Software Design Document (SDD)

Remote Desktop Application

Version: 1.0

Date: October 2024

Table of Contents

1. Introduction 3
1.1 Purpose 3
1.2 Scope 3
1.3 Definitions, Acronyms, and Abbreviations 3
2. System Overview 3
3. Design Considerations 3
3.1 Assumptions 3
3.2 Dependencies 3
4. System Architecture 4
4.1 Client-Server Architecture 4
5. Design Details 4
5.1 Client Application 4
5.2 Server Application 4
5.3 Use Case Diagram 5
5.4 Network Layer 5
6. Security 6
7. User Interface (UI) 6
8. Error Handling 6
9. Testing 6
10. Future Enhancements 6

1. Introduction

1.1 Purpose

This document details the design of a remote desktop application that enables users to share screens in real time, remotely control a computer using mouse and keyboard inputs, and perform system shutdown or restart actions.

1.2 Scope

The application will support secure screen sharing, remote control of mouse and keyboard, and power management (shutdown and restart) actions across networks.

1.3 Definitions, Acronyms, and Abbreviations

- GUI: Graphical User Interface

- RDP: Remote Desktop Protocol

- TCP/IP: Transmission Control Protocol/Internet Protocol

-TLS: Transport Layer Security

2. System Overview

The remote desktop application allows authorized users to connect to a remote computer, share screens, control input devices, and manage power functions (shutdown/restart). It is built with a client-server architecture using secure communication protocols.

3. Design Considerations

3.1 Assumptions

- Both devices are connected to the internet.
- User has permissions for remote control.

3.2 Dependencies

- Secure Socket Layer (SSL) or Transport Layer Security (TLS) for secure communication.
- Libraries for capturing and transmitting screen data and handling remote control inputs.

4. System Architecture

4.1 Client-Server Architecture

The system follows a client-server model where the client initiates a connection to a remote server, enabling data transfer between them. Key components include:

- Client Application: Interface and functionality for the user on the remote device.

- Server Application: Runs on the target machine, managing screen sharing and input handling.

- Network Layer: Handles data transmission over TCP/IP with encryption.

5. Design Details

5.1 Client Application

Modules:

- User Interface (UI): For connecting, initiating screen sharing, and controlling functions like shutdown/restart.
- Connection Manager: Establishes a secure connection using TLS.
- Input Controller: Transmits mouse and keyboard inputs.
- Power Manager: Sends shutdown or restart commands.

