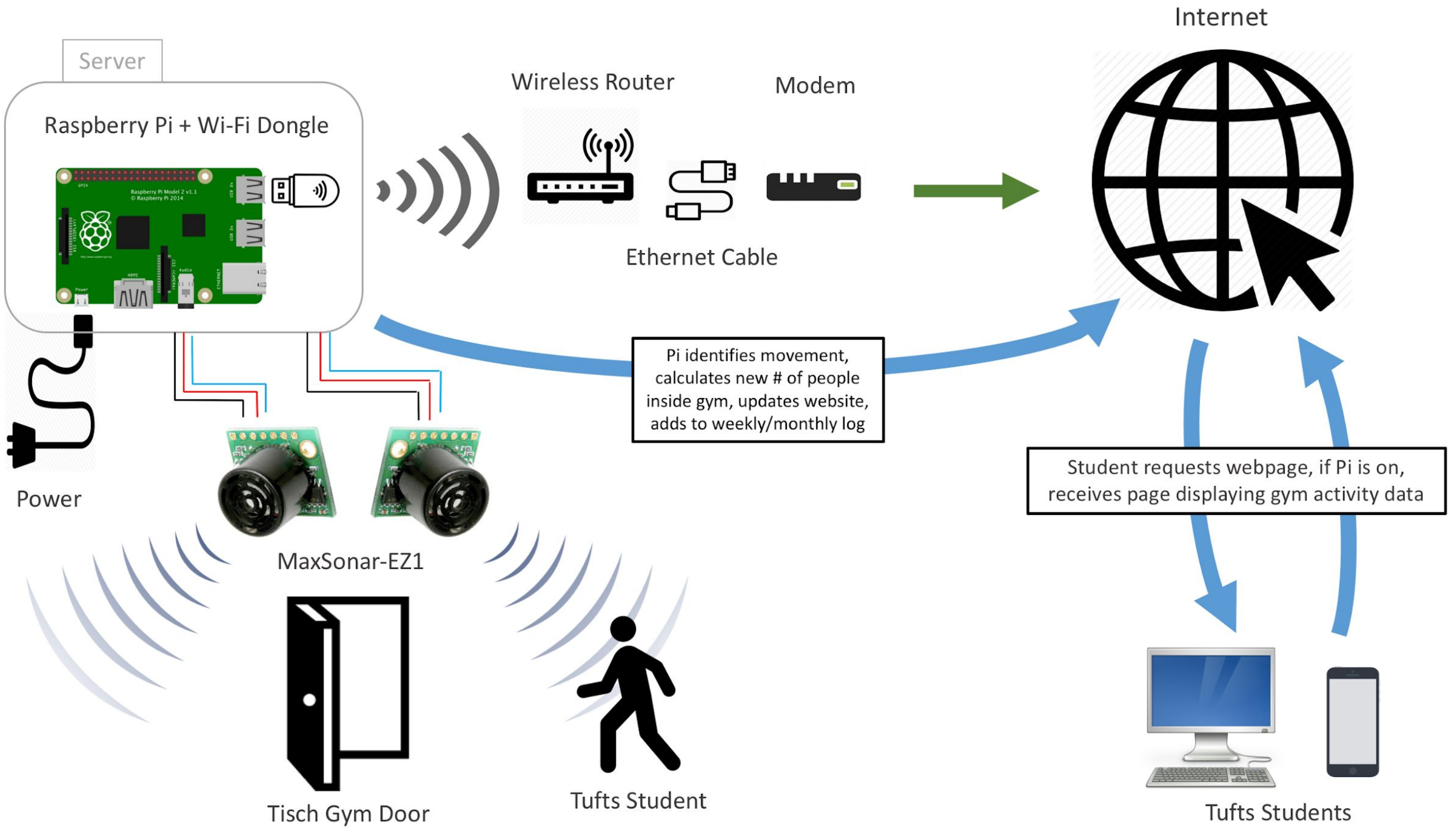
# TUFTS
## GYM COUNT

COMP20 FINAL PROJECT - SPRING 2018 - TEAM 19

Noah Hill - Peter Lam - Benson Cheng - Robert Yang

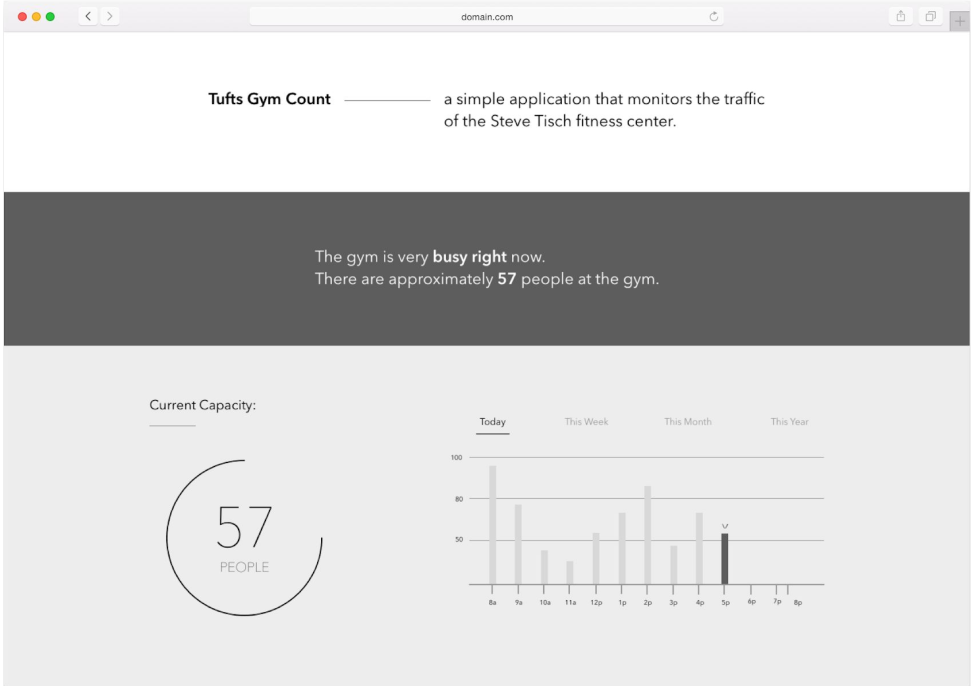
# INITIAL SYSTEM DIAGRAM



Pi identifies movement, calculates new # of people inside gym, updates website, adds to weekly/monthly log

