Can U Cardio?

A Gym Occupancy Tracking System

Team B2: Ian Brito, Nataniel Arocho-Nieves, Ian Falcon

Use Case

- Students at CMU may find themselves wanting to use the cardio machines at the UC Fitness Center
- With Can U Cardio, users can go online and see which machines are in use.

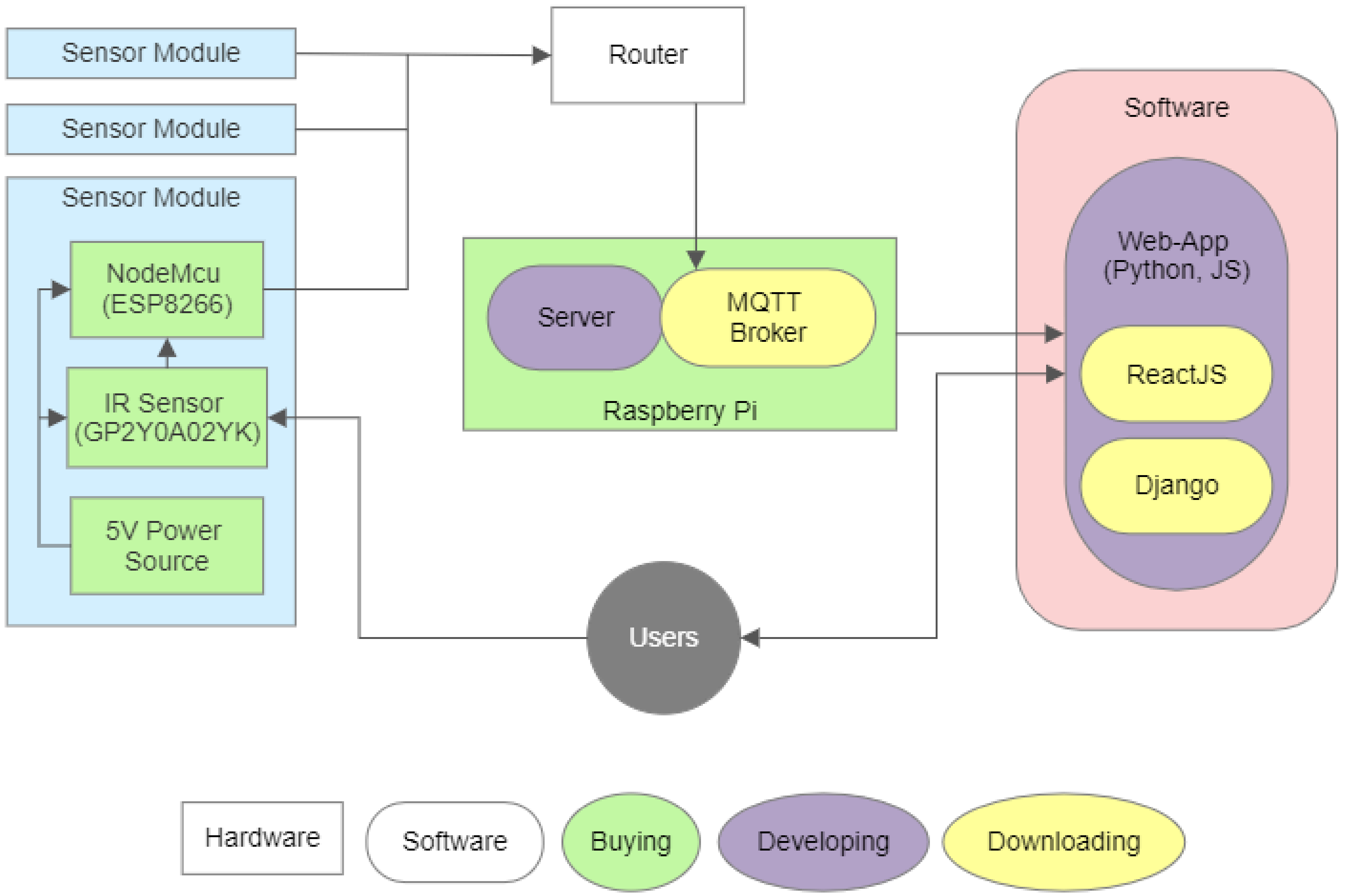
Quantitative Use Case Requirements

Detection delay	5 seconds	Real-time feedback and updates are imperative to attract users to our platform.
Detection Range	3.5 ft.	Farthest back users will typically be is around 4 ft. (on treadmill).
Detection accuracy per machine	> 98%	Customers want to use a reliable system that accurately states occupancy.
Detection Mapping Accuracy	> 98%	We want to ensure that the proper machine is being identified as occupied to avoid conflicts between users.
Battery Life	> 16.5h	UC Gym is open from 6:30 AM to 11 PM on weekdays..
Sensor and module dimensions	60mm x 60mm x 25mm	People at the gym don’t wany anything interfering with the equipment functionalities and with their physical movements. (as least invasive as possible)

Solution Approach

- Proximity based monitoring system
- Physical sensor modules attached to gym equipment’s dashboards
- Wirelessly sends occupancy data to a web-app accessible to gym members and staff.

