LANGUAGE TRANSLATOR

12.10.2022-18.10.2022

Sprint 2

Group 2

INDEX

Overview

Language Translator :

It is a language translator application in which the server has admin access where it can perform various functions like adding new language, adding meaning, modifying and deleting words in the database. The application then correlates this data and returns the respective translated words to the clients based on the input target and source language.

If the words are not present in the database then it gives the error message to the user that "words not found".

Moreover, the application has the capability to generate files which have the information of various word meanings in different languages.

Goals

This project aims at creating a language translator for a client having a required set of features. It aims at smooth functioning and connections between the client and server. All clients should be able to access certain functions.

Purpose

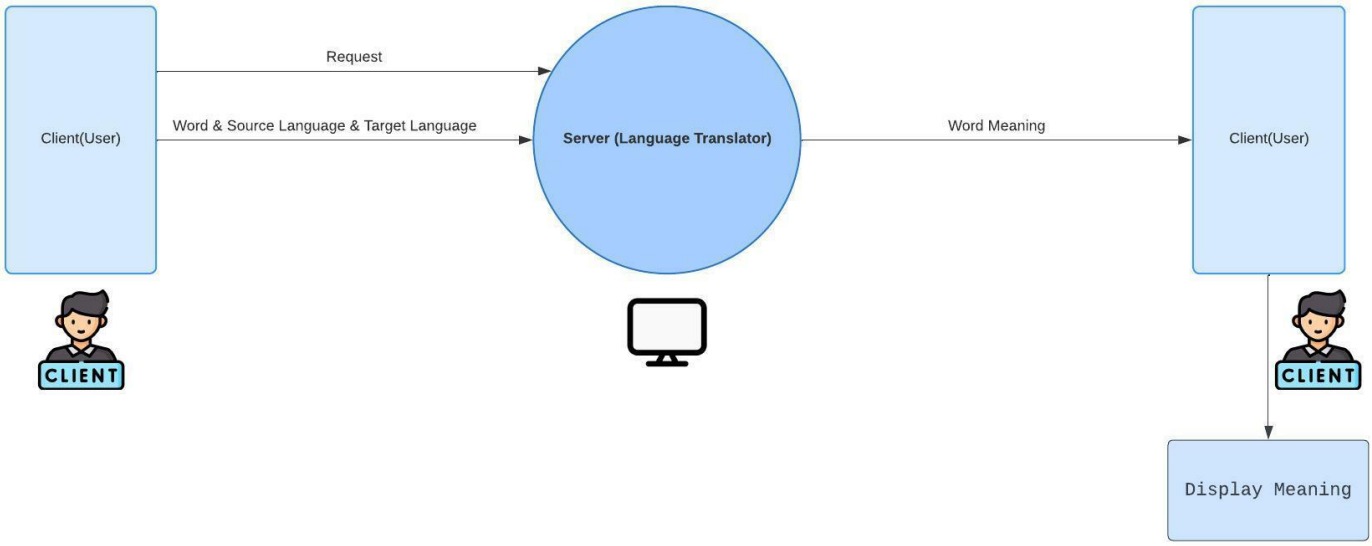
The purpose of this document is to show the requirements for the language translator application, in which clients/users can get the desired translation of the word they provide.

Target audience

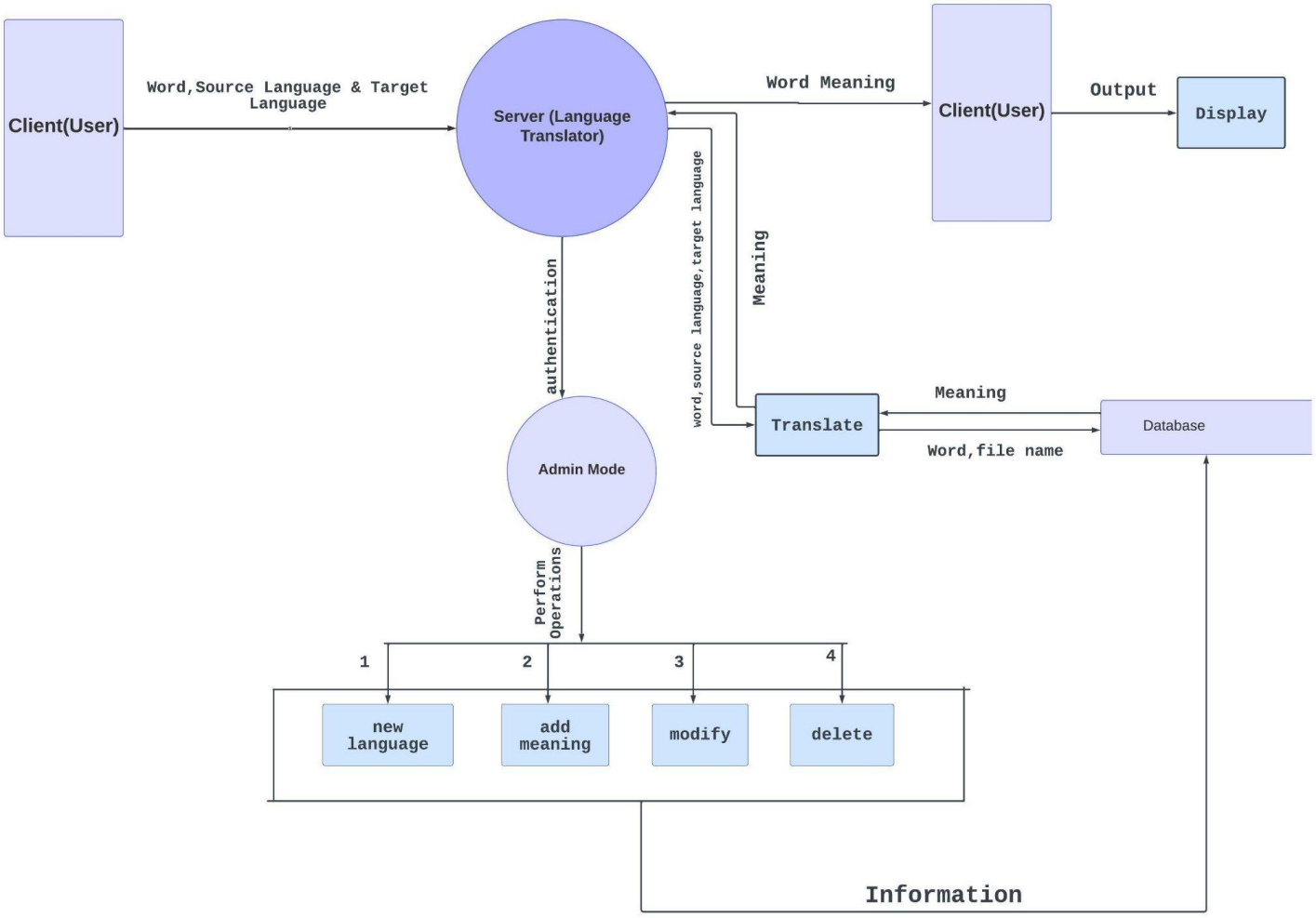
The target audience are the clients who have the rights to translate their language. Also, the server has the authentication rights for themselves and can perform and manage various functionalities.

