**audio-droid™ Charter**

**Vision**

* audioDroid**™** envisions itself to be a free and open source mainstream app

**Mission**

* To develop an app that will be a staple app for mobile audio connectivity.

**Scope**

* This app is limited to streaming audio through a wireless local area network.

**Deliverables**

* Setup Desktop Server
* Setup Mobile Client
* Client – Server Connectivity
* Audio Streaming Capability

**Objectives**

* Within two weeks after the start of the projects. We expect to deliver the first deliverable which the Desktop Server
* Within a week after the delivery of the first deliverable, we expect to deliver the second deliverable which is the Mobile Client.
* Within two weeks after the delivery of the second deliverable, we expect to deliver the third deliverable which is the Client-Server Connectivity module.
* Within two weeks after the delivery of the third deliverable, we expect to finish the final deliverable and integrate it to the project for submission.

**Principles**

* On time delivery is a must.
* Quality of the product (software) is assured.

**Developers: Group 1**

**Danielito Padayhag Jr.**

**Kevin Khalil Reyes**

**Kevin Eric Siangco**

**GitHub – https://github.com/user/perroquiet/audio-droid.git**

**Use Cases & Test Cases**

**Use Case:**

* **Name** : Mobile Client Connection
* **Description**: Connects the mobile device to the desktop server
* **Actor**: Everyone who owns a mobile device and connected to the Wifi network.
* **Pre-condition:** Must be connected to the Wifi network, and the user is currently in the app main menu.
* **Post-condition:** The mobile device is now connected to the desktop server.

**Main Course:**

1. The user clicks the “Connect” button.
2. The user inputs the IP of the server in the textbox provided and clicks the “Connect” button.

**Exception:**

**2.a)** If the user receives a message “Connection Error”. Repeat Step 1.

1. The user’s mobile device is now connected to the desktop server.
2. If there exists an audio stream instance, the app fetches the audio stream.

**Exception:**

**4.a)** If there is no existing audio stream, user receives a message “Connection Error” and user is sent back to the main menu.

**Test Case:**

**Given:**

Francis wants to connect his mobile device to the desktop server using the IP address provided: 192.168.0.1:8554/stream.

**When:**

Francis presses the “Connect” button on the menu and inputs the IP: 192.168.0.1:8554/stream to the text box, and presses the “Connect” button.

**Then:**

Francis’ mobile device is connected and fetches audio stream to the desktop server.

**Use Case:**

* **Name** : Mobile Client Disconnection
* **Description**: Disconnects the mobile device from the desktop server
* **Actor**: Everyone who owns a mobile device and connected to the Wifi network.
* **Pre-condition:** Must be connected to the Wifi network and the app also “Connected” to the desktop server.
* **Post-condition:** The mobile device is now disconnected from the desktop server and returns to the main menu.

**Main Course:**

1. The user clicks the “Disconnect” button.
2. The user is prompted before disconnecting.
3. The user clicks “Yes” button.

**Alternative:**

**3a)** The user clicks “No” button, Use case exits and the client is still connected.

**4.**The user is now disconnected from the server and returns to the main menu.

**Test Case:**

**Given:**

Francis wants to disconnect his mobile device from the desktop server.

**When:**

Francis presses the “Disconnect” button and then presses “Yes” after the prompt.

**Then:**

Francis’ mobile device is diconnected from the desktop server.

**Use Case:**

* **Name**: Server Streaming
* **Description:** The desktop server can stream audio to the clients.
* **Actor**: Users
* **Pre-condition:** The desktop server must be connected to the network.
* **Post-condition:** The desktops server streams the audio to all clients.

**Main Course:**

1. The user handling the server plays an audio stream.
2. The system will broadcast the audio to all clients connected using the mobile app.

**Test Case:**

**Given:**

Francis wants to play an audio file from the desktop server.

**When:**

Francis plays a file, and allow it to stream to the clients.

**Then:**

The audio stream is broadcasted to all the clients.

**Use Case:**

* **Name**: Help Feature
* **Description:** It gives the user simple instructions on how to use the application.
* **Actor**: Users
* **Pre-condition: N/A**
* **Post-condition: N/A**

**Main Course:**

1) The user clicks the Help button.

