



UMassAmherst  
The Commonwealth's Flagship Campus

# SPI – Smart Parking Infrastructure

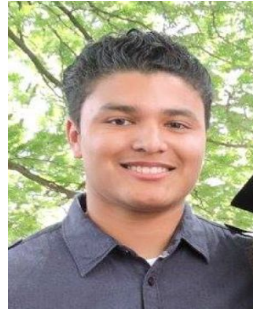
Mid-Year Design Review

December 13, 2019

# Meet The Team



Alexander Donadio



Edwin Munguia



Daniel Perry

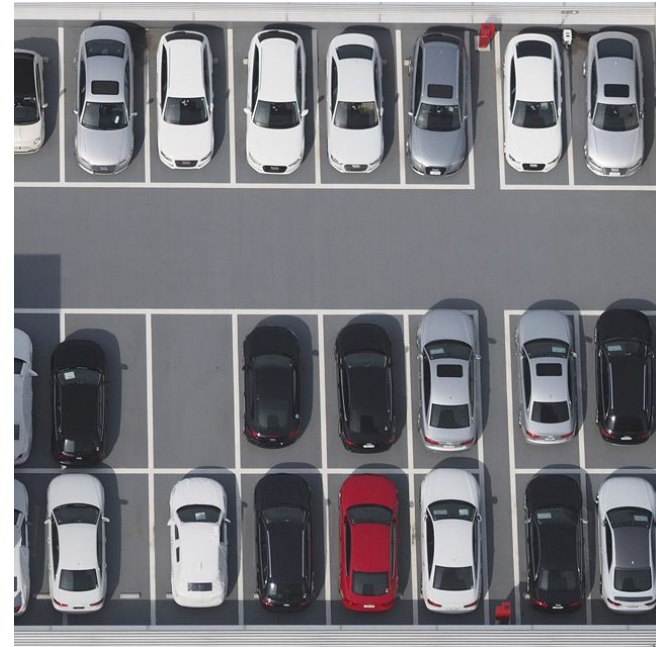


Michael Skaza

**Advisor:** Amir Arbabi

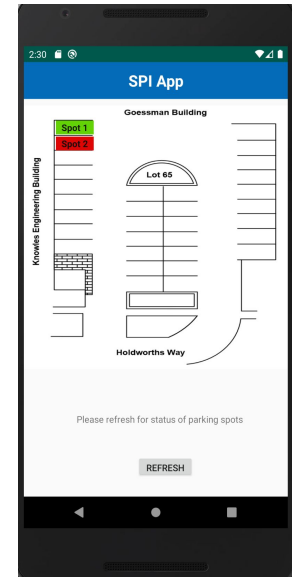
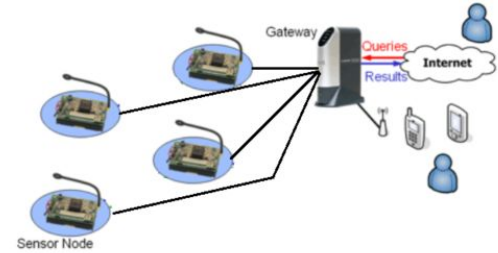
## Problem

- Finding parking, especially remotely before you arrive to your destination is very difficult
- Time, increased CO<sub>2</sub> emissions, and personal stress are all main drawbacks of not having a smart parking infrastructure in place



