Software Requirements Specification (SRS) Document for EZ-Admin – Linux Server Management Application

Prepared by: -

- Krithick Kalyan T
- Vishwa K
- Sai Ram R
- Arvind M Bharadwaj
- Surya K
- Pradeep V
- Aravind R

Contents

1. Introduction

1.1 Introduction

The purpose of this document is to define and describe the requirements of the project and to spell out the system's functionality and its constraints.

1.2 Scope of this Document

The customer and the user for the system are the primarily web server administrators using Apache2 server software and the developers of the system is Group – 1. Our constraints for this section includes our deadline for the document which is due 28th Feb 2022.

1.3 Overview

The product is a console-based menu driven application for Apache2 Server Administration on a Linux distro written in C++, that eases the operations to be performed on the server. The operations include: -

1. Starting/Stopping/Restarting the server
2. Scheduling the server
3. Backing up server files
4. Viewing/Clearing server logs
5. Filtering client logs to monitor server activity

2. General Description

2.1 Product Functions

The product should make server administration process easier and streamlined for the server Admin and time efficient at the same time.

2.2 Similar System Information

The product is being developed with C++, so there are a number of programs that are used for a wide array of purposes. The advantage of our system is usage of Apache2 Server Software, which is open-source and widely used in HTTP servers.

2.3 User Characteristics

The users are mainly web server administrators, as it is based on Apache2. For this application, the user is required to know the basics of Apache2 as well as a fundamental understanding of Linux operating system.

2.4 User Problem Statement

The user is required to memorize various commands for operation of the Apache2 server and typing the commands explicitly consumes a lot of time leading to redundant work and reduced efficiency.

2.5 User Objectives

The user desires a software product that establishes an interface to manage the server. The program must faciliatate the speed and ease of input, while keeping a record of the server activity.

2.6 General Constraints

Constraints include an easy to use interface for the program through terminal on a Linux distro with Apache2 installed. Also, it must be constructed in C++,BASH or another related program that is easily learnable.

3. Functional Requirements

1) **Start / Stop / Restart server**
    a. The application can start/stop/restart the server.
    b. Once the server is running, a real-time client logs window is provided to monitor devices connecting to the server.

2) **Backup the server**
    a. The application can back up the server files in a compressed format
    b. Once the backup is completed, user can view the file using the prompt provided in the application

3) **Scheduling server**
    a. User can schedule basic server operations such as start, stop and restart to execute at a given time on the same day
    b. User can also clear all scheduled jobs using the option provided

4) **View / Clear Server logs**
    a. All server activity such as start, stop and restart performed on the server are logged in a separate file
    b. The file can be viewed to monitor server activity
    c. The logs can be cleared only if the user can provide the appropriate authentication

5) **Filtering client logs**
    a. User can search the client logs for connections established and filter them according to different criteria
    b. The criteria include IP Address, date and month

6) **Help menu**
    a. The user can utilize the help menu (built using the man database in built in Linux) to acquaint themselves with Linux commands

4. Interface Requirements

4.1 User Interfaces

- **4.1.1 GUI**
  There is no graphical user interface

- **4.1.2 CLI**
  The user interface is provided by the Linux Terminal. It has a menu driven interface to operate and monitor the server

- **4.1.3 API**
  There is no API for the product

5. Performance Requirements

The application is designed to be operated on a Linux Terminal, thus no additional system requirements exist beyond those required to run Linux and Apache2. The Linux distro must be based on Debian as core functionalities of the application are designed to run on a Debian-based Linux distro

The minimum requirements for running an Apache2 HTTP server are:

- 1 GHz processor or higher
- 2GB RAM or higher
- 15GB Available Hard Drive Space
- Linux kernel version 2.6 or higher

6. Other non-functional attributes

.

6.1 Security

The system shall be designed with a level of security appropriate for the sensitivity of information enclosed in the server. More interaction is needed with client about the volatility of the information. There is no obvious information that is of a high security level such as credit card information.

6.2 Binary Compatibility

This system will be compatible with any computer that has Linux operating system (based on Debian) and is designed to work with Apache2 web server.

6.3 Reliability

Reliability is one of the key attributes of the system. Back-ups will be made regularly so that restoration with minimal data loss is possible in the event of unforeseen events. The system will also be thoroughly tested by all team members to ensure reliability.

6.4 Maintainability

The system shall be maintained by the server admin or delegated to another employee, who has fundamental understanding of Linux commands and Apache2 server software.

6.5 Portability

The system shall be designed in a way that shall allow it to be run on multiple computers with Linux operating system and Apache2 web server installed.

6.6 Extensibility

The system shall be designed and documented in such a way that anybody with an understanding of Apache2 shall be able to extend the system to fit their needs with the team’s basic instructions.

6.7 Reusability

The system should be designed in a way that allows the console application to be re-used regularly for the various server operations that are to be performed on a daily basis.

6.8 Serviceability

The maintenance of the system should be able to be sufficiently performed by any person with a basic understanding of Linux.

7. Operational Scenarios

Scenario A: Initial Server Operations

The user can select an option from the menu-driven interface, to start, stop or restart the server.

