

# **WORD FREQUENCY COUNT**

**Design Documents WFC - Version 0.1**

**DOCUMENT CONTROL:**

Project Revision History				
Date	Version	Author	Brief Description of Changes	Approver Signature
04-02-2026	Version 0.0		Initial Stage	
04-02-2026	Version 0.1		Some Changes in use case diagram and classes	

**TEAM MEMBERS**


# TABLE OF CONTENTS

## 1) HIGH-LEVEL DESIGN

<b>1. INTRODUCTION</b>	5
1.1 PURPOSE	5
1.2 SCOPE	5
1.3 DEFINITIONS	5
1.4 OVERVIEW	6
<b>2. GENERAL DESCRIPTION</b>	6
2.1 PRODUCT PERSPECTIVE	6
2.2 TOOLS USED	6
2.3 GENERAL CONSTRAINTS	7
2.4 ASSUMPTIONS	7
2.5 SPECIAL DESIGN ASPECTS	7
<b>3. DESIGN DETAILS</b>	7
3.1 MAIN DESIGN FEATURES	7
3.2 APPLICATION ARCHITECTURE	7
3.3 DATA FLOW DIAGRAM	9
3.4 FILES	11
3.5 USER INTERFACE	11
3.6 ERROR HANDLING	12
3.7 HELP	13
3.8 PERFORMANCE	13
3.9 SECURITY	13
3.10 RELIABILITY	13
3.11 MAINTAINABILITY	13
3.12 PORTABILITY	14

3.13 REUSABILITY	14
3.14 APPLICATION COMPATIBILITY	14
3.15 RESOURCE UTILIZATION	14
3.16 MAJOR CLASSES	14
<b>2) LOW-LEVEL DESIGN</b>	<b>15</b>
<b>4. INTRODUCTION</b>	<b>15</b>
4.1 PURPOSE	15
4.2 DOCUMENT CONVENTIONS	16
4.3 INTENDED AUDIENCE AND READING SUGGESTION	16
<b>5. DETAILED SYSTEM DESIGN</b>	<b>16</b>
5.1 DESIGN DESCRIPTION	16
5.2 USE CASE DIAGRAM	16
5.3 CLASS DIAGRAM	17
5.4 SEQUENCE DIAGRAM	17
5.5 DESIGN AND IMPLEMENTATION CONSTRAINTS	18
5.6 USER INTERFACE	18
<b>6. SECURITY</b>	<b>18</b>

# High Level Design Document

## 1. INTRODUCTION

### 1.1 PURPOSE

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding and can be used as a reference manual for how the modules interact at a high level.

### 1.2 SCOPE

This document provides a comprehensive high level design overview of the Word Frequency Counter System. It highlights the high-level flow / use cases in text analysis and word frequency computation and serves as an input to the low-level design documents that would further elaborate on the proposed system design.

### 1.3 DEFINITIONS

#### 1.3.1 WORD FREQUENCY COUNTER

A Word Frequency Counter is a system that analyzes a given text or text file and calculates the number of occurrences of each word. It processes the input text, normalizes the data, and generates frequency counts for effective text analysis.

#### 1.3.2 CLIENT

A client is any computer hardware or software device that requests access to a service provided by a server. In the Word Frequency Counter system, the client is responsible for accepting text or file input from the user and displaying the computed word frequency results.

### **1.3.3 SERVER**

A server is a computer program or device that provides a service to another computer program and its user, also known as the client. In the Word Frequency Counter system, the server processes the received text data, performs word frequency analysis, and returns the results to the client.

## **1.4 OVERVIEW**

This HLD Document is arranged in the following format:

Section 1: Introduction

A brief explanation about the purpose, aim, scope, and design format of the proposed system.

Section 2: General Description

This section is all about the general constraints, assumptions, and design aspects associated with the proposed system. The product perspective will give an overall description of the system.

Section 3: Design Details

This section documents the detailed design of all modules associated with the development of the proposed system.

## **2. GENERAL DESCRIPTION**

### **2.1 PRODUCT PERSPECTIVE**

The Word Frequency Counter System is based on the concept of analyzing text data and calculating the frequency of words in an efficient and simple manner. The main actors in this system are users, client, and server. The system is basically constructed using C++ language. The system will allow multiple users at a time. The client can send text or text files to the server for word frequency analysis, and the server returns the computed results to the client.

### **2.2 TOOLS USED**

1. Linux file system.
2. C++ language is used.

## 2.3 GENERAL CONSTRAINTS

The Word Frequency Counter system should be user-friendly and automated. The client should be able to submit text or text files to the server and receive accurate word frequency results without manual intervention.

