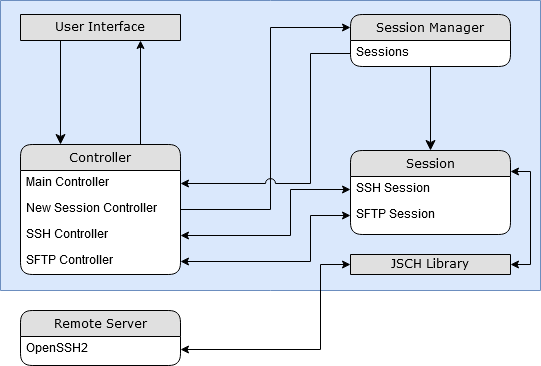
1. **Introduction**
   1. **Abstract**
      1. The following documentation will consist of the major components involved in achieving an SSH-SFTP client for an Android device. The architecture implemented is a Model View Controller (MVC) architecture to interact with the SSH (Secure Shell) and SFTP (Secure Transfer Protocol) protocols. The components which implement the SSH and SFTP sessions will be handled by the JSCH (Java Secure Channel) library. Other components such as the View and Controller will be handled by the Android operating system for interacting with the Model. The JSCH library used for handling the protocols is implemented with a Server-Client architecture and will be used throughout the system. The MVC architecture implemented will allow reliability, usability, supportability, interfacing while the Server-Client architecture implemented in the JSCH library allows performance and security. The tradeoff using the MVC architecture throughout the system is an overweight in the Controller component which handles a considerable chunk having the View interface with the Model.

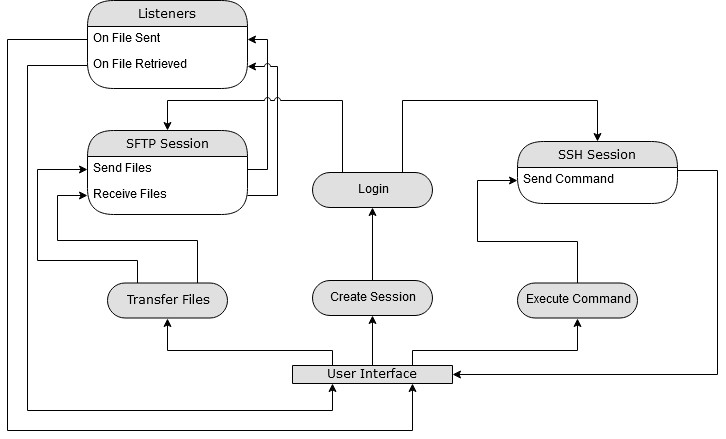
1. **Architecture and Design Philosophy**
   1. **Architecture and Design – Introduction**
      1. One of the key features implemented in the design of the system is the ability to concurrently process multiple sessions. The Model component shall handle all concurrent sessions for the controller to interface with whenever the user wants to interface with a specific session. For optimizing the handling of continuous throughput for the Client-Server architecture, events will allow the Model to update instantaneously whenever a response from the remote server is received. Future implementations may include the support for new protocols, or interfaces with the MVC architecture.
   2. **Architecture and Design – Scope**
      1. The Scope of the architecture is to allow continuous throughput to the user while allowing simultaneous processing in the background for other sessions. The select architecture allows modifications to any component for any maintenance needed.
   3. **Architecture and Design – Architecture purpose**
      1. The purpose of the Model View Controller architecture in this project functionally establishes the following:
         1. The handling of exceptions and errors throughout any use case. Includes presenting errors to the user.
         2. Easily and securely allow the user to interact and create a new connection.
         3. Support multiple simultaneous connections with quick refresh rates.
         4. Allow for flexible development and supportability with Android 8.1
      2. The flowing will also be established when using the JSCH library:
         1. Encrypted transactions with OpenSSH2 support.
2. **Architectural Views**
   1. **Architecture Views – Introduction**
      1. This section will focus on the interaction of the components with one another to achieve the required results. Whenever the user interacts with the controller through the view, the state of the model will change. Depending on the use case being used, the model can either handle a new session or invoke an action within a current session. In a logical view, as requests are being handled, the user interface thread within the controller will continuously update the view by pulling information off the model before or after the handlers update the model. From a process perspective, concurrent threads will only interact with one another when the model needs to be updated. The development view consists of an SSHSession and SFTPSession class’ as the core components to implement the system. The physical view includes the use of parallel processing and concurrency to simultaneously handle multiple sessions and handle the user interface.
   2. **Architectural Views – Logical View**
      1. Logical View Model

