**1. Introduction**

**1.1 Abstract**

* + 1. The SSH-SFTP client-side system is a combination of the SSH (Secure Shell) and SFTP (Secure File Transfer) protocols combined into an Android application. The application will minimize the permissions required by other SSH-SFTP clients and reduce user-level complexity. The system will also provide simple SSH and SFTP interfaces for the user to use. One practical example of use case is to allow the transfer of select media wirelessly via SFTP when there is a lack of storage space on the mobile device the application is installed on. The application will provide a virtual console to the user if the user wishes to connect to a remote SSH hosting server. If the user decides to create an SFTP session, the application will create a simple file system interface allowing the user to select either the remote files to download or the local files to upload. The application will serve consumers who have a background in the Unix environment, to use the SSH client and navigate through both the remote and Android’s file system for the SFTP client. The remote servers will be hosted on a Unix environment but may include other environments for support. The key user needs include secure communications, reliability, performance, and supportability.

1. **User Requirements Specification**
   1. **User Requirements – Introduction**
      1. This is a requirements specification document for an Android application. This document describes the scope, objectives, and goal of the system. In addition to describing non-functional requirements, this document models the functional requirements with use cases, interaction diagrams, and class models. This document is intended to direct the design and implementation of the system in an object-oriented language. The Android application will allow remote access to specific services given a remote server. Within the application, the user shall be able to:
         1. *Create a shell on a remote server.*
         2. *Send and retrieve files with a remote server.*

**2.2 User Requirements – Project Scope**

* + 1. The scope of this project is an Android application that allows remote access with fewer dependencies compared to other Android SSH-SFTP applications. Excessive permissions and advertising products will not be a part of this project. Users will benefit from not having an advertisement popup repetitively and avoid paying for reproducible features.
    2. The system will be available to any potential consumer that is using the internet.

**2.3 User Requirements – Functional Objectives**

1. High Priority
   1. The system shall allow for setting up new connections.
   2. The system shall reflect errors back to the user.
   3. The system shall handle permission requests and permission errors.
      1. Medium Priority
         * 1. The system shall continuously display new updated information.
           2. The system shall display all ongoing connections.

**2.4 User Requirements – Non-Functional Objectives**

* 1. Reliability
     + - 1. The system shall be completely operational at any point in time.
         2. Failures shall restart the application.
  2. Usability
     + - 1. A user should be able to create a new connection within a few clicks.
         2. The mean time to establish a connection should not exceed 10 seconds.

**2.4.3** Performance

* + - * 1. The system should support at least 5 simultaneous sessions.
        2. The system view refresh-rate should not exceed 1 second.

**2.4.4** Security

* + - * 1. All transactions must be transmitted in an encrypted form.
        2. The system shall hide the user’s password from view.

**2.4.5** Supportability

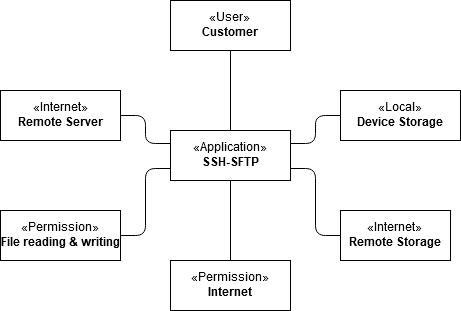
* + - * 1. The system shall be able to accommodate new features without major reengineering.
        2. The system shall be viewable from Android version 8.1 or above.
        3. The system should support any Unix operating system.

1. Interfaces
   * + - 1. The system must interface with the OpenSSH2 tool on the remote server.

**2.5 User Requirements – Context Model**

1. Goal Statement
   1. The goal of the system is to allow the creation of multiple continuous sessions allowing quick remote access to accomplish a task.

**2.5.2** Context Diagram



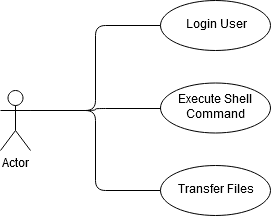
* 1. System Eternals
     + 1. A customer is any user of that system who wishes to connect to a remote server. A customer may create a shell or start to transfer files.
       2. Device storage is the collection of files the customer has access to on the device. A customer should have access to the files through the system.
       3. The remote server is a foreign device that the consumer wishes to connect to when creating a shell or transferring files.
       4. File reading & writing permissions are required for manipulating any foreign or local files.
       5. Remote storage is a foreign collection of files that the customer has access to through the system.
       6. Internet permission is required for transmitting any commands or files or receiving any outputs from the remote server.

**3. System Requirements Specification**

**3.1 System Requirements – Introduction**

* 1. The system is composed of three different use cases, the log in case, execution case, and transfer case. The execution case and the transfer case are the core of the software system and both depend on the log in case. The user must be logged in in order to execute or transfer files.

