

**TRIVIAL FILE TRANSFER PROTOCOL**

**TFTP\_DesignDocument-v0.1**

High Level Design & Low Level Design

The purpose of this document is to provide a template for documenting both HLD & LLD.

**Document Control :**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Project Revision History** | | | | | | | | |
|  |  |  | |  |  |  |  |  |
| **Date** | **Version** | **Author** | **Brief Description of Changes** | | | | **Approver Signature** | |
|  |  |  |  | | | |  | |
|  |  |  |  | | | |  | |
|  |  |  |  | | | |  | |

[**1. Introduction 6**](about:blank)

[1.1. Intended Audience 6](about:blank)

[1.2. Acronyms/Abbreviations 6](about:blank)

[1.3. Project Purpose 6](about:blank)

[1.4. Key Project Objectives 6](about:blank)

[1.5. Project Scope and Limitation 7](about:blank)

[1.5.1. In Scope 7](about:blank)

[1.5.2. Out of scope 7](about:blank)

[1.6. Functional Overview 7](about:blank)

[1.7. Assumptions, Dependencies & Constraints 7](about:blank)

[1.8. Risks 7](about:blank)

[**2. Design Overview**](about:blank) **8**

[2.1. Design Objectives 8](about:blank)

[2.1.1. Recommended Architecture 8](about:blank)

[2.2. Architectural Strategies 8](about:blank)

[2.2.1. Design Alternative 9](about:blank)

[2.2.2. Reuse of Existing Common Services/Utilities 9](about:blank)

[2.2.3. Creation of New Common Services/Utilities 9](about:blank)

[2.2.4. User Interface Paradigms](about:blank) 9

[2.2.5. System Interface Paradigms 9](about:blank)

[2.2.6. Error Detection / Exceptional Handling 9](about:blank)

[2.2.7. Memory Management 9](about:blank)

[2.2.8. Performance 1](about:blank)0

[2.2.9. Security 1](about:blank)0

[2.2.10. Concurrency and Synchronization 1](about:blank)0

[2.2.11. Housekeeping and Maintenance 1](about:blank)0

[**3. System Architecture**](about:blank) **11**

[3.1. System Architecture Diagram. (Not Necessary) 1](about:blank)1

[3.2. System Use-Cases 1](about:blank)1

[3.3. Subsystem Architecture](about:blank) 12

[3.4. System Interfaces](about:blank) 13

[3.4.1. Internal Interfaces](about:blank) 13

[3.4.2. External Interfaces 1](about:blank)3

[**4. Detailed System Design**](about:blank) **14**

[4.1. Key Entities](about:blank) 14

[4.2. Detailed-Level Database Design](about:blank) 15

[4.2.1. Data Mapping Information 1](about:blank)6

[4.2.2. Data Conversion 1](about:blank)6

[4.3. Archival and retention requirements 1](about:blank)6

[4.4. Disaster and Failure Recovery](about:blank) 16

[4.5. Business Process workflow](about:blank) 17

[4.6. Business Process Modeling and Management (as applicable) 1](about:blank)7

[4.7. Business Logic](about:blank) 18

[4.8. Variables 1](about:blank)8

[4.9. Activity / Class Diagrams (as applicable)](about:blank) 19

[4.10. Data Migration](about:blank) 21

[**5. Environment Description**](about:blank) **21**

[5.1. Time Zone Support 2](about:blank)1

[5.2. Language Support 2](about:blank)1

[5.3. User Desktop Requirements 2](about:blank)1

[5.4. Server-Side Requirements 2](about:blank)1

[5.4.1. Deployment Considerations 2](about:blank)1

[5.4.2. Application Server Disk Space 2](about:blank)2

[5.4.3. Database Server Disk Space 2](about:blank)2

[5.4.4. Integration Requirements 2](about:blank)2

[5.4.5. Jobs 2](about:blank)2

[5.4.6. Network 2](about:blank)2

[5.4.7. Others 2](about:blank)2

[5.5. Configuration 2](about:blank)2

[5.5.1. Operating System 2](about:blank)3

[5.5.2. Database 2](about:blank)3

[5.5.3. Network 2](about:blank)3

[5.5.4. Desktop 2](about:blank)3

[**6. References 2**](about:blank)**3**

[**7. Appendix 2**](about:blank)**3**

**1**.**INTRODUCTION**

The Trivial File Transfer Protocol (TFTP) is used to perform upload and download operations between server and multiple clients.

**1.1. Intended Audience**

This project is a prototype for everyone who wants to perform upload and the download operation between the server and the client. This document is intended to read by developers, testers, project managers and customers. This is a technical document, and the terms should be understood by all of them.

**1.2. Acronyms/Abbreviations**

Table

Description automatically generated**1.3. Project Purpose**

Purpose of this document is to describe the requirement to provide file transferring through the UDP (network) between server and client. It aims at uploading or downloading a file from the TFTP server according to the request of the TFTP client.

