[redacted]

High Level Design & Low Level Design

INDEX

1. INTRODUCTION:

The introduction of the software requirement specification provides an overview of the entire software. The entire design with overview description purpose, scope, tools used and basic description. The aim of this document is to gather, analyze and give an in-depth insight into the complete Remote Calculate by defining the problem statement in detail. The detailed requirements of the Remote Calculate are provided in this document

1.1 Purpose: Remote Calculator relieves the user of the need to do mental operations and of the need not to rely on paper.

1.2 Intended Audience: - This document is intended to be read by Client.

2. OVERALL DESCRIPTION:

Connection established between server and client, Then the login is successful in admin mode of server side then only the further process will continue otherwise it will disconnected. After successful login, Server will takes the inputs from the client like number of arguments, operations so on. Server performs some operations like addition, subtraction, so on. After operation to be done then server sends the result to the client.

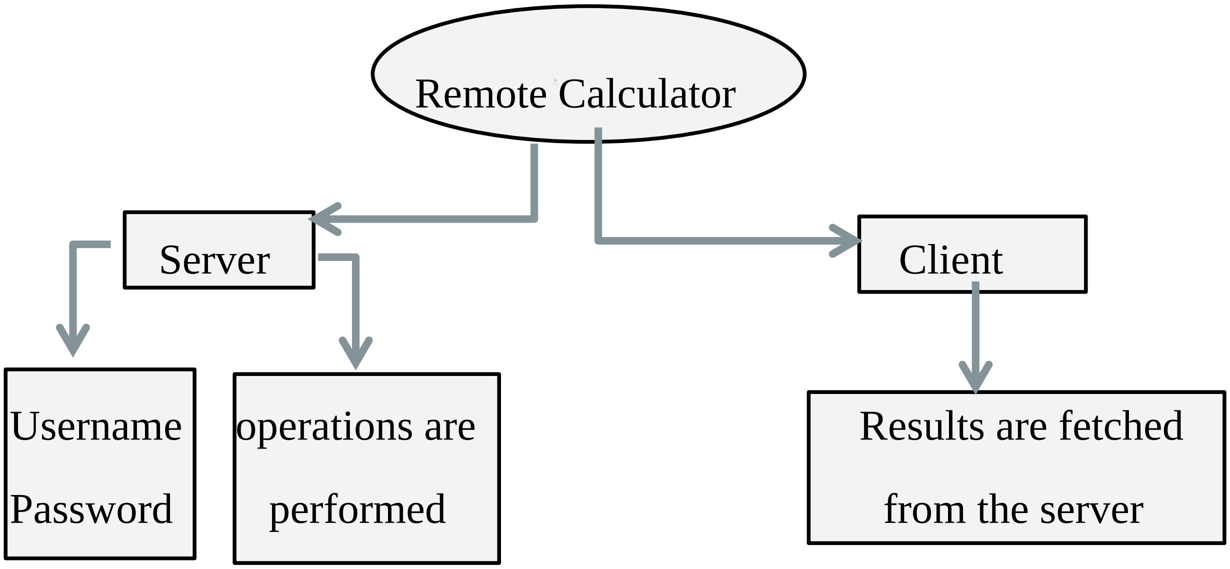
2.1 Assumptions and Dependency:-

- User should have the latest version of Ubuntu Linux installed.
- User should have minimum 4GB RAM.
- The service is available on a desktop or laptop

