HeyJuke

Goal Statement

Our goal is to:

- Create a host program on a Raspberry Pi
- Create a paired phone app

for coordinating the playing of music in a shared environment.

Who We Are

- Students
  - Joe Hirschfeld - hirschjb@mail.uc.edu
  - John D’Alessandro - dalessjj@mail.uc.edu
  - Luis Finke - finkelj@mail.uc.edu
- Advisor
  - Dharma Agrawal - agrawadp@ucmail.uc.edu

Abstract

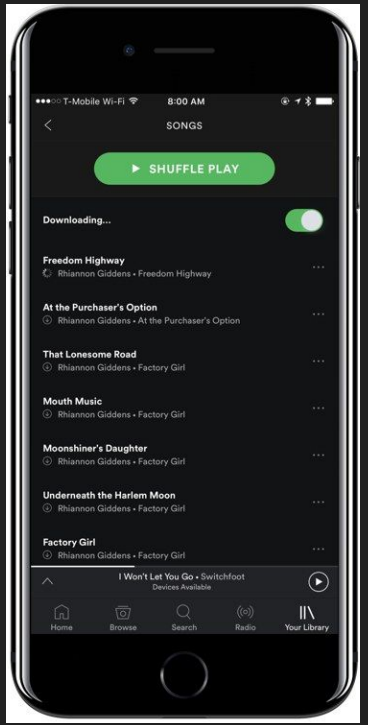
HeyJuke is a home jukebox that is easy for end users to set up and operate in a group setting. HeyJuke consists of a **device** capable of managing the **queue** and an **app** that users download onto their smartphone. Users are able to interact with a playlist that is streamed through the **device** into connected speakers. They do this through the HeyJuke **app** that will communicate with the **device** to add songs to, remove songs from, and manage the **queue**. The **app** and **device** are able to communicate with multiple streaming services to allow the users to add songs from many platforms to the **queue**.

User Story: Base Case

- As a frequent partier, I want to be able to queue or request songs to be played at a party, so that my ears may be blessed with good vibrations
  - HeyJuke should be able to actually play music
  - Music can be queued from the hard drive of the device

User Story: Spotify Permissions

- As a spotify subscriber, I want to listen to songs from my spotify account without overriding whatever music is currently playing, so multiple people can play music
  - HeyJuke must be able to manage queue management from multiple spotify accounts at once
  - Each user should use their own Spotify account, managed through the app



User Story: Other Streaming

- As a music listener, I want to be able to play songs from various streaming services, so that I can have a larger music library available to choose from
  - The app should include ways to search through and queue music from Spotify, YouTube, and the device's internal hard drive
  - The device should be able to switch seamlessly between all three of these formats

