

SOFTWARE REQUIREMENT SPECIFICATIONS

Project Name	: Database Server
Document Title	: SRS
Project Type	: Client Server
Project Timeline	: 02.01.2023 to 12.01.2023



INDEX

1. Introduction

1.1 Purpose

1.2 Intended use

1.3 Scope

2. Overall Description

2.1 Assumption and dependency

3. System Feature and Requirements

3.1 Functional Requirements

3.1.1 R_001 start_Server

3.1.2 R_002 Manage_Client

3.1.3 R_003 Connection Establishment

3.1.4 R_004 Get_Info

3.1.4.1 M_001 Get_put

3.1.4.2 M_002 Get_get

3.1.4.3 M_003 Get_Delete

3.1.4.4 M_004 Get_Quit

3.2.5 R_005 Valid_Set_Index

3.2.6 R_006 Find_Id

3.2 Technical Requirements

3.3 System Requirements

3.3.1 Tool to be used

3.3.2 Hardware Requirements

3.4 Non-Functional Requirements

4. Data Flow Diagram

4.1 Flow chart

1.Introduction:

This Introduction of the Software requirement specification provides an overview of the entire software. The entire SRS with overview description purpose, scope, tools used and basic description. The aim of this document is to gather, analyze and give an in-depth insight into the complete database server application by defining the problem statement in detail. The detailed requirements of database server application is provided in this document.

1.1 Purpose:

The purpose of this document is to show the requirements for the “Database Server”, in which multiple clients will request the Server to store the data and retrieve the data from database depending on the type and format of the data.

1.2 Indented Use:

- Development team
- Maintenance team
- Clients

1.3 Scope:

An application to store and retrieve different type of data such as Employee data, School data, etc. from the database server. A list of supported data types and format is published by the server. All clients connecting to server will send data to store in the server and they can retrieve the data from Server.

2.Overall Description:

This project aims to create development of a Database Server Application. The server should maintain different types and formats of data. The server will receive the type and format of data from the client followed by actual data entry and the server will add to Database. The server should store the data in an appropriate structure and return success or failure to client. The will take the input from user based upon the type and format. The server should accept at least two Clients. If the two clients are trying to modify the structure, the server will throw an error to clients.

2.1 Assumption and Dependency:

- System should have Linux
- Putty to be installed
- System should have either 4GB or more RAM
- The service is used preferably on a desktop or laptop

3. System Features and Requirements:

3.1 Functional Requirements:

3.1.1 R_001 Start_Server (): As the project is based on client and server connection its, hence first we need to invoke the server to connect the client.

3.1.2 R_002 Manage_Client (): As the Server is connected to multiple client since we need to manage the client.

3.1.3 R_003 Connection _Establishment (): Here the system checks where the client and server had successfully connected with client or not. The server will store the data in an appropriate structure and will return the status as either success or failure to the client.

3.1.4 R_004 Get_info (): The client will can retrieve the data on the request of the user and in case of any error, and error message will be sent.

3.1.4.1 M_001 Get_put (): From menu driven the client will choose the option respectively. Here the put option indicated addition of data into the database.

3.1.4.2 M_002 Get_get (): On the request of client the server will get the data from the database and display it.

3.1.4.3 M_003 Get_delete (): On the request of user the existing element will be deleted from the database.

3.1.4.4 M_004 Get_quite (): Client exit functionality

3.1.5 R_005: Valid_Set_Index (): The functionality specifies the index value to retrieve the data from the data server

3.1.6 R_006 Find_Id (): Searching a particular element from the database.

3.2 Technical Requirements:

3.2.1 DS_TR01 – Process Synchronization: It is the way by which processes that share the same memory space are managed in an operating system. Here, the server ensures the protection of data while two clients accessing the same database i.e., Employee data there by avoiding conflicts

3.2.1 DS_TR02 – Shared Memory in Linux : The same database contains different types and formats of data. Mutex can be used for locking when two clients attempt to modify same database.

3.2.3 DS_TR03 - Socket Programming in C – TCP : Socket programming is the way of connecting two nodes, here the client and server on a network communicate with each other to send and receive the data .

3.2.4 DS_TR04 - Support for Statistics : Server is responsible for the display of statistics related to maintain the various type and formats of data.

3.2.5 DS_TR05 – Multiplexing : I/O Multiplexing is the ability to perform I/O operations on multiple file descriptors.

3.3.6 DS_TR06 – Logging and Debugging Framework: Linux logs provide a timeline of events for a valuable troubleshooting tool when encountering issues. When issues arise, analyzing log files facilitates debugging.

3.3 System Requirements:

3.3.1 Software Requirements:

- Operating System - Windows 11
- Server-side script - C Language
- IDE - putty
- Libraries used - C libraries

3.3.2 Hardware Requirements:

- Processor - I3/Intel Processor
- RAM - 4 GB (min)
- Hard Disk - 128 GB
- Keyboard - Standard Windows Keyboard
- Mouse - Two or Three Button Mouse
- Monitor - Any

3.4 Non-Functional Requirements:

- **Supportability:**

The system is easy to use

- **Design Constraints:**

The System is built using only C language

- **Reliability & Availability:**

The system is available 24/7 that is whenever the user would like to use the system, they can use it up to its functionalities.

- **Performance:**

The system will work on the user's terminal.

4. Data Flow Diagrams