Design Overview

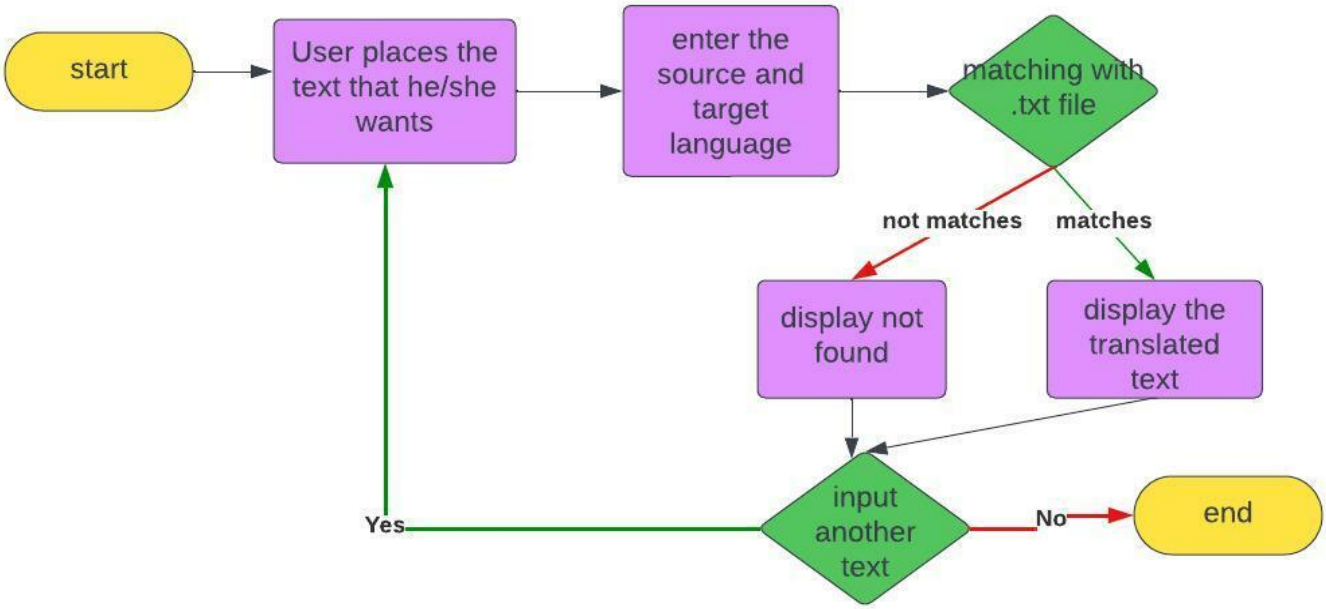
Data Flow Diagram Level 0:



