Use Case/Application

- A virtual music instrument controlled solely by hand movement and gestures.
- CV for pitch control.
- Wireless gloves for volume control and musical effects.
- Synthesizer for customizable tone color, selected using gestures.
- Real-time playing on laptop speaker.
- Makes playing music more affordable and more convenient.

Quantitative Design Requirement

Criteria / Metric	Detection Feedback Speed	Portability	High Gesture Accuracy	High Sound Production Accuracy	Environment Lighting Sensitivity	Battery Life	Plug-and-Playability
Description	Time delay from movement of hand to sound being produced	Expected weight of our whole system	The accuracy of gesture recognition	The produced sound should be within certain error range of the desired frequency and volume	The minimum lighting requirement needed for the system to function	Working time after one full charge	The success rate of setting up on mac OS
Requirement	<= 10 ms	<= 6 pounds	>= 80%	+- 3.6Hz +- 0.5dB Error	25 lumens	>=5 hours	>= 90%

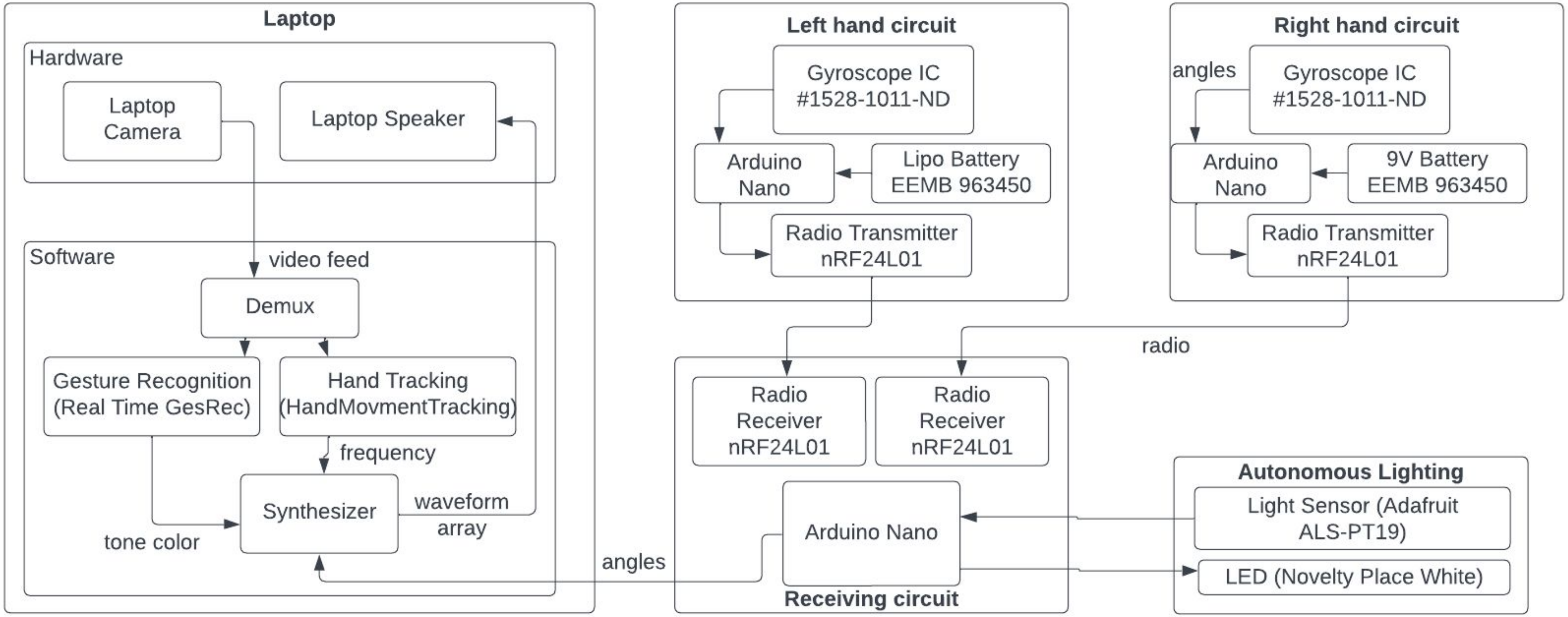
Solution Approach

1. High Gesture/ Movement Detection Accuracy
    a. CV (motion detection) + wearable circuit gloves (rotation detection)
    b. Modifying existing CV tracking algorithm
    c. Use a pretrained model for gesture recognition
