Capgemini

High Level Design & Low Level Design

Index

1.Introduction: -

1.1 Intended Audience: -

It is an excellent translation is to know and understand the target audience the people who will read and use the translation. The translator needs this information to communicate effectively.

1.2 Project Purpose: -

The Language Translator is a application through this application we can translate the source language to the target language. Our purpose system is that , here we have two side one is server site and other one is client site. First the client send the request to server for translating the language after that the server gives the respond to the client. Here, we can translate English language to German and Hindi language. So English is the source language, German and Hindi are the target language. In server after login using the user name and password, there is option for add language, modify and delete.

1.3 Key Project Objectives: -

- Develop a system which able to conversion between the languages.
- Provide an easy and simple for translation.
- Endow good experience to user.
- Translate almost three to four languages.

1.4 Project scope and limitation: -

Primarily, the scope of the language translator is that It will convert written text from one language to another language by writing, analyzing and editing. It develop a system which able to conversion between two languages. It provides easy and simple for translation. It endows good experience for the user. It translates almost three to four languages.

1.5 Functional Overview: -

1.5.1 Following header files are included in the program:

- #include <sys/socket.h>
- #include <netinet/in.h>
- #include <arpa/inet.h>
- #include <stdio.h>
- #include <stdlib.h>
- #include <unistd.h>
- #include <string.h>
- #include <sys/types.h>

1.6 Server site

1.6.1 Login

- Username
- Password

1.6.2 Run

- Add language:
  In add language add the different languages.

- Append new and meaning in existing file

- Modify:
  Make changes or modification of the words.

- Delete:
  Delete the words.

- Exit:
  After the operation done exit from the server.

1.7 Client site

Client send the request to the server for translating the language.

2. Design Overview: -

Language Translator Application comprises of the following modules:

Name of the Module	Login and Run
Handled by	Sudarshan Timshina
Description	Admin can login by using username and password and run the application

Name of the Module	Add languages and Exit modules
Handled by	Harshit Garg
Description	Admin can add new languages and exit from the modules

Name of the Module	Append new word and meaning
Handled by	Subhashree Sahu
Description	Admin can append new word and their meaning in existing files

Name of the Module	Add/Modify words and Delete Words
Handled by	Sudhanshu Sadhwani
Description	Admin can add or modify the existing words and admin can delete the words.

Name of the Module	Translate the words
Handled by	Subhadeep Sadhukhan
Description	It translates the source words to targeted language

2.1 Design Objectives: -

- Login the server with proper username and password
- Allow admin to add new languages
- Allow admin to add/modify the words
- Allow admin to delete the words
- Allow users to translate the words
- Display the proper output to the user

2.2 Design Alternative: -

We have used linked list instead of stack & queue as Insertion and Deletions operations are fast and easier in linked list. Memory allocation is done during run-time.

2.3 User Interface Paradigms: -

The Language translator helps the user to Translate a word from source language to target language. It also helps the customer to Fetch the word usage for a given word in a sentence.

2.4 Error Detection / Exceptional Handling: -

- If the admin gave invalid username or password, it should through an error
- If the user search invalid words or language, it should through an error
- If the admin didn't run the server customer can't translate the words

2.5 Performance: -

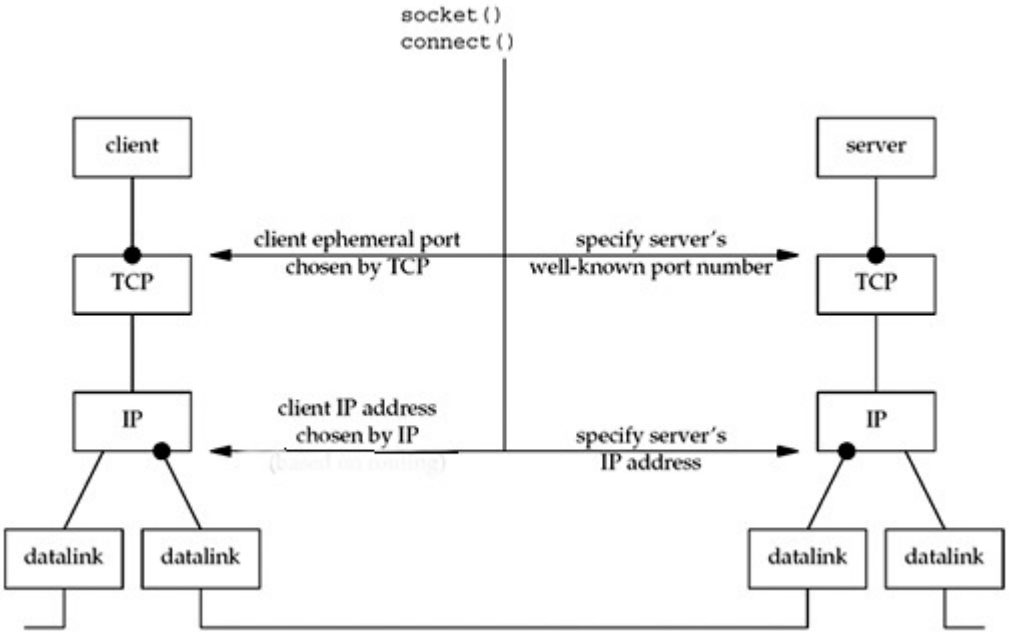
The system will work on the user's terminal. The performance shall depend upon server.

