Problem Statement

**CONCURRENT/MULTITHREADING APPLICATION FOR REAL TIME BOOKING**

Group No. 6

**Design Document**

1. INTRODUCTION:

* It is a web-based application. The main purpose of “Concurrent/Multithreading Application for Real Time Booking” is to provide a convenient way for a customer to book cars and hotels for traveling and staying purposes. The objective of this project is to develop a system that automates the processes and activities of a travel agency. In this project, we will make an easier task of booking cars and hotels. In the present system a customer has to approach various agencies to find details of places and to book tickets. This often requires a lot of time and effort.
* This application is developed by using the sockets to exchange the information between the admin and user and by using the multithreading to enable more than one user at a time without requiring multiple copies of the program running on the computer also using the database to enabling users to store data in a structured form and then access it.
* Each device connected to the Internet has a unique IP address which other machines use to find the device. IP addresses such as 192.168.1.1 (in IPv4).

1. PURPOSE

The purpose of this project is to get a solid grasp on the fundamentals of the Socket. Writing such an application in C++ gives a basic understanding on how the client - server architecture works.

* 1. FUNCTIONALITIES OF THE SYSTEM:

1.Client:

In client terminal first it will login through credentials after giving correct credentials one can see the packages and do bookings according to their respective locations and also we can modify the locations. We can easily search and cancel the reservations.

2.Server:

The server process waits for requests from a client. To do this, the server first establishes (binds) an address that clients can use to find the server. When the address is established, the server waits for clients to request a service. By using threads we are multiple clients at a time.

      1.3 OPERATING ENVIRONMENT:

Operating environment for implementing Concurrent/Multithreading Application for Real Time Booking are:

* Client/server system
* Operating system: Linux
* Platform: Ubuntu/C++

1. SOFTWARE REQUIREMENTS:

* Program: Concurrent/Multithreading for Real Time Booking.
* Purpose: connect with server, Creating functions for bookings, receive client response, accessing Multiple clients at a time

*SR1*: Login credentials:

          Register if a new client and login using the same credentials.

*SR2*: Client-side socket program:

To initiate the communication. Client must create a socket with the server port no.

*SR3*: Establish connection with server:

            We will be assigning default port no. to the server and connection will be established if the default port no. is read by the server using sig\_handler (signal handler).

*SR4*: Send and receive the request:

                    Use the recvfrom () and sendto () file descriptor to send and receive the data from server.

* Program: Concurrent/Multithreading for Real Time Booking.
* Purpose: Accept request from clients, resolve hostnames to IP address

*SR1*:   Structure of Server

Defining a pre-define structure sockaddr, sockaddr\_in which

include predefine elements like sin\_family, sin\_port, sin\_addr.

*SR2*: Server-side socket program

   Server will passively wait for the response from the client. Server is a

Passive Socket.

*SR3*: Bind and listen the local address with socket:

In this function the local address will be bind with the server socket and after binding the listen function will take socketed with maximum number clients it will listen on.

*SR4*:

Trigger the resolver code for sending message to local name server.

*SR5*: Send and receive the request with the help of signals SIGINT(), SIGTSTP().

Use the sendto () and recvfrom () file descriptor to send and receive the

data from client.

SR6: Read the functions of Admin and Customer.

Admin Menu: In this module, We can perform different operations like to

enter new trip, display trip, display reservation, update trip and exit.

Customer Menu: In this module, we can perform different operations like

new reservation, Confirm reservation and cancel reservation and exit

3.   UNIT TEST:

1. Client:

SR1-UT1: Provide valid username and password.

SR2-UT2: Create a valid socket to initiate the communication

SR3-UT3: If the signo==SIGINT function fails then the client must receive an error message of connection failure.

SR4-UT4: If sendto () or recvfrom () function fails then the client won’t be able to send or receive the data

1. Server:

SR1-UT1: While defining the structure mention a valid elements.

SR2-UT2: The port no. in the client side and server side should be same if not connection not establish.

