Project Smartify

Team 9

Hyounjun Chang
Gerald Ko
Karunesh Sachanandani
Justin Suh

https://github.com/180D-FW-2020/Team9

Motivations

- Isn’t it great to be able to listen to your friends’ favorite tunes?
- What if you want to share your favorite songs with your buddies?
- How many of us have wanted to listen to a song we didn’t have downloaded in our library?
- What if you could control your music player with gestures, voice, and even your... EMOTION?

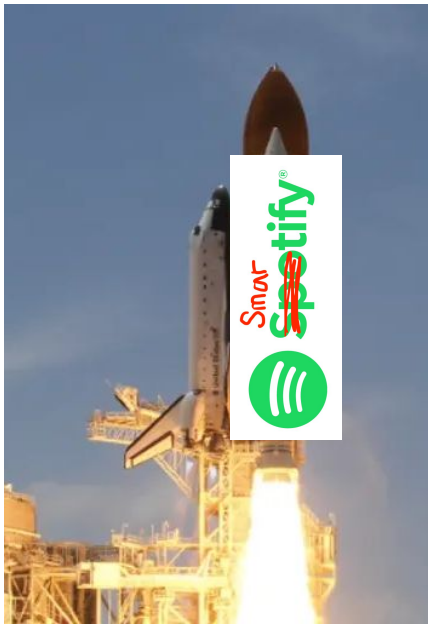
What is Smartify?

- Smartify picks up on your mood and generates the music that you need
- Smartify lets you control your music player from afar (so you can focus on your work)
- Smartify lets you hop on and check out what your friends are listening to

Smar~~Spo~~tify®

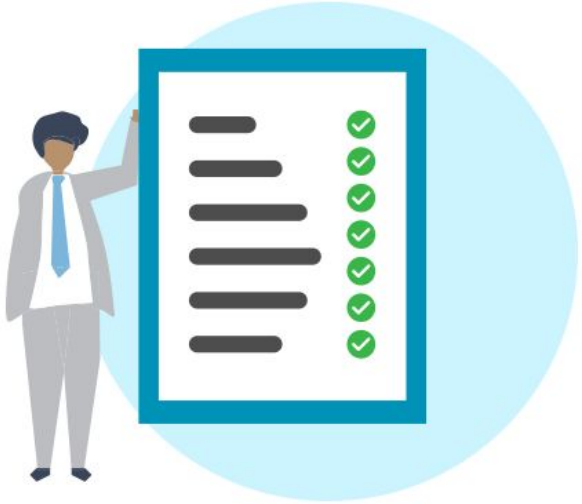
What makes Smartify, **Smartify**?

- FREE
- Quick setup
- Intuitive User Interface
- Control YOUR music player exactly on how YOU want to (Standard Playlist, Voice Commands, Gesture Recognition)
- Instant real time music sharing (no invite codes needed)
- Innovative solutions to find you the best music possible (Computer Vision: Emotion Recognition, Live Song Lookup)



Project Requirements!

- **Gesture Recognition**
  Accepts gesture control using Raspberry Pi with IMU as a remote controller!
- **Speech Processing**
  Recognizes voice commands for player navigation (Python Voice-Recognition)
- **Wireless Communications / “Multiplayer"**
  Allows real-time music sharing with friends through different channels (MQTT: broker-assisted communications)
- **Computer Vision**
  Plays songs that match your feelings using Computer Vision and Emotion Detection. (OpenCV/Tensorflow)
- **User Manual**

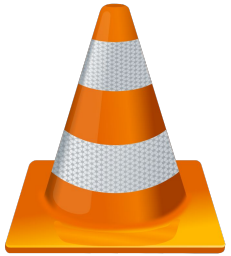


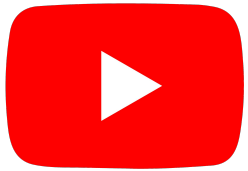
Tech Stack and Integration

Smartify was written entirely in **Python3**!

This means that it can theoretically run on anything that runs Python and following open-source Python libraries!