2.6 Maintenance: -

Very little maintenance should be required for this setup. Initially the admin needs to do the language settings and words settings. Admin needs to run the application then only client can access the application.

3.SYSTEM ARCHITECTURE

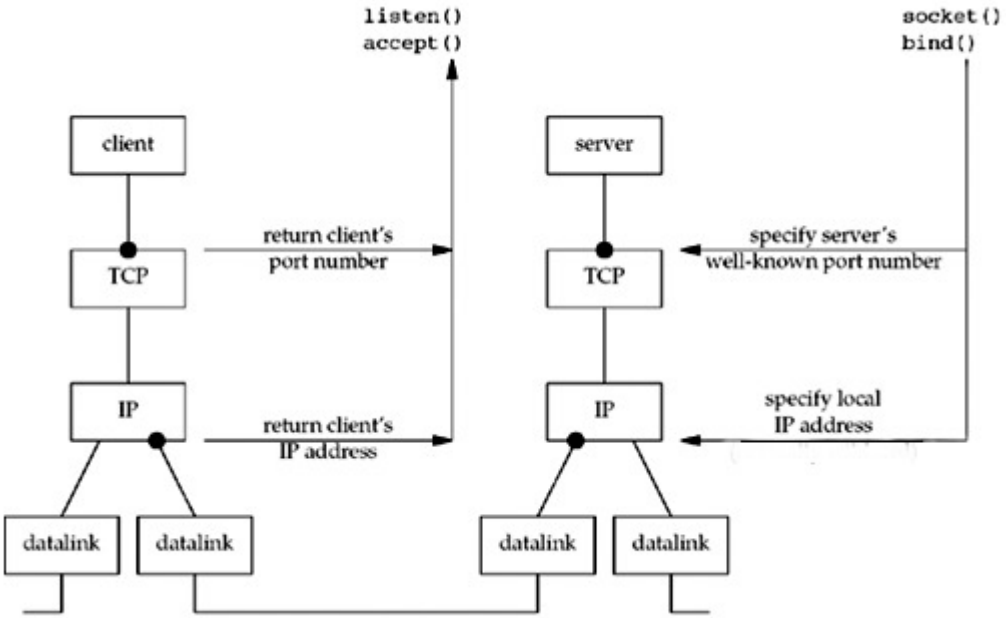
3.1 Client's perspective:



- **Client Socket Creation:** Creating a socket with three parameters (communication domain, type, protocol).
- **Connect:** The connect () call on a stream socket is used by the client application to establish a connection to a server. The IP address and port number must be specified by the client in the call to connect.

- **Sending Message:** The message to translate the language for a particular word is sent after the connection established between the client and the server.

3.2 Server's perspective:

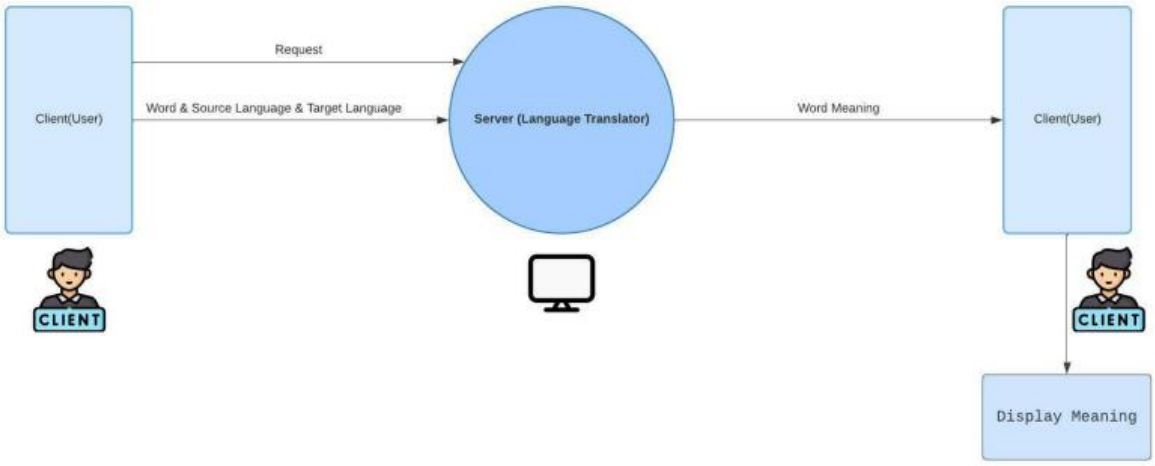


- **Server Socket Creation:** Creating a socket with three parameters (communication domain, type, protocol)
- **Bind:** Bind () function in socket programming is **used to associate the socket with local address** i.e., IP Address, port and address family. The bind () function binds a unique local name to the socket with descriptor socket.