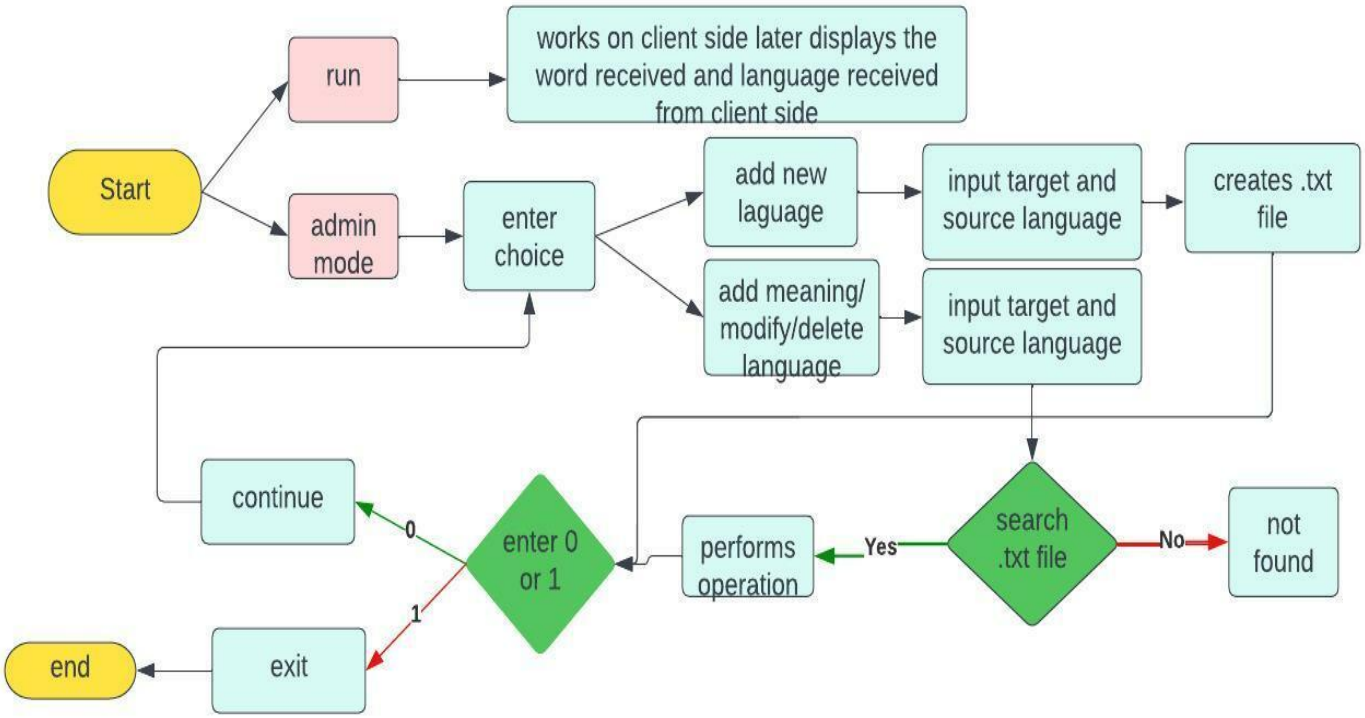
Data Flow Diagram Level 1:



Flowchart for client:



Flowchart for server:



System Architecture

Functions:

SERVER FUNCTIONS:

Server side authentication:

Server program should start with an authentication

translate(): When the client gives the input word, the target and source language, this function matches the word from the text file, if the word is available in the text file , it returns the word meaning.

writeToFile(): This function is used to create new language and also add new meanings in the text file.

modify(): This function is used to replace the meaning of the words that were incorrectly written previously.

delete(): This function deletes the word meaning that is no longer required. It basically deletes the entire line containing that word and in the output screen displays the remaining contents of the text file.

CLIENT FUNCTIONS:

Client side program starting:

The user should start the client program to connect the Server whenever he wants.

Client side connection request:

Client should request a connection to the server on port 8028

Client side user environment:

The user is now ready to give the the word for the language conversion

Client side quit:

The user uses the subcommand exit to terminate the connection.

Tools Report

MAKE FILE:

```
#compiler
GCC = gcc
#output file
LFLAGS = -o
#binary file directory
BIN = ../bin
#source file directory
SRC = ../src
#object file directory
OBJ = ../obj


#compile all source file
all: buildServer buildClient

buildServer: $(OBJ)/server.o $(OBJ)/translate.o $(OBJ)/modify.o $(OBJ)/delete.o $(OBJ)/writetofile.o
	$(GCC) $(LFLAGS) $(BIN)/server $(OBJ)/server.o $(OBJ)/translate.o $(OBJ)/modify.o $(OBJ)/delete.o $(OBJ)/writetofile.o

buildClient: $(OBJ)/client.o
	$(GCC) $(LFLAGS) $(BIN)/client $(OBJ)/client.o

$(OBJ)/server.o: $(SRC)/server.c
	$(GCC) -c  $(LFLAGS) $(OBJ)/server.o $(SRC)/server.c

$(OBJ)/client.o: $(SRC)/client.c
	$(GCC) -c $(LFLAGS) $(OBJ)/client.o $(SRC)/client.c

$(OBJ)/translate.o: $(SRC)/translate.c
	$(GCC) -c $(LFLAGS) $(OBJ)/translate.o $(SRC)/translate.c

$(OBJ)/modify.o: $(SRC)/modify.c
	$(GCC) -c $(LFLAGS) $(OBJ)/modify.o $(SRC)/modify.c

$(OBJ)/delete.o: $(SRC)/delete.c
	$(GCC) -c $(LFLAGS) $(OBJ)/delete.o $(SRC)/delete.c
```

```

$(OBJ)/writetofile.o: $(SRC)/writetofile.c
	$(GCC) -c $(LFLAGS) $(OBJ)/writetofileo $(SRC)/writetofile.c



#run server
server:
	./$(BIN)/server

#run client
client:
	./$(BIN)/client

clean:
	rm $(BIN)/server
	rm $(BIN)/client
	rm $(OBJ)/server.o
	rm $(OBJ)/client.o
	rm $(OBJ)/translate.o
	rm $(OBJ)/modify.o
	rm $(OBJ)/delete.o
	rm $(OBJ)/writetofile.o

