The Smart Security Dash Camera

Group 25

Matthew White
Timothy Deligero
Austin Sturm
Scott Levine
Joseph LaBauve

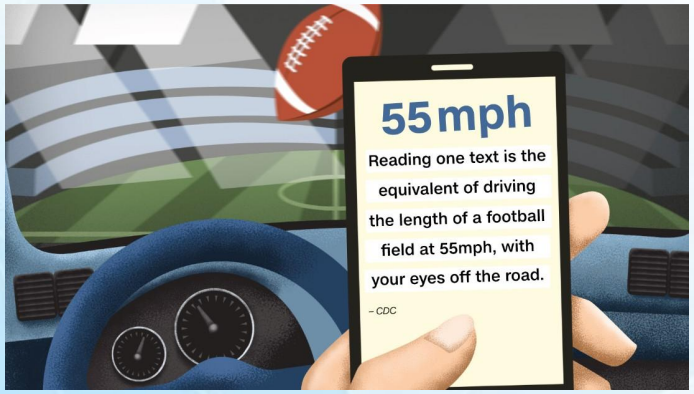
Roles

- Matthew White - Sponsor / Lead Hardware Designer
- Austin Sturm - Lead Software Developer
- Joseph LaBauve - Hardware Designer
- Scott Levine - Embedded Software Design
- Timothy Deligero - Mobile App Development

Introduction

- According to the Department Of Transportation there were 765,484 reported car thefts in the US in 2016
- In March 2017, there was 36 vehicle break-ins on the UCF campus. Costing students and parents hundreds to thousands in damages per vehicle
- What if one car had a Smart Dash Cam?
- The Smart Dash Camera will provide users with piece of mind and instant notification

Goals and Objectives



- Design and implement a Smart Dash Camera to send video data at 1080p 30fps when an incident occurs
- The dash camera will provide a solution to the growing problem of vehicle damages, break ins, texting and driving, vehicle related deaths and overall vehicle thefts
- The dash camera must be calibrated to be able to tell the difference between a true crisis and false positive

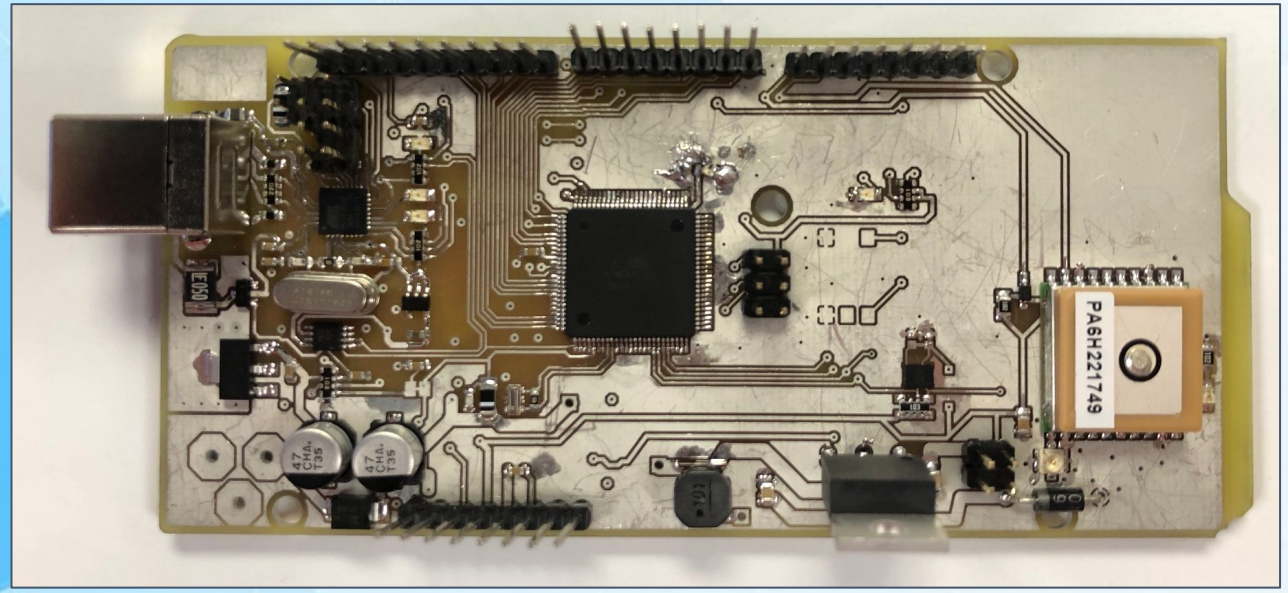
Specifications and Requirements

- **Accelerometer -** Used to detect car movement
- **Cameras** - Two wide angle cameras will be used to view both the front of the vehicle and the inside of the vehicle
- **Android Mobile Application** - The mobile app will be used to receive notifications for the user during cases of car theft or car damage as well as view stored media
- **Data Server -** Remote data server, to maintain statistical history and device management for numerous accounts/devices.
- **GPS Tracker** - Must be able to detect current location of vehicle to assist in locating the vehicle or track stolen vehicles