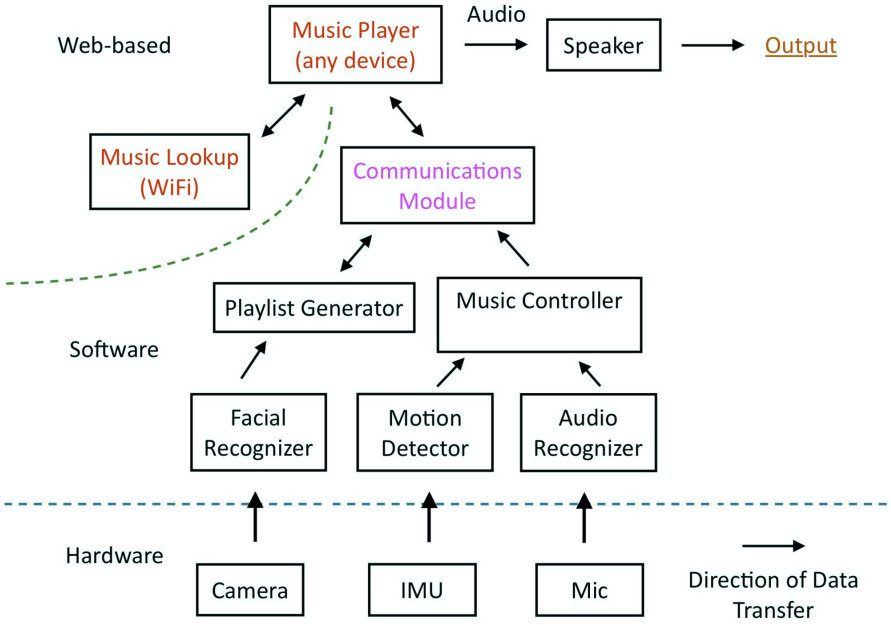
- SpeechRecognition/PyAudio ➔ Audio Commands
- VLC Player (Audio player) ➔ Media(audio) Player
- GUI Libraries (TkInter) ➔ GUI for Smartify Player
- IMU Driver (inside project)➔ IMU Music Controller
- Broker-Assisted Communications (MQTT) ➔ Playlist Sharing, Commands
- Youtube-Query Libraries (Pafy, Youtube-dl) ➔ Online Song SEarch
- Tensorflow, OpenCV ➔ Emotion Playlist





Overall Design

- Two Main Modules:
  - Music Player (Sender/Receiver)
  - Music Controller (Sender)
- Wireless/Local Communication between two modules
- Flexibility with multiple devices
- Music Lookup module: No need to have songs stored in the player anymore!



Smartify Modules! - How it works?

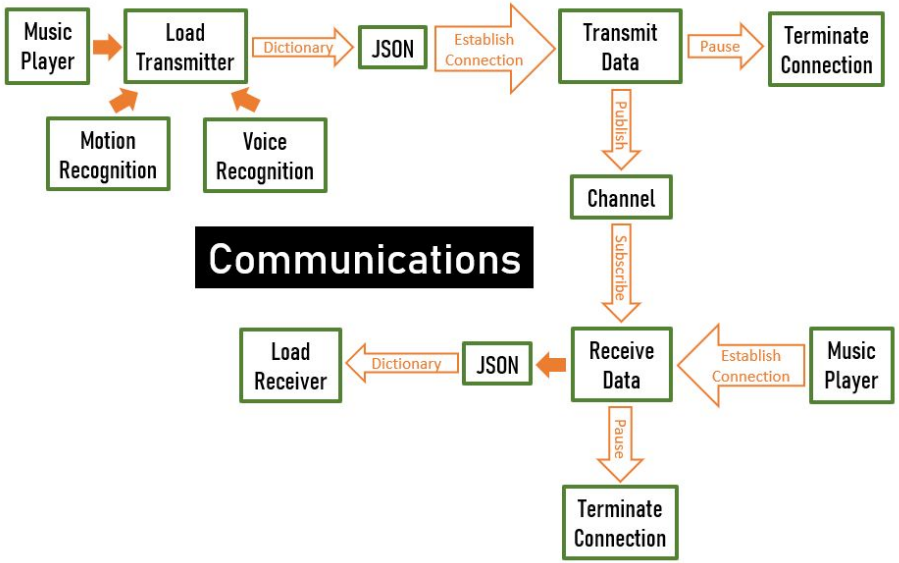
- **Music Player** - Everyone
  The player
- **File System** - Hyounjun
  Data storage, modification, and maintenance
- **Online-Song-Search** - Hyounjun
  Search for songs you don't have on YouTube
- **Emotion Detection** - Gerald
  Detect user emotions and plays songs accordingly
- **Graphic User-Interface** - Gerald & Justin
  Everything you feel, see, and touch **: >**
- **Gesture Control -** Justin
  Navigate controls with gestures
- **User Manual -** Justin
  How to set up Smartify **:)**
- **Communications & MQTT -** Karunesh
  Cross device communication for multi-channel sharing
- **Voice Recognition -** Karunesh
  Voice control - "Mini Alexa"