System Specs/Block Diagram



Implementation Plan

Sensor Module:

- One sensor module will be on each piece of equipment displayed on the web app
- Sharp GP2Y0A02YK IR Sensor
  - Detects when user is within 3.5ft, then microcontroller sends a signal to the RPi to update the web server
- NodeMCU (ESP8266)
  - Code developed for microcontroller will receive data from sensor and communicate with the Raspberry Pi via WiFi

Raspberry Pi 4 Model B:

- Mosquitto will be used as our MQTT Broker to allow for communication between RPi and NodeMCU
- Hosts web app

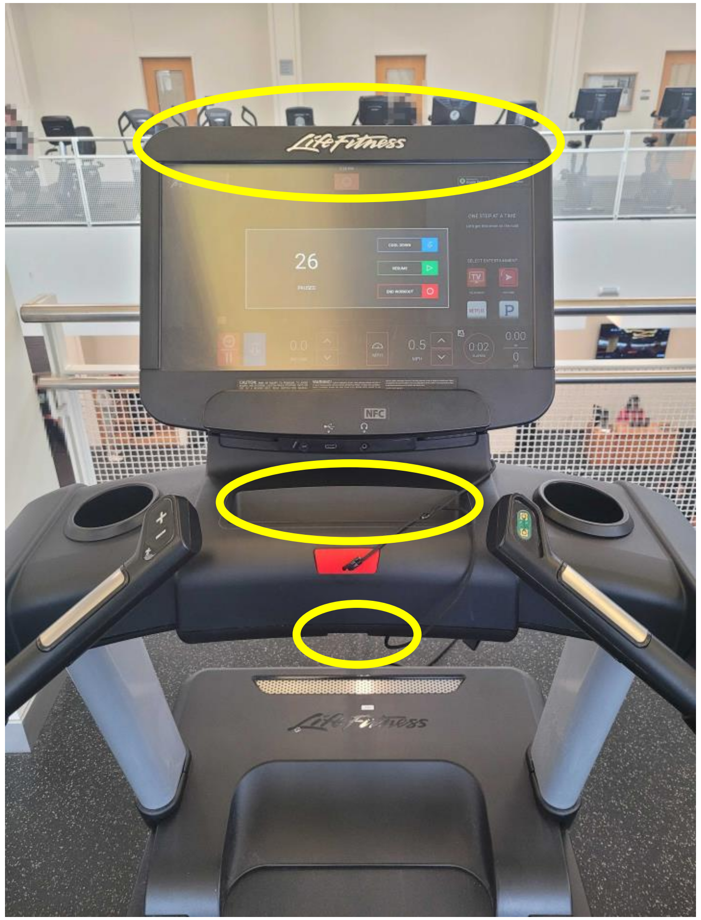
Web Application:

- Django (Python) and React to build web app
- Displays map of gym with occupancy based on data from sensors

Implementation Plan

Mounting mechanism

- Case
  - Dimensions: 60mm x 60mm x 25mm
    - IR Sensor (44.5mm x 20mm x 22mm)
    - NodeMcu (49mm x 26mm x 20mm)
    - Case will be bigger with battery power supply
  - Local fabrication to fit components
- Secure/Fasten
  - Phone bike mount (tube)
  - Adjustable clip (dashboard)
  - Velcro (dashboard & tube)