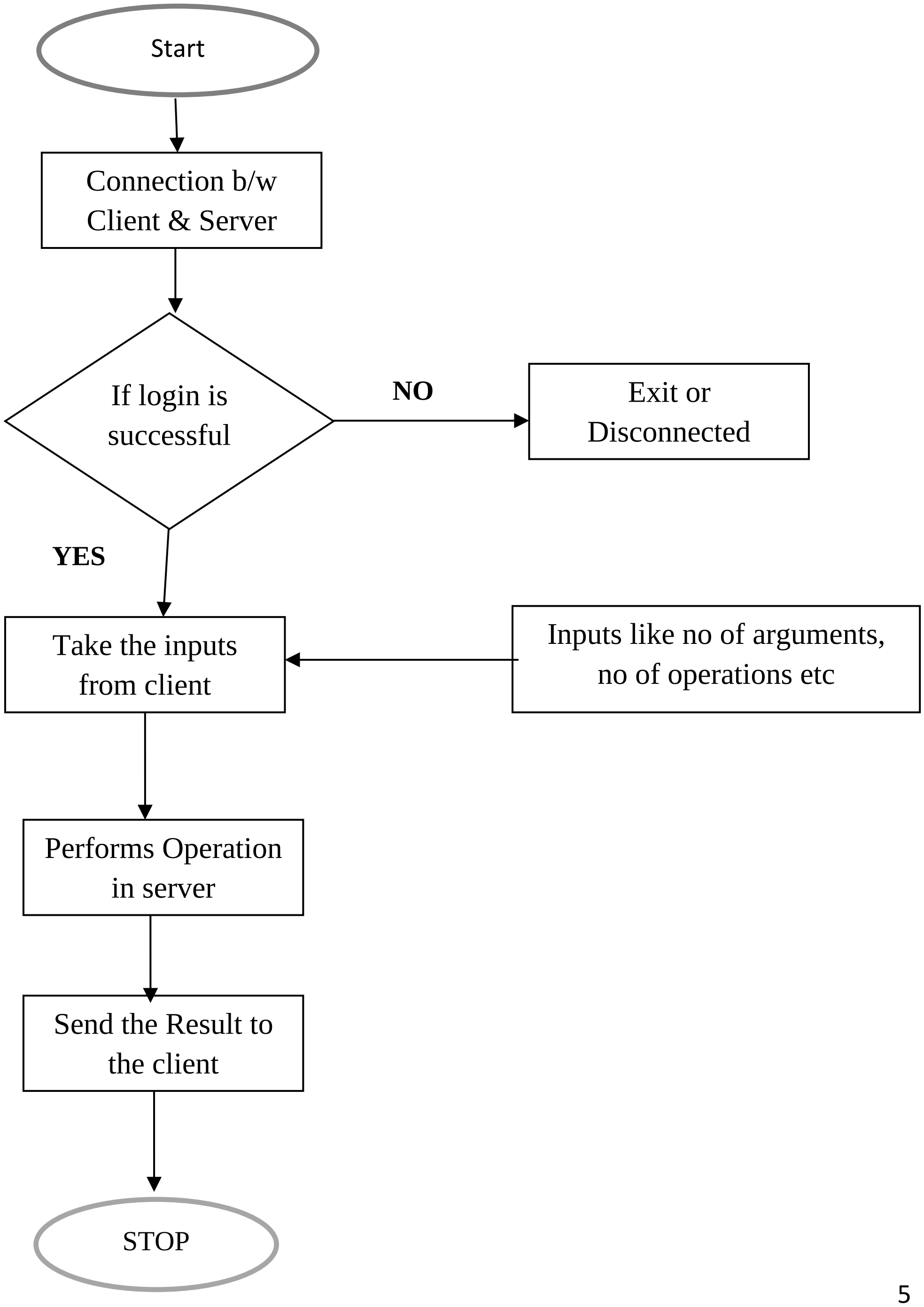
3.DESIGN OVERVIEW

3.1 Data flow diagram

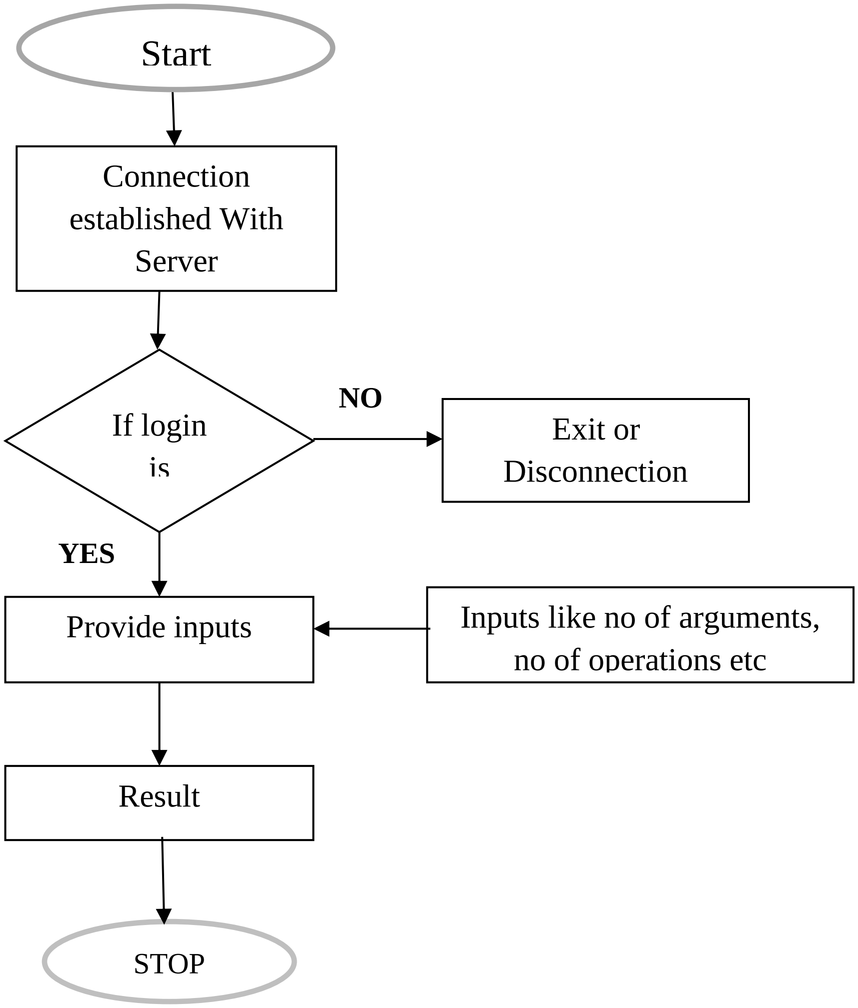