## Solution

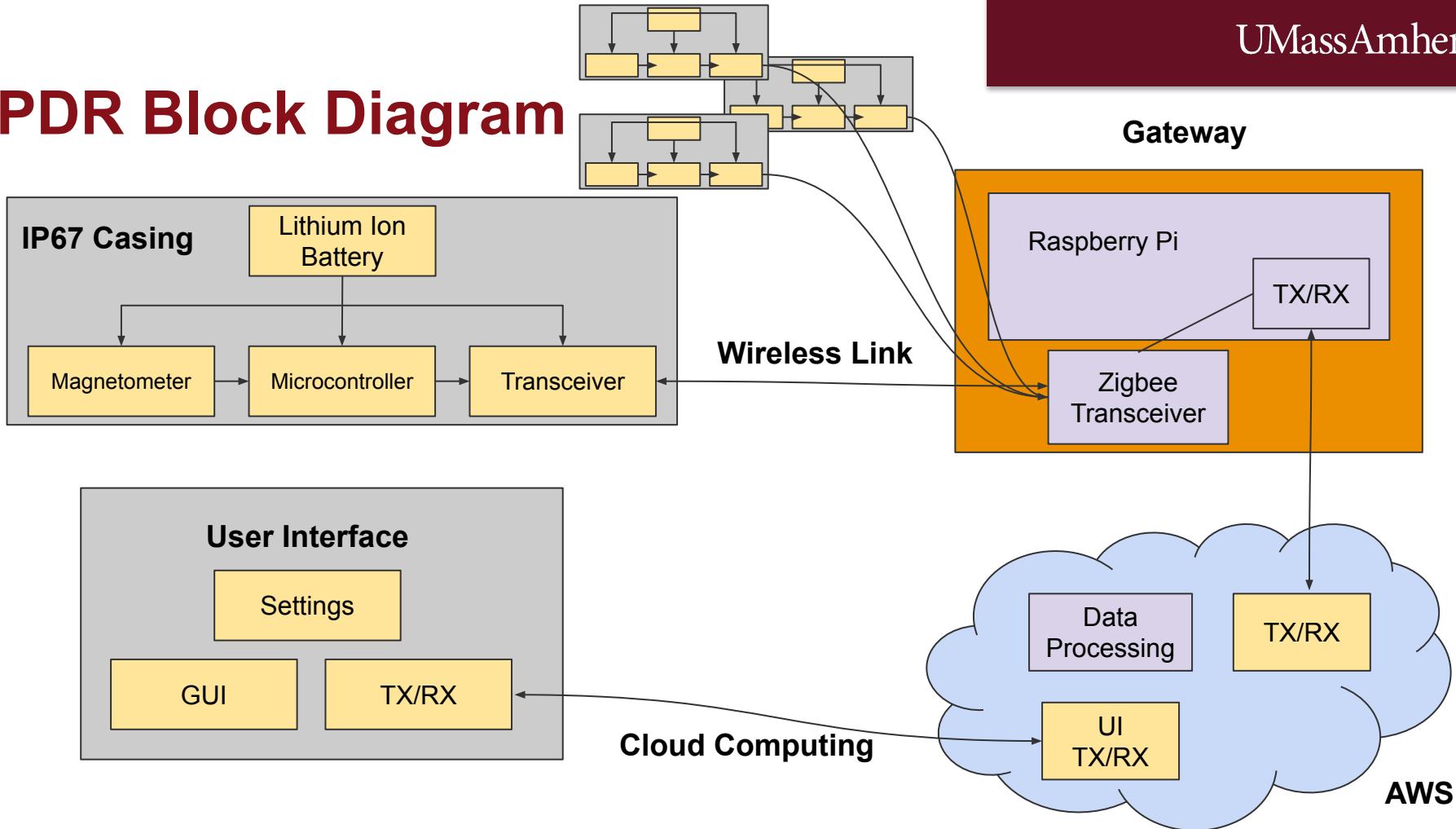
- Wireless Sensor Network of magnetometers and Xbee transceivers
- Android Application
- Cloud computing decision making
- Real time map of parking space availability



