The Smart Security Dash Camera

Group 25

Matthew White
Timothy Deligero
Austin Sturm
Scott Levine
Joseph LaBauve

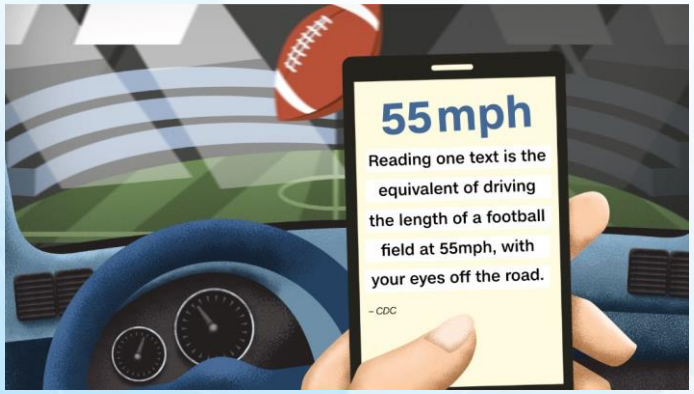
Roles

- 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

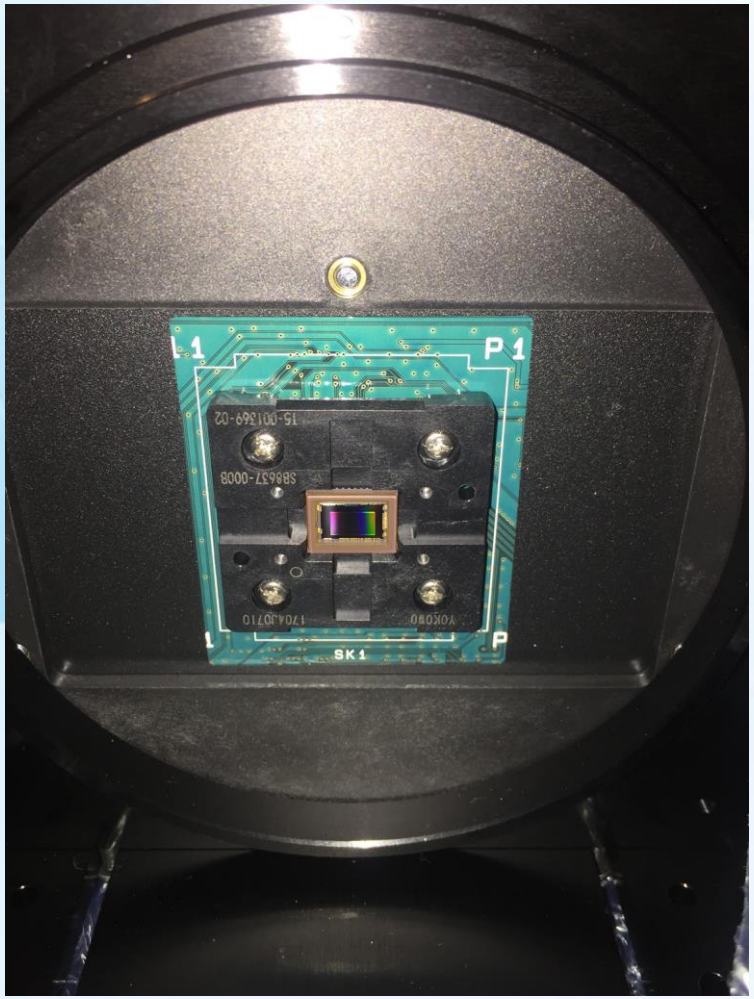
- According to the Department Of Transportation there were 765,484 reported car thefts in the US in 2016
- In March 2017 there was 36 vehicle break-ins on the UCF campus. Costing students and parents hundreds to thousands in damages per vehicle
- What if one car had a Smart Dash Cam?
- The Smart Dash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

Goals and Objectives

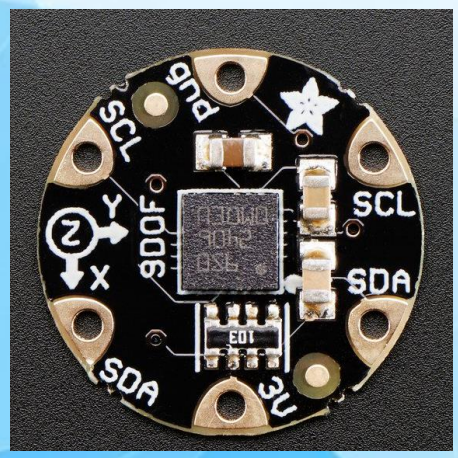


- Design and implement a Smart Dash Camera to send video data at 1080p 60fps when an incident occurs
- The dash camera will provide a solution to the growing problem of vehicle damages, break ins, texting and driving, vehicle related deaths and overall vehicle thefts
- The dash camera must be calibrated to be able to tell the difference between a true crisis and false positive