Student requests webpage, if Pi is on, receives page displaying gym activity data

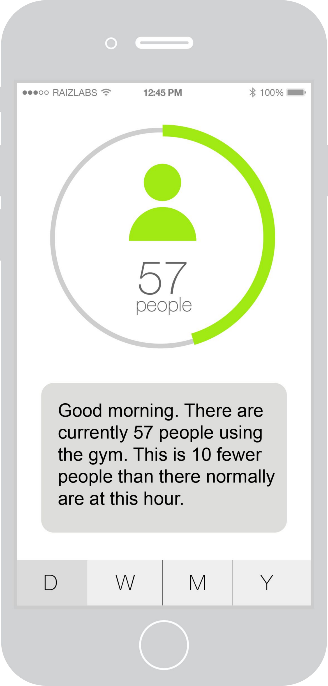
# INITIAL INTERFACE DESIGNS



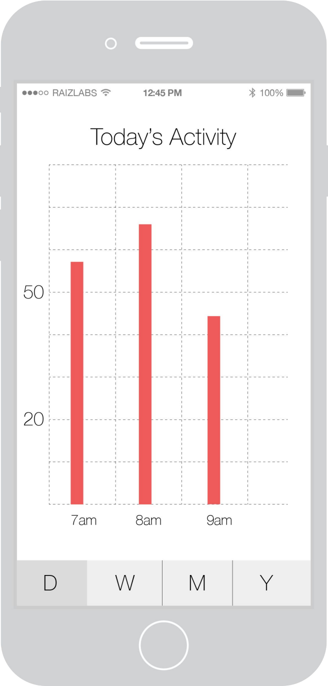
**Tufts Gym Count** — a simple application that monitors the traffic of the Steve Tisch fitness center.

The gym is very **busy right** now.
There are approximately **57** people at the gym.

