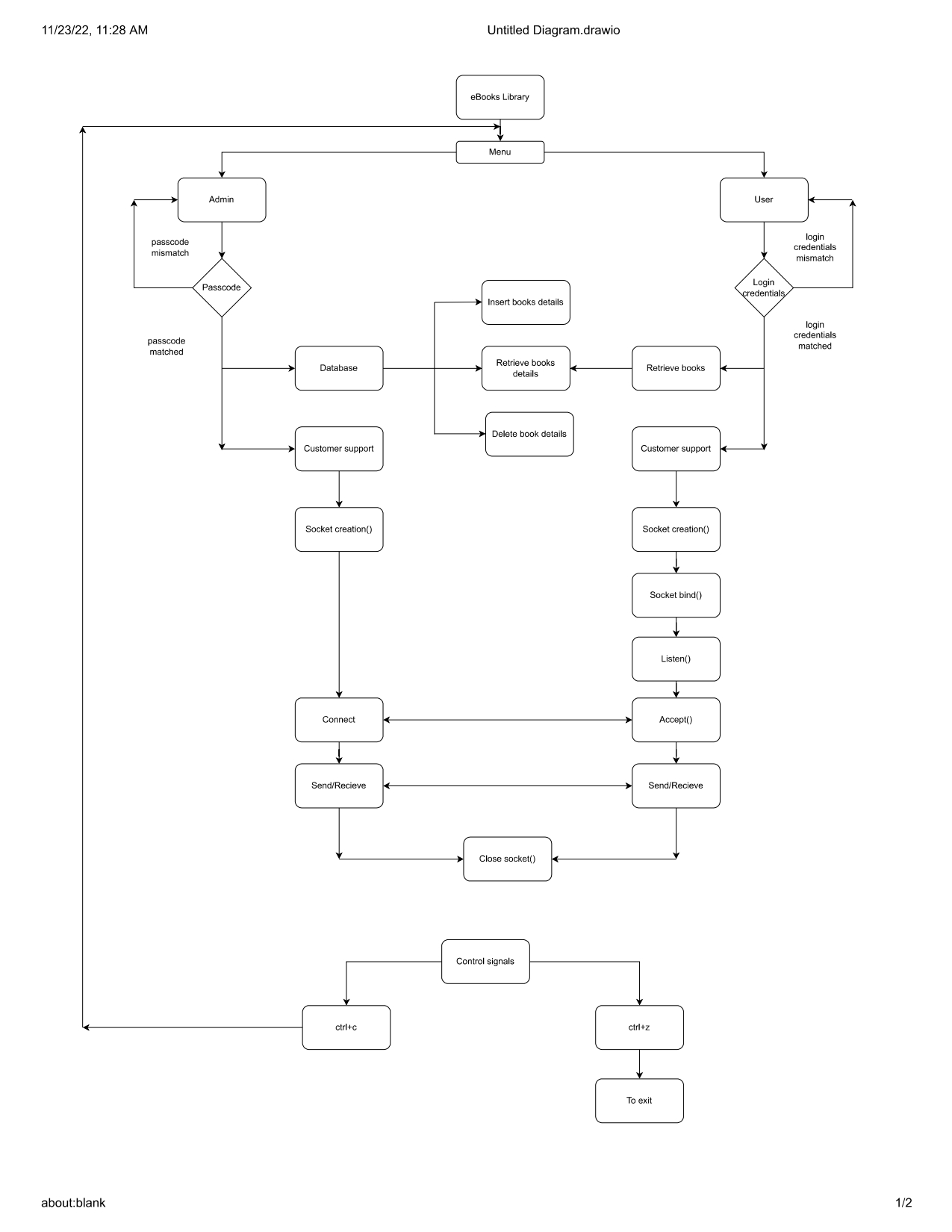
**2. SOFTWARE REQUIREMENTS:**

* **g++/gcc**
* **makefile**
* **valgrind**
* **gcov**
* **git**

**2.1 UNIT TEST:**

|  |  |
| --- | --- |
| SRS\_01 | "Login Credentials:  Admin is required to provide valid passcode for login" |
| SRS\_02 | User registration is required to login |
| SRS\_03 | Login Credentials : Valid User name and password is required for login authentication |
| SRS\_04 | Login Credential: User is required to provide valid login details to access database |
| SRS\_05 | User should be able to get login credentials , if forgot by providing valid login details |
| SRS\_06 | After successful admin login validation admin is permitted to access to Database |
| SRS\_07 | After successful admin login validation admin is permitted to provide customer support for any user issues using sockets |
| SRS\_08 | Admin connect:  The connect() system call connects the socket referred to by the file descriptor sockfd to the address specified by addr. It will try to connect with the server by performing a three-way handshake. |
|  |  |
| SRS\_09 | Creation of Data base, to store book URLs |
| SRS\_010 | Admin should be given access to modify database |
| SRS\_011 | Insertion of Book details into the database |
| SRS\_012 | Retriving of books details from the database |
| SRS\_013 | Deletion of Book details from the database |
|  |  |
| SRS\_014 | User should be able to retrieve book download URLs from database |
| SRS\_015 | User is provided customer support to resolve queries using sockets |
| SRS\_016 | User Socket Creation :Server Socket Creation:  creating a socket with three parameters (communication domain, type, protocol) |
| SRS\_017 | Socket bind:  bind()reserves port for the socket.   It has three parameters socket identifier, struct sockaadr\_in, the port of the machine and size of the sockaddr structure |
| SRS\_018 | Socket listen:  Listening to the binded port for incoming connections keeping a backlog of 5 maximum connections |
| SRS\_019 | Socket listen:  Listening to the binded port for incoming connections keeping a backlog of 5 maximum connections |
| SRS\_020 | Communication is established Between Admin and User |
| SRS\_021 | Exec family is used to provide customer -   Based on user requirement communication is established |

**3. FLOW DIAGRAM:**



**Design Overview:**

**SOCKET:**

Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while the other socket reaches out to the other to form a connection. The server forms the listener socket while the client reaches out to the server