Work Breakdown

Hyounjun Chang	**Integration Manager, Player Features Design/Implementation**	• Worked on Center Framework (Music Player), implemented initial GUI • Integrated Music player with other modules • Implemented Player Features like “Online-Song-Search”
Karunesh Sachanandani	**Main: Voice Recognition MQTT/Communication**	• Audio recognition module, noise filtering, voice command parsing • Integration between multiple Devices through Wireless Communication (MQTT)
Gerald Ko	**Main: Emotion Detection Graphical User Interface, QA**	• Implement emotion detection, trained and fine-tuned the model • Integrate emotion detection with the player • Implement the GUI of the player • Improve player experience based on user feedback
Justin Suh	**Main: UI/UX, Product Manual Motion Detection**	• Create a User Manual • Create gesture recognition, filtering noisy signal • Integrate MQTT to send commands remotely • Contact test users for feedback

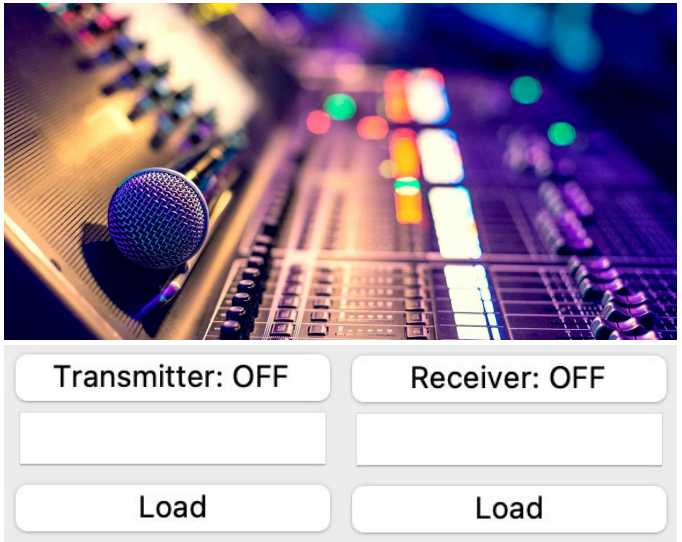
Communications - MQTT

- Only accessible from other music players → Transmit commands from “controllers”
- “Instantaneous” transmitting/receiving → Real time music sharing
- Fields:
  - Command
  - Song name
  - Artist name
  - Song time



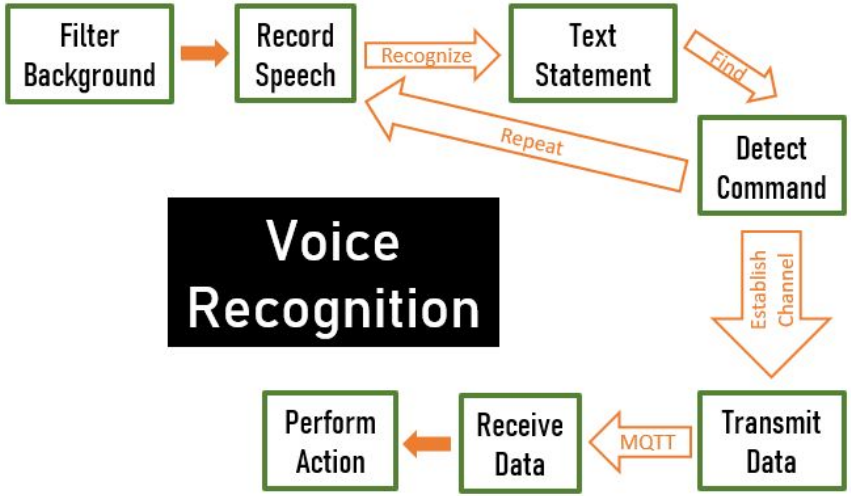
Sharing Rooms

- Prototype only supported 1 “room” for all users
- Now users can join customized “rooms”
- Sharing music is now automatic:
  - Users can turn it “on” or “off”
- Just like a mini-radio station!



Voice Recognition

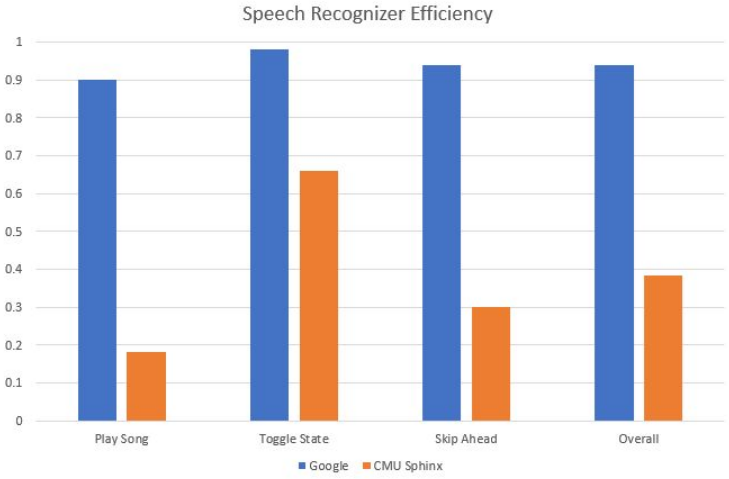
- Python Library of Speech Recognition
- Commands:
  - Play a song
  - Toggle state
  - Skip ahead
- Increased versatility of accepted commands
- Integrated voice recognition directly with the player GUI, along with an additional voice controller
- Smoother user experience during command recognition difficulties