Level 0



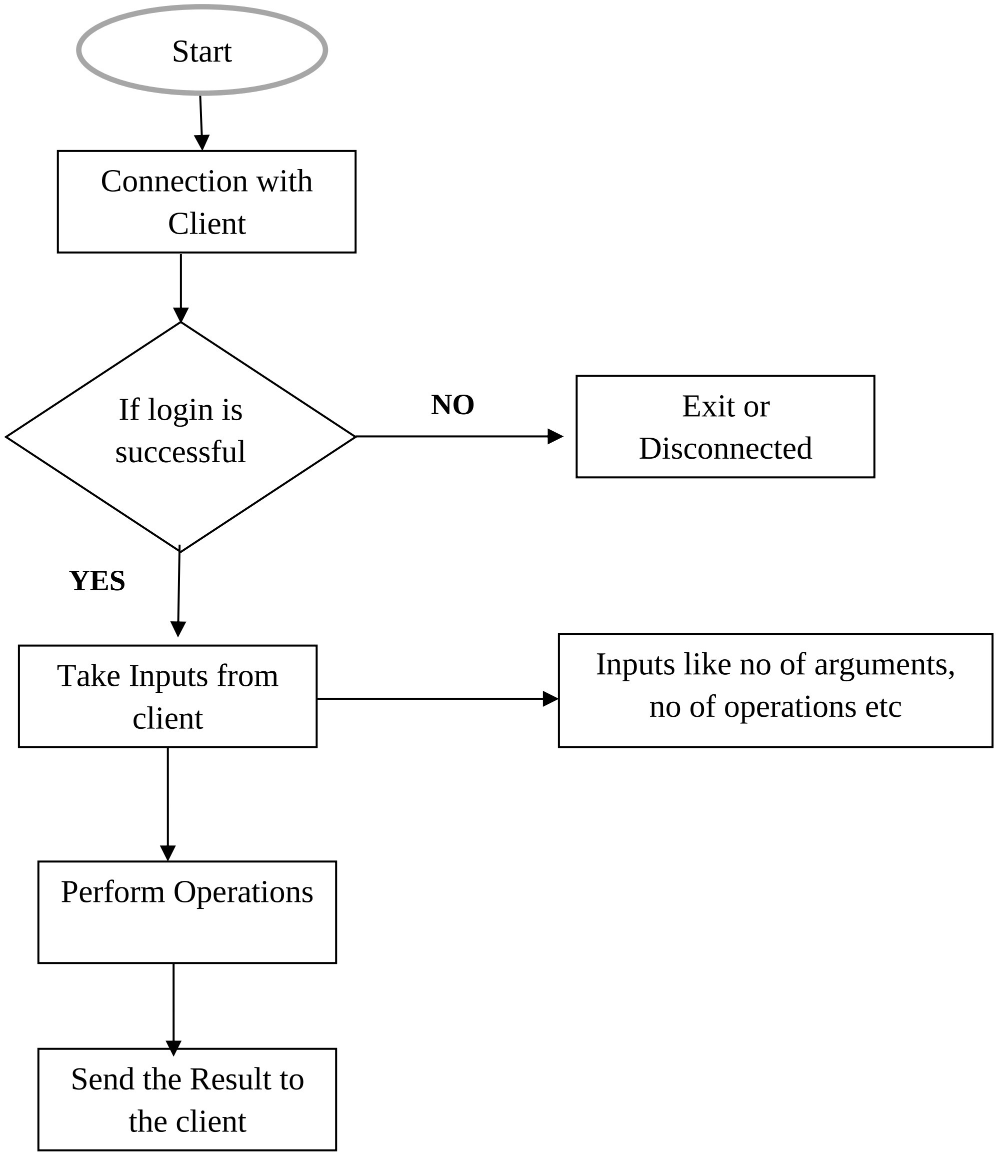
3.2 Flow chart for client server overview