**1.4. Key Project Objectives**

* Establishing connection between TFTP client and TFTP server.
* Providing services to perform download and upload operations.
* Maintaining a log on each upload request.
* Handling several error cases.

**1.5. Project Scope and Limitation**

The scope and limitation of the project are listed,

**1.5.1. In Scope**

This project provides implementation of TFTP Service (UDP based) involving a client and a server. The server handles the client's request and performs several operations (downloading and uploading of the file). Server may optionally handle multiple client’s requests simultaneously.

**1.5.2. Out of scope**

Operating Systems such as Windows are out of scope. Access to the client is only with one port.

**1.6. Functional Overview**

Trivial File Transfer Protocol (TFTP) – TFTP is a simple protocol used for transferring files. TFTP uses the User Datagram Protocol (UDP) to transport data from one end to another. TFTP is mostly used to upload/download files to/from the TFTP server.

**1.7 Assumptions, Dependencies & Constraints**

Issues facing if any to avoid we need a Linux like environment to perform the task and reliable connectivity while file transfer.

**1.8. Risks**

TFTP also sends all data in plain text, meaning there is a risk that this can be sniffed by malicious attackers on the network who will then be able to access username and passwords. In other words, anyone who has access can upload or download files – other risks involve corrupting files or misdirecting them.

**2. DESIGN OVERVIEW**

The system consists of two entities, the remote server, and a multiple client user.

**2.1. Design Objectives**

The server handles the client's request and performs several operations (downloading and uploading of the file). On the server side, it must be configured to start the service and ACK message to the client and provide a file to download the file.

**2.1.1. Recommended Architecture**

The recommended system architecture is as follows.

Server-side architecture:

●    1GB RAM

●    500MHz Processor

●    Database

●    Internet connection

Client-side hardware interface:

●    Desktop or Linux machine

●    Internet connection

●    Terminal

**2.2. Architectural Strategies**

A client-server architecture method was adopted in this project. This method is made up of two parts: many clients and the server. Clients request services from the server, and the server responds by uploading or downloading files from the server. The server is still listening to client requests. A terminal with a variety of file search options will be given.

**2.2.1. Design Alternative**

The UDP protocol is used by the project to link the server and clients in a two-way connection. UDP uses a connectionless method, making it a quicker, simpler, and more efficient protocol to develop than TCP. The project reads the file regardless of whether it is present on the server or not. If the file is missing, the server will send an ACK message to a specific client.

**2.2.2. Reuse of Existing Common Services/Utilities**

The system requires the use of existing common services and utilities which are: UDP (User Datagram Protocol) & other System Calls.

**2.2.3. Creation of New Common Services/Utilities**

TFTP servers allow connections from a TFTP Client for uploading and receiving files by creating or using any new common services or utilities.

**2.2.4. User Interface Paradigms**

A user interface is how a program including the operating system communicates with a client and server. In these types of user interface we use Command Line Interface (CLI).  Desktop or a Linux machine with internet connection.

**2.2.5. System Interface Paradigms**

●    Operating system – Unix.

●    Linux Kernel version - 4.4.0-19041-Microsoft.

●    Bash shell.

**2.2.6. Error Detection / Exceptional Handling**

From the client’s connection to the server will be provided error detection.  Error message for file will also be included if the file is not found, Not defined, Unknown port number, No such user, Access violation, Illegal operation.

**2.2.7. Memory Management**

NA.

**2.2.8. Performance**

In the performance section, we will use UDP for faster speed and decrease latency, system architecture, and with data messages whether they are transmitted or not will satisfy performance criteria.

**2.2.9. Security**

The TFTP server provides an option whereby only files in a specific directory can be accessed.

**2.2.10. Concurrency and Synchronization**

Client and server will be in synchronization and configured in downloading and uploading a file to the server.

**2.2.11. Housekeeping and Maintenance**

The screen is updated as the client begins for an improved user experience. Cleaning the system's memory buffers.

**3.  SYSTEM ARCHITECTURE**

Client/server architecture shares the data processing chores between a server typically, a high-end workstation but quite possibly a mainframe and clients, which are usually PCs.

A multiple client server is a type of software architecture for computer networks where clients, which can be basic workstations or fully functional personal computers, request information from a server computer. The most common type of multiple client server system for small businesses and homes is the single server with multiple clients. One server can handle dozens of information requests from client computers simultaneously.