## 2.4 ASSUMPTIONS

The assumptions and dependencies relevant to the system are as follows.

- Server should be running
- Users should have internet access
- Network should be reliable
- Linux terminal is required

## 2.5 SPECIAL DESIGN ASPECTS

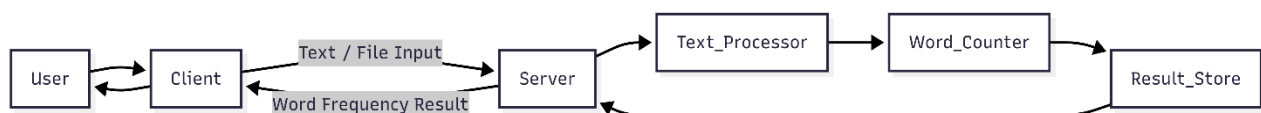
One of the design aspects is that the system will work with multiple clients at a time, allowing simultaneous text analysis requests without affecting performance.

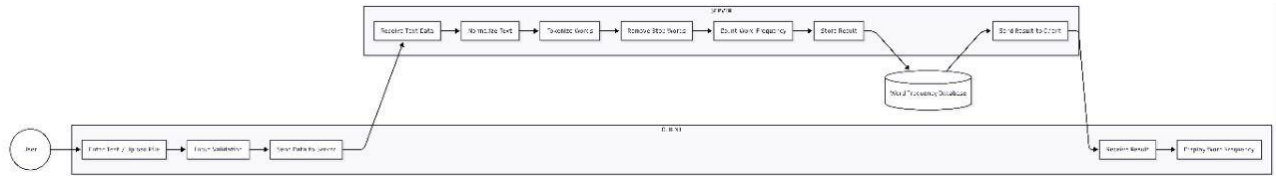
# 3. DESIGN DETAILS

## 3.1 MAIN DESIGN FEATURES

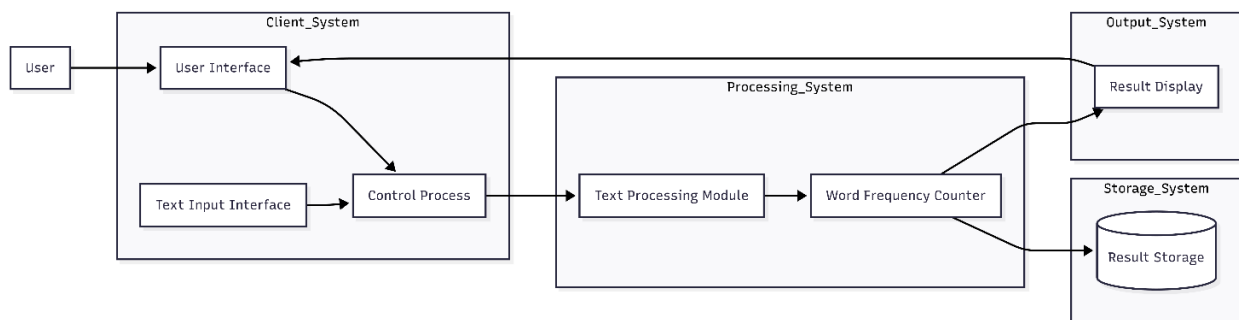
The main design features include four major parts: the architecture, the user interface design, the text processing modules, process relation, and automation. In order to make these designs easier to understand, the design has been illustrated in attached diagrams (Use Case Diagram, Data Flow Diagrams, and Sequence Diagrams).