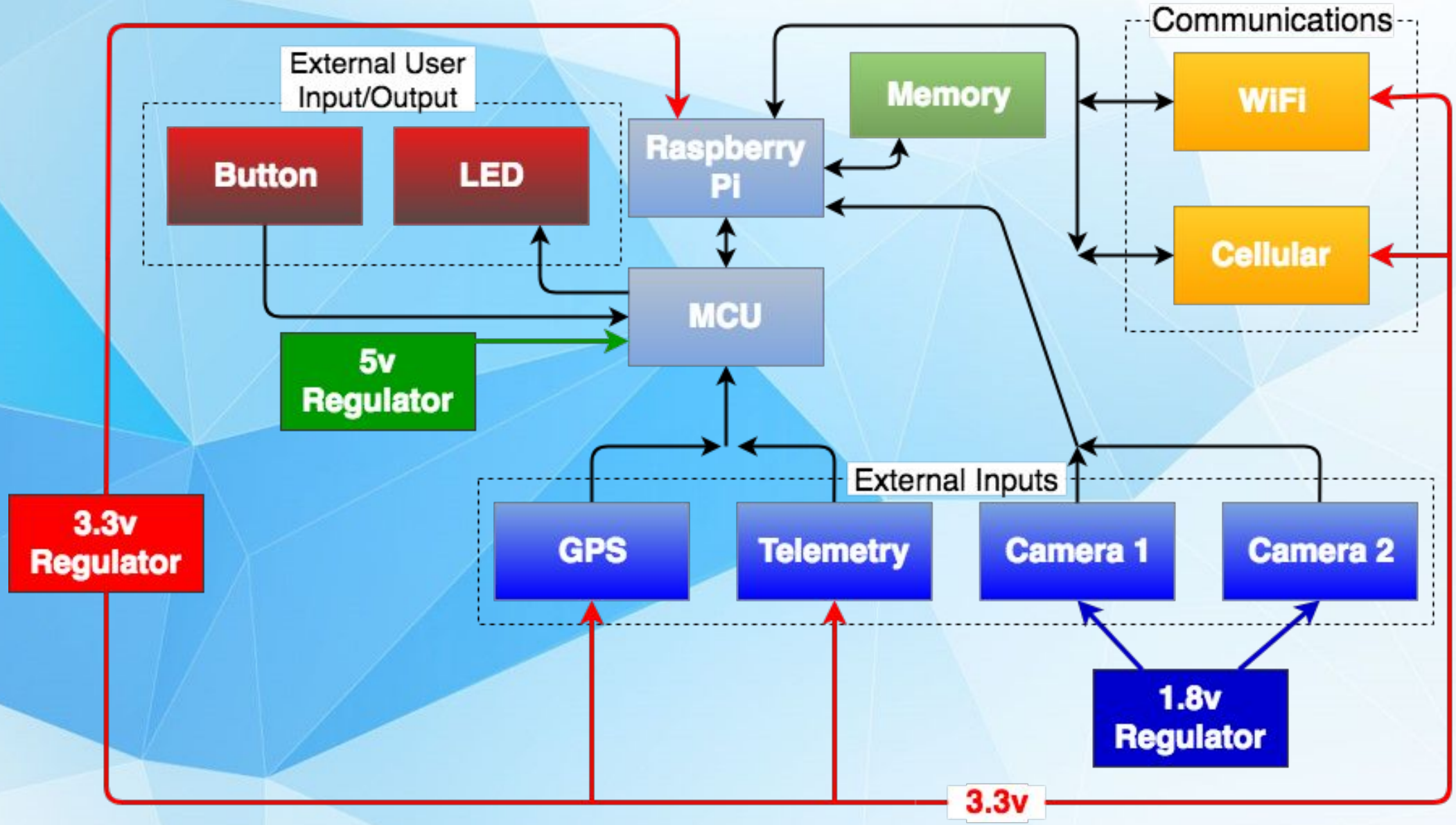
Device Hardware

Parts Selection

- ATmega2560
- LSM6DS3 - Accelerometer
- PA6H221749 - GPS
- Raspberry Pi 3 Model B
- Sierra Wireless Cellular Chip



Hardware Block Diagram



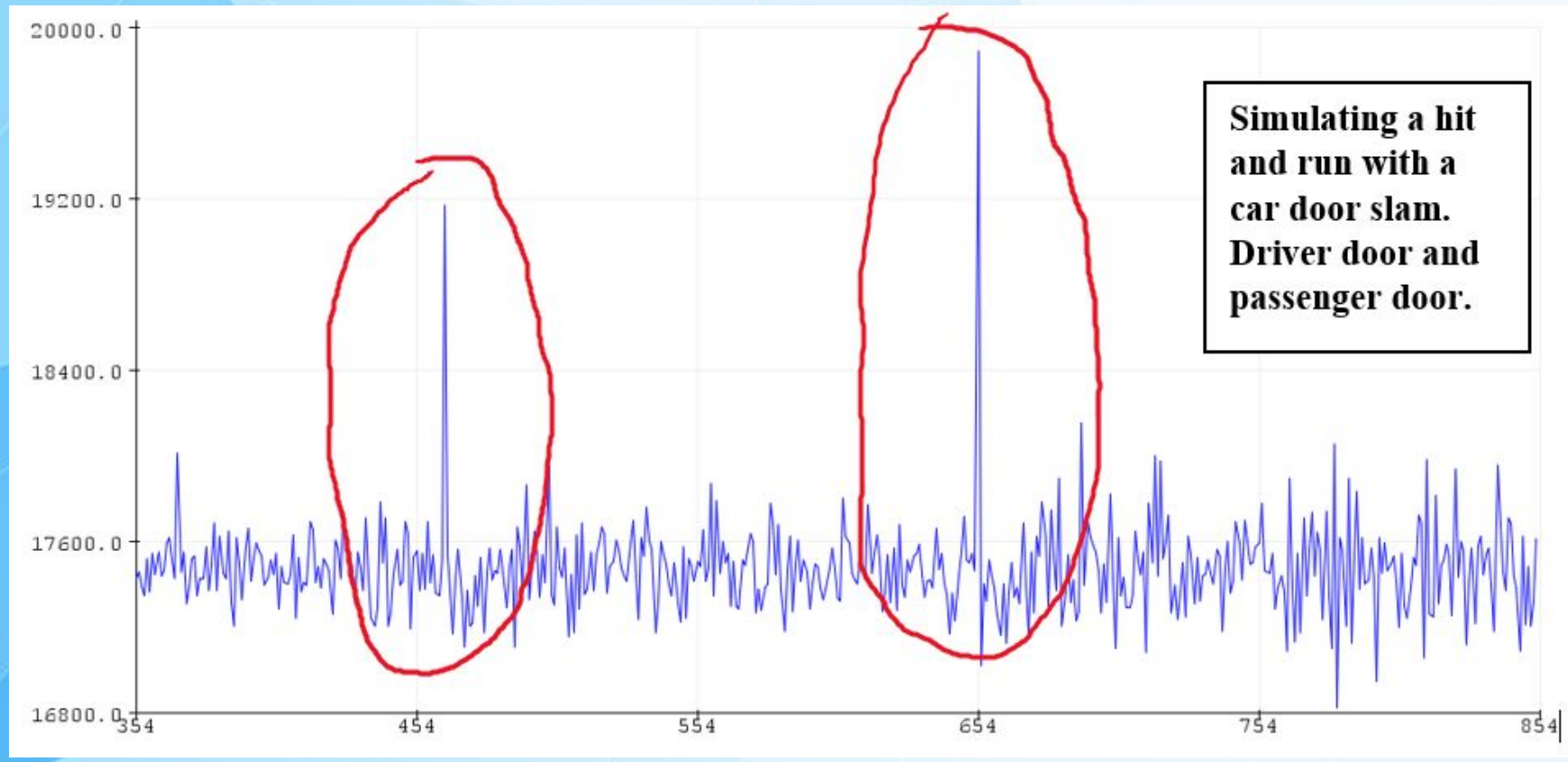
Microcontroller (MCU)

CPU Clock	16 MHz
RAM	8 KB
Program Memory	256 KB
EEPROM	4 KB

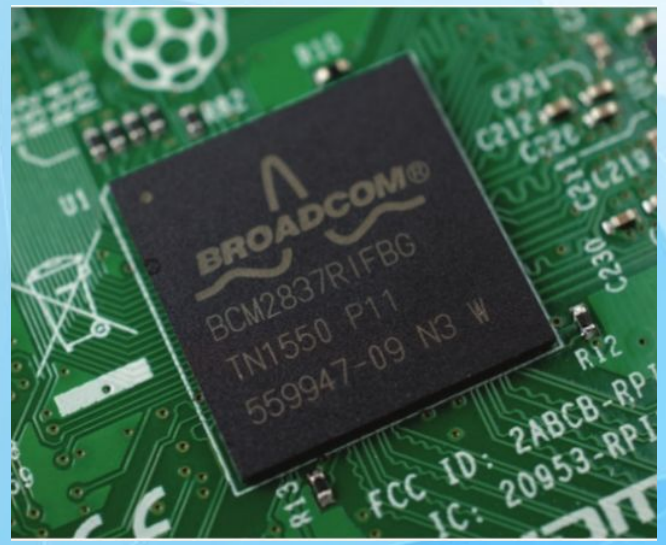
- Communicates with GPS chip and sends coordinate data to data transmit
- Analyzes telemetry sensor data to wake up system.