Start
Connection b/w Client & Server
If login is successful
NO
Exit or Disconnected
YES
Take the inputs from client
Inputs like no of arguments, no of operations etc
Performs Operation in server
Send the Result to the client
STOP

3.3 Flow chart for client side

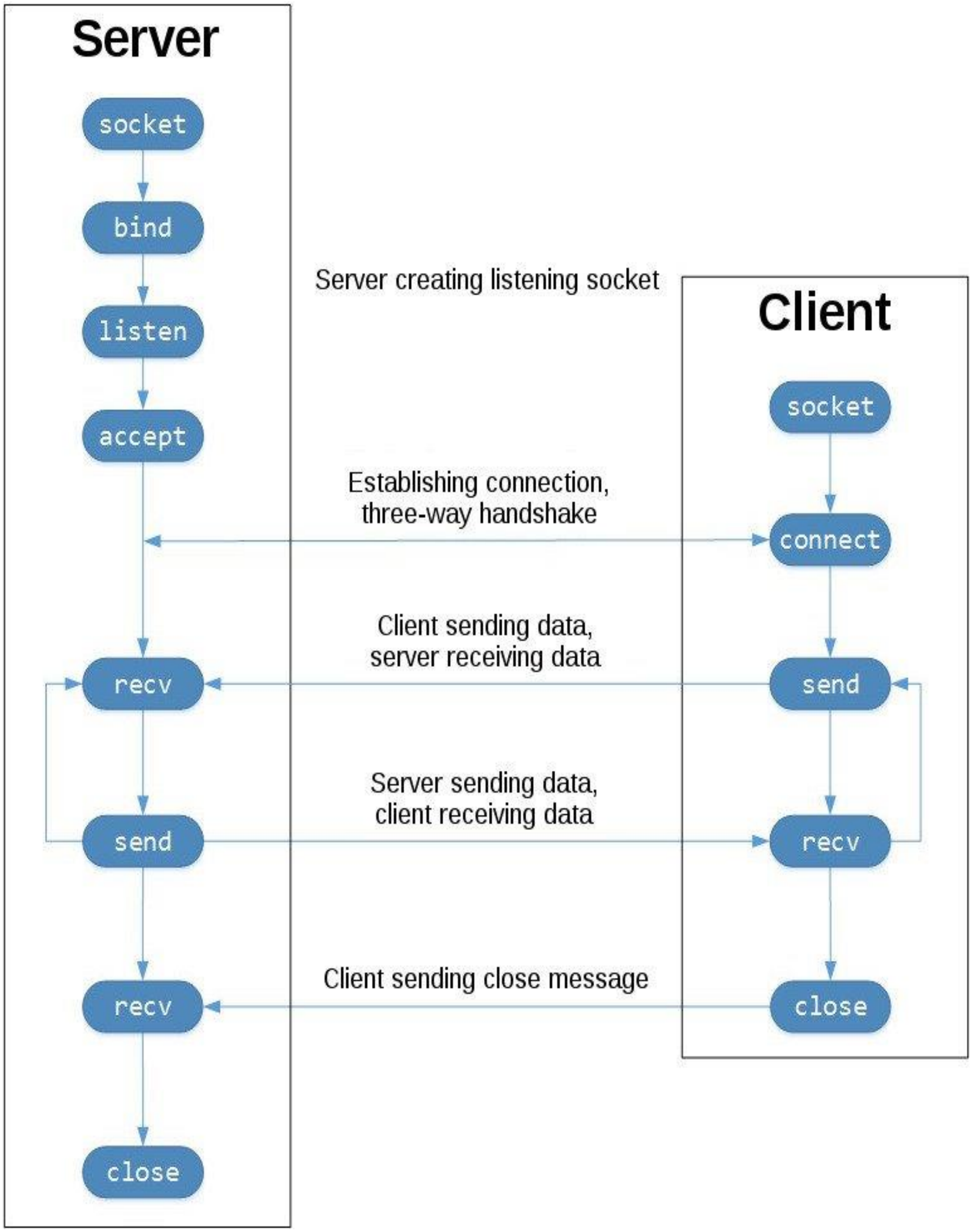

Start
Connection established With Server
If login is
NO
Exit or Disconnection
YES
Provide inputs
Inputs like no of arguments, no of operations etc
Result
STOP