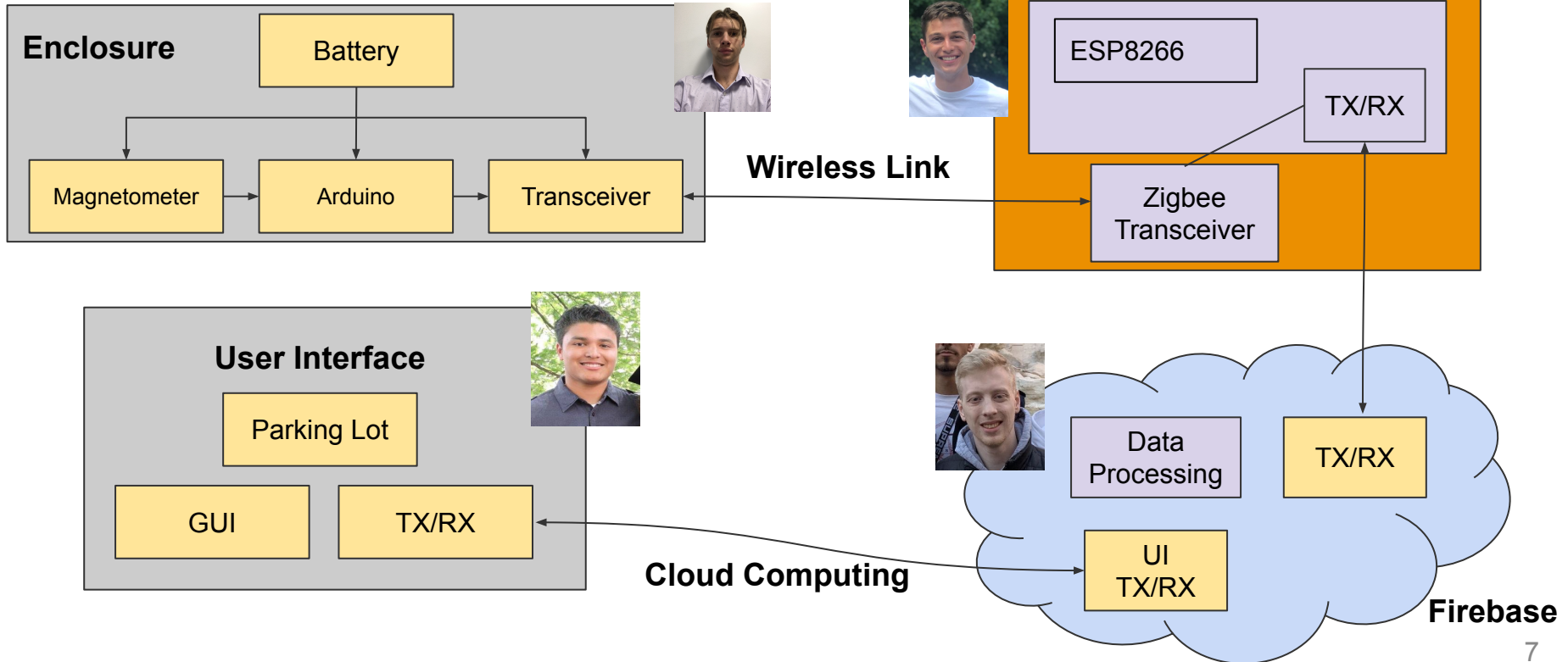
# Final System Specifications

1. Detect the presence of a vehicle with 90% accuracy.
2. The battery should last at least 6 months.
3. Every 1 minute the map is updated
4. User interface is an android application that shows the status of the parking space.
5. Operable at all weather conditions and has IP67 rating enclosure and temperatures ranging from -20°F to 140°F.

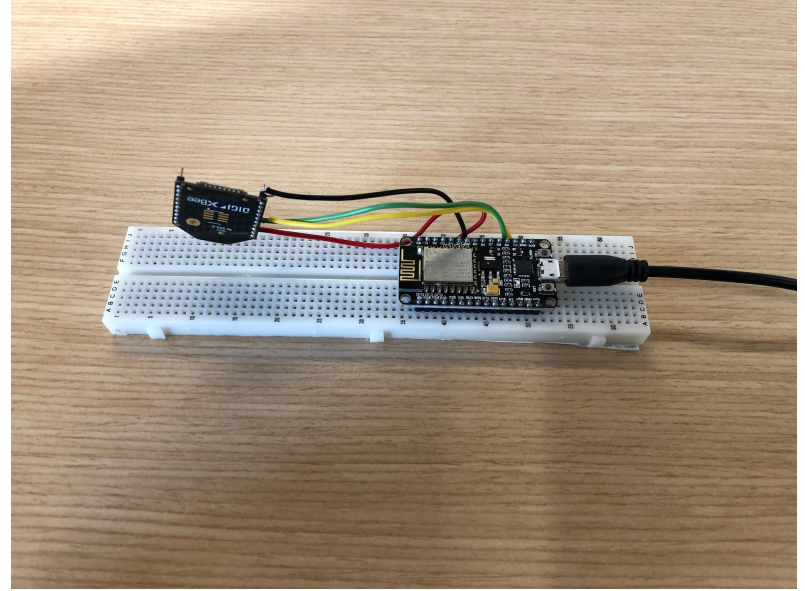
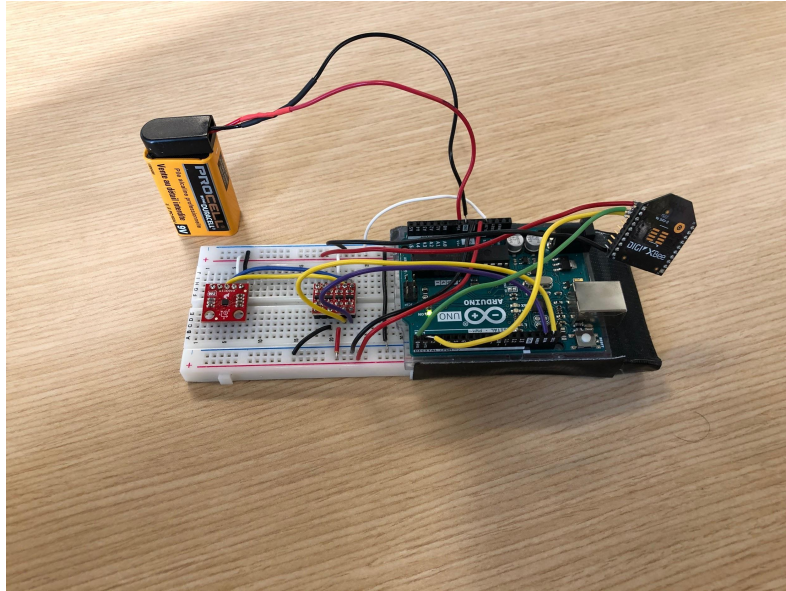
# PDR Block Diagram