****

* + 1. Logical View Table

|  |  |
| --- | --- |
| User Interface: | Displays and receives user input for the controller to manage depending on the use case the user is currently doing. |
| Controller (Login Case): | The Main and New Session controllers handle the Login User use case. The Main controller displays all active selectable sessions to the user and allows the creation of a new session. The New Session controller prompts the user to login to establish a new session through the user interface. |
| Controller (SSH Case): | The SSH Controller utilizes a session passed from the Main controller, or the New Session controller. The controller actively refreshes any output to display to the user and executes commands through the user interface. |
| Controller (SFTP Case): | The SFTP Controller utilizes a session passed from the Main controller, or the New Session controller. This controller actively refreshes for any updates regarding files on the current directory provided. The user may update the remote or device working directory and the controller will update the user interface respectfully. |
| Session Manager: | This component stores all the sessions for the Main controller to use and display all active sessions. |
| Session: | Comprised of two subcomponents, the SSH session used for handling shell commands and the SFTP session to handle the transferring of files. The SSH session is created and stored by the Session Manager and interfaced with the SSH controller. The SFTP session us created and stored by the Session Manager and interfaced with the SFTP controller. Invokes events for the controllers to handle when a session has connected, disconnected, sent a command, sent a file, or received a file. |
| JSCH Library: | Establishes the client-server connection to connect to a remote server for the Sessions. This library interfaces with any toolsets following the Secure Shell protocol or Secure Transfer Protocol. |
| Remote Server: | The remote server consisting of the OpenSSH2 utilities which is interfaced by the JSCH Library. |

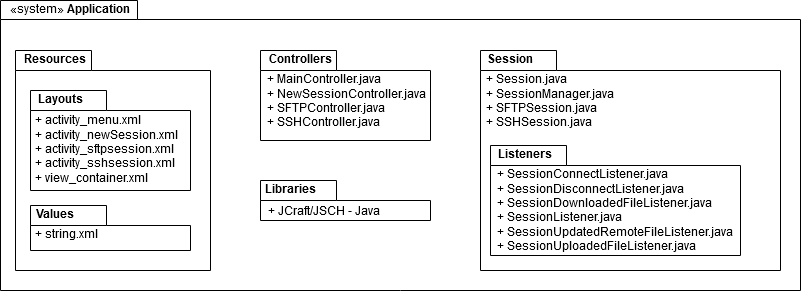
* 1. **Architectural Views – Process View**
     1. Process View Model

****

* + 1. Process View Table

|  |  |
| --- | --- |
| User Interface: | Displays and receives user input for the controller to manage depending on the use case the user is currently doing. |
| Transfer Files: | The files selected will be pulled from the user interface. The selected files will be transferred and have a listener applied to each file in order to update the user interface whenever the file has been sent. Or retrieved |
| SFTP Session: | Have the session thread process the selected files and transfer them to the remote server selected. |
| Listeners: | Contains the listeners which invoke an update whenever a process has been completed. |
| Create Session: | Creates a new thread for the session based on the login credentials provided. |
| Login: | Uses the credentials to log in into the remote server and can create a SSH session or a SFTP session. |
| Execute Command: | The remote command will be pulled from the user interface. The command will be sent across the open session’s stream. |
| SSH Session: | The session thread continuously sends any commands in the queue and has the terminal update anytime the view requests and terminal update. |