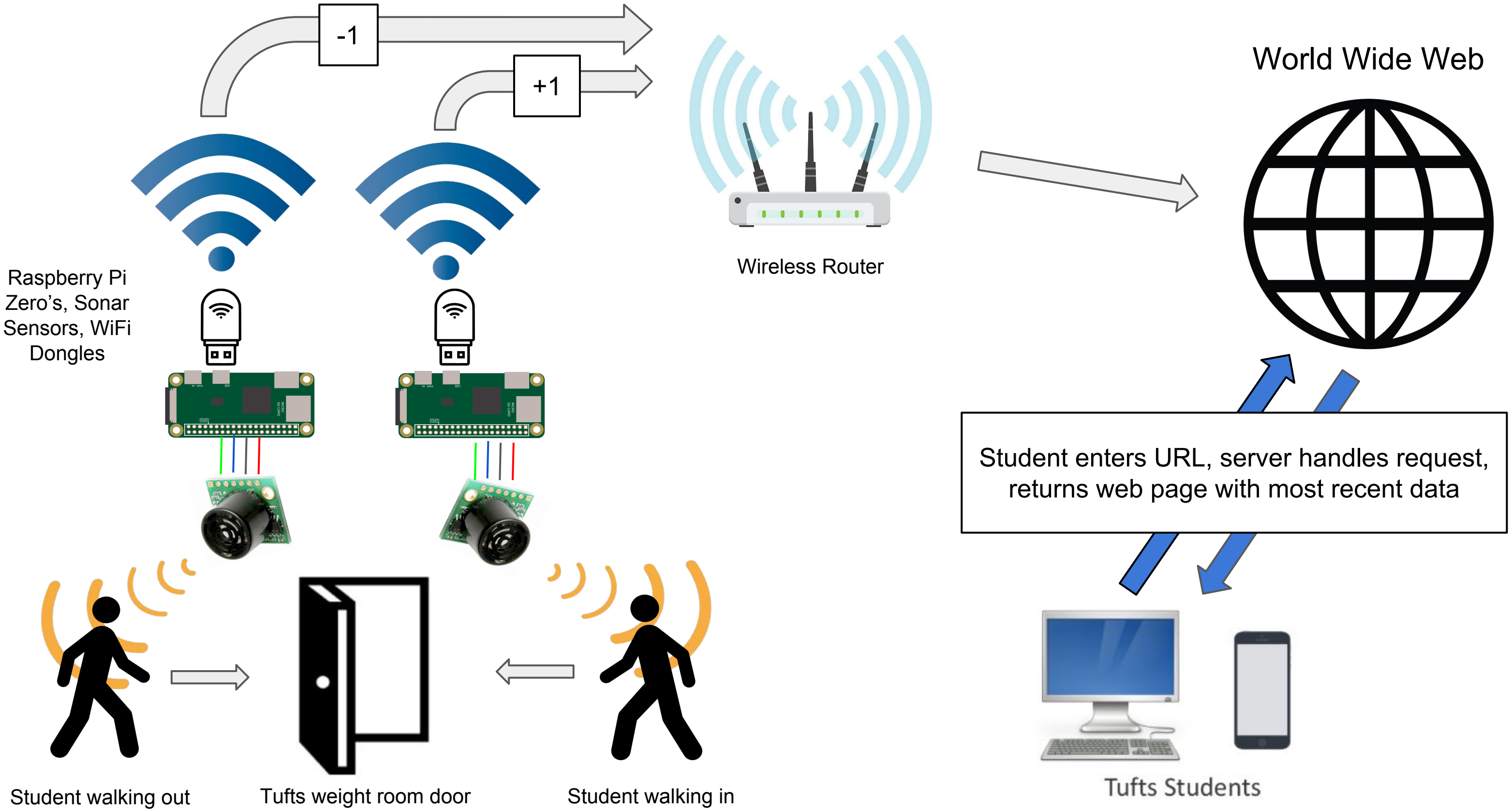
Current Capacity:



Good morning. There are currently 57 people using the gym. This is 10 fewer people than there normally are at this hour.