# Revised MDR Block Diagram



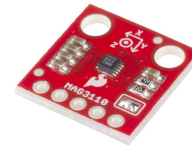
# Sensor Network





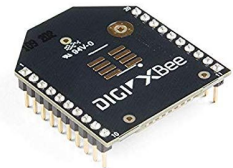
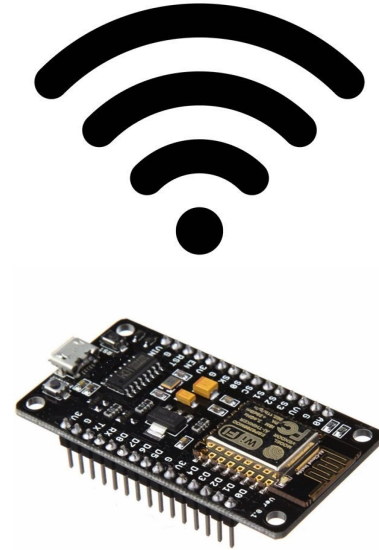
# Sensor Network

- Arduino Uno
  - Awakening and retrieving of data
- MAG3110 magnetometer
  - Senses magnetic field
- Xbee Series 3
  - Sends data once a change is sensed



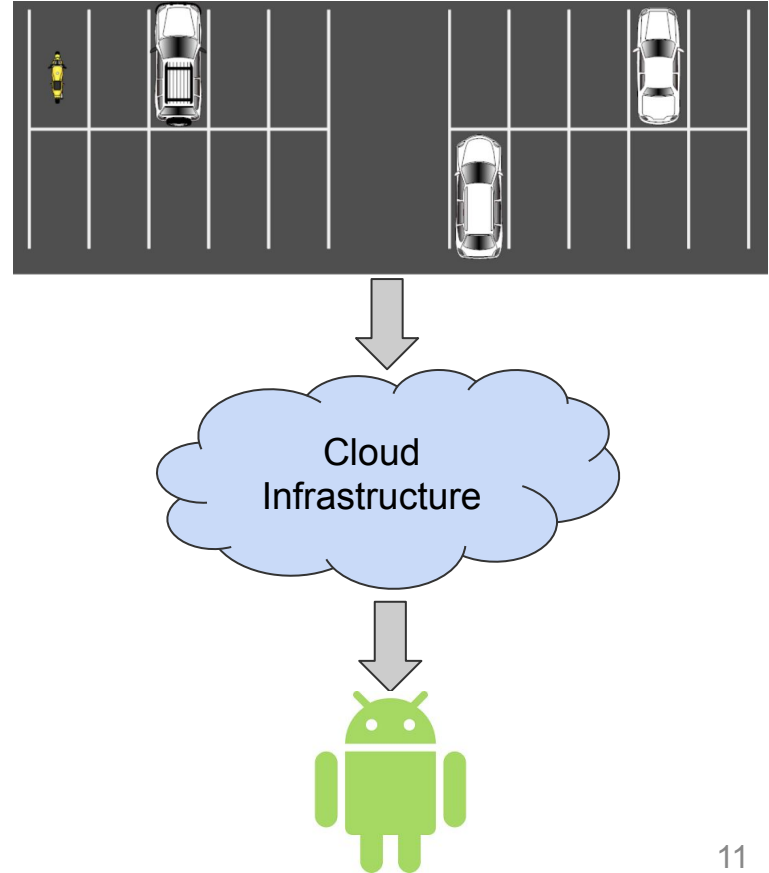
# Gateway

- ESP8266 instead of Raspberry Pi
  - Uses far less power, connects to internet
  - 70mA active as opposed to 250mA
- Xbee Series 3
  - Receives data and transfers to ESP8266
- Serves as relay from sensor to cloud
  - No data processing