Voice Recognition

- Python Library of Speech Recognition
- CMU Sphinx < 40% efficiency

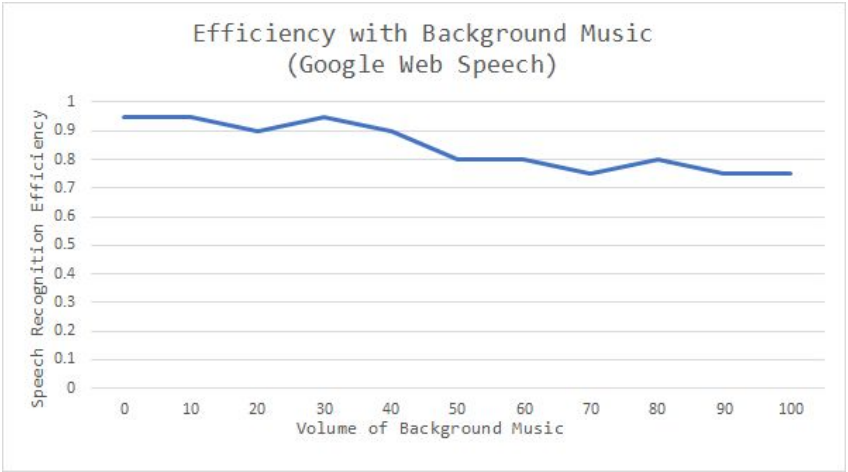
  Google Web Speech > 90% efficiency



(50 samples of each command)

- Background noise filtering:

  ~75% efficiency even at max volume



(20 samples at each volume)

Motion Detection - Gesture Control

Purpose/Function:

- Reads the gestures
- Determines the gestures
- Sends command using MQTT

IMU Accuracy (Rounded):

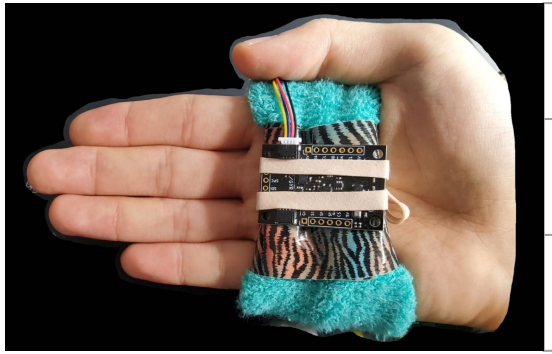
- Toggle - 95%
- Next - 95%
- Previous - 95%

IMU Latency:

- Around 0.3 sec (0.2sec is intentional)

Implementation:

- Dynamic thresholding by speed of hand movement
- Attach IMU to wearable for ease of use

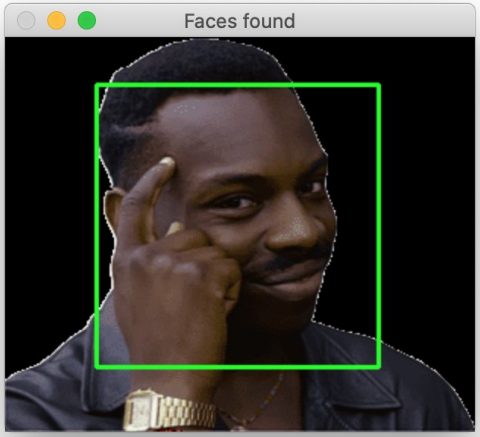


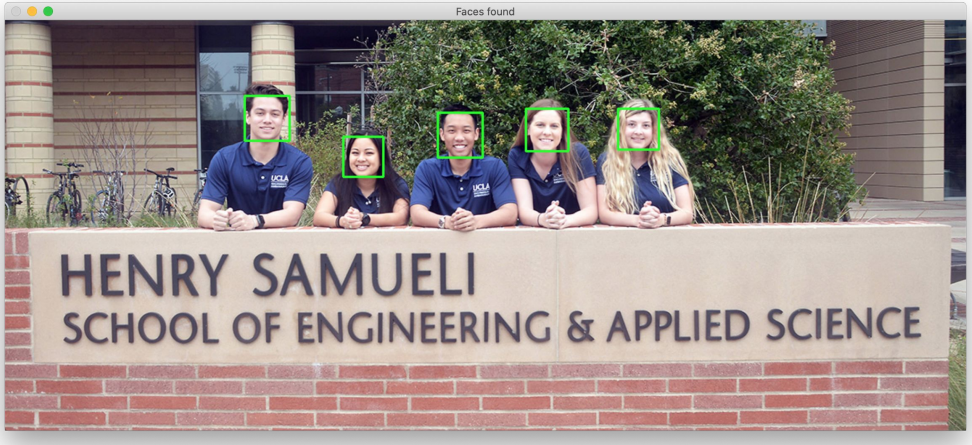
Toggle	Lift up and down
Next	Tilt Right and back
Previous	Tilt Left and back