Possible mounting spots

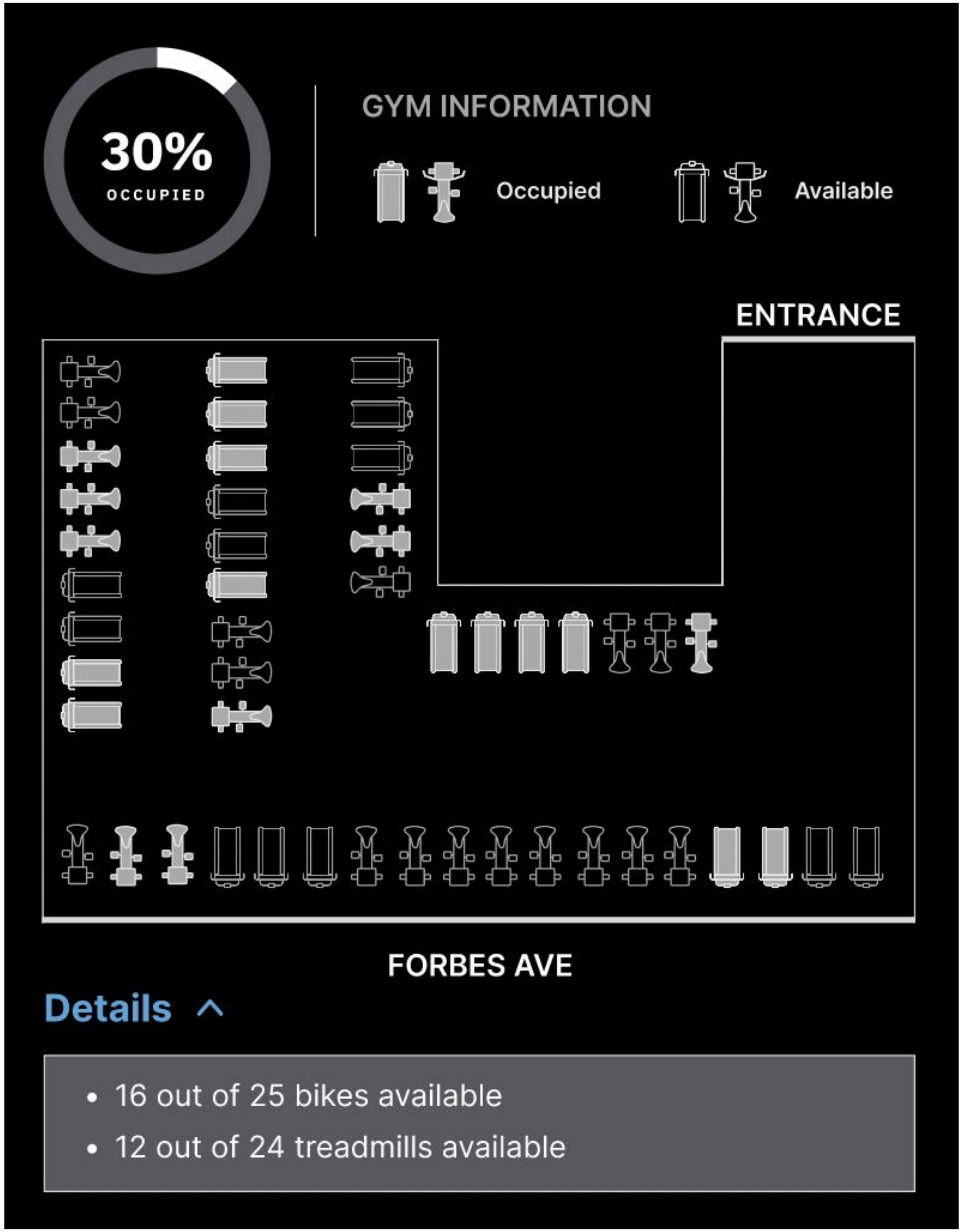


Securing mechanism design inspirations

Implementation Plan

Web app

- Display a layout of the gym depicting occupied vs empty equipment
- Filter by machine
- Display percent of machines in use



Testing, Verification, and Validation

Component	Feature	Method of Testing	Passing Test/Solution
IR Sensor	Proximity detection	Simple LED circuit	Light on
	Calibration	LED circuit with voltmeter, different inclinations, different opacity clothing.	Light on. Measure output voltage between 0V to 3V.
IR/NodeMcu	Communication	Wire NodeMcu and sensors and interface with IDE	IDE reflects data
	Security and safety of mount	Manual stress test on testing setup	No tilt nor interrupted measuring
	Power consumption	Measure current and voltage drawn in a specific time period	Adjust power supply and measure battery life
	Latency (delay)	Send proximity detection data and record time it takes to get to Raspberry Pi (sent & received times)	< 2s

Testing, Verification, and Validation

Component	Feature Testing	Method of Testing	Passing Test
NodeMcu/Raspberry Pi	Communication	Send sample data from NodeMcu to Pi	Sample data accurately received (IDE)
Web App	Displays accurate data	Local test cases	Web app displays machine as occupied. Layout and mapping is consistent.
	Filters	Local test cases	Displays data correctly
	Fluid function	Simulate usage periods of varying durations	No crashes or glitches occur
IR/NodeMcu/Raspberry Pi/Webapp	Detection delay	Measure time sent & time received	< 5s

Project Management

Can U Cardio

Weeks	Week 6	Week 7	Spring Break	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Finals Week
Start Dates	19-Feb	26-Feb	5-Mar	12-Mar	19-Mar	26-Mar	2-Apr	9-Apr	16-Apr	23-Apr	30-Apr
Software											
Order RPi 4											
Create Django web app											
Get web server running on RPi											
Receive data from microcontroller on RPi											
Update web app (backend) with data											
Work on front end											
Hardware											
Order ESP8266											
Develop prototype code for microcontroller											
Develop code for microcontroller											
Develop code to send data to Rpi											
Receive and integrate sensor data into microcontroller											
Signals/Sensors											
Order IR Sensor											
Connecting sensor to microcontroller											
Power sensor module (wired), measure power											
Mounting mechanism for dashboard											
Fabricate testing setup											
Testing											
Slack											
Deliverables											
Design Slides (19-Feb)											
Design Report (3-Mar)											
Ethics Assignment (15-Mar)											
Interim Demo (3-Apr)											
Final Presentation Slides (23-Apr)											
Final Poster (TBD)											
Final Video (TBD)											
Final Report (TBD)											
Public Demo (TBD)											

Key
Ian B.
Nat
Ian F.
Team
Break