

Software Requirements Specification

for

Remote phonebook server for client

**Group 3rd**

**Batch: E2E-Manipal-Systems C Linux System Programming Aug 11th batch**

**SPRINT 2**

**Implementation**

**Project Timeline October 12th 2022 to October 18th, 2022**

# Table of Contents

**Contents**

1. Introduction 1.1

1.2

1.3

1.4

1.5

1. Overall Description 2.1

2.2

2.3

2.4

2.5

2.6

2.7

1. System Features 3.1

3.2

1. External Interface Requirements 4.1

4.2

4.3

4.4

1. **Other Non-functional** Requirements 5.1

5.2

5.3

5.4

6. Data flow diagrams

1

Purpose Document Conventions

Intended Audience and Reading Suggestions

Project Scope References

2

Product Perspective Product Features

User Classes and Characteristics Operating Environment

Design and Implementation Constraints

User Documentation Assumptions and Dependencies

3

System Feature 1 System Feature 2 (and so on)

4

User Interfaces Hardware Interfaces Software Interfaces Communications Interfaces

5

Performance Requirements Safety Requirements Security Requirements Software Quality Attributes

6

1. Introduction

#### Purpose

* + - To provide a comprehensive solution for Remote phone book servers for clients. As we all know People have been relying on the phone book for many years to get basic info such as Phone no, address etc. of the people they need to contact. However, this method is not so efficient since users may not get the desired results.
  1. Document **Conventions**

|  |  |  |
| --- | --- | --- |
| TYPE | FONT SIZE | TYPE STYLE |
| Headings | 18 | BOLD |
| Sub-Headings | 14 | BOLD |
| Contents | 12 | Normal |

### Intended Audience and Reading Suggestions

* + - Developers
    - Testers

#### Project Scope

* + - Client can Add, delete and modify the phonebook with credentials.

### References

* + - https://1000proiects.org/phonebook-application-c-project-report.html
    - RFC 793 standard for Transmission control protocol
    - RFC 959 standard for File transfer protocol
    - [https://www.researchgate.net/publication/239997](http://www.researchgate.net/publication/2399976Context-Phonebook)6 [Context-Phonebook](http://www.researchgate.net/publication/2399976Context-Phonebook)

1. OVERALL **DESCRIPTION**

### Product Perspective

* + - The project to be developed is a Phonebook system.
    - It is to view and edit the contacts in secure programming which leads to storing and retrieving data.
    - It also consists of an authentication system to secure client's data.

### Product Features

* + - It is a phonebook application that contains names and contact numbers of each person who are stored in data structure.
    - The server will support a virtual user with a name to add contact to a public group only without any password. It will also support the concurrent client connection.
    - An administrator can access any contact in addition to adding, removing groups to the phone directory.
    - The client program will start with an authentication.
    - To Quit from the phonebook the user or client use “bye” to end the connection.

## User Classes and Characteristics

This system consists of two login administrators and an anonymous user and authenticated user.

## Administrator

* + - This user is responsible for changing the passwords, include and exclude from the phone directory.
    - They can access any contact in the phone book.

## Anonymous

* + - A valid user can make use of the system to login and view the profile.
    - Login:
    - Users need to login every time before viewing their profile.
    - View Profile:
    - Users can view the profile and different services offered.

## Operating Environment

* + - Operating system Used — Linux Debian 5.10.106-1.
    - Language Used — C, Linux system Programming.

## Design and Implementation Constraints

* + - Login and password are used to identify clients and there is a facility for guests.

## User Documentation

* + - All user information is maintained, and it contains confidential information of the user.

## Assumptions and Dependencies

* + - Stable Internet connections
    - Every user should be comfortable working with a computer.
    - They must have basic knowledge of the English language.

1. **Client Server Connection**

## Description and Priority

* + - TCP Client can be able to connect with remote TCP servers.
    - This is the Highest priority feature.

## Stimulus/Response Sequences

* + - Users need to login using their credentials.
    - On successful login, the system will connect the user with the TCP server.
    - Users enter wrong credentials.
    - System will respond with appropriate error messages.
    - For Anonymous users, no need to login.
    - System will directly connect with the TCP server.