        i. We no longer think that it is reasonable to collect data and train a good recognition model in the short amount of time given since there are more important things we need to focus on.
2. Accessibility
    a. Sending data wirelessly from circuit gloves will give users more freedom in movements.
    b. We will develop our system on macOS.
        i. Windows is not going to be supported because it is too complicated
3. Detection Feedback Speed
    a. We will use radio transceivers for their low latency and power consumption.
    b. We will perform network pruning to minimize the recognition latency.

- Volume control: we decide to use relative instead of absolute number levels. For example a complete tilt (90°) will be 100% volume, and a flat hand (0°) will be silent
  - a reason for this design choice is because the output volume will depend also on the user's laptop speaker, it will be hard to formulate the same thing for every laptop
  - this is easier for the user to remember compared to absolute levels
- Type of battery: Lithium polymer battery to power our gloves
  - rechargeable, lighter, higher power density than alkaline battery
  - more economically friendly
- Arduino selection: arduino nano instead of uno to keep our gloves light
  - arduino nano is 7g, while arduino uno is 25g

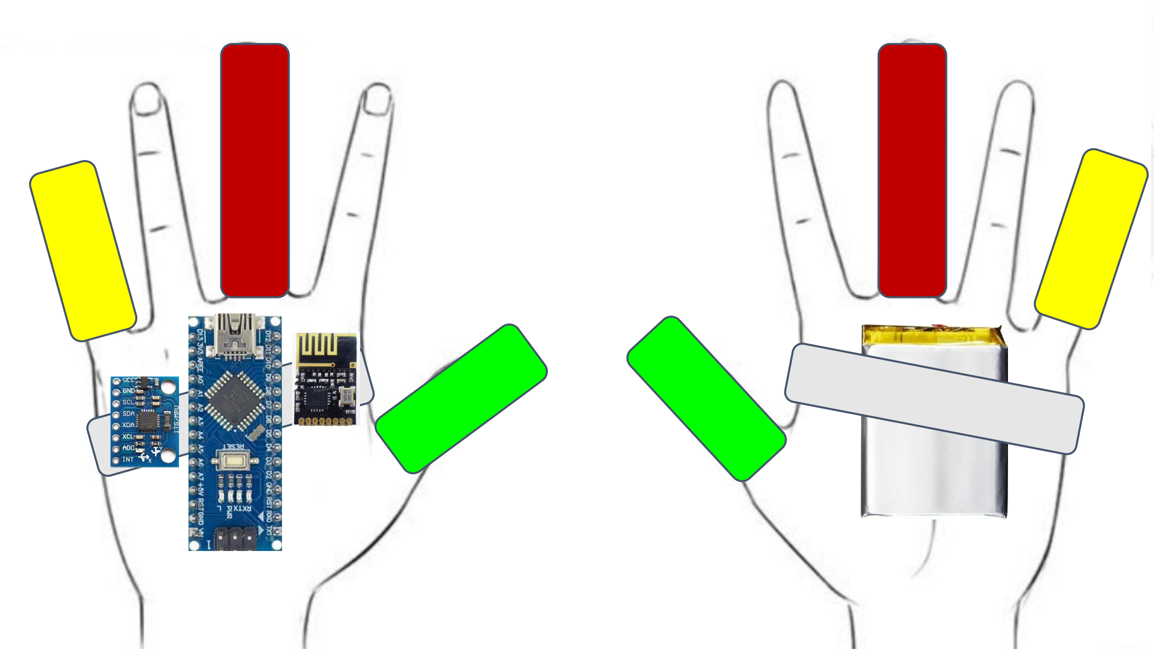
Solution Approach - Specific Design Choices