2) The user will be prompted with a simple instruction on how to use the application.

3) The user clicks OK and the use case exits.

**Test Case:**

**Given:**

Francis wants try audio-droid for the first time but doesn’t know how to use it.

**When:**

Francis clicks the help button and reads the instructions.

**Then:**

Francis then clicks OK.

**Use Case:**

* **Name**: View History
* **Description:** Shows URLs the user attempted to connect to
* **Actor**: Users
* **Pre-condition: N/A**
* **Post-condition: N/A**

**Main Course:**

1) The user clicks the options button.

2) A menu is shown to the user and the user clicks URL History

3) The user is shown a list of URLs the user attempted to connect to

**Test Case:**

**Given:**

Francis wants to know the URLs he has tried to connect to.

**When:**

Francis clicks the option button and is shown the menu. He clicks the URL History.

**Then:**

Francis is then shown a list of URLs the user attempted to connect to

**Use Case:**

* **Name**: Clear History
* **Description:** History is cleared
* **Actor**: Users
* **Pre-condition: N/A**
* **Post-condition:** History is cleared

**Main Course:**

1) The user clicks the option button.

2) The user is shown the menu and clicks the URL History option.

3) The user clicks the option button and clicks the Clear History.

**Test Case:**

**Given:**

Francis wants to clear his URL History.

**When:**

Francis clicks the option button and is shown the menu and pressed the URL History option. The user clicks the option button and clicks the Clear History option.

**Then:**

The history is now cleared.

**Use Case:**

* **Name**: Connection via URL History
* **Description:** The user connects to the server via URL History
* **Actor**: Users
* **Pre-condition:** History must not be empty.
* **Post-condition:** User is connected to the server.

**Main Course:**

1) The user clicks the option button.

2) The user is shown the menu and presses the URL History button.

3) The user selects a URL and the client attempts to connect.

**Test Case:**

**Given:**

Francis wants to connect to a previous URL he already connected to in the past.

**When:**

Francis clicks the option button and is shown the history. He presses the URL History button. He selects 192.168.0.1:8554/stream.

**Then:**

Francis is now connected to 192.168.0.1:8554/stream.

**Feature List and Feature Acceptance Criteria**

**Feature List**

* Client – Server via Wi-Fi Local Area Network Connection
* Client and Server can support Audio streaming with Playback Capabilities

**Feature Acceptance Criteria**

**Client – Server via Wi-Fi Local Area Network Connection**

**Server**

* The server (desktop computer) can create a host where the clients where connect to.
* The server must use a Wi-Fi Local Area Network Connection.
* The server can play audio.
* The server can stream the audio to the clients.
* The server has playback capabilities.

**Client**

* The client (mobile device) can connect to the server using the server’s IP Address in the network.
* The client must connect via Wi-Fi Local Area Network Connection.
* The client can disconnect from the server.
* The client can accept audio stream and play it.
* The client can change streaming protocols to connect to.

**System Architecture**

Desktop Server Application

Media Player

Data (Audio) Stream

**Mobile Phone**

Wifi Connection

Mobile Client

Router

The mobile application relies on a Desktop as a server to stream music to the mobile client through a Wi-Fi network connection. The music stream is first passed through the router then sent to the mobile client. Since this is real-time streaming, the media controls (i.e. play/stop/pause/seek) can be controlled from the server.

**Deliverables Schedule**

Setup Desktop Server - February 6, 2013

Setup Mobile Client - February 13, 2013

Client – Server Connectivity - February 27, 2013

Audio Streaming Capability – March 15, 2013

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Setup Desktop Server |  |  |  |  |  |  |  |  |  |
| Setup Mobile Client |  |  |  |  |  |  |  |  |  |
| Client – Server Connectivity |  |  |  |  |  |  |  |  |  |
| Audio Streaming Capability |  |  |  |  |  |  |  |  |  |
|  | Week 1 | Week 2 | Week 3 | Week 4 | Week 5 | Week 6 | Week 7 | Week 8 | Week 9 |