Getting to Motion Detection:

- Download modules folder into raspberry
- Connect to Raspberry Pi with IMU connected
- Run
  python -m IMUControl.gesture

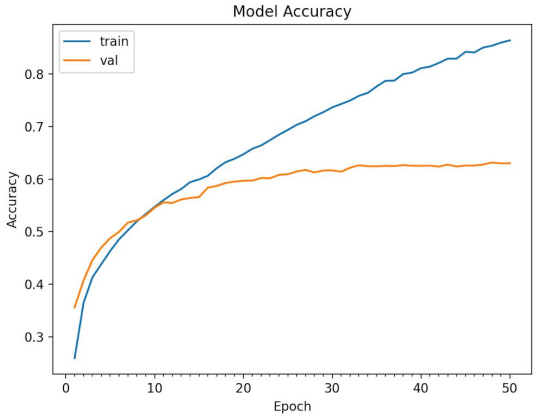
Face Detection - Computer Vision

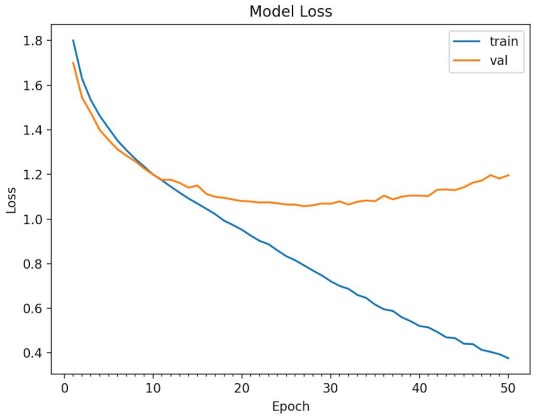


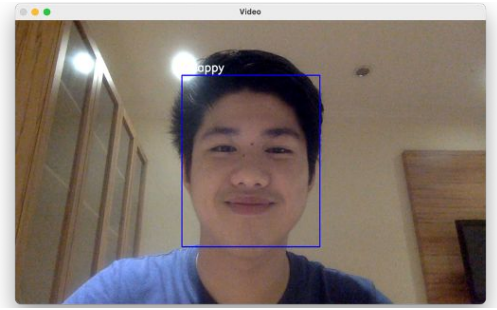


Emotion Detection

- Identification of 7 different emotions: neutral, happy, angry, sad, disgusted, fearful, and surprised, using deep convolutional neural network (4-layer CNN)
- The trained model achieved a test accuracy of 63.2% using the **FER-2013** dataset published by the International Conference on Machine Learning (ICML).
- To improve accuracy, three identical emotions have to be detected consecutively.