Accelerometer Test Data

- Parking garage and car crash
- Determine alert threshold
- Alert when data abnormal from threshold



Raspberry Pi



CPU	4xARM Cortex-A53 @ 1.2 GHz
RAM	1 GB
Storage	microSD (16 GB)
Networking	10/100 Ethernet 2.4 GHz 802.11n WiFi

- Video processing/encoding
- Server communication
- Wireless
- Cellular

Hardware

Raspberry Pi 3 - Microcontroller to manage the data of the two cameras and communicates with server

MCU - Processor that controls sensors and interrupts, and sends start/stop commands to cameras through SPI

WiFi - Wirelessly communicate with app

LTE Cell Chip - Allows the user to use the device in the US or in other countries. Delivers data only

Camera

Requirements: 1080p 2.0MP, 60FPS, close to a 180 degree view, H.264 compression

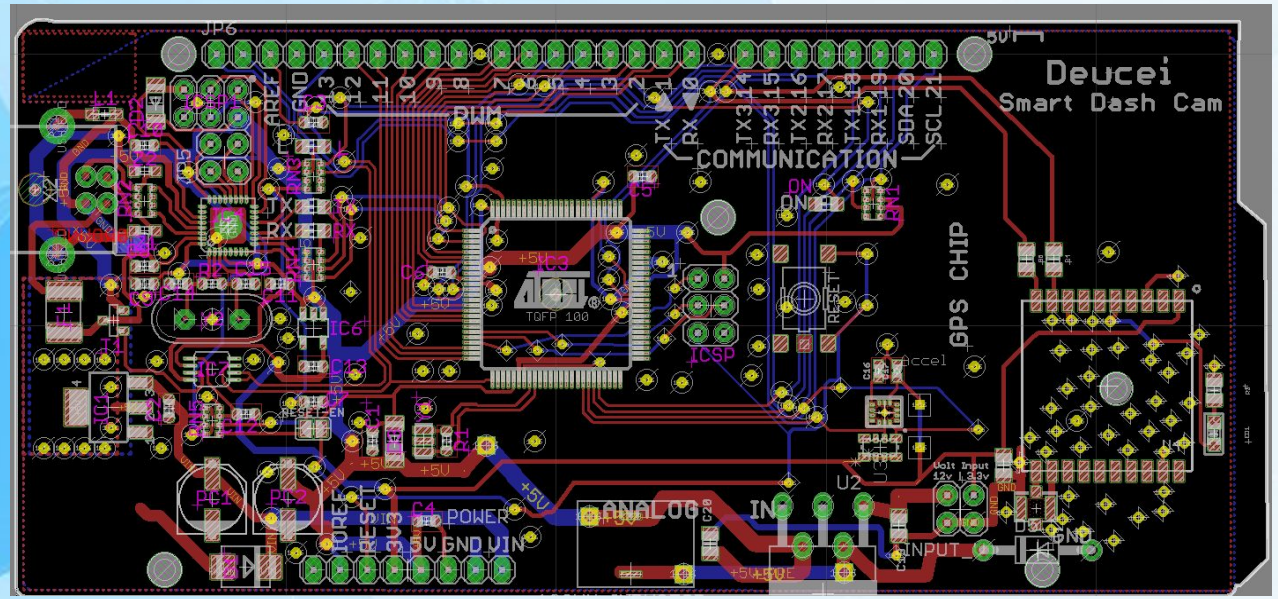
Calculation of Mbps (Megabits per second) for 1 camera:

compression ratio * #cameras * Height of Image * Width of Image * FPS * Motion Rank