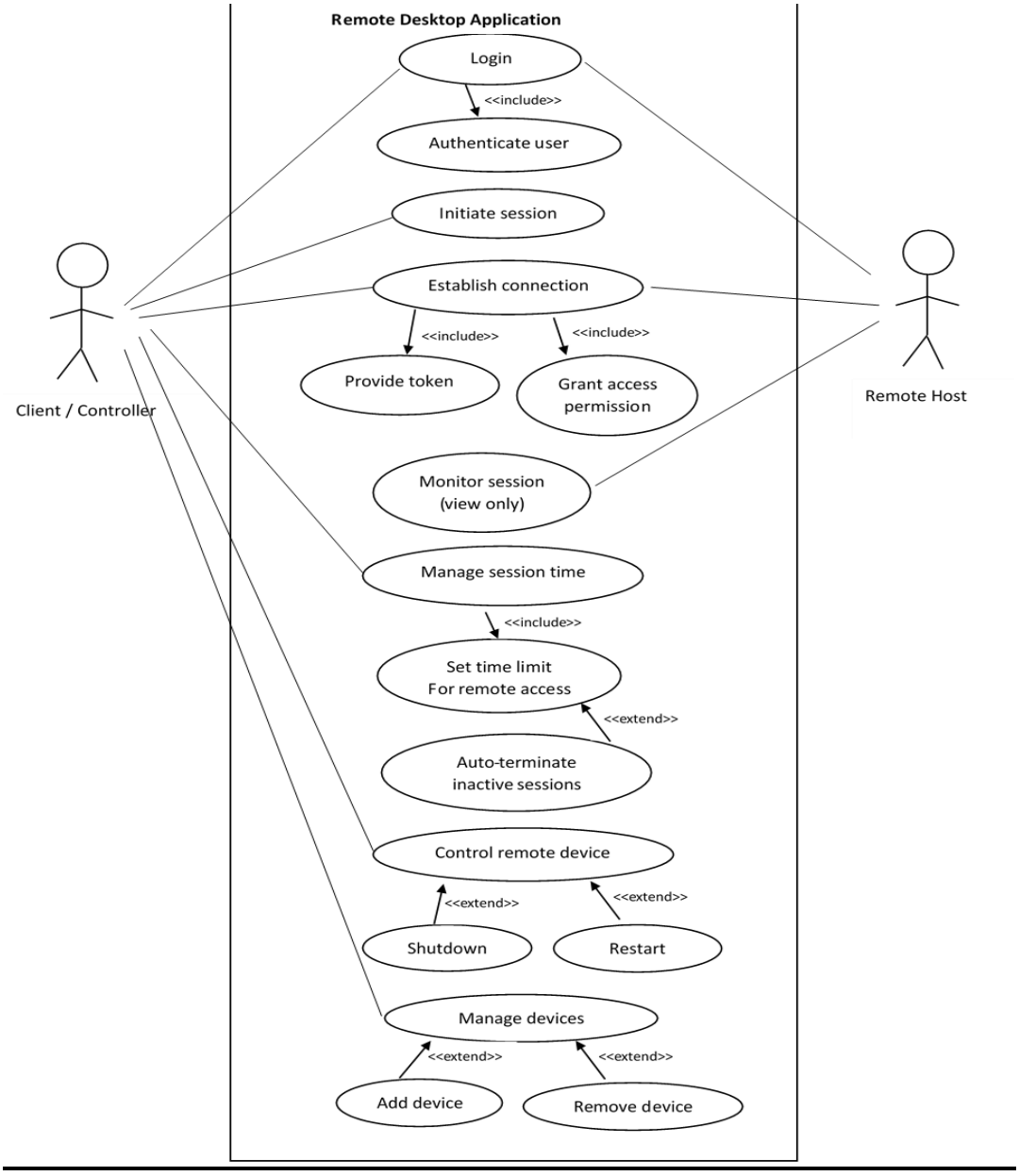
Data Flow:
1. User authenticates and initiates a connection to the server.
2. UI provides real-time screen feed and input controls.
3. Input events are captured and transmitted to the server.
4. Power management requests are transmitted securely.

5.2 Server Application

Modules:
- Screen Capture: Captures and streams the screen in real-time to the client.
- Input Listener: Listens for mouse and keyboard inputs from the client.
- Command Executor: Processes commands for shutdown or restart.

Data Flow:
1. Server authenticates the client upon connection.
2. Screen data is captured and transmitted to the client.
3. Input Listener processes client commands and updates screen accordingly.
4. Command Executor handles shutdown/restart requests.

5.3 Use Case Diagram



5.4 Network Layer

The network layer ensures secure and reliable data transmission between the client and server, employing TLS over TCP/IP.

6. Security

Authentication: Only authenticated users can access a remote session.
Encryption: All data transfers are encrypted using TLS.
Access Control: Input and power management requests are only accepted from authenticated sessions.

7. User Interface (UI)

The UI will provide:
1. Login Screen for authentication.
2. Main Dashboard for initiating screen share, controlling inputs, and managing power options.
3. Status Indicator showing connection state and remote session information.

8. Error Handling

Connection Timeout: Retry connection or prompt the user.
Authentication Failure: Display an error and prompt for re-authentication.
Command Errors: Log and inform the user if shutdown/restart fails.

9. Testing

Unit Testing: Test individual components like screen capture, input control, and command execution.
Integration Testing: Test client-server interactions, including secure connections.
System Testing: Ensure complete functionality in a controlled environment.

10. Future Enhancements

Multi-User Support: Allow multiple clients to view/control the same session.
File Transfer: Enable secure file transfer between client and server.
Audio Sharing: Add remote audio streaming.