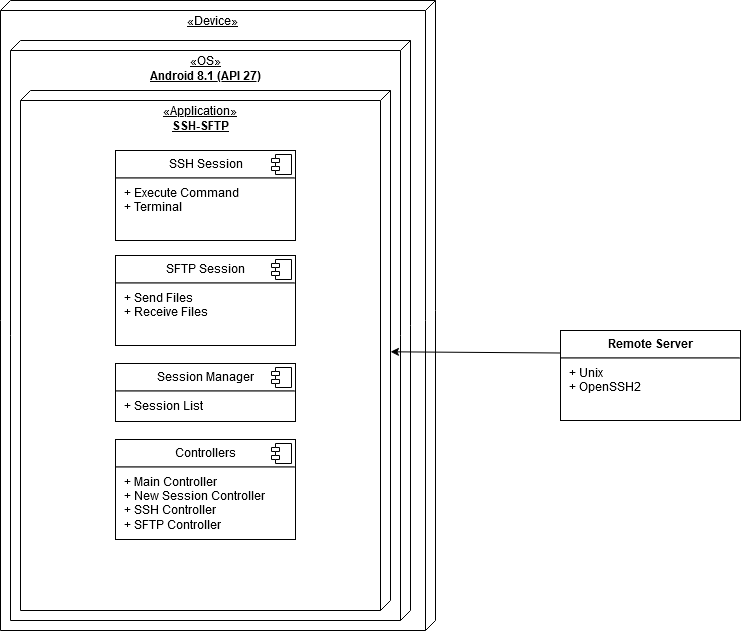
* 1. **Architectural Views – Design View**
     1. Design View Model

****

* + 1. Design View Table

|  |  |
| --- | --- |
| Resources: | The resources are composed of xml documents for view presentation. |
| Controllers: | The controllers are developed with the Java language. The controller’s methods will be called by the view within the application when the user invokes an event. |
| Model: | The sessions and the listeners within the model are developed with the Java language. The model also includes the JSCH library for secure communications. |

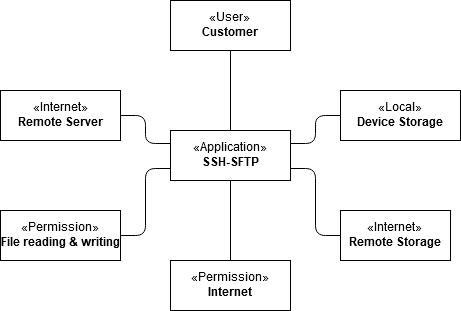
* 1. **Architectural Views – Physical View**
     1. Physical View Model