# FINAL SYSTEM DESIGN



-1
+1
Wireless Router
World Wide Web
Raspberry Pi Zero's, Sonar Sensors, WiFi Dongles
Student enters URL, server handles request, returns web page with most recent data
Student walking out
Tufts weight room door
Student walking in
Tufts Students

# IMPORTANT SPECS

## Movement Sensing

- **Range:** 10.5 feet away from door
- **Sampling rate:** 65us
- **Accuracy:** +/- 4cm
- **Algorithm:**
  1) Run lin. reg. on 4 data points
  2) If slope is negative, lin. reg. on next 15 data points
  3) If  negative slope & $r^2 > 0.65$, HTTP POST

## Server & Data Analysis

- Hosted on Heroku w/MongoDB
- Background function to keep server "alive"
- Several HTTP GET routes, one POST
- Dynamically updates html with most recent data when served
- Daily data compiled into averages of weekly/monthly trends at end of day
- Daily data cleared at 12:00am

# LIBRARIES USED

## Raspberry Pi

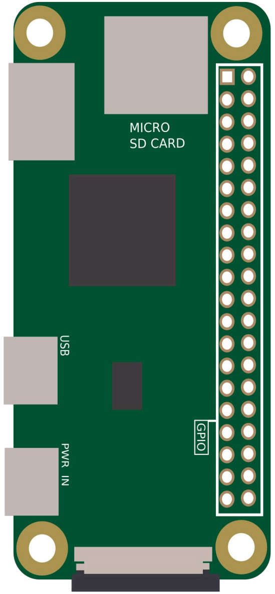
- ARM-version of node.js http://node-arm.herokuapp.com/node_latest_armhf.deb
- NPM regression-js https://www.npmjs.com/package/regression
- NPM node-rpi-ws281x-native https://github.com/beyondscreen/node-rpi-ws281x-native
- NPM forever https://www.npmjs.com/package/forever
- NPM forever-service https://www.npmjs.com/package/forever-service

## Server and HTML

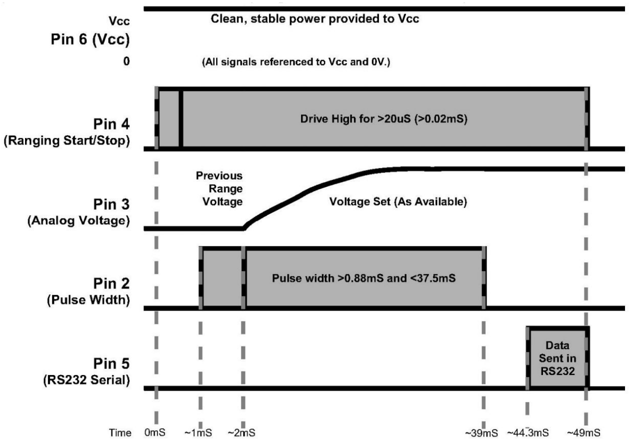
- AnyChart API https://github.com/AnyChart/AnyChart
- NPM Schedule https://www.npmjs.com/package/node-schedule

# HARDWARE USED

Raspberry Pi Zero x2



MaxSonar EZ-MB1010 x2



12 NeoPixel Ring x2

Wifi USB Adapter x2

Many Adapter Cables

# MOST CHALLENGING PARTS

- Circular “live feed” graphic was surprisingly difficult to create and make dynamic with css and html
- Our Heroku server wouldn’t stay “alive” for extended periods, had to make background function make GET requests to keep active
- Reading an analog PWM signal from the MaxSonar EZ-1 without an ADC by manually initiating sonar pulses, timing how long it takes for pulse to return, and calculating distance of objects
- Manipulating NeoPixel strip using node.js, required a confusing wrapper library with many dependencies
- Making Pi’s connect to wifi, turn on NeoPixels, then collect & send data automatically on boot with no user-input
- Cloning disk image from one Pi onto another Pi - ***PLEASE*** don’t get me started on Kernel Panics or file system partitioning

# FUTURE IMPROVEMENTS

## Back-end

- Improved security
- More refined algorithm
  - Edge cases
  - Multiple people at once
  - Line of people
- Build for scalability
- Movement library for others to use and more elegant local use
- More reliable wireless communication
  - Personal Router
  - Static IP

## Front-end

- Student authentication with Shibboleth for security
- Cross-browser stability
- iOS/Android App
- Integrate with Tufts Mobile
- More filtering options and data analysis/trends
- Modular such that other universities can implement
- Update data w/o refreshing page

## Aesthetics

- Higher resolution 3D print
- More discrete sensor housing shape
- Better way to supply power
- Color matched to wall or surroundings
- Designed such that the housing can be easily opened and closed
- Variable-angle mechanism for sonar sensor for doors of any height