

Software Requirements Specification (SRS)

Remote Desktop Application

Version: 1.0

Date: October 2024

Table of Contents

1. Introduction.....	2
1.1 Purpose	2
1.2 Scope.....	2
1.3 Definitions, Acronyms, and Abbreviations	2
1.4 References	2
2. Overall Description	3
2.1 Product Perspective.....	3
2.2 Product Features.....	3
2.3 User Classes and Characteristics.....	3
2.4 Operating Environment	3
2.5 Design and Implementation Constraints.....	4
2.6 Assumptions and Dependencies.....	4
3. System Features	4
3.1 Real-Time Screen Sharing.....	4
3.2 Remote Control	4
3.3 Remote Shutdown/Restart	5
4. External Interface Requirements	5
4.1 User Interfaces.....	5
4.2 Hardware Interfaces	5
4.3 Software Interfaces	5
4.4 Communication Interfaces.....	5
5. Non-Functional Requirements.....	6
5.1 Performance Requirements	6
5.2 Security Requirements.....	6
5.3 Usability Requirements	6
5.5 Maintainability Requirements	6
6. Other Requirements.....	6
6.1 Legal and Regulatory Requirements.....	6

1. Introduction

1.1 Purpose

The purpose of this SRS document is to outline the functional and non-functional requirements for the development of a Remote Desktop Application. This application will enable users to access, control, and manage a remote computer by sharing the screen in real time and providing functionalities such as remote shutdown or restart.

1.2 Scope

The Remote Desktop Application allows a user to remotely access another computer, view its screen, and interact with it as though they were physically present. The app will enable secure, real-time screen sharing, mouse, and keyboard control, as well as remote shutdown or restart functionalities. It is intended for both personal and professional use, offering support for Windows, macOS, and Linux platforms.

1.3 Definitions, Acronyms, and Abbreviations

- RDP: Remote Desktop Protocol
- SSH: Secure Shell
- GUI: Graphical User Interface
- LAN: Local Area Network
- FR: Functional Requirement
- GDPR: General Data Protection Regulation

1.4 References

- IEEE Standard for Software Requirements Specifications (IEEE 830-1998).
- User manuals for Windows, macOS, and Linux systems.
- Data Protection and Privacy Act of Uganda, 2019.

2. Overall Description

2.1 Product Perspective

The Remote Desktop Application will act as a client-server system where the client (user) can access the server (remote computer) via the internet or LAN. It will utilize RDP or other secure protocols like SSH for remote access. The system will generate special tokens to each device.

2.2 Product Features

- Real-Time Screen Sharing: The remote desktop's screen will be shared in real time with the user.
- Remote Control: The user can control the mouse and keyboard of the remote computer.
- Remote Shutdown/Restart: The user can initiate a shutdown or restart of the remote machine.
- Multiple Platform Support: Available on Windows, macOS, and Linux.
- Secure Connection: All connections will be encrypted to ensure data protection.

2.3 User Classes and Characteristics

- Basic Users: Individuals who need to access their home or work computers remotely.
- IT Professionals: Administrators and support personnel who need to troubleshoot systems remotely.
- Organizations: Businesses that require remote access for employees working from different locations.

There are two primary user groups for this system:

- Client/Controller: The computer that will control another computer (host).
- Host: The computer that is being controlled remotely.

2.4 Operating Environment

The application will function on the following environments:

- Client Devices: Windows, macOS, Linux, iOS.
- Host (Remote Machines): Windows, macOS, Linux.
- Network: Must support internet or LAN connectivity for real-time data transmission.

2.5 Design and Implementation Constraints

- Latency Issues: Must handle network delays without significantly affecting user experience.
- Cross-Platform Compatibility: The application needs to be optimized for multiple platforms.
- Security Requirements: Encrypted communication must be ensured to prevent unauthorized access.

2.6 Assumptions and Dependencies

- The remote machine must have a stable internet or LAN connection.
- The necessary permissions and software must be installed on the remote computer.

3. System Features

3.1 Real-Time Screen Sharing

Description: The user will see the remote computer's desktop as it appears in real time.

- Input: Initiate a connection from the client device.
- Output: The remote computer's screen will be displayed on the client device.

Functional Requirements:

- FR1.1: The system must capture and transmit the screen data of the remote computer.
- FR1.2: The system must maintain synchronization between the client and the remote machine.

3.2 Remote Control

Description: The user can interact with the remote desktop using their local keyboard and mouse.

- Input: User input from the local device.
- Output: Actions are executed on the remote desktop.

Functional Requirements:

- FR2.1: The system must capture mouse and keyboard inputs from the client device and transmit them securely to the remote machine.
- FR2.2: The system must handle inputs in real-time, minimizing lag.

3.3 Remote Shutdown/Restart

Description: The user can initiate a shutdown or restart of the remote machine.

- Input: Shutdown or restart command from the client device.
- Output: The remote machine will perform the shutdown or restart.

Functional Requirements:

- FR3.1: The system must allow the user to send shutdown or restart commands to the remote machine.
- FR3.2: The system must verify user authorization before executing the command.

4. External Interface Requirements

4.1 User Interfaces

- Login Screen: Users will input their credentials to log in to the app.
- Main Dashboard: Users can view a list of remote computers they have access to.
- Control Screen: This screen will display the remote desktop and provide options for remote control, shutdown, and restart.

4.2 Hardware Interfaces

- Client Device: Must have a functional display, keyboard, and mouse.
- Remote Computer: Must have the necessary remote desktop services enabled.

4.3 Software Interfaces

- Operating System: Compatible with Windows, macOS, Linux for both client and server.
- Communication Protocols:
 - HTTP: Used for standard REST API requests like login and issuing shutdown or restart commands.
 - WebSocket: Used for real-time, bidirectional communication, particularly for sending/receiving events and screen sharing data between the server and the client.

4.4 Communication Interfaces

- Internet or LAN: The system will use internet or local networks to connect the client and server devices.
- Port Requirements: Must have port 5000 open.

5. Non-Functional Requirements

5.1 Performance Requirements

- The system must maintain a screen refresh rate of at least 30 frames per second under normal network conditions.
- The system must handle latency of up to 300ms without noticeable degradation in user experience.

5.2 Security Requirements

- All communication between the client and server must be encrypted (using TLS or SSH).
- The system must support multi-factor authentication (MFA) for logging in.

5.3 Usability Requirements

- The user interface must be intuitive and easy to navigate, especially for non-technical users.
- The system should provide clear error messages and tooltips to guide users.

5.4 Reliability Requirements

- The system must handle sudden disconnections and allow users to resume their sessions without data loss.

5.5 Maintainability Requirements

- The system should allow for updates and patches to be deployed without significant downtime.

6. Other Requirements

6.1 Legal and Regulatory Requirements

- The application must comply with data privacy laws, such as GDPR, especially for data transmitted between client and remote machines.