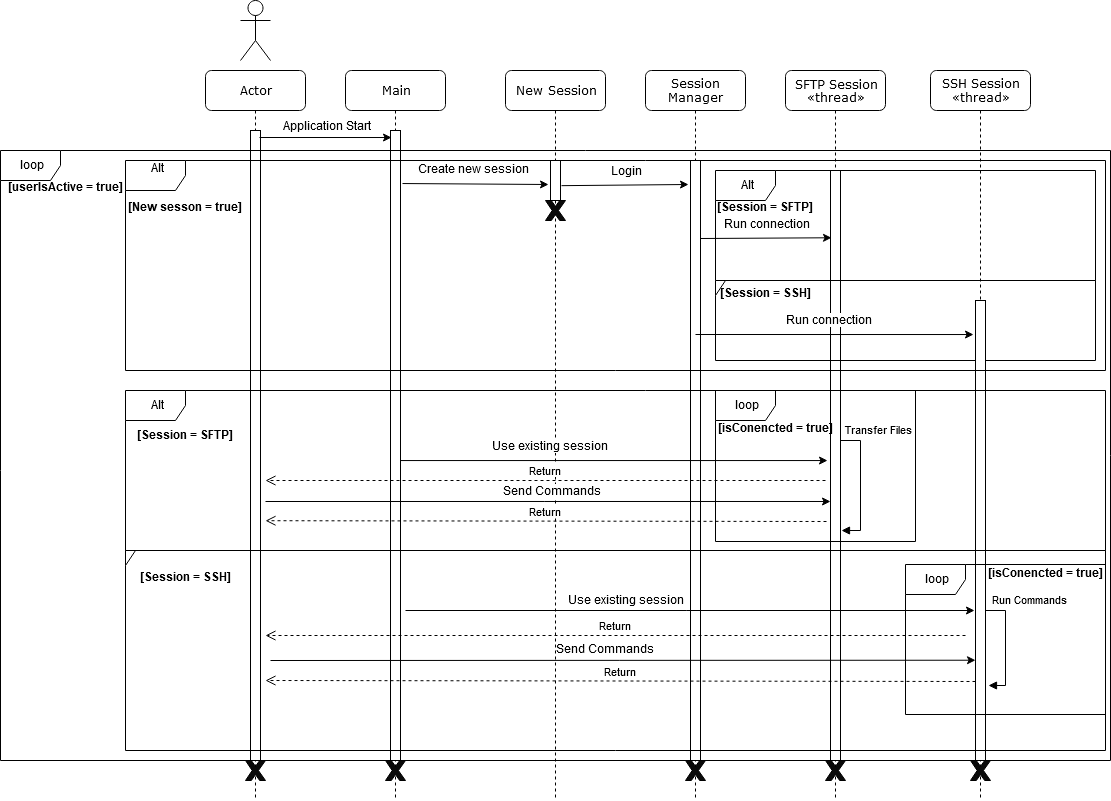
* + 1. Physical View Table

|  |  |
| --- | --- |
| Device: | The device the system is running on. The architecture of the system should be flexible with any system running an Android 8.1 or better. |
| Operating System (OS) | The operating system is Android 8.1 as specified in the requirements document. The API level used is 27. |
| SSH-SFTP | This is the system being developed, consists of Sessions, Controllers, and a view for the user. |

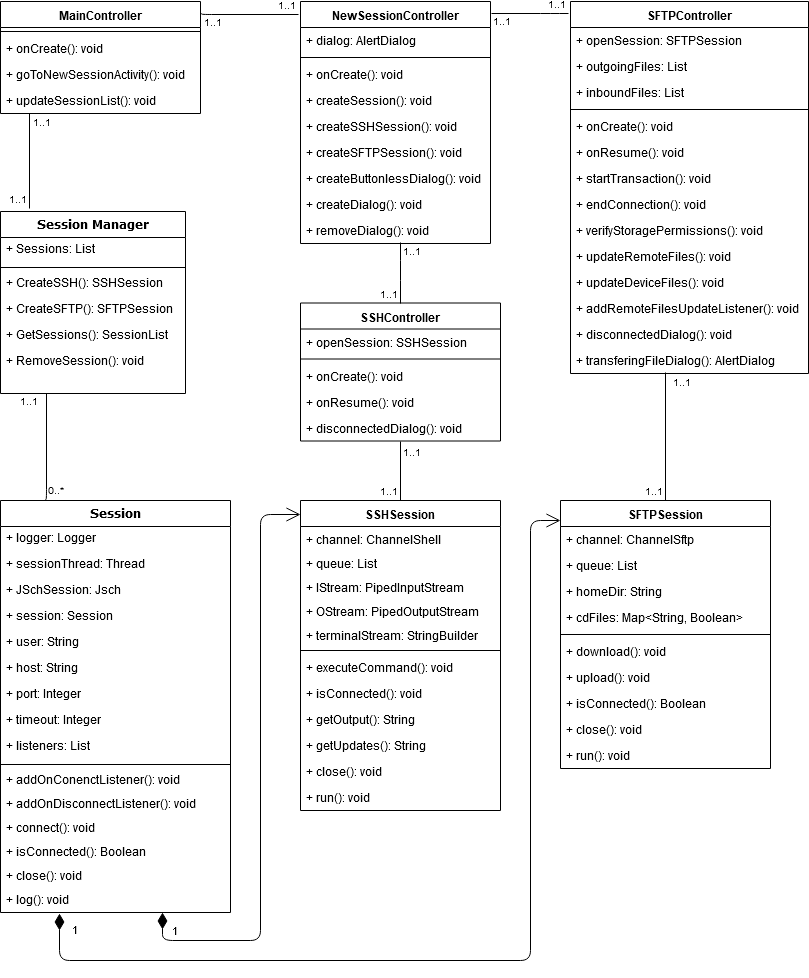
1. **Design Models**
   1. **Design Models – Models**
      1. Context Model