# Cloud Infrastructure

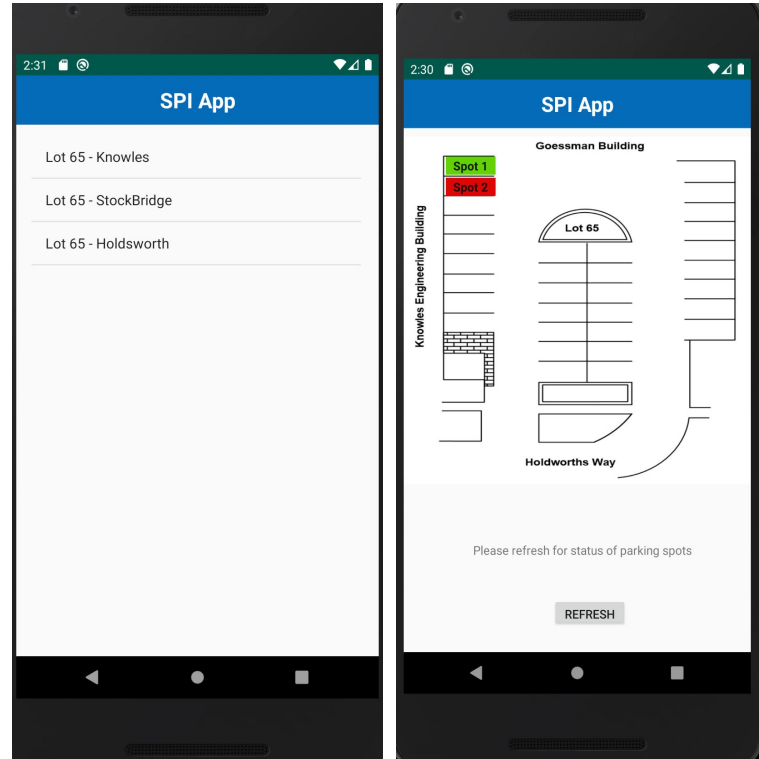
- Responsible for data processing
  - Google Firebase
  - Receives sensor information from Gateway
  - Performs calculations to determine if a spot is taken
  - Updates Database accordingly
- Back-end to android application
  - Send receive/packets
  - Process database requests



# Application

## Basic UI Implemented

- Can see a live map with available parking spaces
  - Green = Available
  - Red = Unavailable



# Interface for Cloud Computing

How does it work?

- Simple connection through App Engine

Advantages:

- Easy to implement
- Reliable



# MDR Deliverable

- Sensor Network

- ✓ We will show the ability to detect vehicle presence with use of a magnetometer at an accuracy rating of 90%
- ✓ We will be able to send XYZ coordinates of the magnetometers readings via Zigbee protocol to the receiver and ESP8266

- Application

- ✓ Show basic map/layout of parking area and show the appropriate status of the parking spot as read by the magnetometer.
- ✓ Graphical User Interface will be in basic stages, showing green as open and red as taken

## MDR Deliverable

- Cloud Computing

- ✓ Receive relayed Sensor Data from ESP8266 over internet
- ✓ Update Database in accordance with received sensor magnetometer reading
  - Includes processing of magnetometer XYZ coordinate reading
  - Update spot status based on processing outcome
- ✓ Push out requested database information to Android Application
  - Includes parking lot status and spot timers.

- ✓ **• Whole system will be functional, meaning we will be able to detect a vehicle's presence, and update the map on a phone application.**