**3.2 System Requirements - Use Case Diagram**

****

**3.3 System Requirements - Use Case Descriptions (for select cases)**

1. Notes:
   * + 1. For all use cases, the user can terminate at any step that requires user input. This action will end the use case. Any open session shall continue when that use case ends unless specified otherwise.
       2. For all use cases that require a logged-in user, the current login session is updated during that use case to reflect the navigation paths of the use case.
2. Login User

|  |  |
| --- | --- |
| Use Case Name: | Login User |
| Summary: | In order to execute shell commands or transfer files, a user must log in so that the system can connect to the remote host. |
| Glossary: | URL – Universal Resource Locator.  Port – Communication endpoint. Ranges from 0 to 65535. |
| Basic Flow: | 1. The use case starts when a user wants to create a session. 2. The system requests the type of session to create, Username, URL, Password, and Port the remote system is running on. 3. The user enters at least their username, URL, and password. The default port is 22. The default session to create is SSH. 4. The system will verify the credentials of the user. 5. The system will start a respective session depending on whether they want to execute a shell command or transfer files. |
| Alternate Flows: | Step 4: If the username is invalid, the use case goes back to step 2.  Step 4: If the password is invalid, the use case goes back to step 2.  Step 4: If the URL is invalid, the use case goes back to step 2.  Step 4: If the port is invalid, the use case goes back to step 2. |
| Extension Points: | None |
| Preconditions: | The user is connected to the internet and the user is registered on the remote server. |
| Postconditions: | The user can now send commands or transfer files. |
| Business Rules: | None |

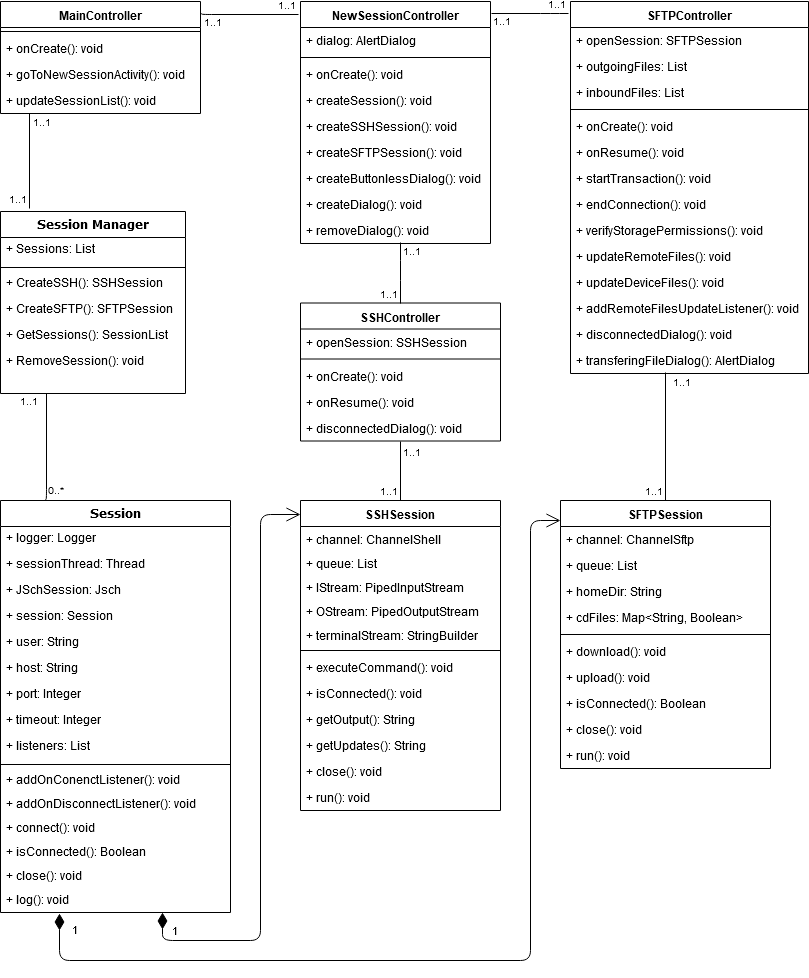
* 1. Execute Shell Command

|  |  |
| --- | --- |
| Use Case Name: | Execute Shell Command |
| Summary: | This use case allows a logged-in user to execute commands on the remote server they have connected to. |
| Basic Flow: | 1. The use case starts when an SSH session has been created. 2. The system will continuously display the remote server’s outputs. 3. The user may enter a command. 4. The user may send the command. 5. The user may leave the active session. |
| Alternate Flow: | Step 2: If the connection is lost, leave the session. |
| Extension Points: | None |
| Preconditions: | The session is connected, and the user is logged in. |
| Postconditions: | The user can execute another shell command. |
| Business Rules: | None |

* 1. Transfer Files

|  |  |
| --- | --- |
| Use Case Name: | File Transfer |
| Summary: | This use case allows a logged-in user to transfer files between the device and the remote server. |
| Basic Flow: | 1. The use case starts when a user wants to create an SFTP session. 2. The system will continuously display the remote server’s files and device files. 3. The user may enter a new remote directory path or a new device directory path. 4. The user may select files from either the remote server or the device. 5. The user may transfer the files. 6. The user may end the session. |
| Alternate Flow: | Step 2: If the connection is lost, leave the session.  Step 5: If the files fail to be sent or retrieved, do not continue sending that file. |
| Extension Points: | None |
| Preconditions | The session is connected, and the user is logged in. The system must have permission to read and write files. |
| Postconditions | The user can select another set of files to transfer. |
| Business Rules: | None |

**3.4 System Requirements – Class Model**

****