## 3.2 APPLICATION ARCHITECTURE

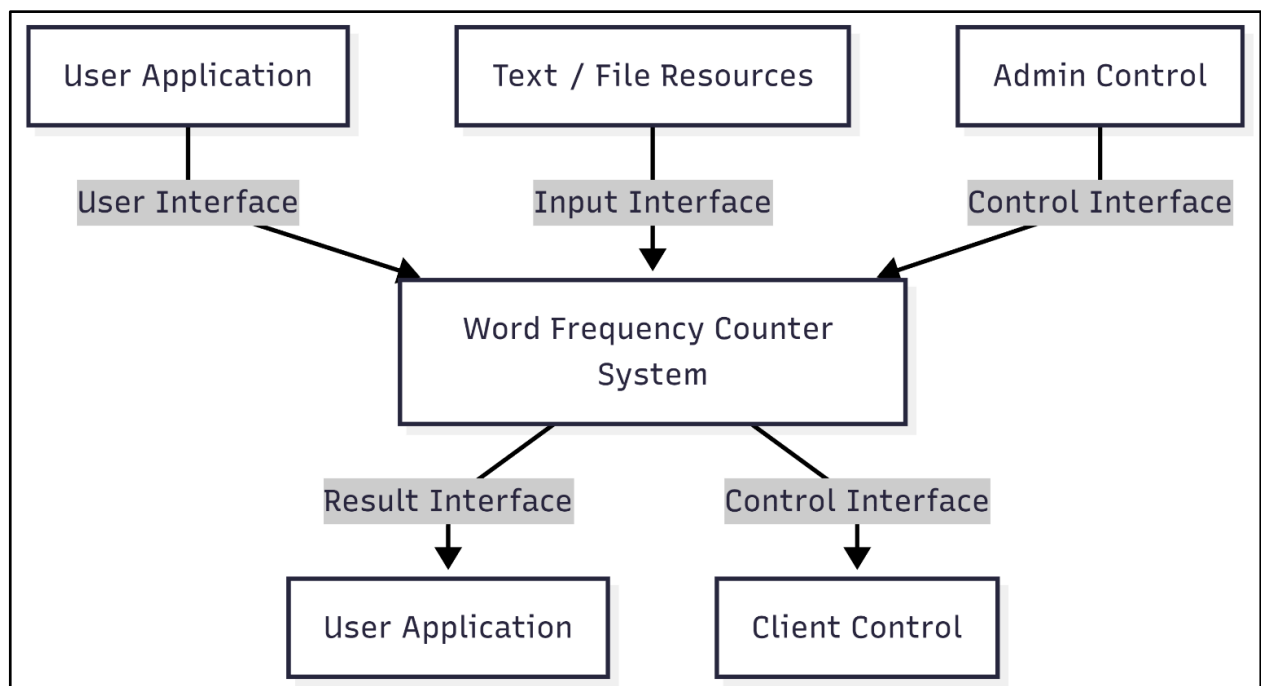




### 3.2.1 SYSTEM ARCHITECTURE



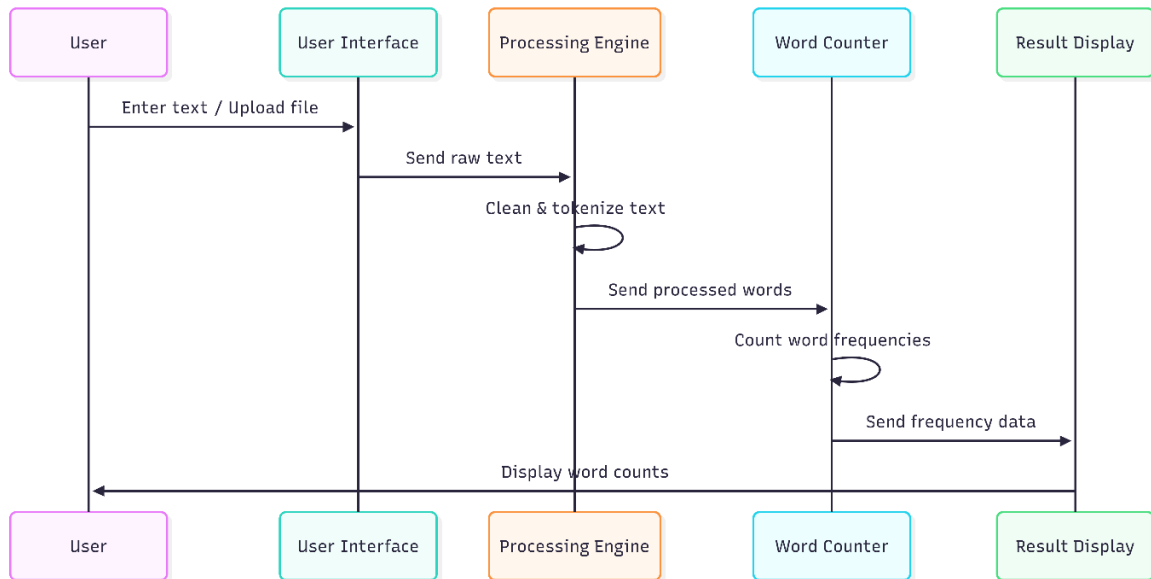
### 3.2.2 SYSTEM INTERFACES



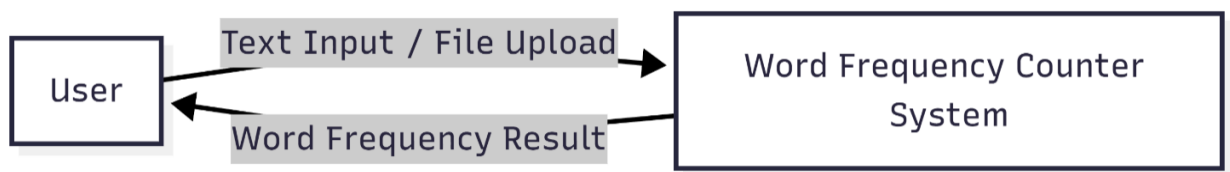


### 3.3 DATA FLOW DIAGRAM

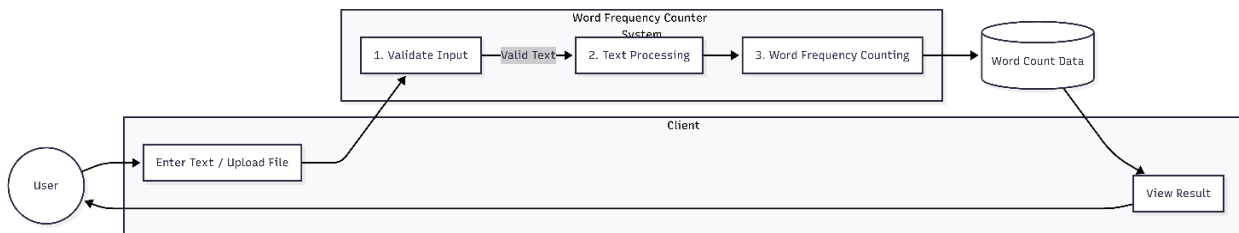
