
High Level Design & Low Level Design

1. INTRODUCTION	3
1.1. INTENDED AUDIENCE	3
1.2. PROJECT PURPOSE	3
1.3. KEY PROJECT OBJECTIVES	3
1.4. PROJECT SCOPE AND LIMITATION	3
1.5. FUNCTIONAL OVERVIEW	3
2. DESIGN OVERVIEW	4
2.1. DESIGN OBJECTIVES	5
2.2. ARCHITECTURAL STRATEGIES	5
2.2.1. <i>Design Alternative</i>	5
2.2.2. <i>User Interface Paradigms</i>	5
2.2.3. <i>Error Detection / Exceptional Handling</i>	5
2.2.4. <i>Performance</i>	6
2.2.5. <i>Maintenance</i>	6
3. SYSTEM ARCHITECTURE	6
4. DETAILED SYSTEM DESIGN	7
4.1. DFD LEVEL 0	7
4.2. DFD LEVEL 1	7
5. ENVIRONMENT DESCRIPTION	8
5.1. TIME ZONE SUPPORT	8
5.2. LANGUAGE SUPPORT	8
5.3. USER DESKTOP REQUIREMENTS	8
5.4. SERVER-SIDE REQUIREMENTS	8
5.4.1. <i>Deployment Considerations</i>	8
5.4.2. <i>Application Server Disk Space</i>	8
5.4.3. <i>Database Server Disk Space</i>	8
5.4.4. <i>Integration Requirements</i>	8
5.4.5. <i>Network</i>	8
5.5. CONFIGURATION	9
5.5.1. <i>Operating System</i>	9

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 best thing about this application is it helps users to produce an error free result within a few seconds.
- This application is compatible with almost every smart device having an internet connection.
- This application don't require any installation process and provides the best options for editing tasks

1.3. Key Project Objectives

- Client id is displayed whenever connection is established.
- Signal Handling when server shutdowns abruptly.
- Anonymous and Authenticated User.
- Multiple clients connected to a single server.
- Server shows the changes received from the edit command.
- Signal Handling for client shutdown.

1.4. Project Scope and Limitation

This project aims for users to edit there text files easily from any device having internet connection. The user can easily access this application by just there User Id and Password from any device. This application don't require any installation process and provides the best options for editing tasks and it is compatible with almost every smart device.

1.5. Functional Overview

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>`

2. Design Overview

- Remote Line Editor comprises of the following modules in maintain database:

Name of the Module	Change Directory command
Handled by	
Description	The function is used to change the directory.

Name of the Module	list Directory command
Handled by	
Description	The function is used to list the contents of the directory.

Name of the Module	print command
Handled by	
Description	This function is used to print the contents inside the file.

Name of the Module	Edit command
Handled by	
Description	This function is used to edit a particular line in the filename

Name of the Module	Select command
Handled by	
Description	This function is used to select the filename.

Name of the Module	Help command
Handled by	
Description	This command is used to list down all the commands inside the editor.

Name of the Module	help ls,print,edit
Handled by	

Description	This command is used to tell the user about the syntax of the commands
-------------	--

Name of the Module	Bye command
Handled by	
Description	This command is used to disconnect from the server.

Name of the Module	clear
Handled by	
Description	This command is used to clear the screen.

2.1. Design Objectives

- Add different commands in the Line Editor
- Start the connection
- Accept the connection
- Different commands should be able to print the desired output.
- Remote Line Editor should be able to edit line
- Signal Handling for server and client disruptions.

2.2. Architectural Strategies

2.2.1. Design Alternative

We have used three structure to store data i.e. User, Server and Client and further typedef it and used in the program.

2.2.2. User Interface Paradigms

The Remote Line Editor should be able to allow the user to select files , print the contents and edit the particular line. Multiple clients should be able to connect one server.

2.2.3. Error Detection / Exceptional Handling

- If the user gives invalid username and password, it shows user not authenticated error.
- If the user gives command that is not listed in help command then it shows invalid command.
- If both the server and client side ends abruptly then it is handled by signal handling.

2.2.4. Performance

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

2.2.5. Maintenance

Very little maintenance should be required for this setup. Initially the admin needs to login whether as authenticated or as anonymous user. Admin needs to run the application then only clients can access the application and remotely edit and print the contents in files.

3. System Architecture

3.1 Database Architecture

The database used inside our program is the user.txt file. The user.txt file contains username and password separated by space. We have used strtok with a delimiter as space to store username and password. Also for the present directory we can use username for the purpose.

The architecture used in this system is like that we store all the users inside a data folder in the home directory. And all the information of the authenticated and anonymous user stored in user.txt files.

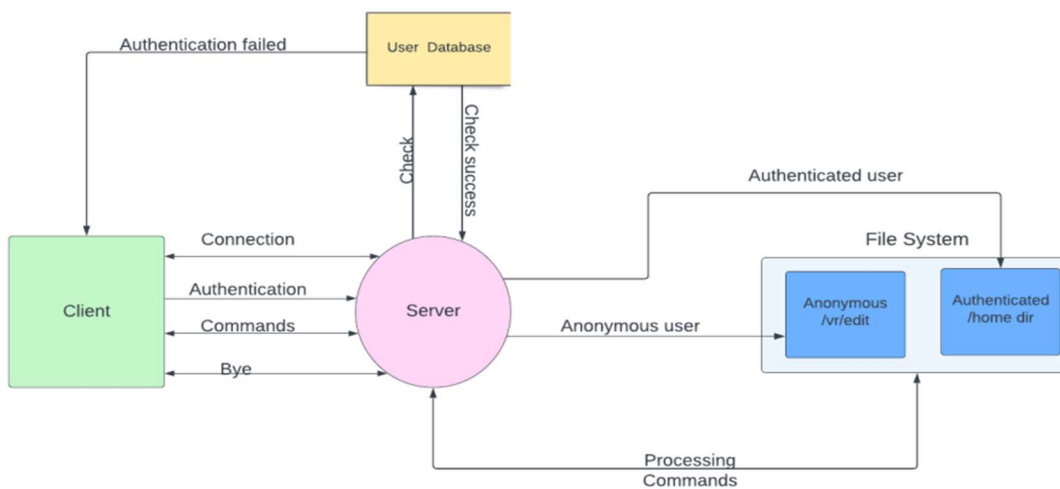
```
1  anonymous password
2  suraj 1234
3  nikita 1234
4  dinesh 1234
5  tejas 1234
6  munesh 1234
```

4. Detailed System Design

4.1. DFD Level 0



4.2. DFD Level 1



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

5.4. Server-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

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 txt file to store the records of users.

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 txt file to store the records of users.

5.4.4. Integration Requirements

- Language: - C
- Tools: - Valgrind, Makefile, Gcoverage, Gprof
- Compiler: - gcc
- Linux Environment

5.4.5. Network

End to End

5.5. Configuration

5.5.1. Operating System

Linux environment

Change Log

QMS Template Version Control (Maintained by QA)
--

Date	Version	Author	Description
28-May-2015	1.0	QA Team	Initial Version