**BIND:**

When a socket has both an IP address and a port number it is said to be 'bound to a port', or 'bound to an address'. A bound socket can receive data because it has a complete address. Binding is the process of allocating a port number to a socket.

**LISTEN:**

The listen () function applies only to stream sockets. It indicates a readiness to accept client connection requests, and creates a connection request queue of length backlog to queue incoming connection requests. Once full, additional connection requests are rejected**.**

**ACCEPT:**

The accept () call is used by a server to accept a connection request from a client. When a connection is available, the socket created is ready for use to read data from the process that requested the connection. The call accepts the first connection on its queue of pending connections for the given socket.

**CONNECT:**

The connect () call on a stream socket is used by the client application to establish a connection to a server. The server must have a passive open pending. A server that is using sockets must successfully call bind () and listen () before a connection can be accepted by the server with accept ().

**SEND/REC:**

The server uses the socket that is returned from the accept () call.

These functions return the amount of data that was sent or received. Because stream sockets send and receive information in streams of data, it can take more than one send () or rec () to transfer all of the data. It is up to the client and the server to agree on some mechanism to signal that all of the data has been transferred.

**EXEC:**

The execv function is most commonly used to overlay a process image that has been created by a call to the fork function. file. is the filename of the file that contains the executable image of the new process. argv is a pointer to an array of pointers to null-terminated character strings. exec is a functionality of an operating system that runs an executable file in the context of an already existing process, replacing the previous executable.

**DATABASE:**

A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS).

**SIGNALS:**

A signal is a software generated interrupt that is sent to a process by the OS because of when user press ctrl-c or another process tell something to this process. There are fix set of signals that can be sent to a process. signal is identified by integers.

**4.CONCLUSION:**

Digital book downloading using URLs provided in database which can be updated by Admin is successfully done by user. Sockets are used to provide customer support and signals to control and exit the code.