Block Diagram



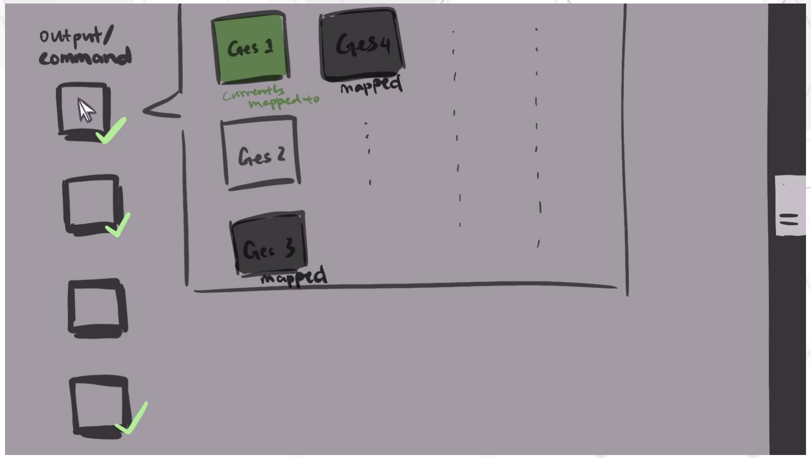
Additional Diagrams

Glove Diagram



Screen area labeling (16:9 ratio)

Mapping



Implementation Plan - Item list (Physical)

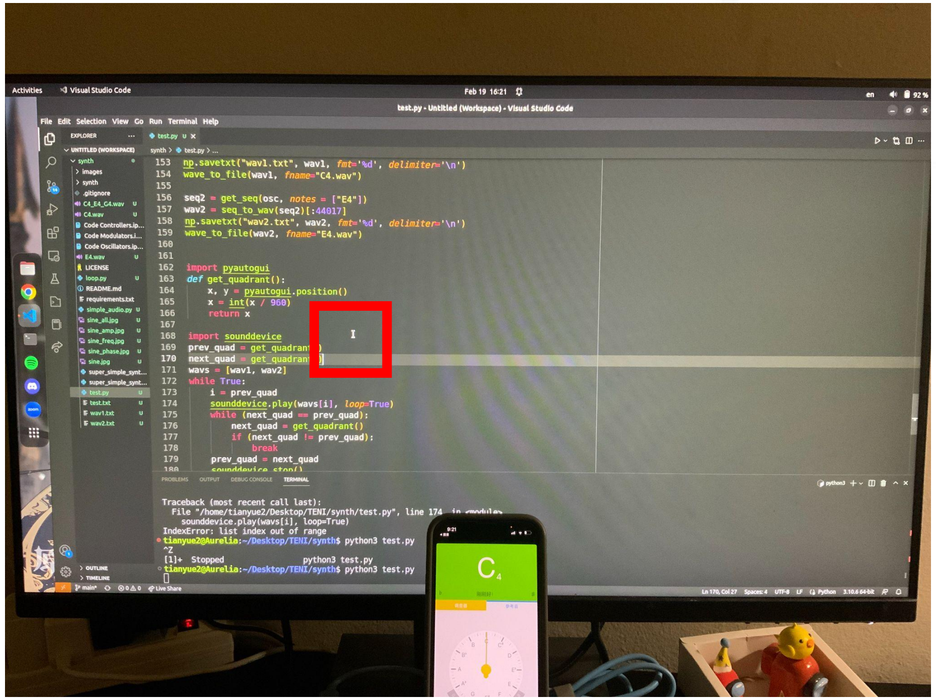
- Light sensor - Adafruit ALS-PT19
- LED - Novelty Place Ultra Bright Clear 3V 20mA LED
- Arduino Nano
- Tilt sensor - #1528-1011-ND from Digi-Key
- Radio transmitter - NRF24L01
- Radio receiver - NRF24L01
- Basic circuit components - from ECE Lab
  - Wires, power source, cables, breadboard, etc.
- Battery - 3.7V 1800mAh Lipo Rechargeable Battery Pack
- Gloves
- Laptop - Normal Macbook