**3.1 System Architecture Diagram**

Diagram

Description automatically generated

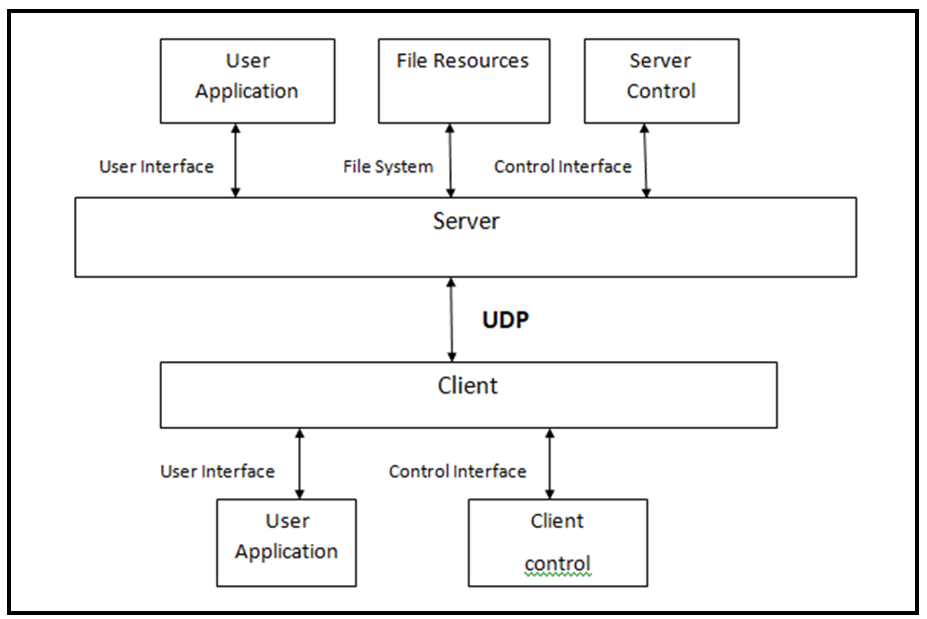
**3.2 System Use Cases**

* Display packet insufficiency if the packet is lost.
* If not able to find the directory display message not found.
* Display the termination error if the transferred data packet is fewer than 512 bytes.
* Should display errors when the server is not able to upload the files.
* Should display errors when the server is not able to upload the files.
* Should display the debug log errors accordingly.

**3.3 Subsystem Architecture Diagram

Description automatically generated**

**3.4 System Interfaces**

****

**3.4.1 Internal Interfaces**

The internal interface is nothing but a **connection to a device inside the computer's cabinet and it** comprises interfaces through which the system interacts with the clients through which it provides them services.

* Cloud System
* Internet

**3.4.2 External Interfaces**

The external interface comprises interfaces through which the users interact with the system.

* Internet
* The required file should be downloaded from the server side by the client.
* The required file should be uploaded from the client side to the server.

**4. DETAILED SYSTEM DESIGN**

The Trivial File Transfer Protocol (TFTP) includes server and client connection. The connection between server and the client is established using UDP protocol.

The client can,

1. Download the file or data from the server.
2. Upload the file or data to the server.

**4.1 Key Entities**

The key entities associated with the system are:

**Server**

* The server is a remote entity that stores the files.
* It verifies the IP address of the client making the connection.
* Provides service on the client request.
* Logs are maintained regarding server functions.

**Client**

* The client is an entity that is run by the user on their system.
* It requests to connect with the server.
* Client requests the server to download or upload a file from the server based on the port number.

**4.2 Detailed-Level Database Design**

* + 1. **Level - 0 Diagram**

**Diagram

Description automatically generated**

* + 1. **Level – 1 Diagram**

**Diagram

Description automatically generated**

**4.2.3 Level - 2 Diagram**

Diagram

Description automatically generated

**4.2.4 Data Mapping Information**

**Diagram

Description automatically generated**

**4.2.5 Data Conversion**

NA

**4.2.6 Requirement Specifications**

**4.2.6.1 Feature to download and upload files from/to the server**

The TFTP server allows multiple clients to download files from the server and upload the files to the server.

* + - 1. **Feature to handle multiple client’s request**

The system shall allow the server to handle multiple client’s requests.

* + - 1. **Appropriate Error Handling**

Provides appropriate error messages namely file not found, access violation and packet loss etc.

**4.2.6.4 Data Security**

TFTP server provides an option whereby only files in specific directory can be accessed.

* + - 1. **Re-transmission of the lost packet**

If a data packet gets lost in the network, then the intended server or receiver will timeout and send an acknowledgment, thus causing the sender to retransmit that lost packet.

* + - 1. **Log maintenance**

The TFTP server maintains a log on upload operation performed by its clients.

**4.3 Archival and retention requirements**

Logs have been maintained on the server side, according to what function server is performing.

TFTP includes debug log message with at least 4 levels

FATA, INFO, WARNING, DEBUG.

* 1. **Disaster and Failure Recovery**

NA

**4.5 Business Process Workflow**

**Diagram

Description automatically generated**

**4.6 Business Process Modelling and Management (as applicable)**

NA

**4.7 Business Logic**

**Diagram

Description automatically generated**

**4.8 Variables**

int portNo.

int sockfd.

struct sockaddr\_inserver\_addr, client\_addr.