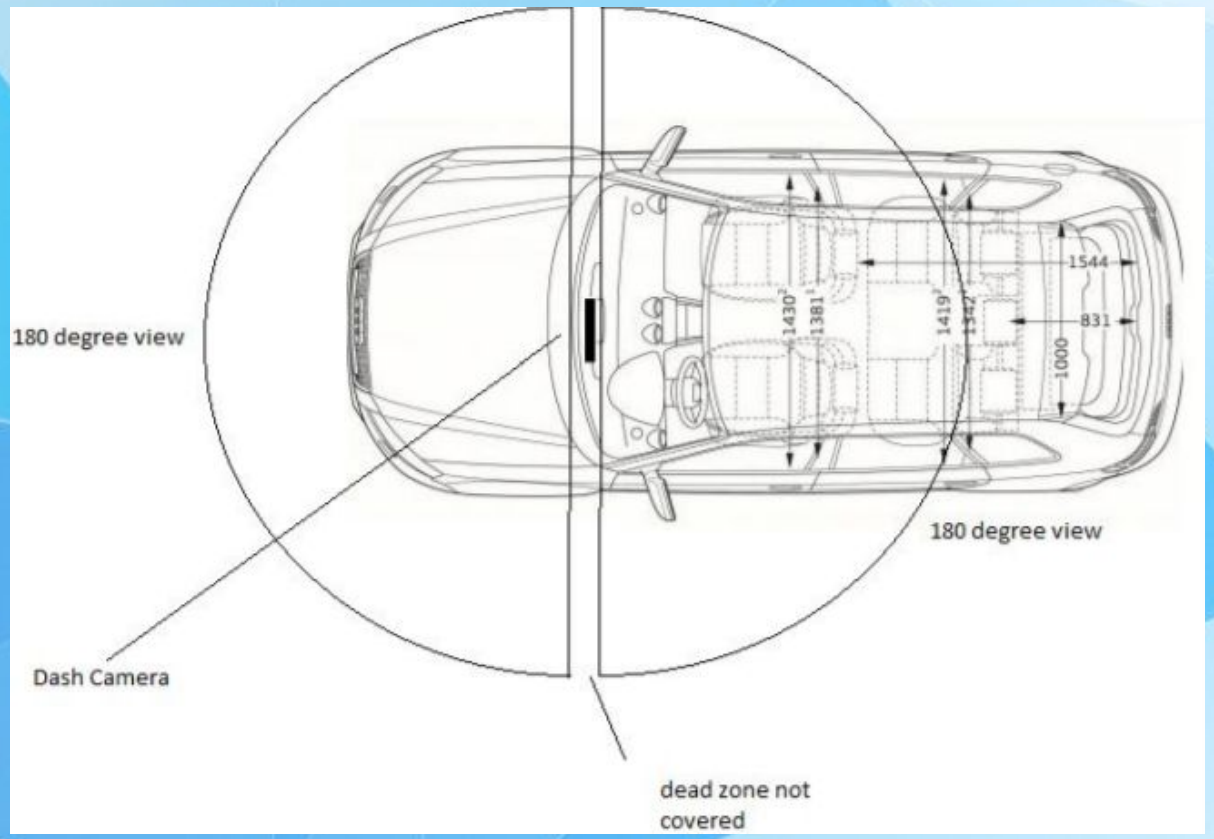
= 17.4Mbps

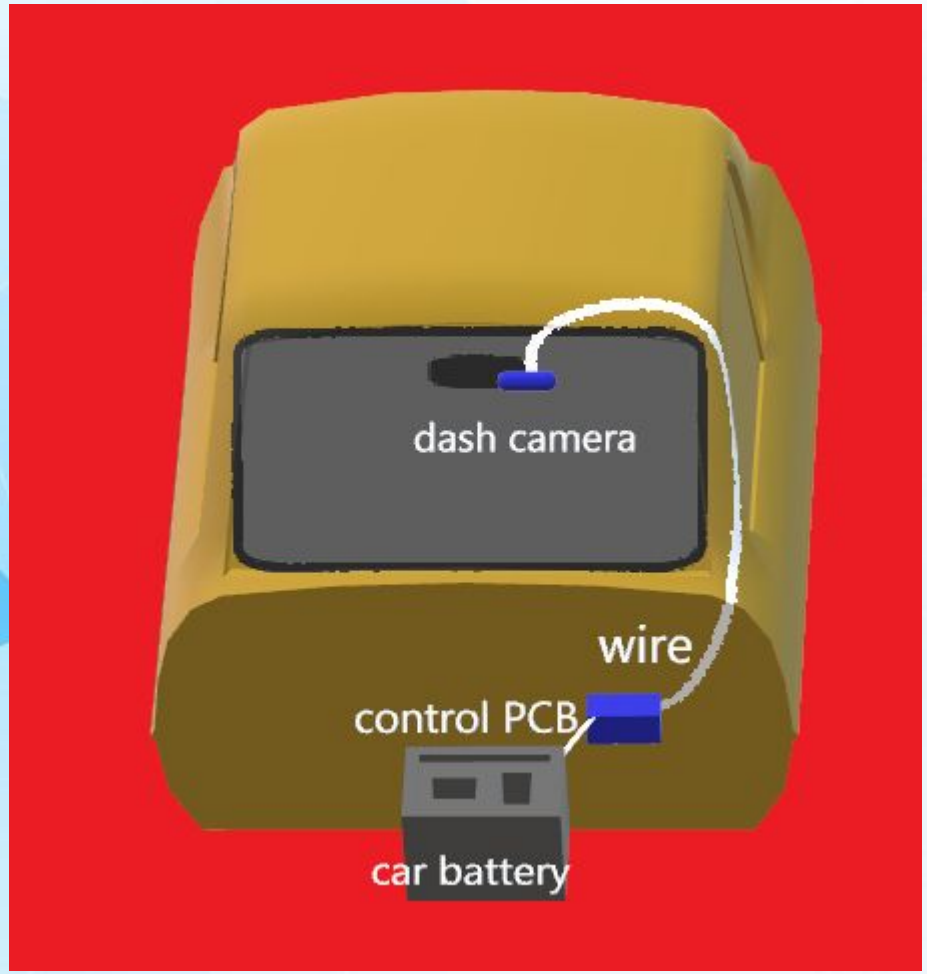
PCB Design

- Atmega2560 MCU
- FGPMMOPA6H stand alone GPS chip SPI
- LSM6DS3 (X,Y,Z) axis I2C accelerometer chip
- 12v to 5v 3A Step-Down Voltage Regulator