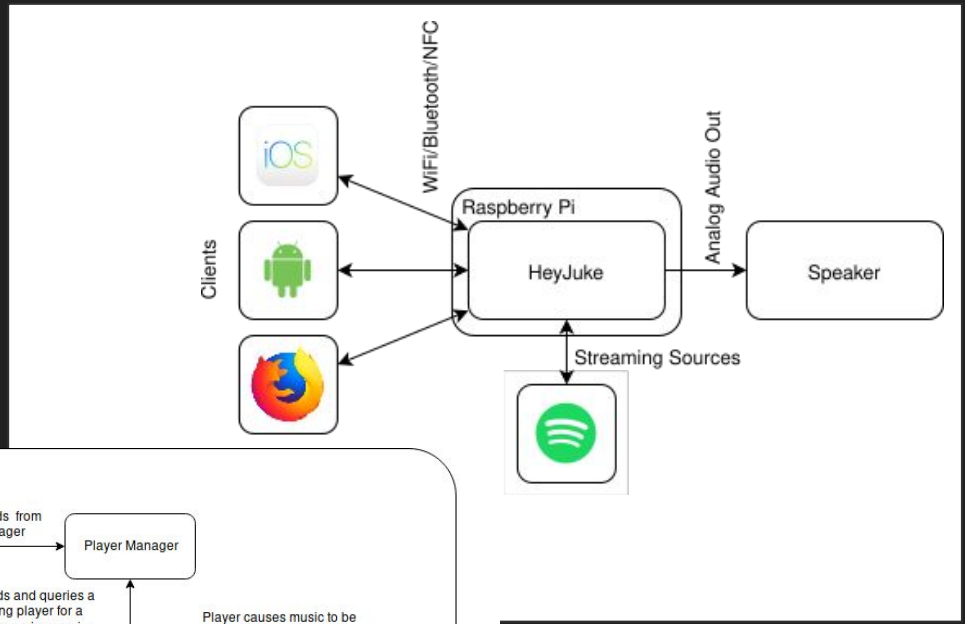
Project Progress

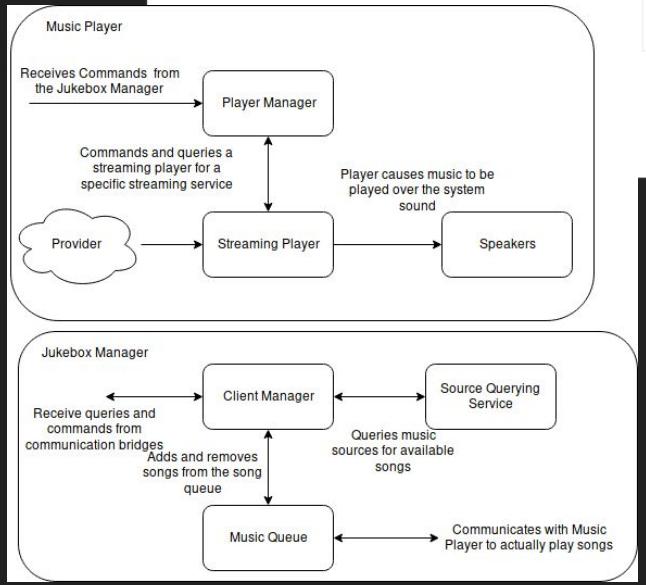
Preliminary Technology Research

User Stories

System Engineering Diagrams

Tasklist and Timeline





Task List - Raspberry Pi

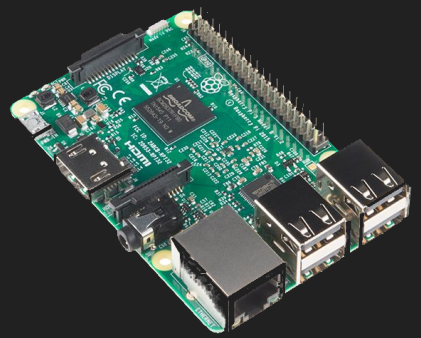
- Delegation
  - John
- Task
  - Acquire Raspberry Pi
  - Research an OS to use for it
  - Install OS on the Pi
- Cost
  - About $35
- Requirements for OS
  - Lightweight
  - Can run Node.JS
  - Can run a web browser
  - Can run securely on a network

Task List - Core Music Player

- Delegation
  - Luis
  - John
  - Joe
- Task
  - Write a music player program to play from
    - Spotify (Luis)
    - Youtube (Joe)
    - Local library (John)

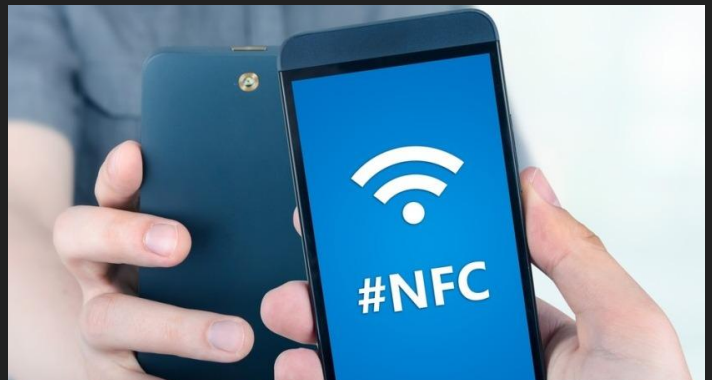
Task List - Web Server

- Delegation
  - John
- Task
  - Write a web server to run on the Raspberry pi to handle queueing and voting requests from the local network
  - Document web server endpoints for use in a mobile app
  - Web server will send commands to the core music player



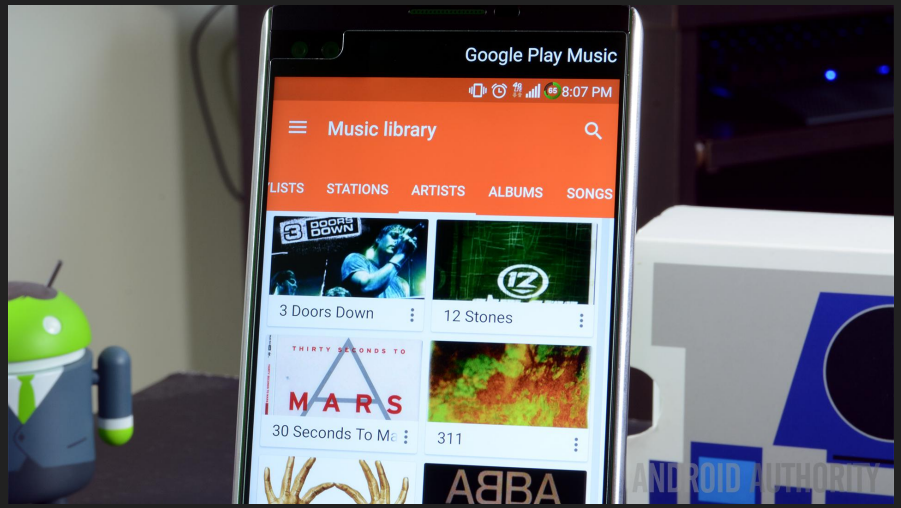
Task List - NFC Reader

- Delegation
  - Luis
- Task
  - Write a program for the raspberry pi that will allow usage of NFC to queue songs
- Necessary as a fallback for the web server, in case user does not have access to the network
- Will allow tapping a phone on the device to add songs



Task List - Mobile App

- Delegation
  - Joe
- Task
  - Write a mobile app to manage jukebox
  - Allow queueing songs through network calls or NFC
  - Allow voting on queued songs
  - Allow searching available music



Task List - Extras

- We may get to these things if we have time
  - physical shell for the Pi
  - touchscreen on the pi with nice UI
  - usage manual