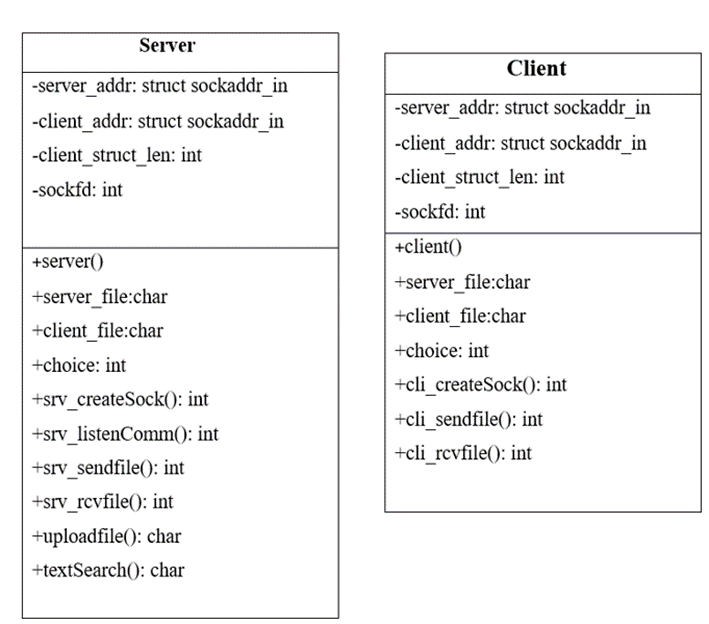
char server\_file[1000].

char client\_file[1000].

int client\_struct\_len.

     string filename.

**4.9 Activity/Class Diagrams (as applicable)**

****

**4.9.1 Activity Diagram**

**4.9.1.1 Socket creation diagram**

**Diagram

Description automatically generated**

**4.9.1.2 Socket end operation diagram**

**Diagram

Description automatically generated**

**4.9.1.3 Sequence diagram**

**Diagram

Description automatically generated**

**4.10 Data Migration**

Data is migrated between the client and server.

**5. ENVIRONMENT DESCRIPTION**

 The Environment description allows the TFTP client to connect to the TFTP server and perform uploading and downloading operations within the TFTP server and also allows the server to maintain a log on uploaded files.

## 5.1 Time Zone Support

It will support time zones as per Indian standard time (IST) in (GMT +5:30) and UST standard.

## 5.2 Language Support

C++ language and compilation using g++. The Linux commands to do that task we can specify the commands.

## 5.3 User Desktop Requirements

User desktop requires a Linux environment, Operating system of Linux Debian, or Terminal x86\_64 GNU/Linux and kernel version 4.4.0-19041-Microsoft #1237 Microsoft and reliable internet connectivity.

## 5.4 Server-Side Requirements

In server side,

● Disk space – Minimum 150GB

● Uninterrupted connectivity 24x7

### 5.4.1 Deployment Considerations

Deployment considerations are,

● 500Mhz Processor

● 120GB HDD CPU

● 4GB RAM

● Network connectivity

### 5.4.2 Application Server Disk Space

    Disk space – Minimum 150GB.

### 5.4.3 Database Server Disk Space

   NA.

### 5.4.4 Integration Requirements

The PWD command displays the current working directory on the server for the logged in user.

### 5.4.5 Jobs

NA.

### 5.4.6 Network

The network connects the system for the purpose of uploading and downloading of files therefore stable Internet connectivity is required.

### 5.4.7 Others

NA.

## 5.5 Configuration

The TPTP follows the stop and wait protocol. The TFTP Server provides the acknowledgment for the client for the upload operation, using system calls and UDP protocol then the client proceeds with an upload operation of the required file, and the client can also download files from the TFTP server. TFTP shall work in all standard LINUX-like Operating systems.

### 5.5.1 Operating System

●      Operating system – Linux.

●      RAM - 8GB.

●      Processor - i3/i5/i7.

**5.5.2 Database**

NA.

### 5.5.3 Network

The following are the network details regarding the project:

●      The client and server communicate over a UDP protocol.

●      The IP address used can be either IPv4 or IPv6.

### 5.5.4 Desktop

A Linux-like environment is required.

**6. REFERENCES**

1. <https://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=GIOD_udp>

for UDP protocol.

1. System Requirements Specification Document
2. <https://linuxhint.com/grep_command_linux/> Grep Command in Linux

**7. APPENDIX**

NA.

**Change Log**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **QMS Template Version Control (Maintained by QA)** | | | | | |
|  |  |  |  |  |  |
| **Date** | **Version** | **Author** | | **Description** | |
|  |  |  | |  | |
|  |  |  | |  | |
|  |  |  | |  | |
|  |  |  | |  | |