```

Gcov report:

```
harsh@DESKTOP-NJGS31F ~/sprint_2/src
$ gcc -fprofile-arcs -ftest-coverage writetofile.c delete.c translate.c modify.c server.c -o server

harsh@DESKTOP-NJGS31F ~/sprint_2/src
$ gcc -fprofile-arcs -ftest-coverage client.c -o client

harsh@DESKTOP-NJGS31F ~/sprint_2/src
$ gcov client.gcno
File 'client.c'
Lines executed:82.61% of 46
Creating 'client.c.gcov'

Lines executed:82.61% of 46

harsh@DESKTOP-NJGS31F ~/sprint_2/src
$ gcov server-delete.gcno
File 'delete.c'
Lines executed:87.93% of 58
Creating 'delete.c.gcov'

Lines executed:87.93% of 58

harsh@DESKTOP-NJGS31F ~/sprint_2/src
$ gcov server-modify.gcno
File 'modify.c'
Lines executed:97.14% of 35
Creating 'modify.c.gcov'

Lines executed:97.14% of 35
```

```
harsh@DESKTOP-NJGS31F ~/sprint_2/src
$ gcov server-writetofile.gcno
File 'writetofile.c'
Lines executed:97.67% of 43
Creating 'writetofile.c.gcov'

Lines executed:97.67% of 43


harsh@DESKTOP-NJGS31F ~/sprint_2/src
$ gcov server-translate.gcno
File 'translate.c'
Lines executed:52.17% of 23
Creating 'translate.c.gcov'

Lines executed:52.17% of 23
```

Gprof report:

```
$ gcc -Wall -pg server.c translate.c writetofile.c delete.c modify.c -o server
server.c: In function 'main':
server.c:35:17: warning: 'choice' may be used uninitialized [-Wmaybe-uninitialized]
   35 |                 gets(choice);
      |                 ^~~~~~~~~~~~
writetofile.c: In function 'writetofile':
writetofile.c:53:29: warning: unused variable 'i' [-Wunused-variable]
   53 |                         int i;
      |                             ^

$ gprof ./server.exe
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        2     0.00     0.00  translate

 %         the percentage of the total running time of the
time       program used by this function.

cumulative a running sum of the number of seconds accounted
 seconds   for by this function and those listed above it.

 self      the number of seconds accounted for by this
seconds    function alone.  This is the major sort for this
           listing.

calls      the number of times this function was invoked, if
           this function is profiled, else blank.

 self      the average number of milliseconds spent in this
ms/call    function per call, if this function is profiled,
           else blank.

 total     the average number of milliseconds spent in this
ms/call    function and its descendents per call, if this
           function is profiled, else blank.

name       the name of the function.  This is the minor sort
           for this listing. The index shows the location of
           the function in the gprof listing. If the index is
           in parenthesis it shows where it would appear in
           the gprof listing if it were to be printed.
```

```
Copyright (C) 2012-2022 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.


                     Call graph (explanation follows)


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

index % time    self  children    called     name
                0.00    0.00       2/2          main [51]
[1]     0.0    0.00    0.00       2          translate [1]
-----------------------------------------------

 This table describes the call tree of the program, and was sorted by
 the total amount of time spent in each function and its children.

 Each entry in this table consists of several lines.  The line with the
 index number at the left hand margin lists the current function.
 The lines above it list the functions that called this function,
 and the lines below it list the functions this one called.
 This line lists:
     index      A unique number given to each element of the table.
                Index numbers are sorted numerically.
                The index number is printed next to every function name so
                it is easier to look up where the function is in the table.

     % time     This is the percentage of the `total' time that was spent
                in this function and its children.  Note that due to
                different viewpoints, functions excluded by options, etc,
                these numbers will NOT add up to 100%.

     self       This is the total amount of time spent in this function.

     children   This is the total amount of time propagated into this
                function by its children.

     called     This is the number of times the function was called.
                If the function called itself recursively, the number
                only includes non-recursive calls, and is followed by
                a `+' and the number of recursive calls.

     name       The name of the current function.  The index number is
                printed after it.  If the function is a member of a
                cycle, the cycle number is printed between the
                function's name and the index number.
```

For the function's parents, the fields have the following meanings:

self This is the amount of time that was propagated directly from the function into this parent.

children This is the amount of time that was propagated from the function's children into this parent.

called This is the number of times this parent called the function `/' the total number of times the function was called. Recursive calls to the function are not included in the number after the `/'.

name This is the name of the parent. The parent's index number is printed after it. If the parent is a member of a cycle, the cycle number is printed between the name and the index number.

If the parents of the function cannot be determined, the word `<spontaneous>' is printed in the `name' field, and all the other fields are blank.

For the function's children, the fields have the following meanings:

self This is the amount of time that was propagated directly from the child into the function.

children This is the amount of time that was propagated from the child's children to the function.

called This is the number of times the function called this child `/' the total number of times the child was called. Recursive calls by the child are not listed in the number after the `/'.

name This is the name of the child. The child's index number is printed after it. If the child is a member of a cycle, the cycle number is printed between the name and the index number.

If there are any cycles (circles) in the call graph, there is an entry for the cycle-as-a-whole. This entry shows who called the cycle (as parents) and the members of the cycle (as children.) The `+' recursive calls entry shows the number of function calls that were internal to the cycle, and the calls entry for each member shows, for that member, how many times it was called from other members of the cycle.