Scenario B: Server Backup

The user can take a backup of the server files in a compressed format on the machine and the files will be saved according to a time stamp.

E.g: backup_15-02-2022.tar.gz

Scenario C: Server Scheduling

The user or server admin can schedule server activity that should take place according to the input time. These activities include start, stop and restart operations

Scenario D: Server Logs Maintenance

All the server operations performed by a particular user are logged into a log file. The logs can be viewed for monitoring server activity. With authorized access, the logs can be cleared as well.

Scenario E: Client Logs Maintenance

The connections to the server by a client can be monitored real time. The activity can be filtered according to IP Address, date or month

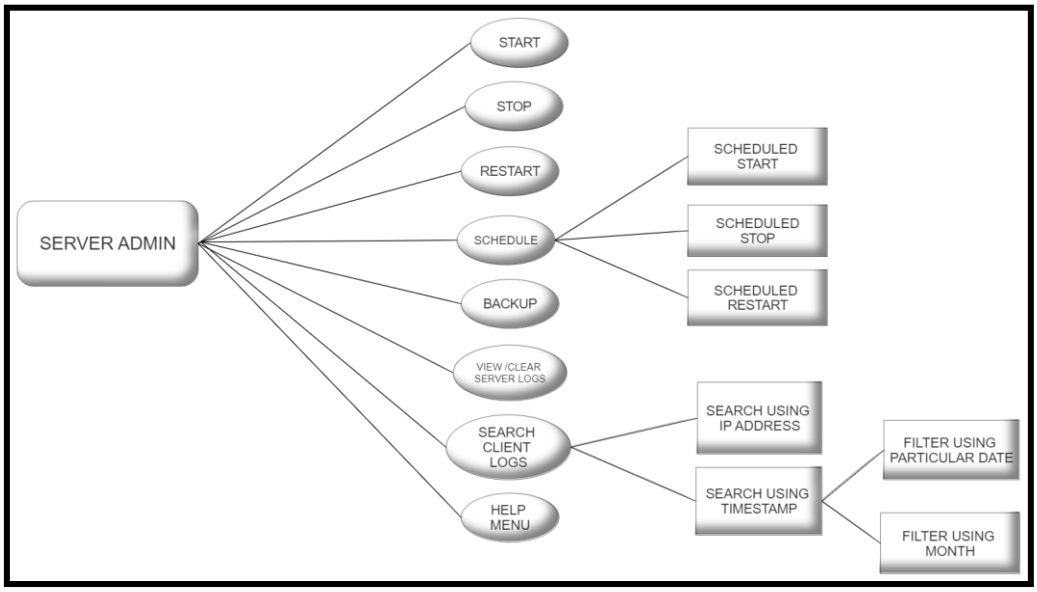
Scenario F: Help Menu

The menu driven interface has an inbuilt help menu to assist the user in case any commands are unfamiliar

8. Preliminary Use Case Models and Sequence Diagrams

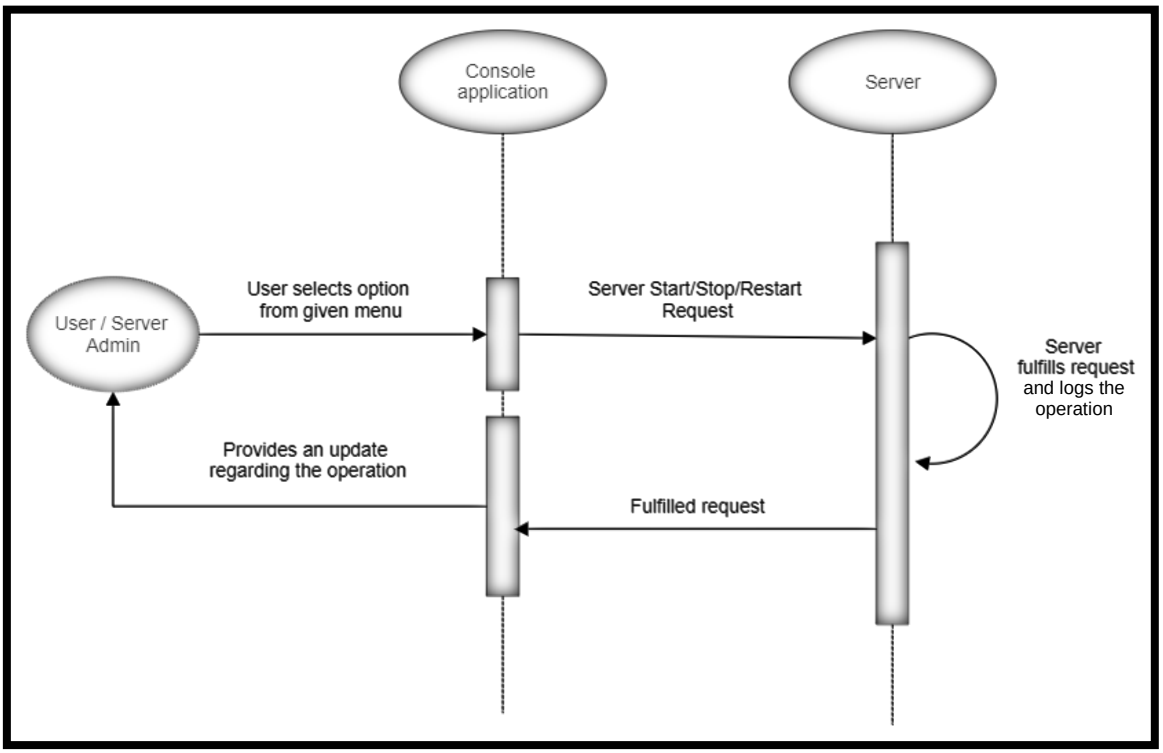
This section presents a list of the fundamental sequence diagrams and use cases that satisfy the system’s requirements. The purpose is to provide an alternative, "structural" view of the requirements stated above and how they might be satisfied in the system.