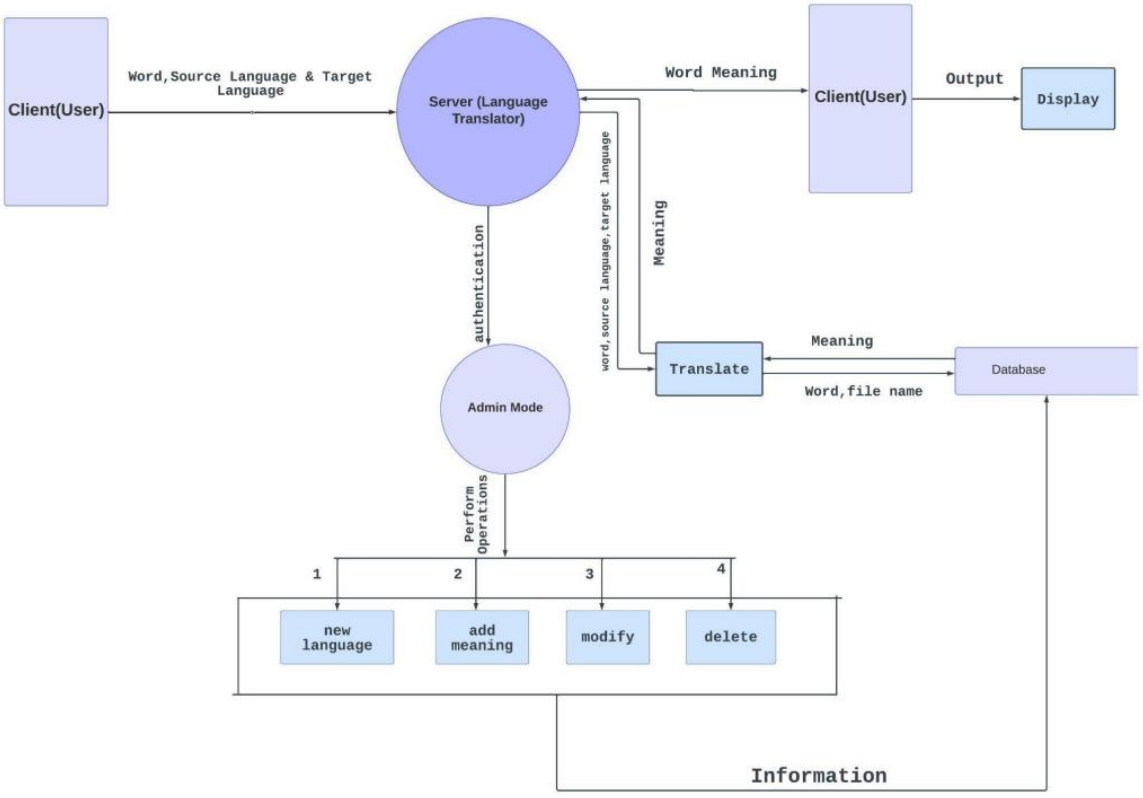
- **Socket listen:** Listening to the binded port for incoming connections keeping a backlog of 5 maximum connections.
- **Accept**: Accepting the connection from client and perform the Language translation and send back the response to the client.
- **Server side authentication:**
- **Server program 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.**

4.Detailed system design

Data Flow Diagram Level 0:

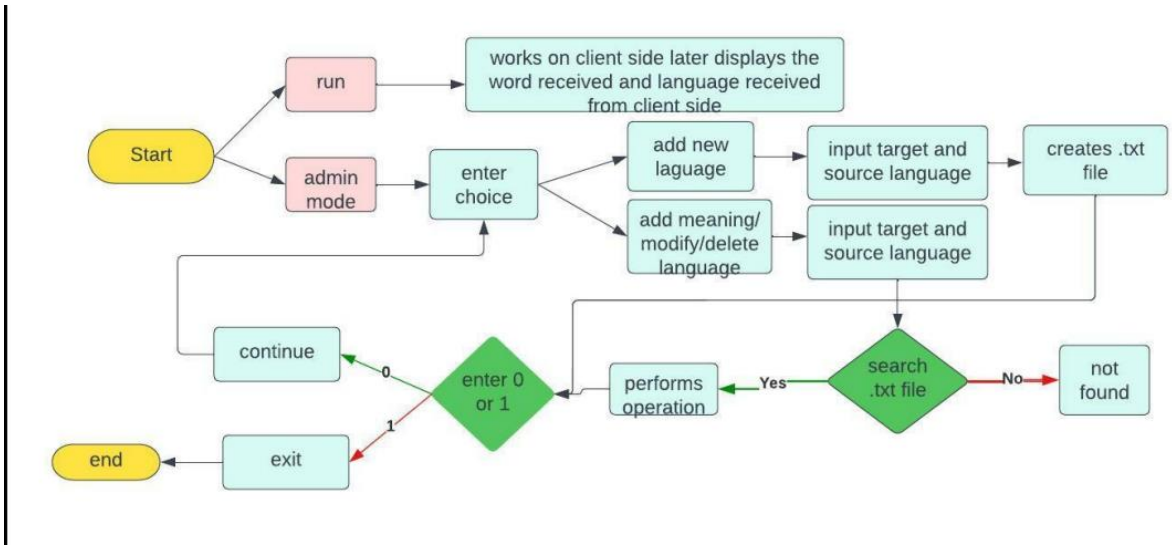


Data Flow Diagram Level 1:

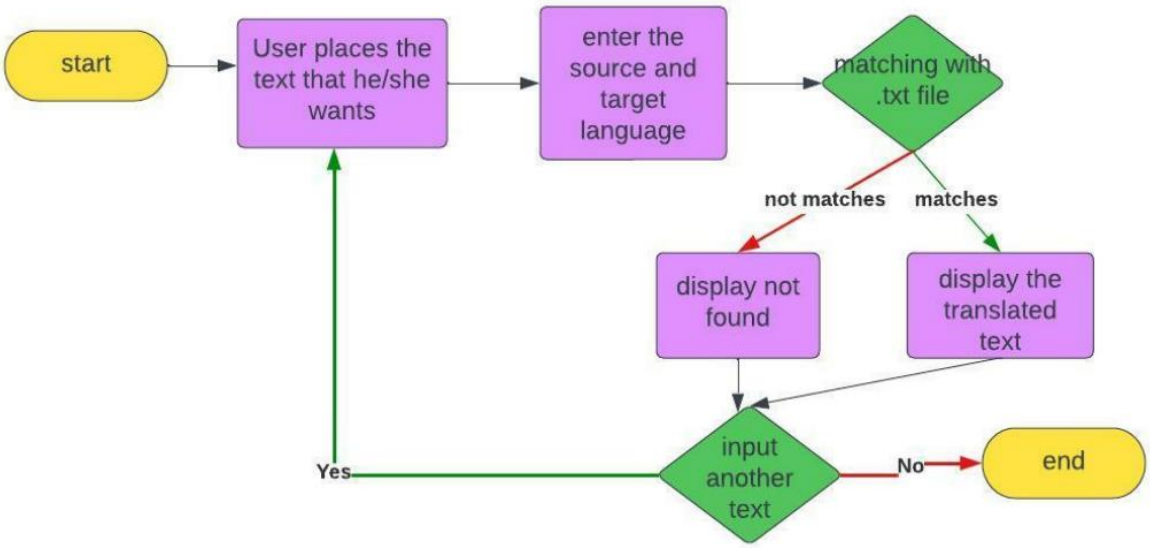


FLOW CHARTS:

->FOR SERVER:



->FOR CLIENT:



5. Environment Description: -

5.1 Time Zone Support: -IST-Kolkata

5.2 Language Support: -English

5.3 User Desktop Requirements: -

- 64-bit processor, 1 GHz or faster
- At least 10 GB free hard drive space
- At least 1 GB RAM**Server**

5.4 -Side Requirements: -

- 64-bit processor, 1 GHz or faster
- At least 2GB free hard drive space
- At least 1GB RAM

5.4.1 Deployment Considerations: -

- Local storage is used
- No network latency to consider
- To scale buy a bigger CPU, more memory, larger hard drive, or additional hardware 9

5.4.2. Application Server Disk Space: -

No such disk space is required as the program is fully functional on online IDE(s) as well. Local Operating System is required and twotxt file to store the records of processes.

5.4.3. Database Server Disk Space: -

No such disk space is required as the program is fully functional on online IDE(s)as well. Local Operating System is required and twotxt file to store the records of processes.

5.4.4. Integration Requirements: -

- Language: -C
- Tools: -Valgrind, Makefile ,splint, gcoverage,Gprof
- Complier: -gcc
- Linux Environment

5.4.5. Network: -End to End

5.5 Configuration: -

5.5.1. Operating System: -Linux environment