#### 3.3.1 DATA PACKET FLOW DIAGRAM



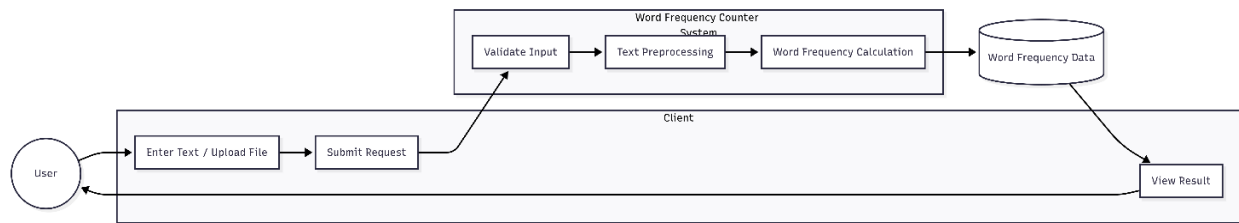
#### 3.3.2 LEVEL 0 DESIGN DIAGRAM



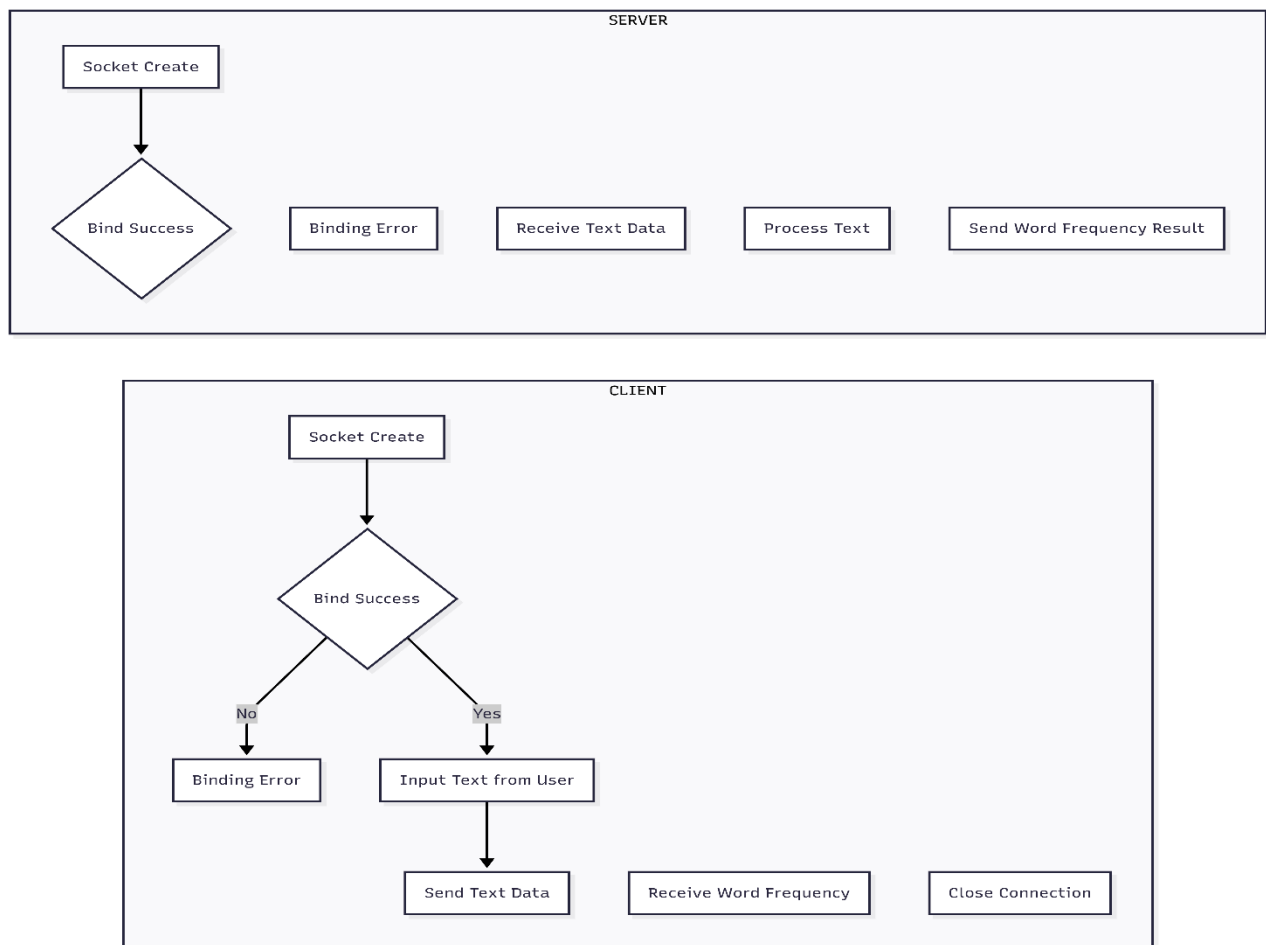
#### 3.3.3 LEVEL 1 DESIGN DIAGRAM



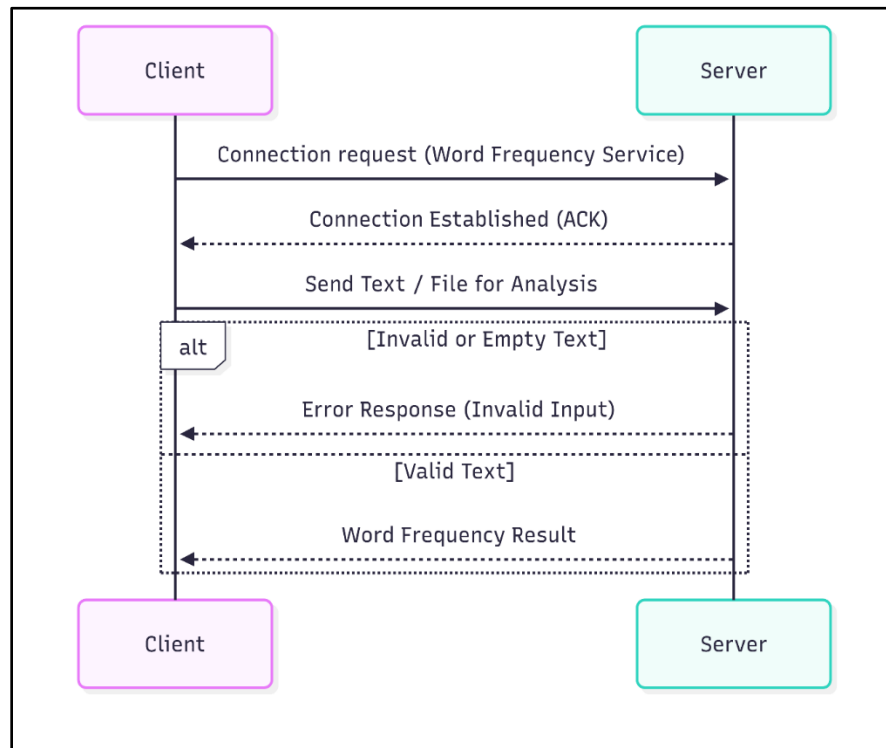
### 3.3.4 LEVEL 2 DESIGN DIAGRAM



### 3.3.5 DATA MAPPING INFORMATION



### 3.3.6 SEQUENCE DIAGRAM



### 3.4 FILES

The Word Frequency Counter system reads text input either directly from the user or from text files provided by the client. The system processes these files to analyze and compute word frequencies. The text files can be read, analyzed, and reprocessed as needed since no authentication is required.

