

# TEAM HK

# FTP CLIENT

## **1. INTRODUCTION**

### 1.1 Purpose:

The document is aimed to describe the requirements of an FTP client software. Information regarding why our FTP Client would differ than current FTP software that lacks likability from a large amount of users.

### 1.2 Background

Existing FTP software is embedded with tons of settings and advanced features that are hardly used by the average user. TeamHK is coming up with a simple graphical user interface enabling the user to connect to any FTP server. We took a micro-kernel design approach in designing the FTP Client, as only core functionalities will be included such as: transferring files from local machine to FTP server and vice versa. Able to traverse through directories on both local and FTP server with nice Icon Viewing.

### 1.3 Definitions, Acronyms, and Abbreviations

- **FTP** - File Transfer Protocol
- **SSH** - Secure Shell
- **IP** - Internet Protocol
- **TCP** - Transmission Control Protocol
  - (Type of protocol used in our Transport Layer of our FTP Client software)

## **2. SYSTEM REQUIREMENTS**

### 2.1 Actors

FTP User(s) - Able to connect to any FTP server and manage uploading or downloading any file from the server. File manipulation is also a feature that users are allowed to use.

### 2.2 High level Flow

FTP user's need to transfer files from a server on a daily basis. This task requires the user to connect to a specific server and later being able to disconnect from the connection established to the server. Connecting to the FTP server requires several pieces of information from the user: host or IP address, username, password, and port. Usually the port will be connecting to the reserved port 21, which is specifically for FTP servers. The user will be able to choose files and directories through an icon viewer. Concluding transfer of files successfully, the user will be able to terminate connection and end the client by pressing the exit button. Before exit, the user's connection will be safely terminated and account will be logged out.

## 2.3 Basic Use Cases

1. User makes a connection to server
2. FTP Client makes connection to server (connect button)
3. User clicks to highlight specific file displayed on an Icon view
4. User clicks file and highlights the currently selected file on local side (left side)
5. User clicks file and highlights the currently selected file on FTP server side (right side)
6. User can upload by selecting a file and pressing the Upload Button (Upload)
7. User can download by selecting a file and pressing the Download Button (Download)
8. Connection indication on Login Screen whether the connection to FTP server has been established (on top)
9. Error indications regarding incorrect information entered or empty fields when attempting to connect to FTP server, such as Red Background on Text Fields.
10. User able to traverse through icon view of files and directories in local side (right side)
11. User able to traverse through list view of files and directories in FTP server side (left side)
12. User adding/creating a new file to server
13. User being able to refresh currently viewed directory using refresh/"List Files" button
14. User able to remove file on server side, by highlighting file to remove and clicking "remove" button
15. User able to traverse to previous directory using ".." button.
16. User able to rename a file or directory on server side
17. User able to rename a file or directory on local side

## 2.4Extended Use Cases

1. User can login using history log, enabling users to connect faster
2. User being able to remove directory by highlighting directory and pressing remove button.
3. User able to create directory by entering desired name of directory and pressing create button.