# Sensor Range





# Videos



Cars surrounding sensor



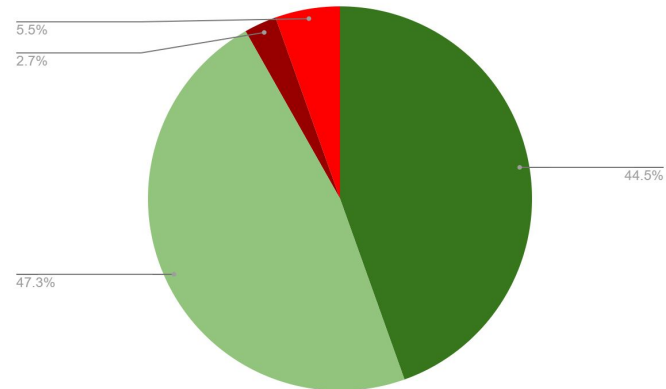
Pulling in



Backing in

# Testing Data - Accuracy of System

- 110 tests were performed
- Able to detect across parking lot **over 200 feet**
  - Things that were varied:
    - time of day, temperature (20°F - 60°F), location of parking spot, types of vehicles driven over the sensor
  - Results:
    - 52/58 with no vehicle present
    - 49/52 with a vehicle present
    - **101/110 = 92% Accuracy**



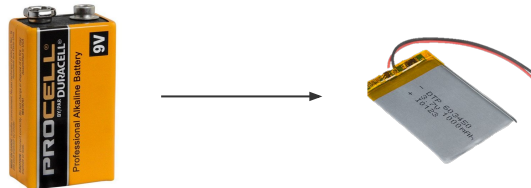
# Battery Data

- 9V - D Cell Battery

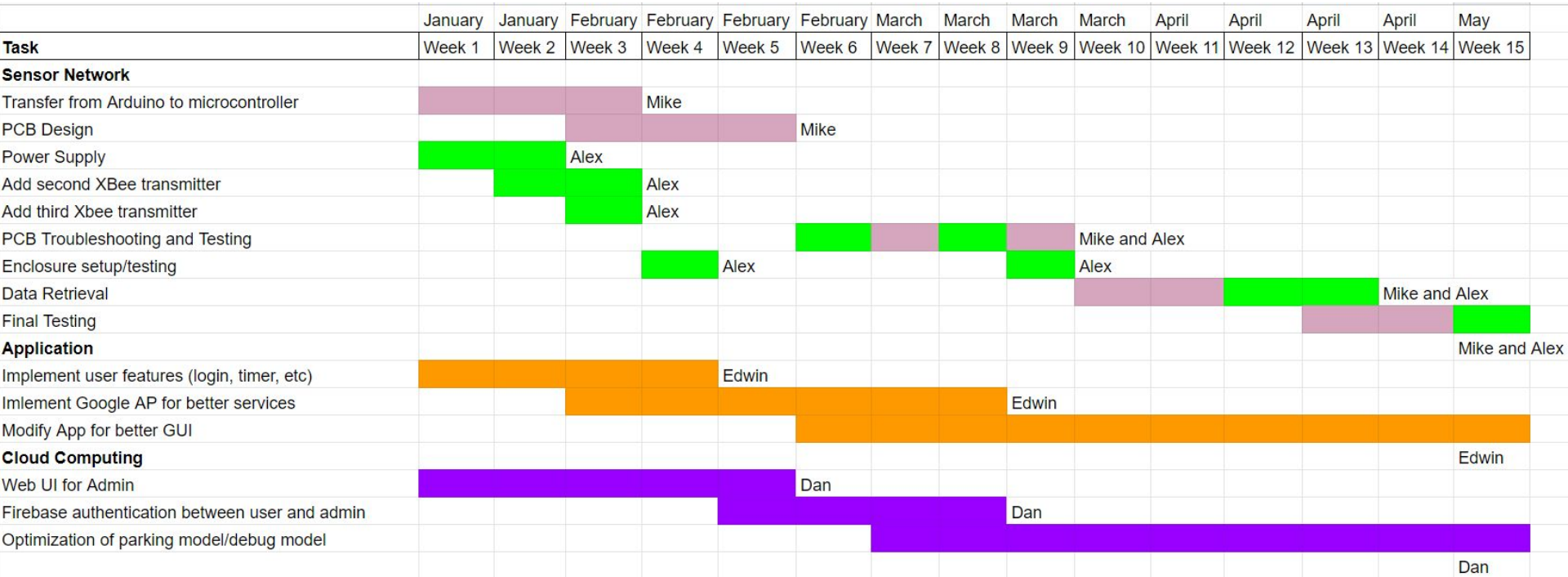
Component	Idle Current mA	Active Current mA
Sensor	.002	.1
Arduino	.05	25
Transceiver	.002	19

Current Arduino Setup Expected Battery Life: ~ 11 days

Battery Life Expected with 2500mA and no Arduino: ~ **7 months**



# Future Team Responsibilities - Gantt Chart



# DEMO