3.4 Flow chart for server side



4. SYSTEM TURE:

STOP



4.1 FUNCTIONS:

4.1.1 LOGIN functionality

Admin Mode (Server side) which takes

- Login
  - ✓ Username
  - ✓ Password
- Register
- Exit

4.1.2 CALCULATION

Admin Mode (Server Side)

It is required to perform operations of addition, multiplication, subtraction, division with postfix, prefix and infix expression for calculation.

4.1.3 ARGUMENTS

Admin Mode (Server Side)

To add the number of arguments to be inputted followed by numbers using a colon, as a separator for the arguments as well as the operands.

4.1.4 OPERATIONS

Admin Mode (Server side)

It is to add the number of operations to be performed like for example: Addition, Subtraction, Multiplication, Division.

2.1.4.1 Input the operators using (;) semicolon as separators.

2.1.4.2 It is required to formulate the problem to be calculated to get the required output.

4.1.5 RESULT

User Mode (Client Side)

After the required calculation processed, It is required to fetch the results from the Server and provide it to the Client as per the request.

4.2 STRUCTURE

First enable the connection between server and the client.

SERVER SIDE: LOGIN

- ✓ USERNAME
- ✓ PASSWORD

REGISTER
EXIT

After successful login, Enter the arguments with operands.
Now select the operators from (+,-,/,*) by entering the operation using (;) semicolon as separator.
The answer is then provided by the server to the client.

CLIENT SIDE: The answer is fetched from the server and the result is displayed in the client.

5. SYSTEM FEATURES AND REQUIREMENTS:

5.1 FUNCTIONALITY:

5.1.1 LTR_01-> **Login Functionality:** In admin mode of server side, client should login by providing username and password.

5.1.2 LTR_02-> **Add the number of arguments to be inputted followed by the numbers:** In admin mode of server side, client should provide the inputs like number of arguments followed by the numbers.

5.1.3 LTR_03-> **Add the number of operations to be performed:** In admin mode of server side, Client should gives the input to the server like number of operation to be performed.

5.1.4 LTR_04-> **Input the operator using ; separator:** In admin mode of server side, Client should gives the input to the server like operator which is separated by semicolon.

5.1.5 LTR_05-> **Formulate the problem to be calculated:** In user mode of server side, client should specifies which operation to be performed.

5.1.6 LTR_06-> **Perform the required calculations:** In User mode of server side, the operation is performed by the server.

4.1.7 LTR_07-> **Fetch the results from the server:** In user mode of client side, server send the results to the client.

5.2 SYSTEM REQUIREMENTS:

- Linux based operating system with GCC compiler
- Coding language: C Programming
- System: Intel IV 2.4 GHz
- Hard Disk: 100 GB
- Ram: 4 GB

5.3 SYSTEM FEATURES:

- Supportability: The system is easy to maintain.

- Design Constraints: The system is built using only C language. So the constraints of C language are applied to it.