Design Diagrams





PCB and Camera Enclosures

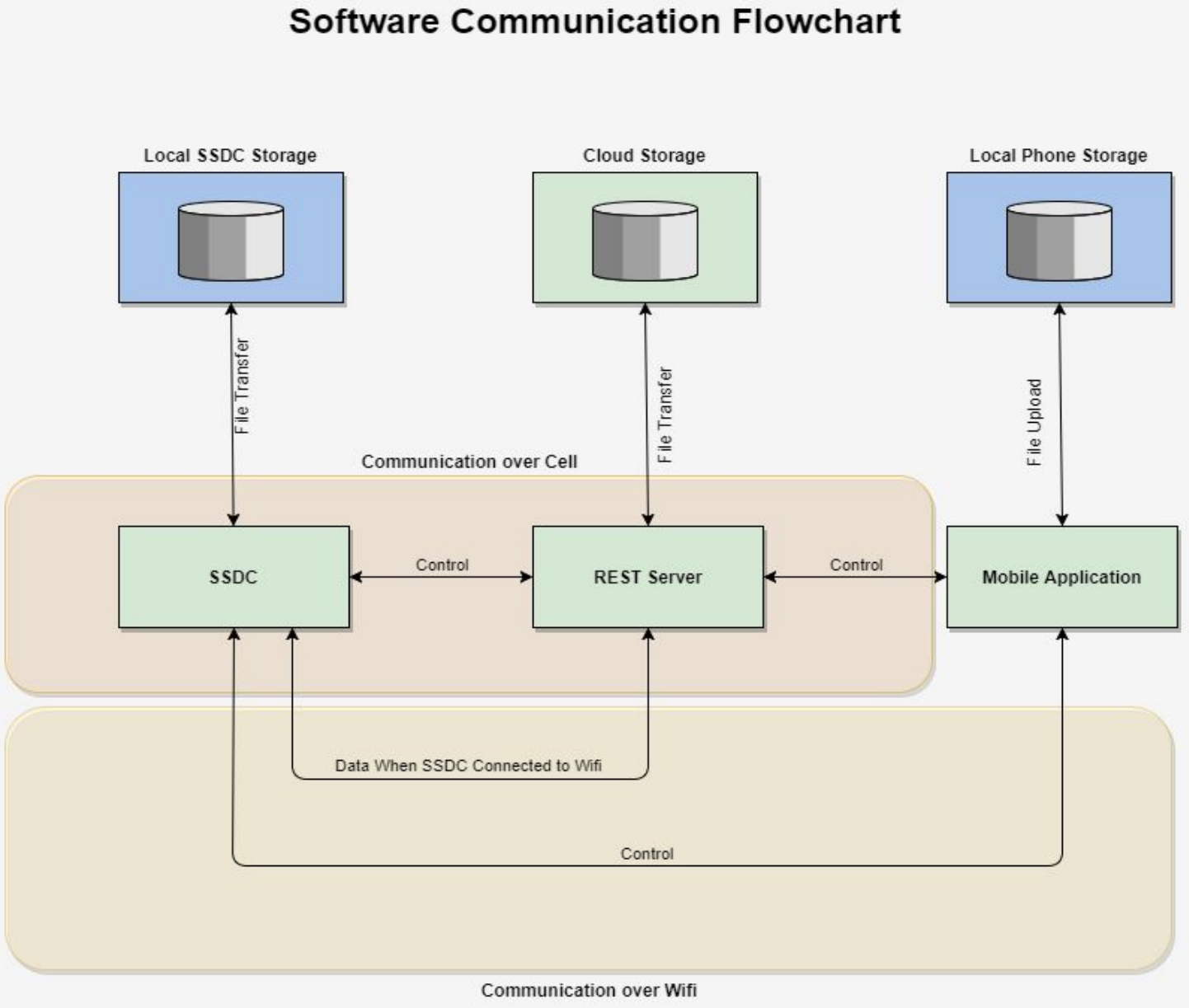
Main Case

Camera Case

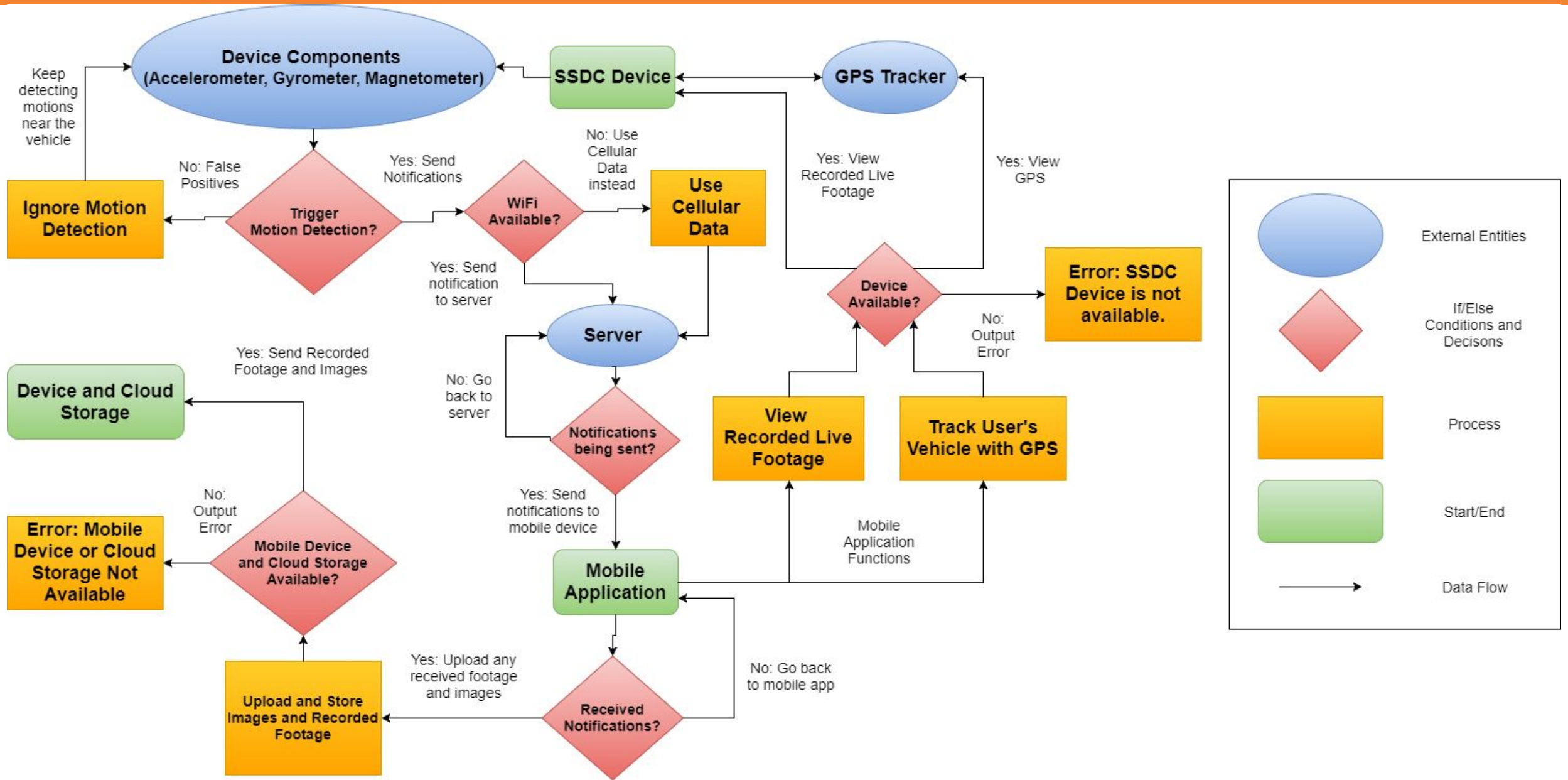
Server

Software Communication

- Django Rest Framework
- AWS Hosting
- Dev Environment: Local Server
- Python 3.5
- Java Mobile App

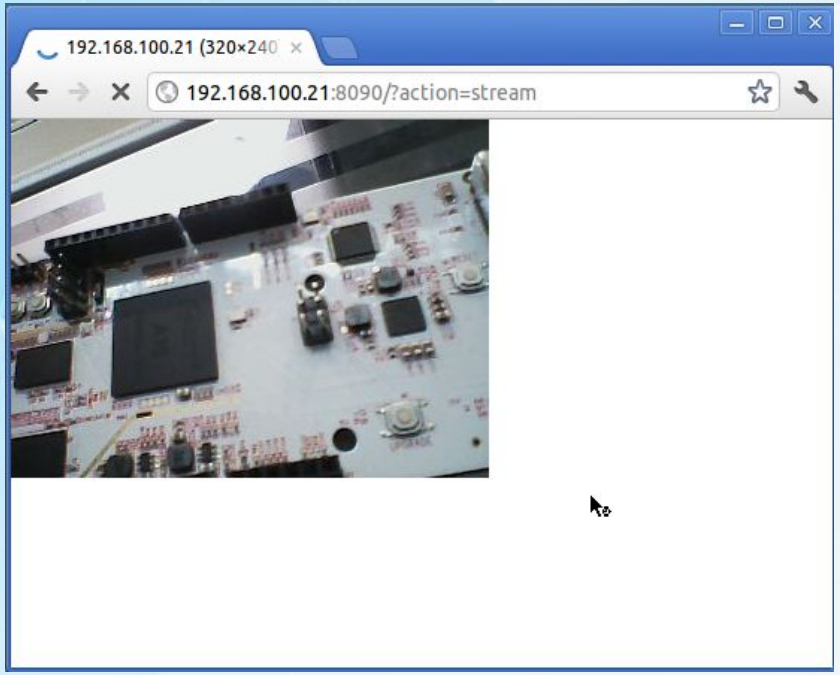


Data Flowchart



Video Streaming

- Local Streaming:
  - mjpegStreamer
  - Faster streaming, local web service
- Remote Streaming
  - RTMP through ffmpeg library
  - Nginx RTMP relay server



Security

- TLS enabled on all endpoints
- Encrypted video transfer
- Hashed database credentials
- Token Based Authentication
- WPA2/PSK for access point