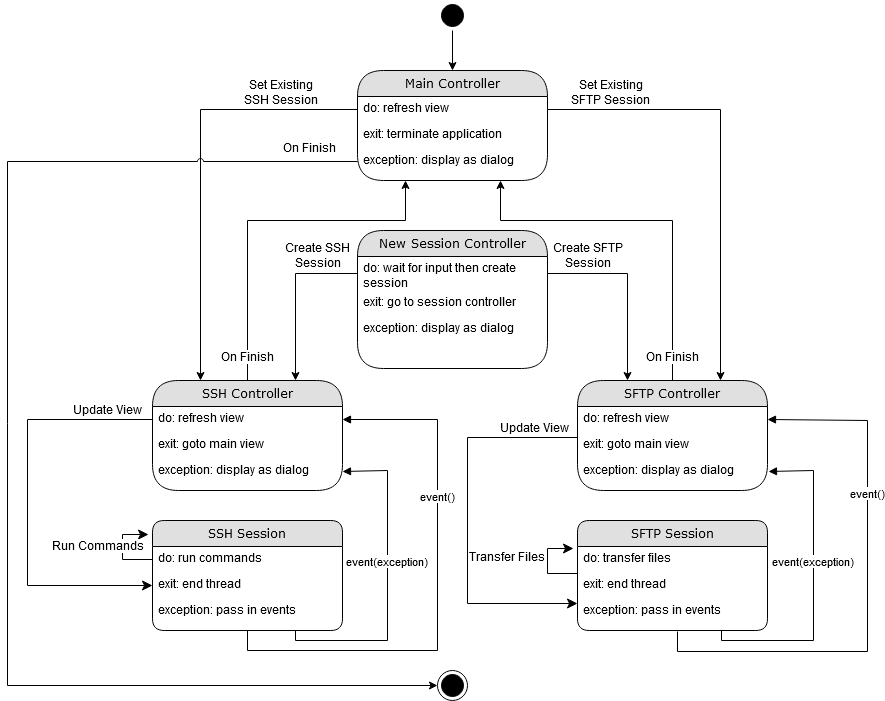
* + 1. The system is focused on several key contexts for the system to work. This includes, the user, the local device storage, the remote server’s storage, internet permission, file and reading permissions, and the remote server itself.
    2. Sequence Model

****

* + 1. The model above displays the interaction between the user and other components. When the actor creates a new session, they create a new thread with a new active session. When the Session has been created, switch to it, if the session was not, switch to an existing one. Repeat process until the user closes the system (when use is no longer active).
    2. Class Model

****

* + 1. The model above states the relationships between every component of the system. The main controller displays all the sessions stored in a list within the Session manager. Each session in this system must be created through the new session controller which can be either an SSH session or SFTP session but not both. Every session controller has only one active session is it currently viewing. Important data structures are lists. The list in the session manager maintains all active sessions that can be displayed to the user. The list in the Session class stores all listeners to invoke upon key events shown in diagram 3.3.1. The SSH Session class’ list queues all runnable commands to have the remote server process on the session’s thread. The SFTP Session class’ list queues all files which are to be either downloaded or uploaded. The SFTP Controller class’ list is synonymous to the SFTP Session class’s list as it stores all selected files to transfer only upon the startTransaction() method.
    2. Behavioral Model

****

* + 1. The model above displays how each controller handles events. The events will be handles by the SSH session and on the SFTP session respectfully and they come in two forms, events with exceptions and events without exceptions. Events with exceptions are called when an exception has occurred while the session has encountered an unrecoverable exception. These exceptions include any throwable exceptions from the JSCH library. The only manually thrown exception is an invalid argument exception that is thrown when an invalid port is entered by the user. Events without exceptions would inform the controller that the session has updated depending on the type of event invoked. All listeners are present in the 4.1.5 structural model.