Splint report:

```
cg83-user35@instance-1:~/Language-Translator/Harshit Sprint_2/CUT/Code$ splint -I../header ./src/server.c
Splint 3.1.2 --- 21 Feb 2021

src/server.c:1: Include file <sys/socket.h> matches the name of a POSIX
    library, but the POSIX library is not being used.  Consider using +posixlib
    or +posixstrictlib to select the POSIX library, or -warnposix to suppress
    this message.
  Header name matches a POSIX header, but the POSIX library is not selected.
  (Use -warnposixheaders to inhibit warning)
src/server.c:1: Include file <netinet/in.h> matches the name of a POSIX
    library, but the POSIX library is not being used.  Consider using +posixlib
    or +posixstrictlib to select the POSIX library, or -warnposix to suppress
    this message.
src/server.c:5: Include file <unistd.h> matches the name of a POSIX library,
    but the POSIX library is not being used.  Consider using +posixlib or
    +posixstrictlib to select the POSIX library, or -warnposix to suppress this
    message.
< Location unknown >: Field name reused:
  Code cannot be parsed.  For help on parse errors, see splint -help
  parseerrors. (Use -syntax to inhibit warning)
< Location unknown >: Previous use of
src/server.c: (in function main)
src/server.c:15:2: Return value (type char *) ignored: fgets(ct, 100, s...
  Result returned by function call is not used. If this is intended, can cast
  result to (void) to eliminate message. (Use -retvalother to inhibit warning)
src/server.c:23:9: Return value (type char *) ignored: fgets(LoginId, 1...
src/server.c:26:9: Return value (type char *) ignored: fgets(Password, ...
src/server.c:31:8: Test expression for while not boolean, type int: 1
  Test expression type is not boolean or int. (Use -predboolint to inhibit
  warning)
src/server.c:35:3: Return value (type char *) ignored: fgets(choice, 25...
src/server.c:65:15: Function memset expects arg 2 to be int gets char: '0'
  A character constant is used as an int. Use +charintliteral to allow
  character constants to be used as ints.  (This is safe since the actual type
  of a char constant is int.)
src/server.c:77:18: Function memset expects arg 2 to be int gets char: '0'
src/server.c:85:44: Function bind expects arg 3 to be socklen_t gets size_t:
                      sizeof((server))
  Types are incompatible. (Use -type to inhibit warning)
src/server.c:85:2: Return value (type int) ignored: bind(fd, (struct...
  Result returned by function call is not used. If this is intended, can cast
  result to (void) to eliminate message. (Use -retvalint to inhibit warning)
src/server.c:89:2: Return value (type int) ignored: listen(fd, 10)
src/server.c:90:18: Null storage passed as non-null param:
                      accept (..., (struct sockaddr *)NULL, ...)
  A possibly null pointer is passed as a parameter corresponding to a formal
  parameter with no /*@null@*/ annotation.  If NULL may be used for this
  parameter, add a /*@null@*/ annotation to the function parameter declaration.
```

```
parameter, add a /*@null@*/ annotation to the function parameter declaration.
  (Use -nullpass to inhibit warning)
src/server.c:90:42: Null storage passed as non-null param: accept (..., NULL)
src/server.c:94:8: Argument to exit has implementation defined behavior: 1
  The argument to exit should be 0, EXIT_SUCCESS or EXIT_FAILURE (Use -exitarg
  to inhibit warning)
src/server.c:96:8: Test expression for while not boolean, type int: 1
src/server.c:99:4: Unrecognized identifier: bzero
  Identifier used in code has not been declared. (Use -unrecog to inhibit
  warning)
src/server.c:101:4: Assignment of ssize_t to int: n = recv(in, buff, 256, 0)
  To allow arbitrary integral types to match any integral type, use
  +matchanyintegral.
src/server.c:105:9: Argument to exit has implementation defined behavior: 1
src/server.c:110:47: Possibly null storage word passed as non-null param:
                       printf (..., word, ...)
   src/server.c:109:15: Storage word may become null
src/server.c:112:58: Possibly null storage sl passed as non-null param:
                       printf (..., sl, ...)
   src/server.c:111:13: Storage sl may become null
src/server.c:114:58: Possibly null storage tl passed as non-null param:
                       printf (..., tl, ...)
   src/server.c:113:13: Storage tl may become null
src/server.c:115:11: Variable nbuff shadows outer declaration
  An outer declaration is shadowed by the local declaration. (Use -shadow to
  inhibit warning)
   src/server.c:62:7: Previous definition of nbuff: char [1024]
src/server.c:118:5: Return value (type ssize_t) ignored: send(in, nbuff, ...
src/server.c:122:3: Fresh storage nbuff not released before scope exit
  A memory leak has been detected. Storage allocated locally is not released
  before the last reference to it is lost. (Use -mustfreefresh to inhibit
  warning)
   src/server.c:115:41: Fresh storage nbuff created
src/server.c:123:4: Return value (type int) ignored: close(in)
src/server.c:123:4: Unreachable code: close(in)
  This code will never be reached on any possible execution. (Use -unreachable
  to inhibit warning)

Finished checking --- 29 code warnings
cg83-user35@instance-1:~/Language-Translator/Harshit Sprint_2/CUT/Code$ splint -I ./src/client.c
Splint 3.1.2 --- 21 Feb 2021

Finished checking --- no code processed
cg83-user35@instance-1:~/Language-Translator/Harshit Sprint_2/CUT/Code$ splint -I../header ./src/client.c
Splint 3.1.2 --- 21 Feb 2021

src/client.c:1: Include file <sys/socket.h> matches the name of a POSIX
    library, but the POSIX library is not being used.  Consider using +posixlib
    or +posixstrictlib to select the POSIX library, or -warnposix to suppress
    this message.
```

```
    this message.
  Header name matches a POSIX header, but the POSIX library is not selected.
  (Use -warnposixheaders to inhibit warning)
src/client.c:3: Include file <netinet/in.h> matches the name of a POSIX
    library, but the POSIX library is not being used.  Consider using +posixlib
    or +posixstrictlib to select the POSIX library, or -warnposix to suppress
    this message.
src/client.c:7: Include file <unistd.h> matches the name of a POSIX library,
    but the POSIX library is not being used.  Consider using +posixlib or
    +posixstrictlib to select the POSIX library, or -warnposix to suppress this
    message.
< Location unknown >: Field name reused:
  Code cannot be parsed.  For help on parse errors, see splint -help
  parseerrors. (Use -syntax to inhibit warning)
< Location unknown >: Previous use of
src/client.c: (in function main)
src/client.c:17:15: Function memset expects arg 2 to be int gets char: '0'
  A character constant is used as an int. Use +charintliteral to allow
  character constants to be used as ints.  (This is safe since the actual type
  of a char constant is int.)
src/client.c:29:18: Function memset expects arg 2 to be int gets char: '0'
src/client.c:43:53: Function connect expects arg 3 to be socklen_t gets size_t:
                      sizeof((server))
  Types are incompatible. (Use -type to inhibit warning)
src/client.c:54:7: Unrecognized identifier: bzero
  Identifier used in code has not been declared. (Use -unrecog to inhibit
  warning)
src/client.c:55:7: Return value (type char *) ignored: fgets(buff, 255,...
  Result returned by function call is not used. If this is intended, can cast
  result to (void) to eliminate message. (Use -retvalother to inhibit warning)
src/client.c:58:3: Return value (type char *) ignored: fgets(nbuff, 255...
src/client.c:61:3: Return value (type char *) ignored: fgets(mbuff, 255...
src/client.c:69:7: Assignment of ssize_t to int:
                     in = send(fd, buff, strlen(buff), 0)
  To allow arbitrary integral types to match any integral type, use
  +matchanyintegral.
src/client.c:78:7: Assignment of ssize_t to int: in = recv(fd, buff, 255, 0)
src/client.c:87:2: Return value (type int) ignored: close(fd)
  Result returned by function call is not used. If this is intended, can cast
  result to (void) to eliminate message. (Use -retvalint to inhibit warning)
src/client.c:10:14: Parameter argc not used
  A function parameter is not used in the body of the function. If the argument
  is needed for type compatibility or future plans, use /*@unused@*/ in the
  argument declaration. (Use -paramuse to inhibit warning)
src/client.c:10:26: Parameter argv not used

Finished checking --- 16 code warnings
```