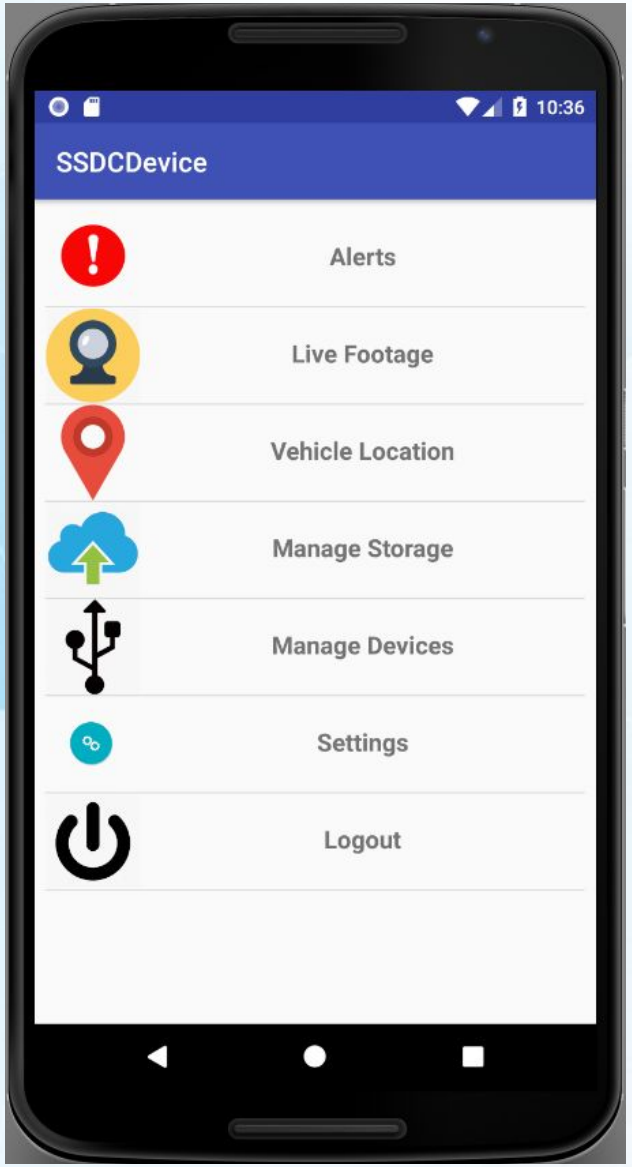
Mobile Application

Android Studio

- The mobile application will be made using the system software Android Studio
- The software will be used to create the app with multiple activities/Java classes and xml files
- An emulator will be used to test and debug the mobile application

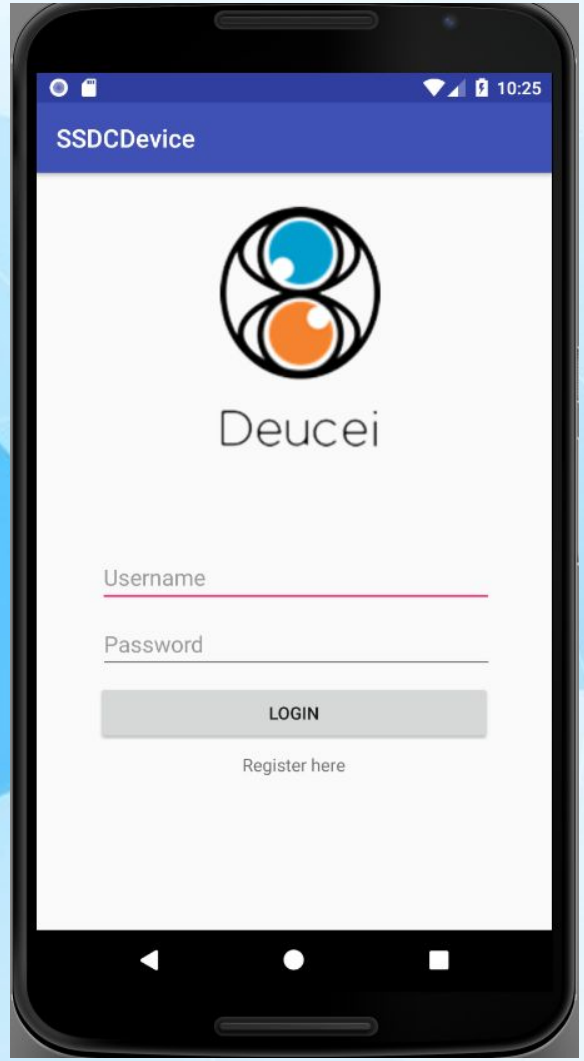
Mobile App Features

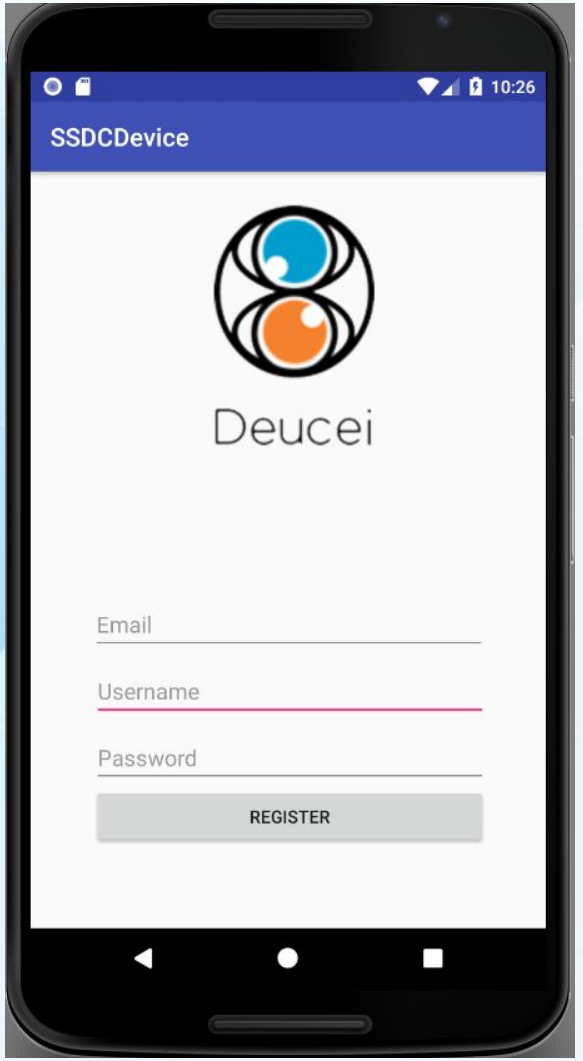
- Login/Register
- Alerts
- View Live Feed
- GPS (Vehicle Location)
- Manage Devices/Networks
- Settings (Edit User Information)



Login/Register GUI

- Login - User logins to a previously made account with a username and password
- Register - User creates a new account with email, username, and password