8.1 Use Case Models

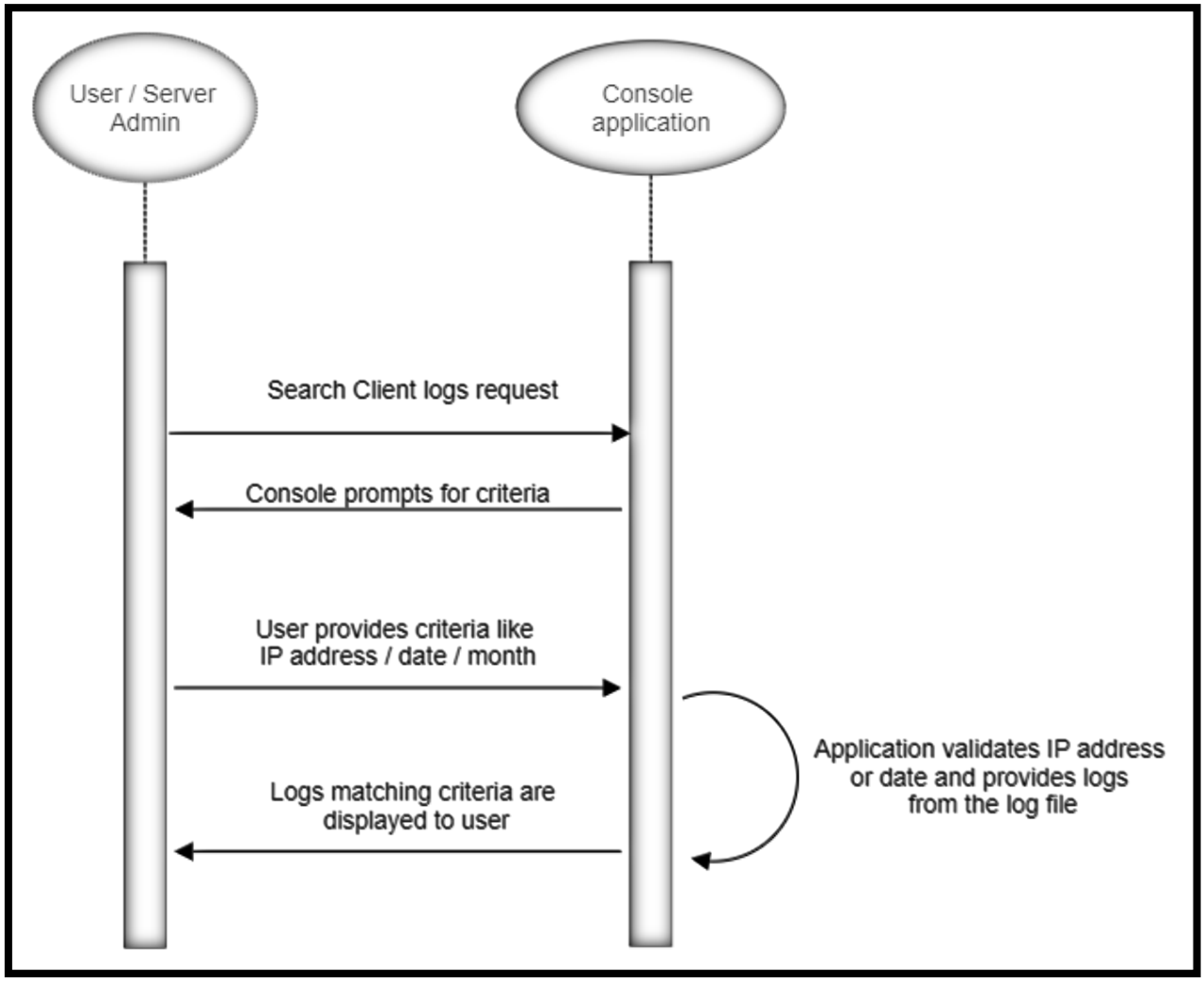


8.2 Sequence Diagrams

Flow diagram of start / stop / restart operation



Flow diagram of searching / filtering client logs



9. Appendices

9.1 Definitions, Acronyms, Abbreviations

GUI – Graphical User Interface

CLI - Command Line Interface

API – Application Programming Interface

HTTP – Hyper Text Transfer Protocol

BASH – Bourne Again Shell