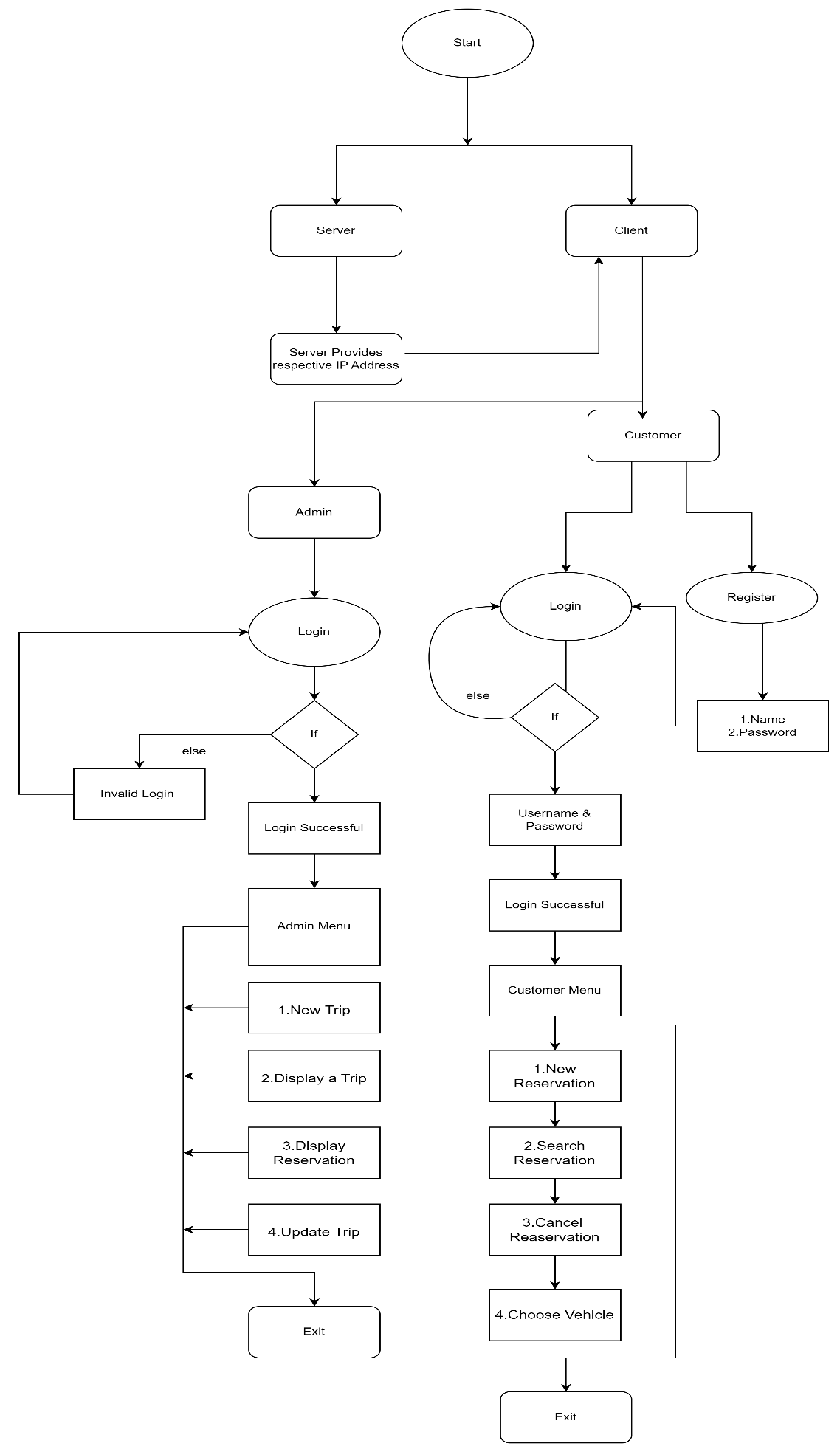
SR3-UT3: The bind() and listen() should bind the client and server if fail must return the Binding error while listen() function must listen on particular port number.

SR4-UT4: If the binding fails should return an error of Binding failed.

SR5-UT5:  Send and receive the request

                   Use the sendto () and recvfrom () file descriptor to send and receive the data from server.

**Flow Diagram:**



1.Start

This is the start block which indicates the start of the program.  
which will accept the client and server credentials (like the username and password). On validation of these credentials the system will allow the client and server to further communicate else if the credentials are wrong it will display an error message indicating “Invalid Credentials”.

1. Server Login

This is the module where server socket is created and listens to the client with the help of signals.

It provides the respective IP address and connects with the client.

1. Client Login

This is the module used for the client login where the client, if new, has to register with a username and password and then login by entering the same credentials (username and password). If already registered the client can login with the credentials.

1. Login Credentials

In this module the credentials entered by the server/client are then validated by the system. If the server/client enters valid credentials then it will move to the further step else the system will prompt the server/client with an error message.

5. Admin Login

In this module, as we know admins are fixed initially we take login attempts

is equal to zero and by using if conditions we can check admin login.

6. Admin Menu

In this module, we can perform different operations like to enter new trip,

display trip, display reservation, update trip and exit.

7. Customer Login

In this module, we have multiple number of users, so we have used text file to save the existing accounts. If the user entered details matches with the details in file then we can proceed to dashboard. If the account is new, then by entering details like name, password then we can create a account for customer

8. Customer Menu

In this module, we can perform different operations like mew reservation, confirm reservation and cancel reservation and exit

1. Server provides respective IP address

When a domain name corresponding to the client entered domain name is found, the server will give the respective IP address to the client. 

1. Display error message
2. End

This ensures that the program has terminated.

CONCLUSION:

This web application was successfully created and stored all the travel admin tourism packages booking, creation managing and tour details into the database using this application. This system thus provides an easy way to automate all the functionalities.