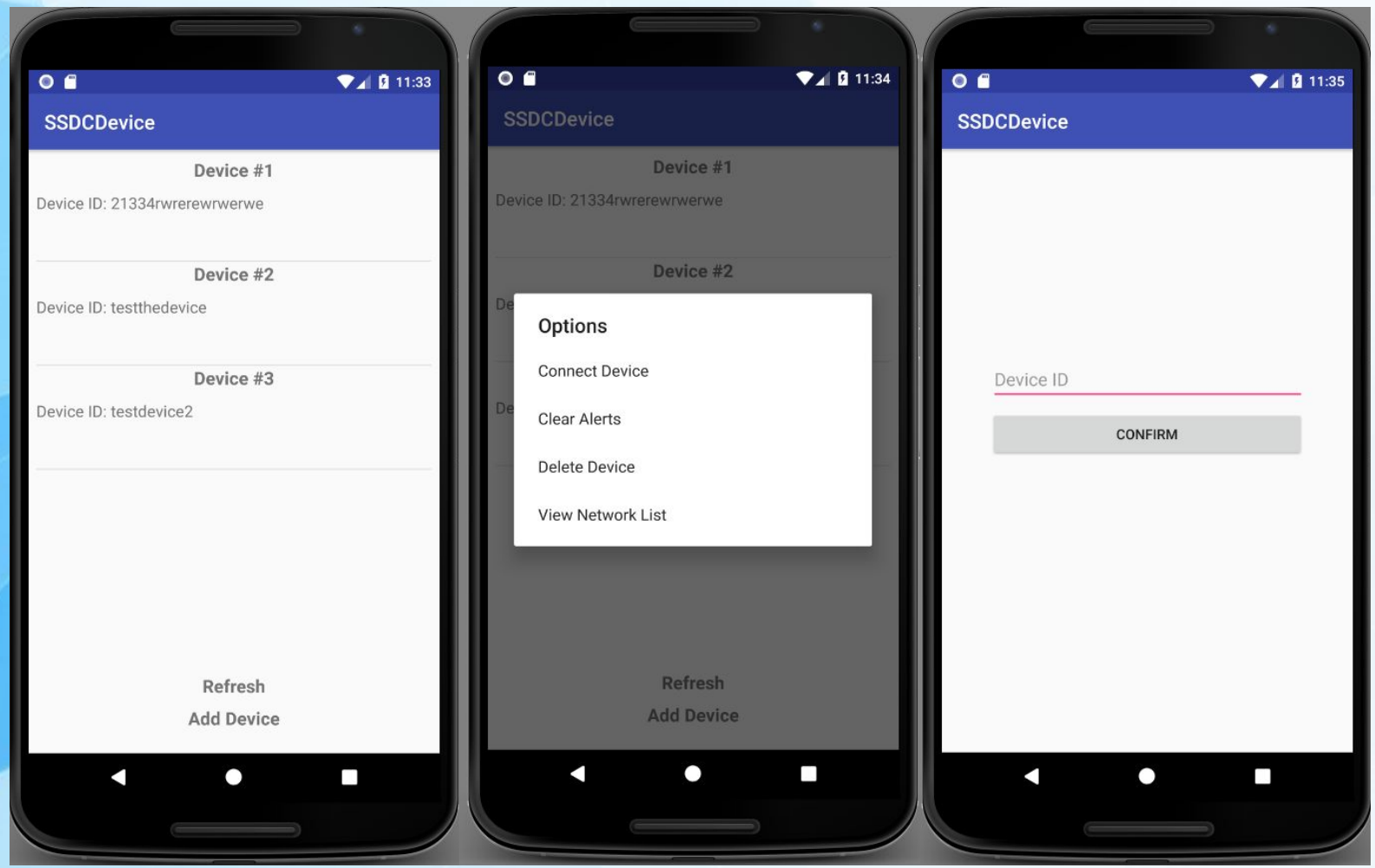




Manage Devices/Networks GUI

Device Options:

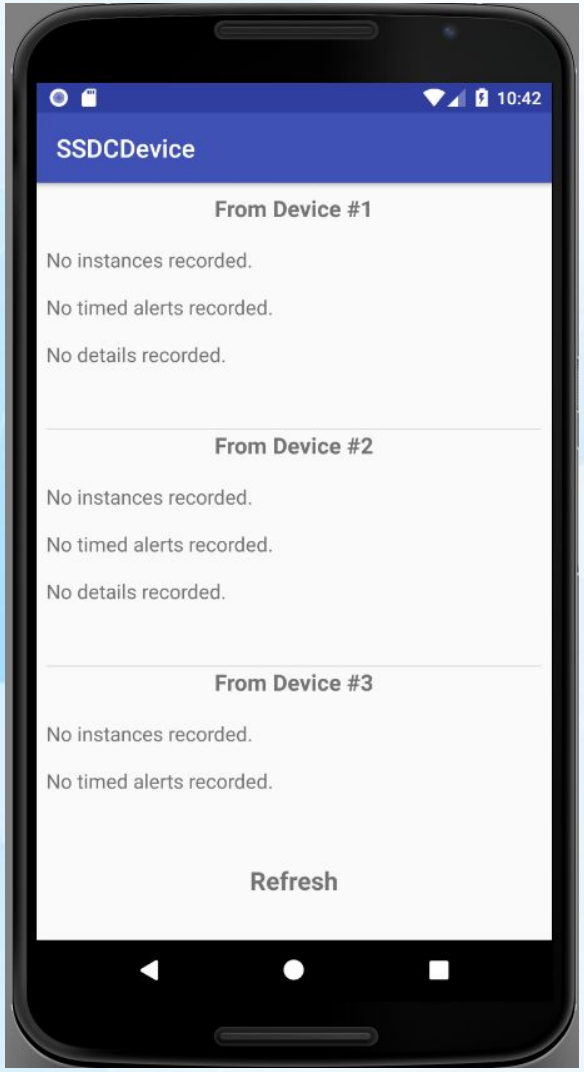
- Add Device
- Connect Device
- Clear Alerts
- Delete Device
- Networks



Alerts GUI

Alerts List Information

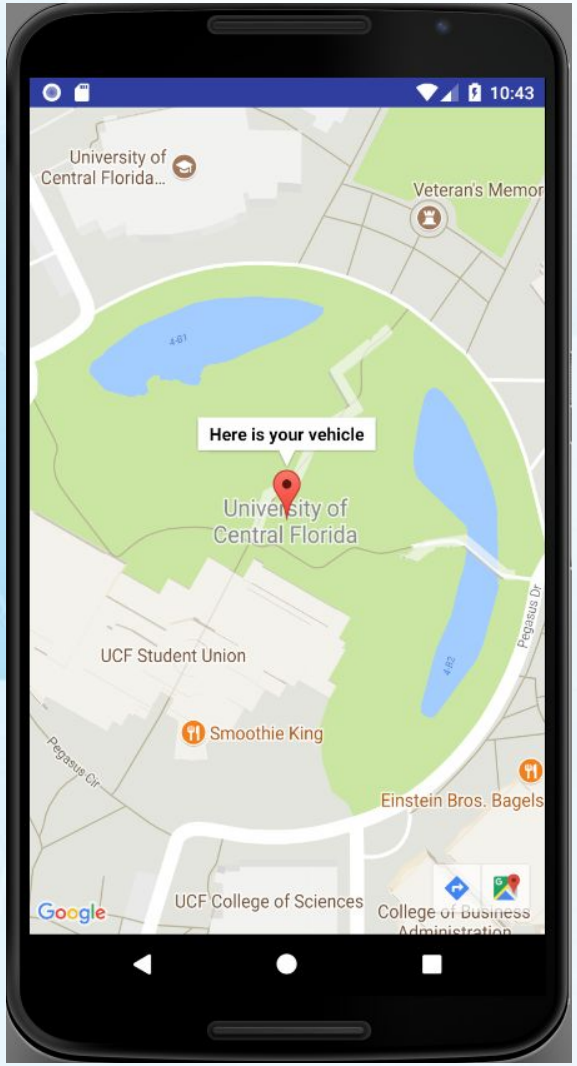
- Device
- Alert Title
- Alert Time
- Alert Details
- Send Push Notifications