Valgrind report:

```
cg83-user35@instance-1:~/Language-Translator/Harshit Sprint_2/CUT/Code/src$ cd ../bin/
cg83-user35@instance-1:~/Language-Translator/Harshit Sprint_2/CUT/Code/bin$ valgrind ./server
==122091== Memcheck, a memory error detector
==122091== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==122091== Using Valgrind-3.16.1 and LibVEX; rerun with -h for copyright info
==122091== Command: ./server
==122091==
1) Login
2) Run
2
1


^Zserver start runningError in accepting request==122091==
==122091== HEAP SUMMARY:
==122091==     in use at exit: 0 bytes in 0 blocks
==122091==   total heap usage: 2 allocs, 2 frees, 2,048 bytes allocated
==122091==
==122091== All heap blocks were freed -- no leaks are possible
==122091==
==122091== For lists of detected and suppressed errors, rerun with: -s
==122091== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
cg83-user35@instance-1:~/Language-Translator/Harshit Sprint_2/CUT/Code/bin$ valgrind ./client
==122114== Memcheck, a memory error detector
==122114== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==122114== Using Valgrind-3.16.1 and LibVEX; rerun with -h for copyright info
==122114== Command: ./client
==122114==
Client Error: Connection Failed.: Connection refused
==122114==
==122114== HEAP SUMMARY:
==122114==     in use at exit: 0 bytes in 0 blocks
==122114==   total heap usage: 2 allocs, 2 frees, 1,496 bytes allocated
==122114==
==122114== All heap blocks were freed -- no leaks are possible
==122114==
==122114== For lists of detected and suppressed errors, rerun with: -s
==122114== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
cg83-user35@instance-1:~/Language-Translator/Harshit Sprint_2/CUT/Code/bin$
```

UNIT TESTING

1. For writetofile function:

Test_Case 1-6

```
void writetofile_sunnycases(void)
{
        CU_ASSERT_EQUAL(writetofile("hello","eng","gem","halllo","1"), 1);
        CU_ASSERT_EQUAL(writetofile("hola","fre","gem","halllo","1"), 1);
        CU_ASSERT_EQUAL(writetofile("bonjour","gem","eng","welcome","1"), 1);


}

void writetofile_rainycases(void)
{
        CU_ASSERT_EQUAL(writetofile("hello","","","","",""), 0);
        CU_ASSERT_EQUAL(writetofile("hola","eng","","hi","1"), 0);
        CU_ASSERT_EQUAL(writetofile("welcome","gem","","bonjour","1"), 0);

}
```