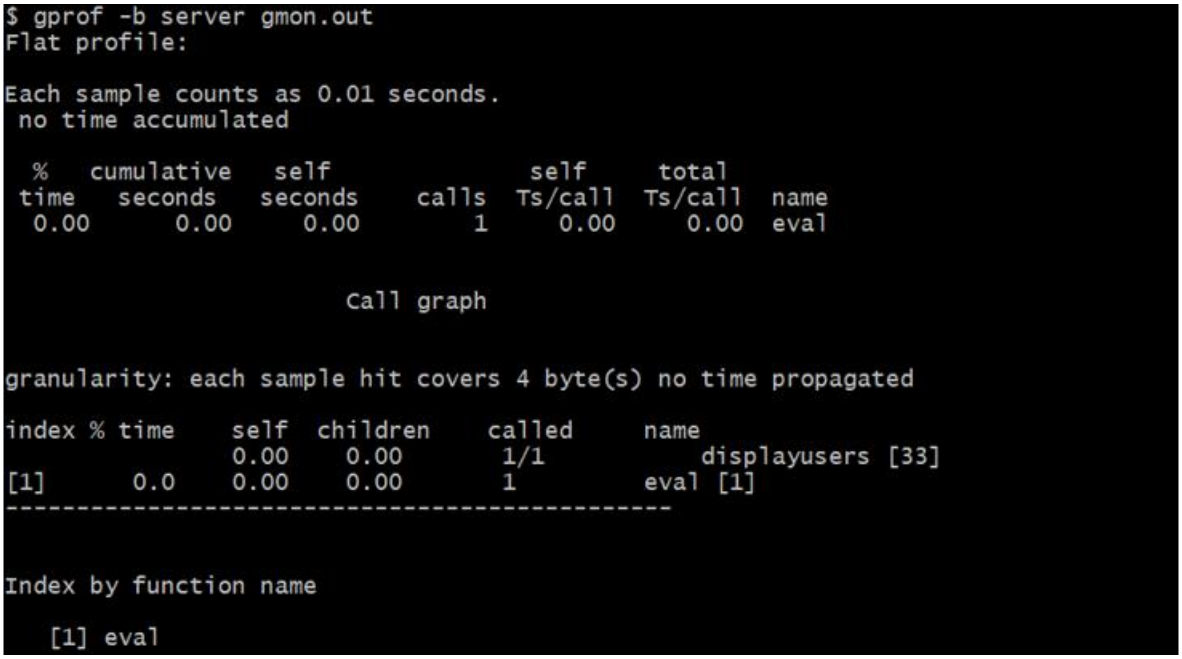
- Usability: In this project, we have developed a remote calculator for exact real number computation and performed the calculations. It is clear that it relieves the user of the need to do mental operations and of the need not to rely on paper. Here, the client sends the request to the server in simple arithmetic equations and after processing the request, the server will respond back with the answer to the equation.

- Reliability & Availability: The system is available when the user is requested for service. The system is available 24/7.

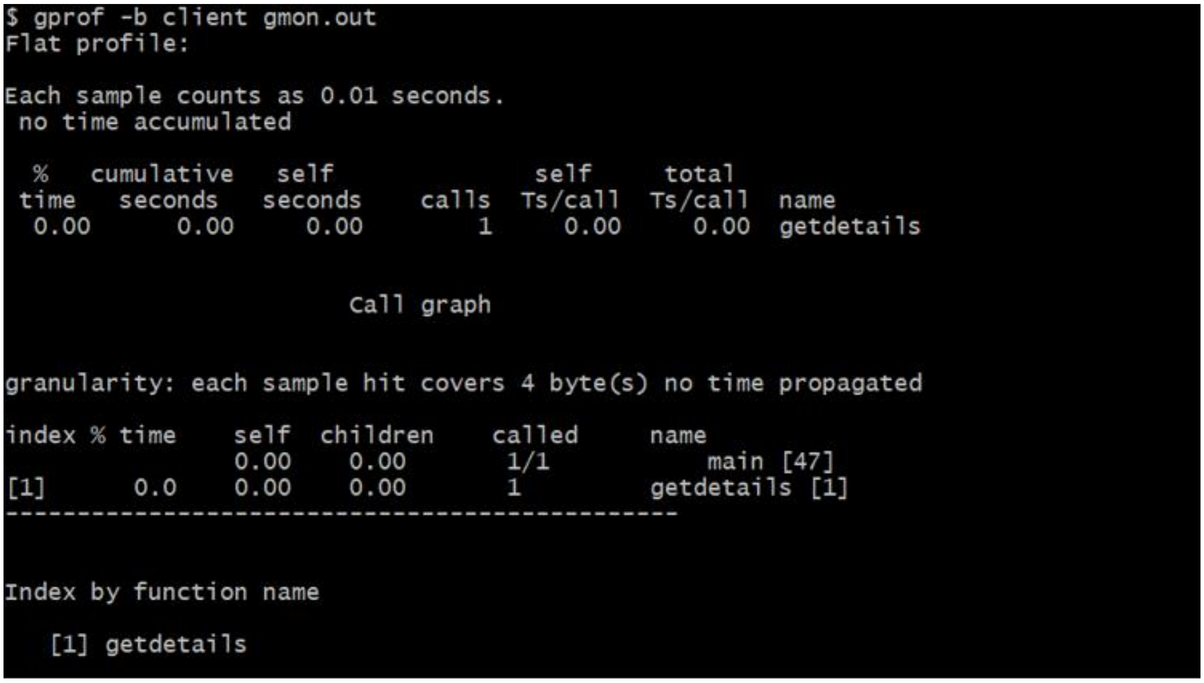
- Performance: The system will work on the user's terminal.