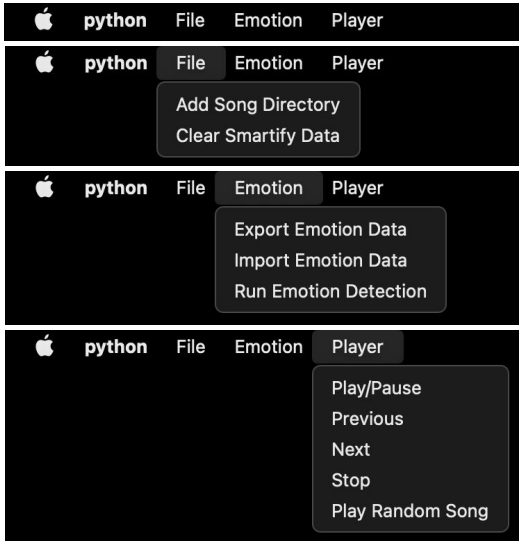


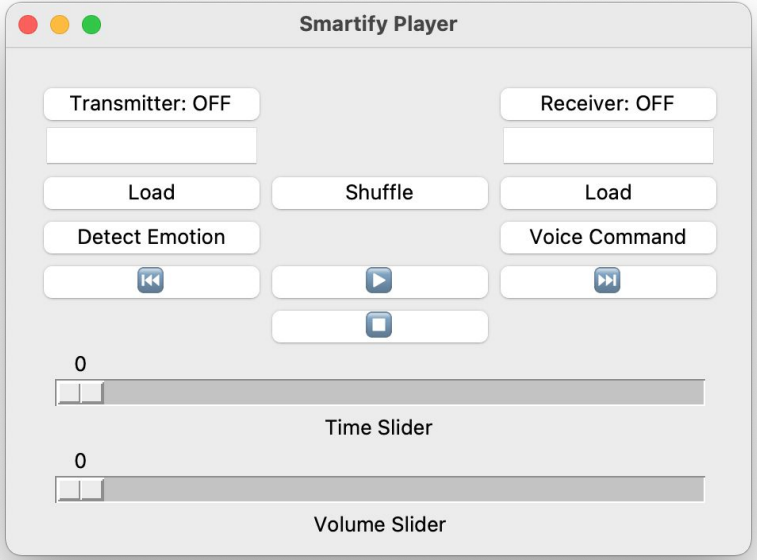




Graphical User Interface (GUI)

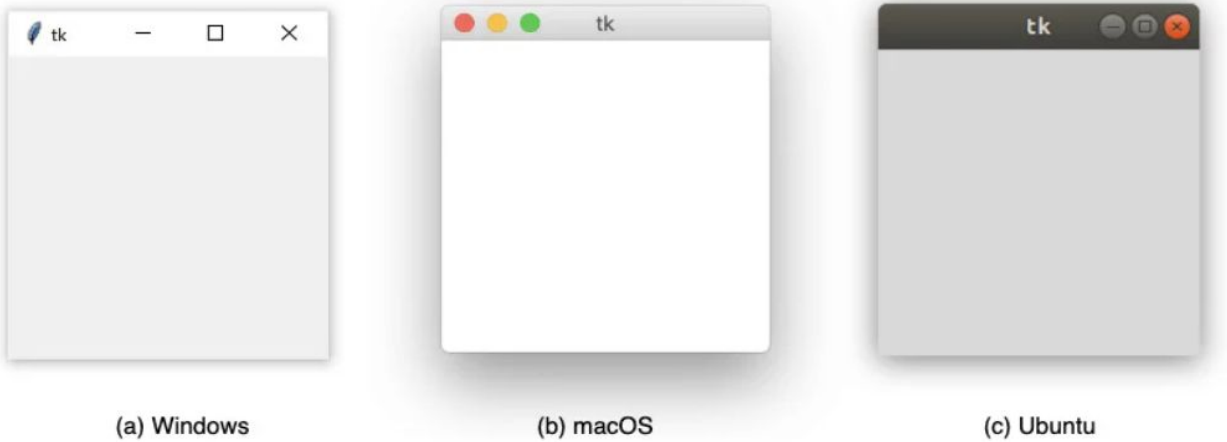
- Menu bar to simplify navigation.
- Buttons that change based on state of the player.





Cross Platform Support

- It works on Windows, Mac, Linux (basically all consumer-level)!
- All dependent libraries compatible with three OS's
- And in theory, on device that can run python and VLC (open-source player)