2. For modify() function

Test_Case 7-12

```
void modify_sunnycases(void)
{
        CU_ASSERT_EQUAL(modify("English_German.txt","Happy","Make"), 1);
        CU_ASSERT_EQUAL(modify("German_English.txt","Wald","Make"), 1);
        CU_ASSERT_EQUAL(modify("French_German.txt","Heureux","Make"), 1);
}

void modify_rainycases(void)
{
        CU_ASSERT_EQUAL(modify("engh_erman.txt","Happy","Make"), 0);
        CU_ASSERT_EQUAL(modify("abcd","Happy","Make"), 0);
}
```

3.For delete() function

Test_Case 13-18

```
void delete_sunnycases(void)
{
        CU_ASSERT_EQUAL(delete("English_German.txt","Make"), 1);
        CU_ASSERT_EQUAL(delete("German_English.txt","Make"), 1);
        CU_ASSERT_EQUAL(delete("French_German.txt","Make"), 1);


}

void delete_rainycases(void)
{
        CU_ASSERT_EQUAL(delete("engh_erman.txt","Happy"), 0);
        CU_ASSERT_EQUAL(delete("French_German.txt","xyz"), 0);
```

4.For translate() function

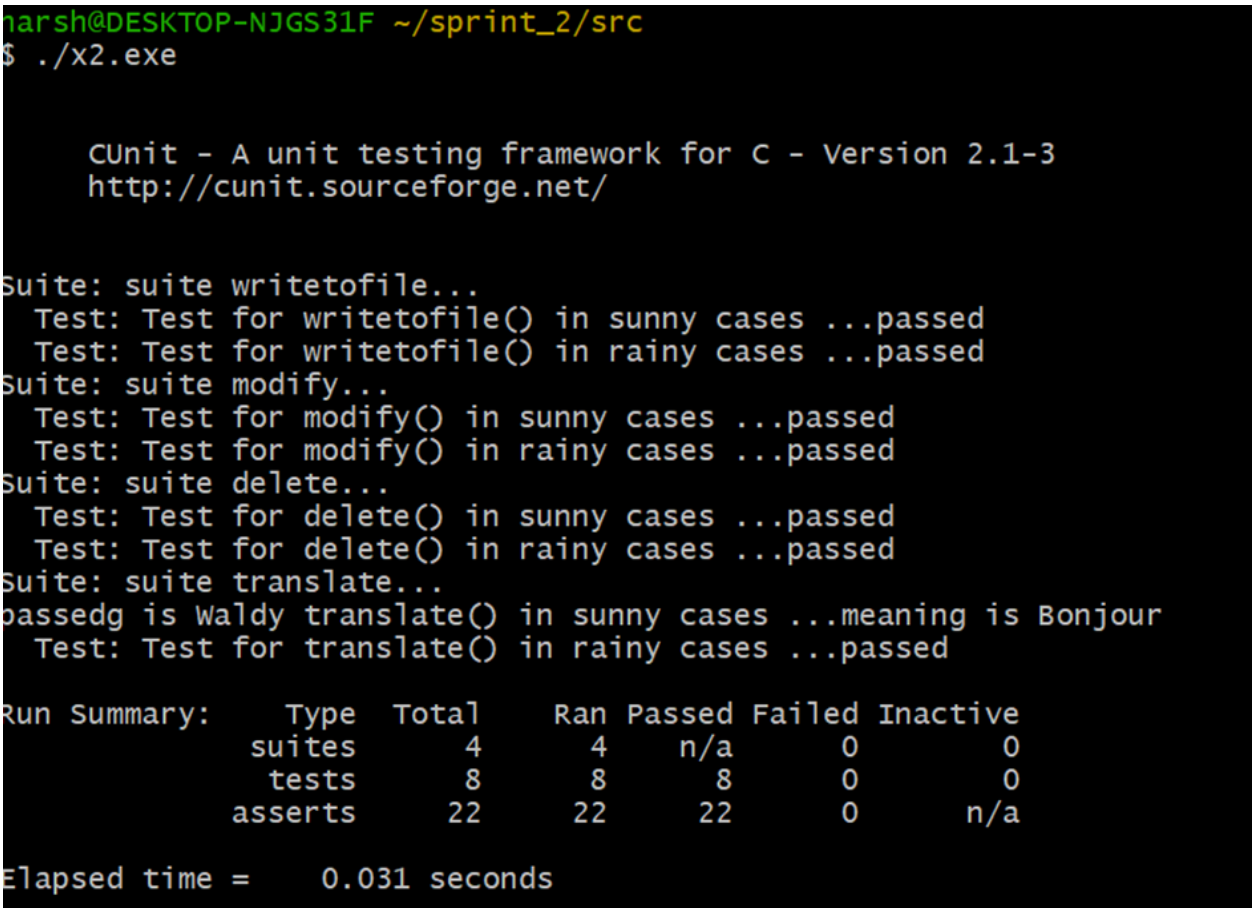
Test_Case 19-25

```
void translate_sunnycases(void)
{
        CU_ASSERT_EQUAL(translate("Hallo","German","French"), 1);
        CU_ASSERT_EQUAL(translate("Contento","Spanish","English"), 1);
        CU_ASSERT_EQUAL(translate("Bosque","Spanish","German"), 1);


}

void translate_rainycases(void)
{
        CU_ASSERT_EQUAL(translate("","Happy","abc"), 0);
        CU_ASSERT_EQUAL(translate("healo","Spanish","German"), 0);
        CU_ASSERT_EQUAL(translate("Hello","panish","German"), 0);
```