### 3.5 USER INTERFACE

- Desktop or a Linux machine with internet connection.
- Command Line Interface (CLI).

### 3.6 ERROR HANDLING

Error handling is used for showing error messages when an error occurs while executing commands. This helps users to understand the flow and can identify where the system is going wrong and attempt to rectify the issue.

- It will check whether the input text or file is valid.
- Errors related to input handling, text processing, and communication between client and server are handled.
- **1. E\_SUCCESS:** When the requested operation is completed successfully, this message is displayed.
- **2. E\_INPUTEMPTY:** When the user provides empty text input, this error message is displayed.
- **3. E\_FILEEXISTS:** When a user tries to create or save a result file with a name that already exists, this message is displayed.
- **4. E\_FILENOTFOUND:** When the user tries to access a text file that does not exist, this error message is displayed.
- **5. E\_INVALIDINPUT:** If the input contains unsupported or invalid characters, this error message is displayed.
- **6. E\_ABORTCMD:** When the operation is not performed properly or is interrupted by the user, the abort command is executed to stop the process.
- **7. E\_BADFILESELECTED:** When the selected file format is not supported, this message is displayed.
- **8. E\_PROCESSFAIL:** When an error occurs during text processing or word frequency calculation, this error message is displayed.
- **9. E\_FILETOOBIG:** When the selected file is too large to be processed by the system buffer, this error message is displayed.
- **10. E\_BADFORMAT:** When the input text or file is not in the proper desired format, this error message is displayed.

### 3.7 HELP

Help will come in the form of all the documentation created prior to coding, which explains the intended usage of the Word Frequency Counter system. Should time allow, detailed instructions will be written on how to create, run, and implement the system with the intention of publishing it as an Open Source solution.

### **3.8 PERFORMANCE**

The performance of the system is expected to provide quick responses for word frequency computation. The performance requirements include efficient software design and optimized coding so that text analysis and result generation are executed with minimal delay.

### **3.9 SECURITY**

The Word Frequency Counter system does not require a username or password for accessing its basic functionality. Since the system is designed for text analysis, security risks are minimal. To prevent misuse, access can be restricted to specific directories from which text files can be read. The system can be executed with limited user permissions to ensure that only allowed files are processed.

### **3.10 RELIABILITY**

The Word Frequency Counter system is available whenever the server is running and the client is connected through a valid network connection. The system provides consistent and reliable results for valid text inputs, although network-related issues may affect communication between client and server.

### **3.11 MAINTAINABILITY**

Very little maintenance is required for this system. Initial configuration is the only required interaction after deployment. Any future maintenance may include updating text processing logic or modifying system settings. Hardware or software upgrades may result in temporary downtime.

### **3.12 PORTABILITY**

This system is portable and can be executed on different Linux-based environments. The source code can be compiled and run across various Linux distributions, provided the required compiler and libraries are available.

### **3.13 REUSABILITY**

The code written and components used in the Word Frequency Counter system are reusable and modular. With minimal modification, the same code can be extended or integrated into other text analysis or natural language processing applications.

### **3.14 APPLICATION COMPATIBILITY**

The Word Frequency Counter system operates as a client-server application and can be executed over standard network protocols. It is compatible with Linux-based systems and can be integrated with other applications that require text analysis or data processing functionality.

### **3.15 RESOURCE UTILIZATION**

NA

### **3.16 MAJOR CLASSES AND CHARACTERISTICS**

The major classes in the Word Frequency Counter system include:

- Text\_Input
- Word\_Frequency\_Processor
- Client\_Module
- Server\_Module

# Low Level Design Document

## 4. INTRODUCTION

The Word Frequency Counter system is a multithreaded C++ application that analyzes multiple text files and counts the frequency of unique words, while leaving out common and invalid words. The system processes each input file at the same time using POSIX threads, which makes good use of system resources. The application keeps a shared hash table to store word frequencies and records invalid words in a separate file. This Low Level Design (LLD) document gives a detailed technical description of the internal components, class responsibilities, interactions, and constraints of the system.

### 4.1 PURPOSE

- To describe the low-level design and internal working of the Word Frequency Counter system.
- To translate the functional requirements and technical requirements defined in the SRS into detailed design elements.
- To explain the internal components, classes, and modules used in the system.
- To describe the data structures (such as hash tables) and algorithms used for word processing and frequency counting.
- To define the threading model, including child thread creation and synchronization using mutexes.
- To provide clear guidance on file handling, word validation, and error handling mechanisms.
- To serve as a technical reference for developers during implementation.
- To assist testers in understanding internal logic for unit and integration testing.
- To help evaluators and reviewers verify that the system design correctly satisfies the SRS requirements.