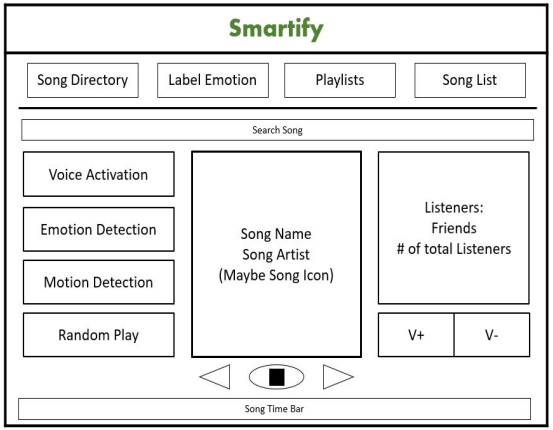
(a) Windows (b) macOS (c) Ubuntu

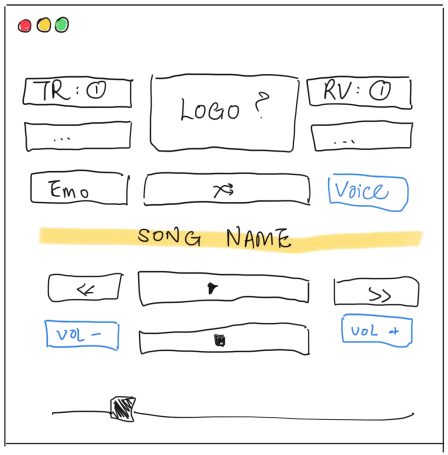
Sidenote: Visuals may vary by OS

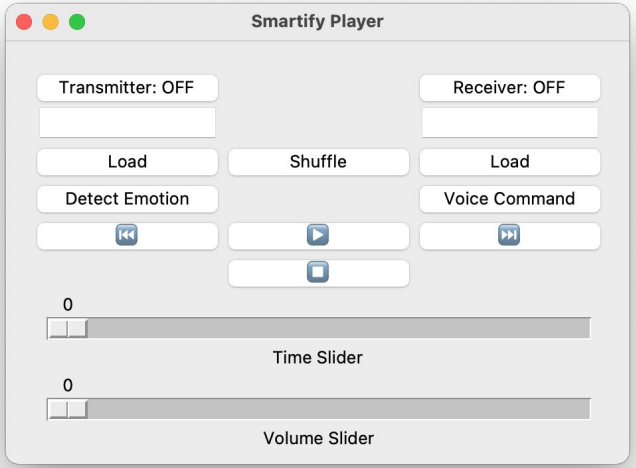
Growth of User Interface (GoUI)

Goal:

Create a simple and intuitive UI with all the functionalities we want.







Smartify Player & Files Module

- Smartify Player is based upon the open-source VLC-media player! (VLC needs to be installed on device as well)
- Player features (ie. play, next song, online audio-stream) are from VLC player itself
- Songs are stored in a .csv like-structure (updated and stored as .csv/Excel File)
- User will need to update the .csv file for best experience with emotion-based song playlist!

	A	B	C	D
1	path	artist	title	emotion
2	/Users/GeraldKo	Dua Lipa	IDGAF	0
3	/Users/GeraldKo	Shaggy	It Wasn't Me	6
4	/Users/GeraldKo	SAYGRACE	Boys Ain't Shit	1
5	/Users/GeraldKo	A Great Big Wor	Say Something	5
6	/Users/GeraldKo	Harry Styles	Watermelon Sug	3
7	/Users/GeraldKo	Lucy Park	Rain	4
8	/Users/GeraldKo	Eminem	Not Afraid	2

Online Song-Search

- Tired of having to store your song to listen to music?
- Any song not in the player can be looked up online! (Warning: increased bitrate usage)
- Song search via Youtube-Search (same as searching on Youtube!)
- Smooth transition between offline-songs and online songs!
- Smartify saves bitrate by only playing songs not found offline!

User Manual

Goals

- Downloads required software and libraries
- UI Labels
- Feature list/Function
- Feature Instructions
  - Basic functions
  - Emotion
  - Voice
  - Motion
  - Transmit/Receive

Usability Testing - Feedback

Library:
- Issues with compatibility

➜ Conda Environment for Windows/Mac Provided! (PyAudio needs to be manually installed since Smartify requires 3.8+)

Ubuntu:
- Issues loading UI
- Issues with Voice Recognition

➜ Bug Fixes Implemented

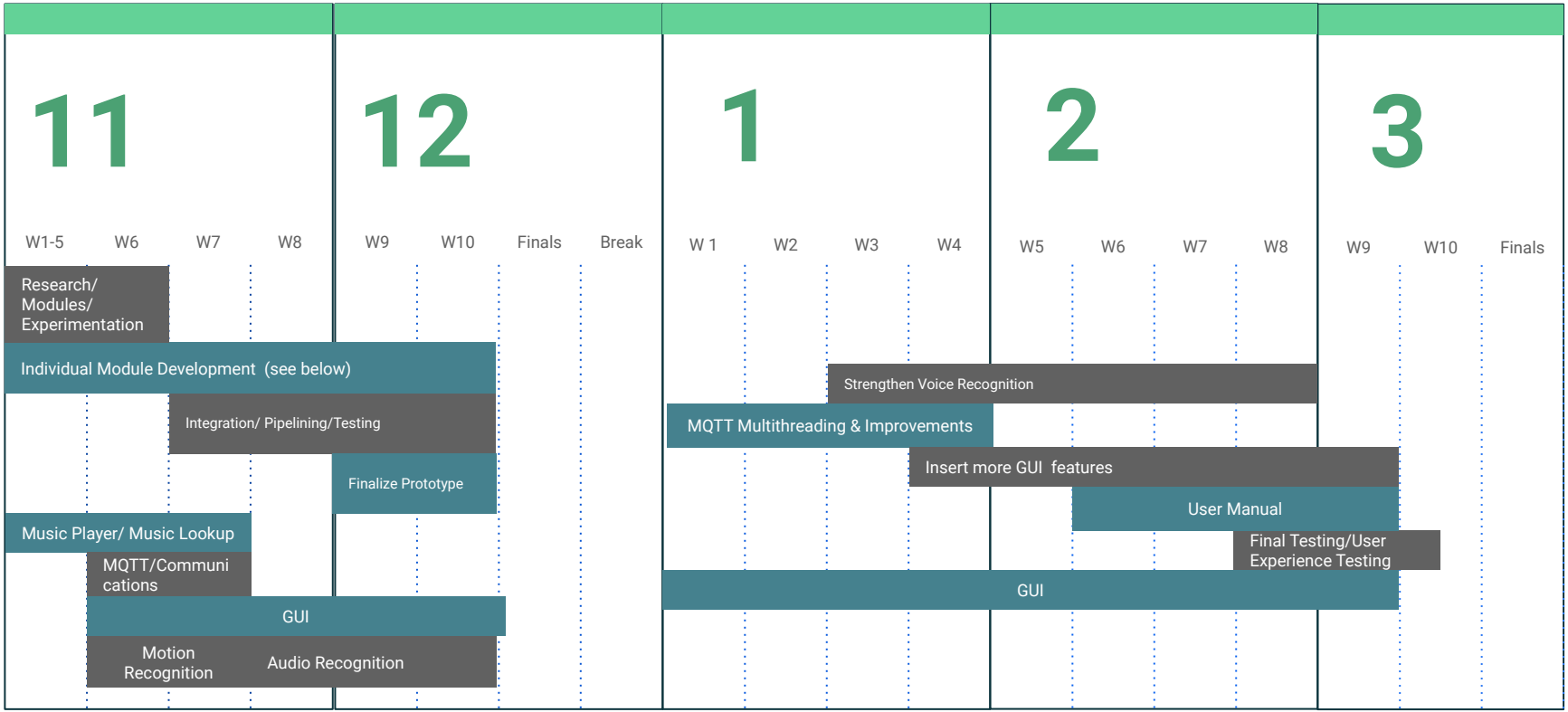
User Manual:
- Post on github Wiki
- Transmit/Receiver unclear instructions
- Emotion Detection unclear instructions