Output:


harsh@DESKTOP-NJGS31F ~/sprint_2/src
$ ./x2.exe
CUnit - A unit testing framework for C - Version 2.1-3
http://cunit.sourceforge.net/
Suite: suite writetofile...
Test: Test for writetofile() in sunny cases ...passed
Test: Test for writetofile() in rainy cases ...passed
Suite: suite modify...
Test: Test for modify() in sunny cases ...passed
Test: Test for modify() in rainy cases ...passed
Suite: suite delete...
Test: Test for delete() in sunny cases ...passed
Test: Test for delete() in rainy cases ...passed
Suite: suite translate...
passedg is Waldy translate() in sunny cases ...meaning is Bonjour
Test: Test for translate() in rainy cases ...passed
Run Summary: Type Total Ran Passed Failed Inactive
suites 4 4 n/a 0 0
tests 8 8 8 0 0
asserts 22 22 22 0 n/a
Elapsed time = 0.031 seconds

INTEGRATION TESTING

IT_Case1: Running the Server - User Login

```
Subha@SUBHA-X360-PAVILION ~/SystemProgramming/Sprint2/sprint_2/sprint_2/bin
$ ./server.exe
1) Login
2) Run
1
Welcome to the Login_System
Please Enter your LoginID
User
Please Enter your Password
12345

1) Add New Language
2) Add New Word_Meaning
3) Modify Words
4) Delete Words
5) Exit
enter your choice
```

IT_Case2: Add New Language

```
1) Add New Language
2) Add New Word_Meaning
3) Modify Words
4) Delete Words
5) Exit
enter your choice
1
please enter your source language: English
please enter your target language: Russian
Enter your word
Yellow
Enter your meaning
Желтый
 press 0 for continue or  1 for exit: 0
Enter your word
Blue
Enter your meaning
Синий
 press 0 for continue or  1 for exit: 1

1) Add New Language
2) Add New Word_Meaning
3) Modify Words
4) Delete Words
5) Exit
enter your choice
```

IT_Case3: Add New Word_Meaning

```
1) Add New Language
2) Add New Word_Meaning
3) Modify Words
4) Delete Words
5) Exit
enter your choice
2
please enter your source language: English
please enter your target language: German
Enter your word
Remember
Enter your meaning
Denken Sie daran
 press 0 for continue or  1 for exit: 1

1) Add New Language
2) Add New Word_Meaning
3) Modify Words
4) Delete Words
5) Exit
enter your choice
```

IT_Case4: Modify Words

```
1) Add New Language
2) Add New Word_Meaning
3) Modify Words
4) Delete Words
5) Exit
enter your choice
3
Enter path of source file: English_French.txt
Enter word to replace: Hello
Enter the New word: Hellllloooo

Successfully replaced all occurrences of 'Hello' with 'Hellllloooo'.
1) Add New Language
2) Add New Word_Meaning
3) Modify Words
4) Delete Words
5) Exit
enter your choice
```

IT_Case5: Delete Words

```
1) Add New Language
2) Add New Word_Meaning
3) Modify Words
4) Delete Words
5) Exit
enter your choice
4


 Delete a specific line from a file :
----------------------------------------
 Input the file name to be opened : English_French.txt
Enter your word
Hello

1 Now the content of the file English_French.txt is :
Good Morning:Bonjour
Happy:Heureux
Forest:Forêt

1) Add New Language
2) Add New Word_Meaning
3) Modify Words
4) Delete Words
5) Exit
enter your choice
```

IT_Case6: Running the Client - Translator

```
Subha@SUBHA-X360-PAVILION ~/SystemProgramming/Sprint2/sprint_2/sprint_2/bin
$ ./server.exe
1) Login
2) Run
2
```

```
Subha@SUBHA-X360-PAVILION ~/SystemProgramming/Sprint2/sprint_2/sprint_2/bin
$ ./client.exe

Please Enter The Word to Translate: Hello
Please Enter Your Source Language: English
Please Enter Your Target Language: Hindi

      Meaning: नमस्ते

Please Enter The Word to Translate: Good Morning
Please Enter Your Source Language: English
Please Enter Your Target Language: Spanish

      Meaning: Buenos días
```

Requirement Traceability matrix(RTM):

Table 1

Requirement	Design Mapping	Code Mapping	IT Mapping	UT Mapping
LTR_01	2.1	Login Functionality(Server Side)	IT_01	
LTR_02	2.1.1	Add a new Language and Add a word(Server Side)	IT_02	Test_Case 1-6
LTR_03	2.1.2	Append a new word and its meaning into a specific language(Server Side)	IT_03	Test_Case 1-6
LTR_04	2.1.3	Edit Word and their Meaning(Server Side)	IT_04	Test_Case 7-12
LTR_05	2.1.4	Delete Word(Server Side)	IT_05	Test_Case 13-18
LTR_06	2.2	Translate a word from source language to target language(Server Side)	IT_06	Test_Case 19-25