## 4.2 DOCUMENT CONVENTIONS

The document uses the “Times New Roman ” font with bold for the heading. The main heading size is 18. The Sub-heading size is 14pointand sub heading under the Subheading is of 12points. The content under the sub-headings is of 12 points size. All the spacing are normal/default spacing of MS Word.

## 4.3 INTENDED AUDIENCE AND READING SUGGESTIONS

This project is a prototype for everyone who wants to make file transfer between client and server. This document is intended to be read by developers, testers, Project Managers and Customers. This is the technical document and the terms should be understood by all of them.

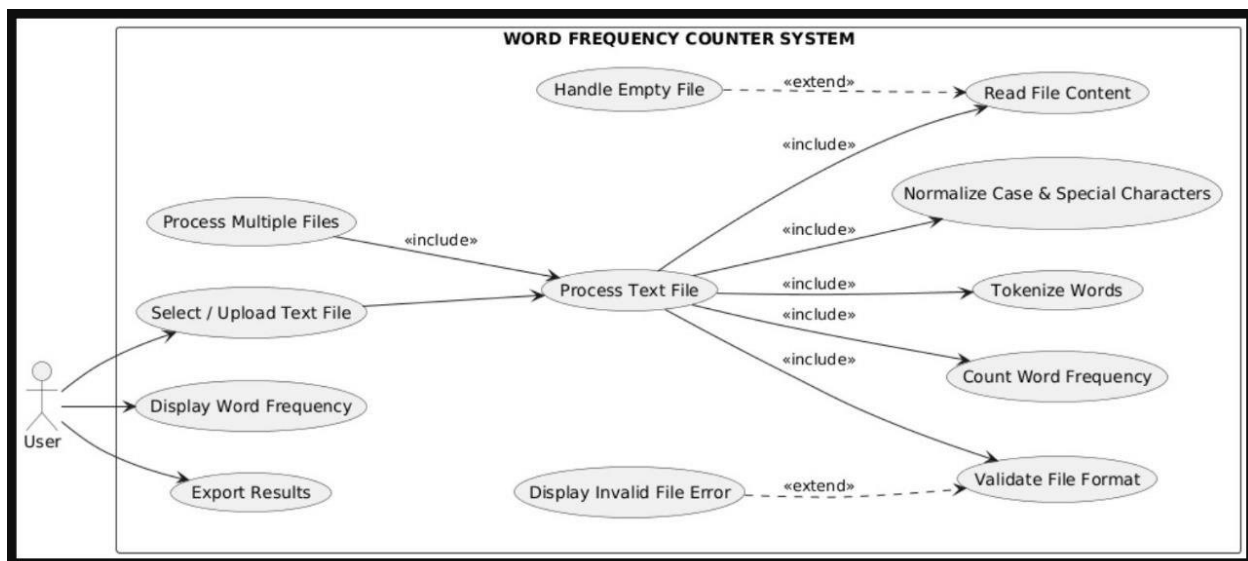
## 5. DETAILED SYSTEM DESIGN

The system design is as follows:

### 5.1 DESIGN DESCRIPTIONS

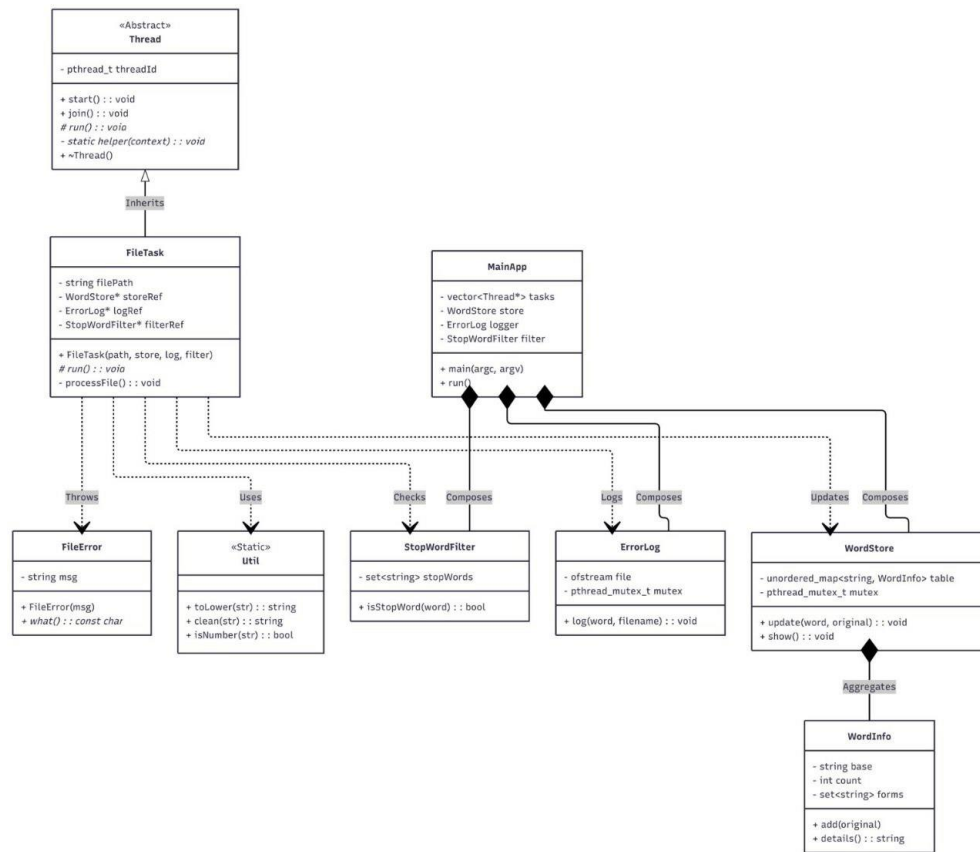
The main design features include four major parts: the architecture, the user interface design, the files, process relation, and automation. In order to make these designs easier to understand, the design has been illustrated in attached diagrams ( Use Case, Data flow diagrams).

### 5.2 USE CASE DIAGRAM

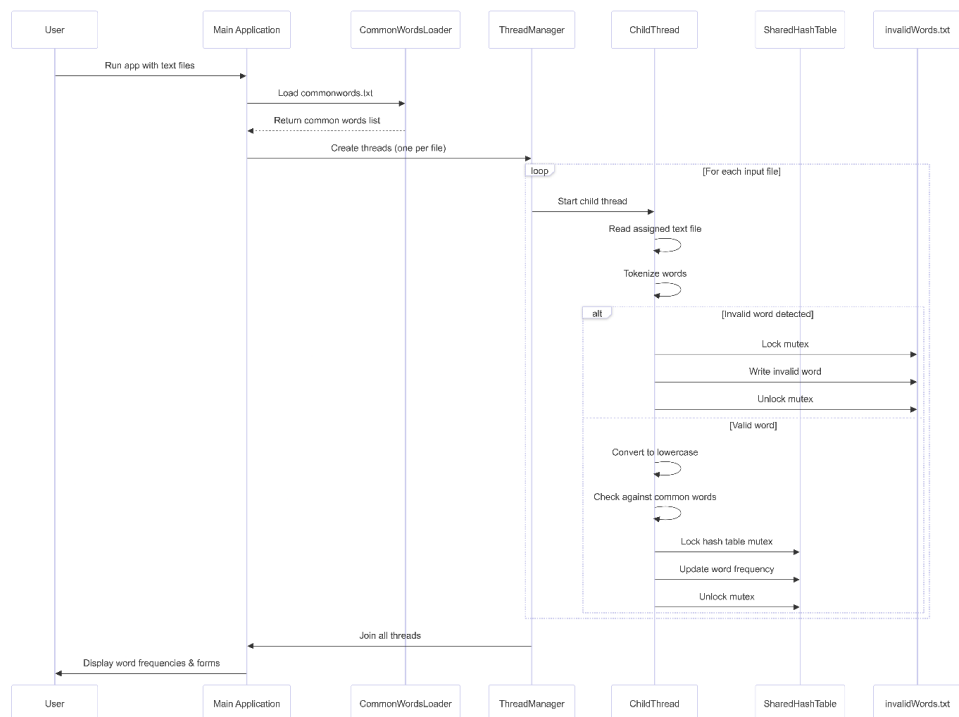




## 5.3 CLASS DIAGRAM



## 5.4 SEQUENCE DIAGRAM



## **5.5 DESIGN AND IMPLEMENTATION CONSTRAINTS**

The system must be implemented using C++ only

## **5.6 USER INTERFACE**

- Desktop or a linux machine with internet connection.
- Command Line Interface (CLI).

## **6. SECURITY**

Since the Word Frequency Count System processes user-provided text files, care must be taken to prevent unauthorized file access and ensure data integrity. The system is designed to read only permitted text files from predefined locations and does not allow modification of system or protected files. Input validation is performed to avoid processing malicious or unsupported file formats. Access to shared resources is controlled to prevent unintended data corruption during concurrent processing.