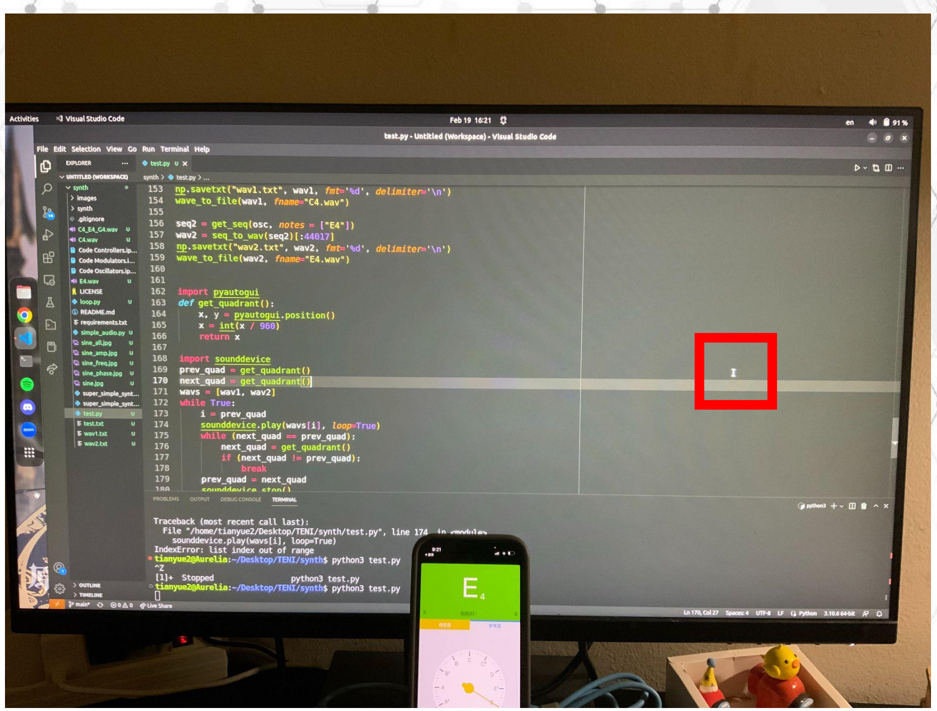
Implementation Plan - Item list (Software)

- Hand gesture recognition model
  - Real-time-GesRec
  - Hand-gesture-recognition-mediapipe
- Hand movement tracking model
  - Hand Movement Tracking
  - Yoha
- Synthesizer
  - Packages: librosa, itertools, seaborn, scipy, sounddevice, numpy
  - Back up plan: Surge, an open source synthesizer
- Interactive software mapping sound, command and gesture
  - Front end & UI
  - Data saving and mapping