6.TOOLS REPORT

6.1 Gprof

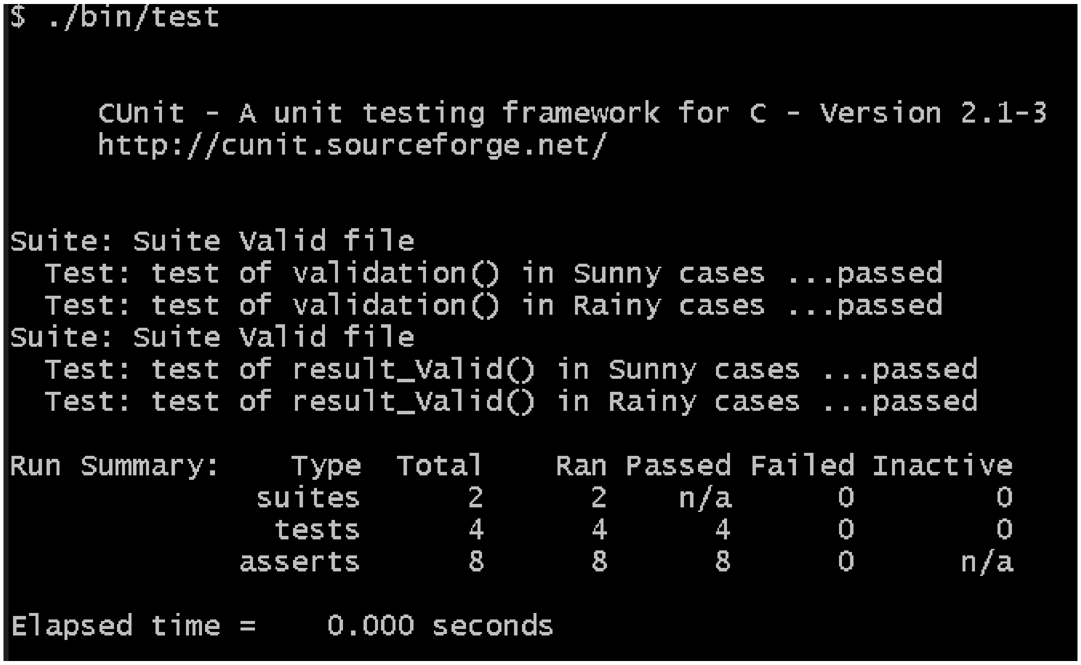


```
$ gprof -b server gmon.out
Flat profile:

Each sample counts as 0.01 seconds.
 no time accumulated

  %   cumulative   self              self     total
 time   seconds   seconds    calls  Ts/call  Ts/call  name
  0.00      0.00     0.00        1     0.00     0.00  eval


                        Call graph


granularity: each sample hit covers 4 byte(s) no time propagated

index % time    self  children    called     name
                0.00    0.00       1/1           displayusers [33]
[1]      0.0    0.00    0.00       1         eval [1]
-----------------------------------------------


Index by function name

   [1] eval
```



```
$ gprof -b client gmon.out
Flat profile:

Each sample counts as 0.01 seconds.
 no time accumulated

  %   cumulative   self              self     total
 time   seconds   seconds    calls  Ts/call  Ts/call  name
  0.00      0.00     0.00        1     0.00     0.00  getdetails


                        Call graph


granularity: each sample hit covers 4 byte(s) no time propagated

index % time    self  children    called     name
                0.00    0.00       1/1           main [47]
[1]      0.0    0.00    0.00       1         getdetails [1]
-----------------------------------------------


Index by function name

   [1] getdetails
```