➜ Edited the user manual on those areas

Final Feedback:
- UI is intuitive
- Audio output is clear
- Playlist load/Shuffle works

Usability Testing with Team 8

Project Timeline



DEMO

Drop the tunez.

DEMO

Gesture Control

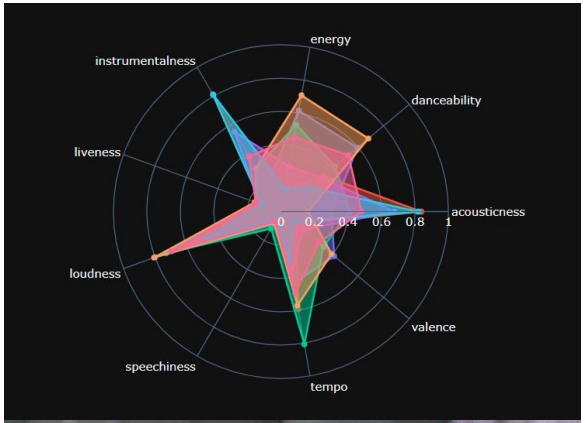
Smartify's Future

There are still Improvements that could be made to Smartify!

Just to name a few:

- Better GUI Design (More colors!)
- Self-Learning Song Suggestion Algorithms using ML
- Using Signal Processing to assist song suggestions!
- Better motion detection/controls system

However, it was a good learning experience for all of us!



Attributes Spotify uses to characterize songs

Emotion-based song suggestion using Signal Processing

- Fast-Fourier Transforms can be used to define musical characteristics such as Intensity and Timbre! (Music is related to frequencies!)
- We can use ML algorithms (ie. K-Nearest Neighbors) to suggest similar songs!
- This model was dropped from Smartify Project due to time/resource constraint.
- With servers (ie. AWS) and song licenses, we could store these values and suggest songs based upon these characteristics like Spotify!

Source: https://sites.tufts.edu/eeseniordesignhandbook/2015/music-mood-classification/

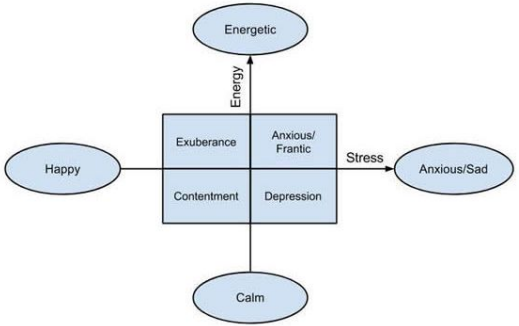


Figure 1
Thayer's mood model. Source: Derived from Bhar (2014).

Mood	Intensity	Timbre	Pitch	Rhythm
Happy	Medium	Medium	Very High	Very High
Exuberant	High	Medium	High	High
Energetic	Very High	Medium	Medium	High
Frantic	High	Very High	Low	Very High
Anxious/Sad	Medium	Very Low	Very Low	Low
Depression	Low	Low	Low	Low
Calm	Very Low	Very Low	Medium	Very Low
Contentment	Low	Low	High	Low

Thank You!

Questions?