Implementation Plan - Progress

A working synthesizer: https://youtu.be/GxFPV0mgCcU





Carnegie Mellon University
Computing Services

Testing - Gesture Input & Sound Output

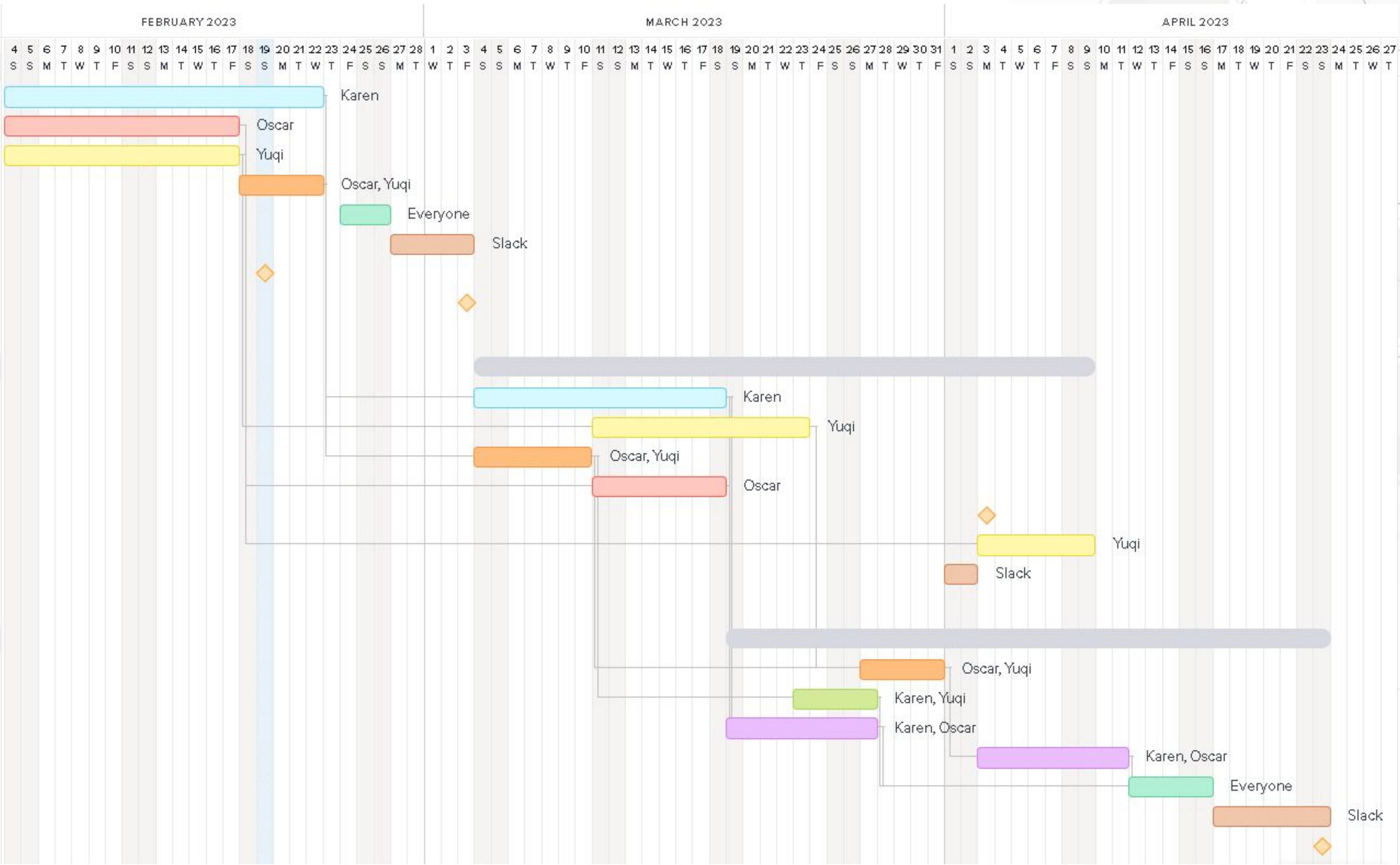
- Input a random sequence of all gestures for recognition accuracy
  - pass if accuracy >= 80%, repeat test multiple times for a more accurate recognition accuracy
  - otherwise use a different recognition model from our back up list
- Use a slow-motion camera to time the sound production latency.
  - pass if trigger and sound output is less than 10 ms, otherwise swap to faster components
- Use a spectroid (audio spectrum analyzer) to test pitch accuracy.
  - pass if output pitch is within +- 3.6Hz of the predetermined pitch
- Use decibel meter to test volume accuracy.
  - pass if halving the tilt angle reduces the sound by 10dB +- 0.5dB
  - absolute volume between 0dB ~ 60dB

Testing - Other Metrics

- Weigh the overall system (every component together) on a scale
  - if the system is >= 6 pounds, use smaller Lipo batteries
- Light meter (light intensity app) to test performance in dark environment
  - >= 80% accuracy in environments above 25 lumens, otherwise use more/brighter LED
- Battery-life test to determine the gloves' battery life
  - record the time for lipo batteries to fall below 7V (nano's Vin threshold).
- Ask participants to set up our system on macOS to test plug-and-playability.
  - if setup fails, fix it and add solution to our installation workflow

Project Management - Task & Schedule

	Assigned	Progress
▾ Design		0%
Hand monitoring system design	Karen	0%
Synthesizer design	Oscar	0%
Wearble circuits design	Yuqi	0%
Arduino design	Oscar, Yuqi	0%
Design review	Everyone	0%
Slack	Slack	0%
Design slides		
Design document		
▾ Build & Component Testing		0%
Hand monitoring system	Karen	0%
Wearable circuits(RH)	Yuqi	0%
Arduino	Oscar, Yuqi	0%
Synthesizer	Oscar	0%
Interim demo		
Wearable circuits(LH)	Yuqi	0%
Slack	Slack	0%
▾ Integration		0%
RH + Arduino	Oscar, Yuqi	0%
Arduino + Hand monitoring system	Karen, Yuqi	0%
Hand monitoring system + Synthesi...	Karen, Oscar	0%
LH+ RH + Arduino	Karen, Oscar	0%
Final Integration Test	Everyone	0%
Slack	Slack	0%
Final Presentation		



Carnegie Mellon University
Computing Services