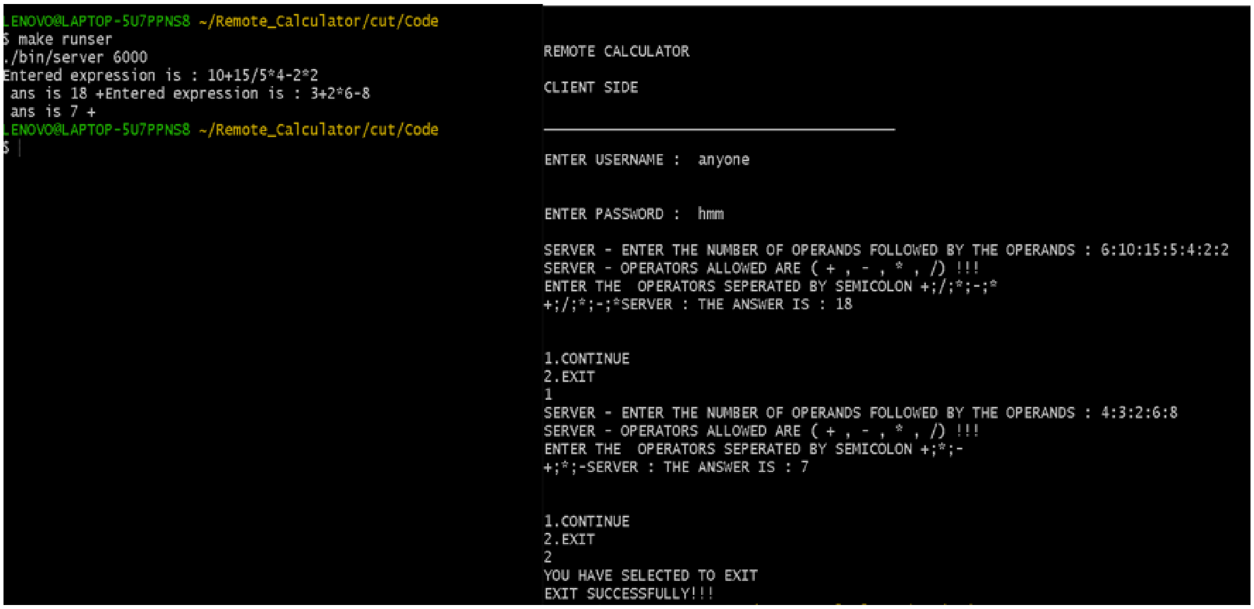
6.2 CUnit



```
$ ./bin/test


     CUnit - A unit testing framework for C - Version 2.1-3
     http://cunit.sourceforge.net/


Suite: Suite Valid file
  Test: test of validation() in Sunny cases ...passed
  Test: test of validation() in Rainy cases ...passed
Suite: Suite Valid file
  Test: test of result_Valid() in Sunny cases ...passed
  Test: test of result_Valid() in Rainy cases ...passed

Run Summary:    Type  Total    Ran Passed Failed Inactive
              suites      2      2    n/a      0        0
               tests      4      4      4      0        0
             asserts      8      8      8      0      n/a

Elapsed time =    0.000 seconds
```

7. TESTING

7.1 Unit testing



```
LENOVO@LAPTOP-5U7PPNS8 ~/Remote_Calculator/cut/Code
$ make runser
./bin/server 6000
Entered expression is : 10+15/5*4-2*2
 ans is 18 +Entered expression is : 3+2*6-8
 ans is 7 +
LENOVO@LAPTOP-5U7PPNS8 ~/Remote_Calculator/cut/Code
$ |
```

```
REMOTE CALCULATOR

CLIENT SIDE

________________________________________

ENTER USERNAME :  anyone


ENTER PASSWORD :  hmm

SERVER - ENTER THE NUMBER OF OPERANDS FOLLOWED BY THE OPERANDS : 6:10:15:5:4:2:2
SERVER - OPERATORS ALLOWED ARE ( + , - , * , /) !!!
ENTER THE  OPERATORS SEPERATED BY SEMICOLON +;/;*;-;*
+;/;*;-;*SERVER : THE ANSWER IS : 18


1.CONTINUE
2.EXIT
1
SERVER - ENTER THE NUMBER OF OPERANDS FOLLOWED BY THE OPERANDS : 4:3:2:6:8
SERVER - OPERATORS ALLOWED ARE ( + , - , * , /) !!!
ENTER THE  OPERATORS SEPERATED BY SEMICOLON +;*;-
+;*;-SERVER : THE ANSWER IS : 7


1.CONTINUE
2.EXIT
2
YOU HAVE SELECTED TO EXIT
EXIT SUCCESSFULLY!!!
```

8. REQUIREMENT TRACEABILITY MATRIX(RTM)

Activity	Expected Start Date	Expected end date	Actual Start Date	Actual end date
SRS Release	13th October	18th October	13th October	19th October
Design	14th October	14th October	14th October	14th October
Coding	15th October	15th October	15th October	15th October
UT Plan	16th October	17th October	16th October	17th October
UT	16th October	17th October	16th October	17th October
IT Plan	17th October	18th October	17th October	18th October
IT	18th October	18th October	18th October	18th October
Sprint Demo	19th October	19th October	19th October	19th October