### **Functional Requirements:**

* + - REQ-1: - RPBC\_01| Client Server Connection: A connection between client and server must be established first.
    - REQ-2: - RPBC\_02| User Authentication: The server should authenticate the user from a poll of registered user data kept with appropriate data structure.
    - REQ-3: - RPBC\_03| Server Handshaking: The user should get the customize message from server entering the user id but before providing the password by which the user get conformation that it got connected to the proper server.
    - REQ-4: - RPBC\_04| Server-side Firewall Protection: The server should listen to port 8028 for client connection and make the client side computer access the server port
    - REQ-5: - RPBC\_05| Server side Security: Before allowing a client connection the server should match the client IP with the blacklist IP specified in /etc/phone/ client\_blacklist .
    - REQ-6: - RPBC\_06| Service Starting at Boot: The server must configure to start the service automatically at boot.
    - REQ-7: - RPBC\_07| Anonymous user access: Server should support an virtual user with name anonymous to add contact to public group only without any password.
    - REQ-8: - RPBC\_08| Authenticated User Access: All the authenticated user should be allowed to add, view and delete contact from phone directory for the group they belongs to. They should view the content of public group.
    - REQ-9: - RPBC\_09| Authenticated Admin Access: An administrator can access any contact in addition to add, remove group to phone directory
    - REQ-10: - Server side Concurrency: The server should support the concurrent client connection.
    - REQ-11: - RPBC\_11| Client side program starting: The user should start the client program to connect the Server whenever he wants.
    - REQ-12: - RPBC\_12| Client Side Connection request: Client should request a connection to server on port 8028.
    - REQ-13: - RPBC\_13| Client Side Authentication: Client program should start with an authentication (default to anonymous).
    - REQ-14: - RPBC\_14| Client Side User Environment: On successful authentication the user should placed on public group for phone book access.
    - REQ-15: - RPBC\_15| Client Side Environment Customization: User can use chgrp GRPNAME to change his Working group on phone book.
    - REQ-16: - RPBC\_16| Client Side Browsing Contact: User can use list sub command to browse the contact list in the working Group. He can use list [A] sub command to list all contact start with A in the working Group and [Ab] to list all contact starting with A followed by b and so on.
    - REQ-17: - RPBC\_17| Client Side contact add: User can add an contact to working Group by using Add sub command with NAME & PHONENO.
    - REQ-18:- RPBC\_18| Client Side contact deletion: User can remove an contact from working Group by using rm sub command with NAME
    - REQ-19:-RPBC\_19| Client Side Group Add & Remove: Admin user must use the addgrp rmgrp sub command with GRPNAME to add and remove group.
    - REQ-20:-RPBC\_20| Client Side Quit: The user should use a sub command bye to terminate the Connection.

# External Interface Requirements

### User Interfaces

* + - The application interacts with CLI (Command Line Interface). The interface is simple and easy to handle. The user can easily come into the flow with the application and use all interface properly

### Hardware Interfaces

#### Client Side:

* + - * Laptop interface which is used in client side

### Server Side:

* + - * Files stored in a db is used from the server side

### Software Interfaces

* + - Client software, web browser
    - Linux operating system
    - A kernel shell which is a command line interpreter

#### Communications Interfaces

* + - TCP Sockets which are used for communication between the server and the client.

1. OTHER NON-FUNCTIONAL REQUIREMENTS

### **Performance Requirements**

* + - It is dependent on how the system is constructed; customers of the products may not be aware of how the software is designed; hence, we require a user-friendly and highly efficient system.
    - As a result, good programming will assist in boosting the efficiency of the system, resulting in efficient performance.

### **Safety Requirements**

* + - When privacy concerns are taken into account, i.e., if a harmful part of society gains access to a person's personal information, the application can be used for an unethical purpose. As a result, extremely high access security levels must be implemented.

## Security Requirements

* + - When the intended software is used, it is vulnerable to all possible dangers, such as misuse or attempted modification.
    - A decent user interface to access data should be employed in these circumstances, such as requiring the client to input credentials in order to access information from the server.
    - Any user who wants to see his profile must provide his credentials so that scammers cannot access the information.

## Software Quality Attributes

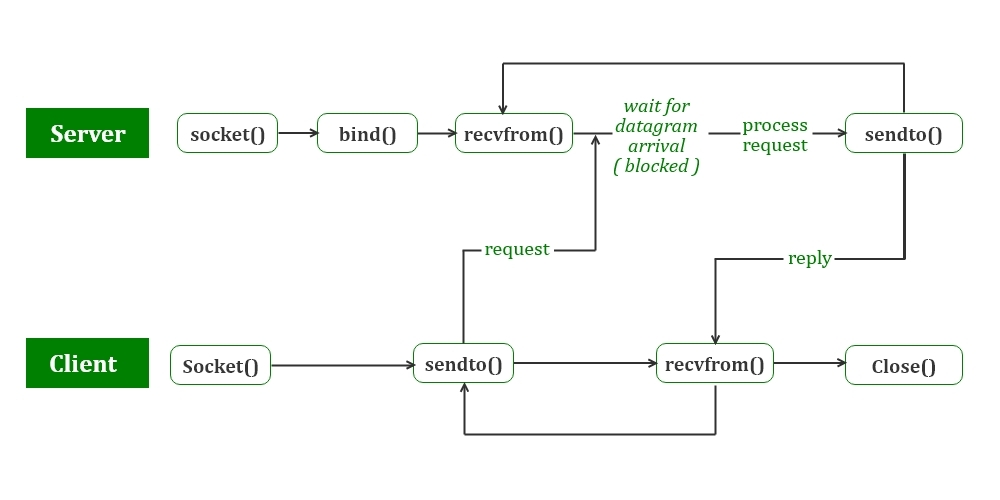
* + - Additional quality characteristics for the product that will be important to either the customer or the developer, such as reliability, maintainability.

**Reliability:**

* + - The software is used by users to access information, and it is critical that the information viewed is accurate.
    - If a person's address or phone number changes and the change is not reflected in the data structure, it may eventually result in data inconsistency.
    - As a result, the software must be more dependable, which can be accomplished through data logs.

**Maintainabilitv:**

* + - There may come a moment when the software needs to be improved or upgraded; in such case, the code must be very adaptable, i.e., updating the code should not degrade the application.

1.  DATA FLOW DIAGRAMS