GPS GUI

GPS Location

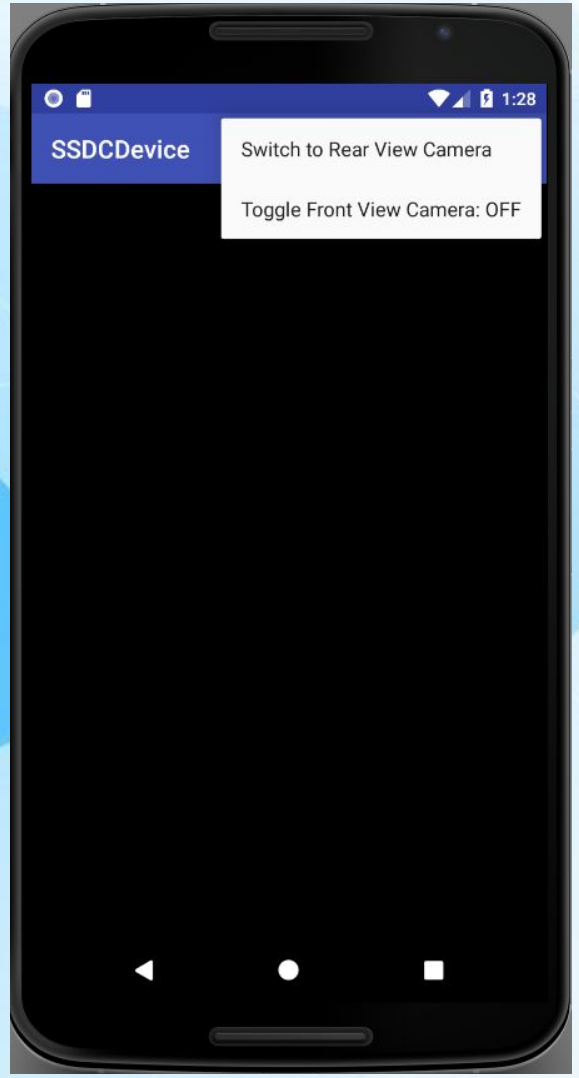
- Google Maps View
- Marker of vehicle location
- Timer - Update location of vehicle periodically



Live Feed GUI

Live Feed Options

- Switch between camera views
- Toggle the cameras ON/OFF
- View Live Feed when cameras are ON



Administrative Content

Design Difficulties

- Adjusting device triggers to account for false positives in parking garages, subway stations, train stations
- Correct selection of parts that can properly communicate with one another
- Size of unit and amount of data being transferred
- Finding quality parts

The Pivot

Sony Optical Sensor

DM388 Dev board

Budget and Financing

Development Costs			
Item	**Quantity**	**Cost/Part**	**Total Cost**
Atmel MCU Chip	3	$12.71	$38.13
TI Video Processor Dev board & Cam	1	N/A	$412.00
Arduino	1	N/A	$44.99
Rental of Sony Sensors	1	N/A	$150.00
PCB	8	$33.00	$372.00
Digikey Components	3 sets of parts	$45.00	$155.00
Sierra Wireless Board	1	$160.00	$160.00
Raspberry Pi 3	1	$45.00	$45.00
Cameras	2	$50.00	$100.00
Sierra Wireless LTE modem	1	$70.00	$70.00
Total	**22**	**$415.71**	**$1,547.12**

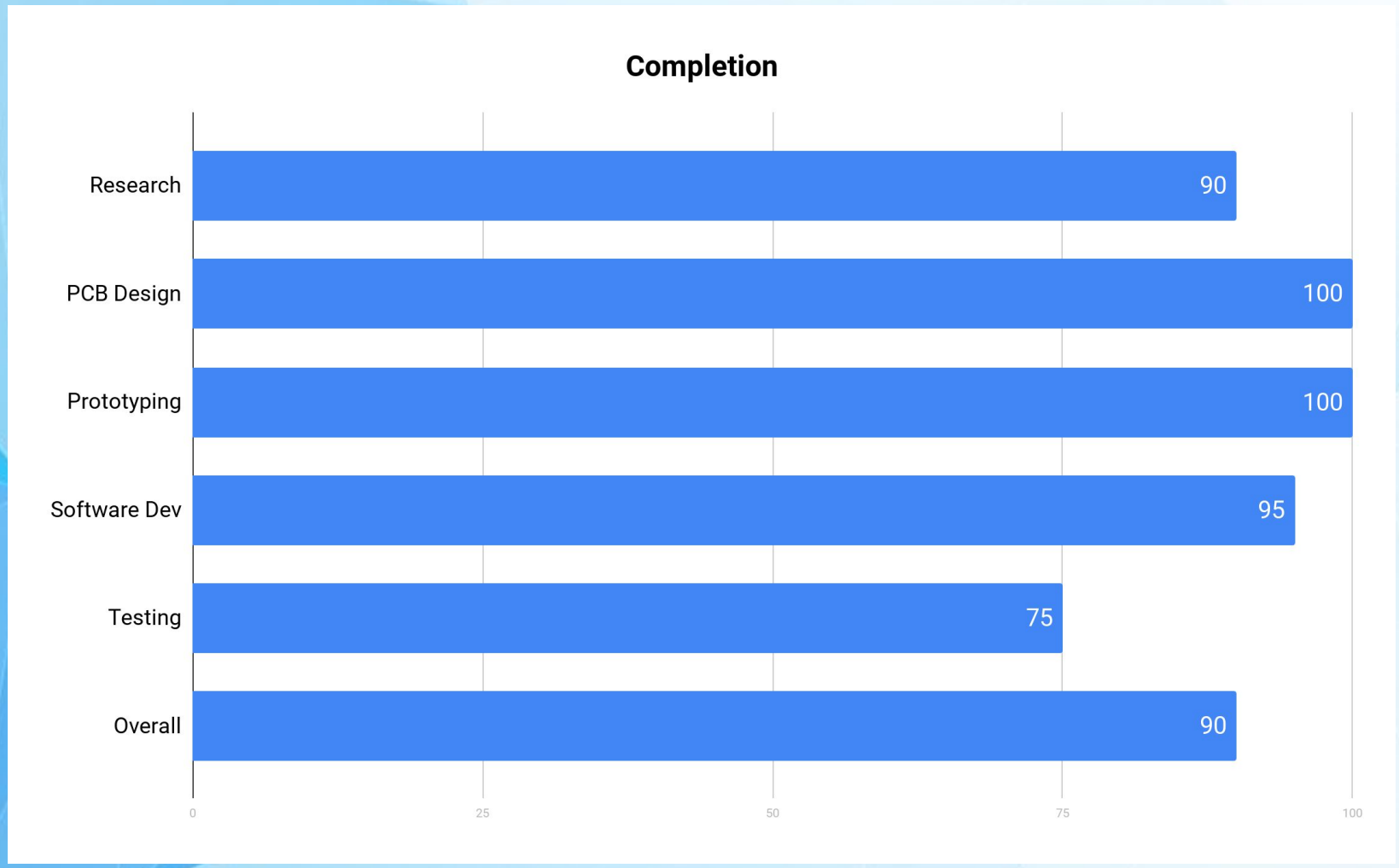
Budget and Financing

Large Scale production (10,000 units):

Retail Price: $349.99

Costs: $59.99

Current Progress



Questions?

We’ve got answers.