Optical Sensor Testing



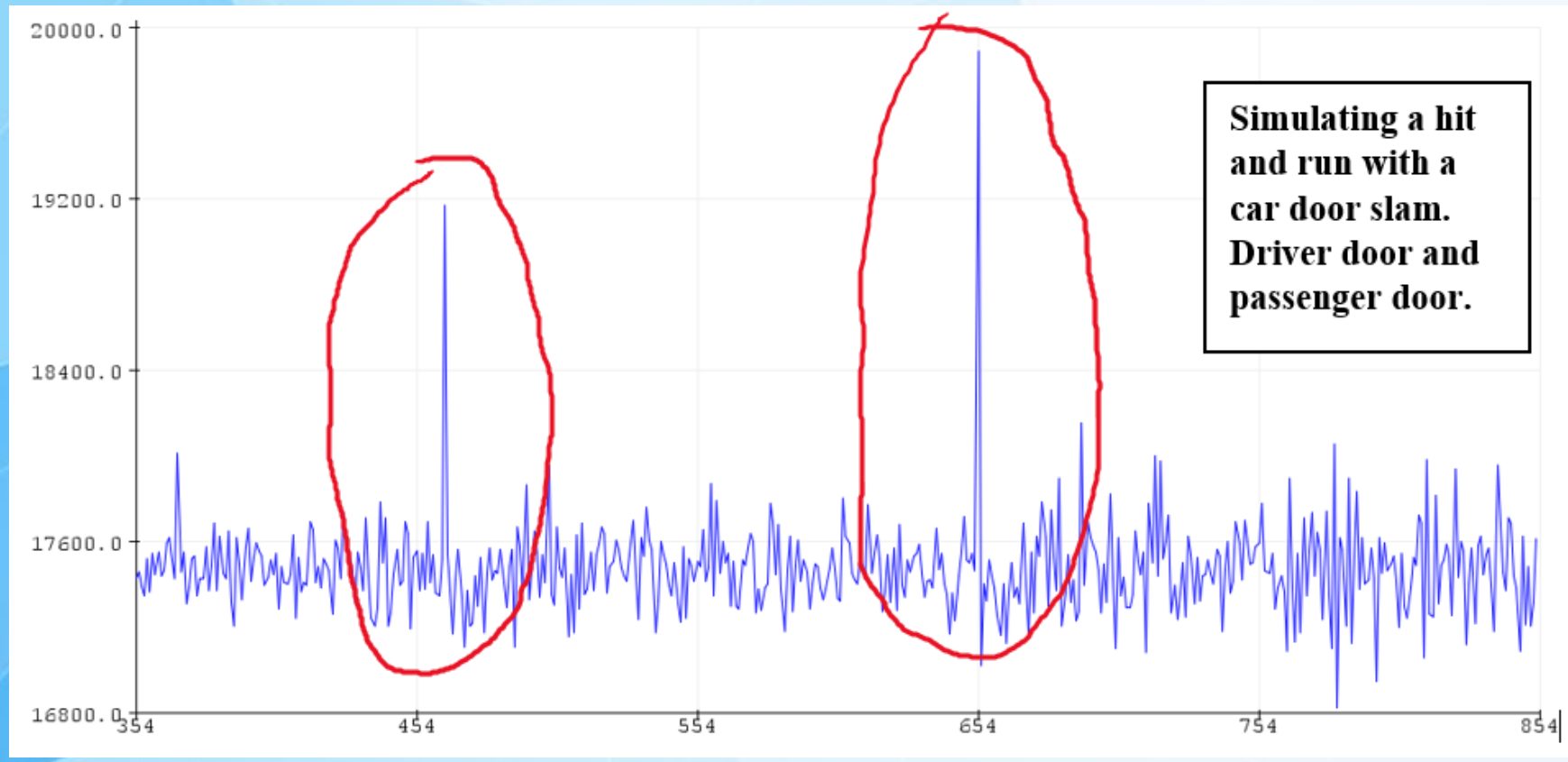
Specifications and Requirements



- **Accelerometer -** Used to detect car movement when the car is off
- **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 images and recorded footage
- **GPS Tracker** - Must be able to detect current location of vehicle to assist in location of vehicle or track stolen vehicles

Accelerometer Test Data

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



Device Hardware

Parts Selection

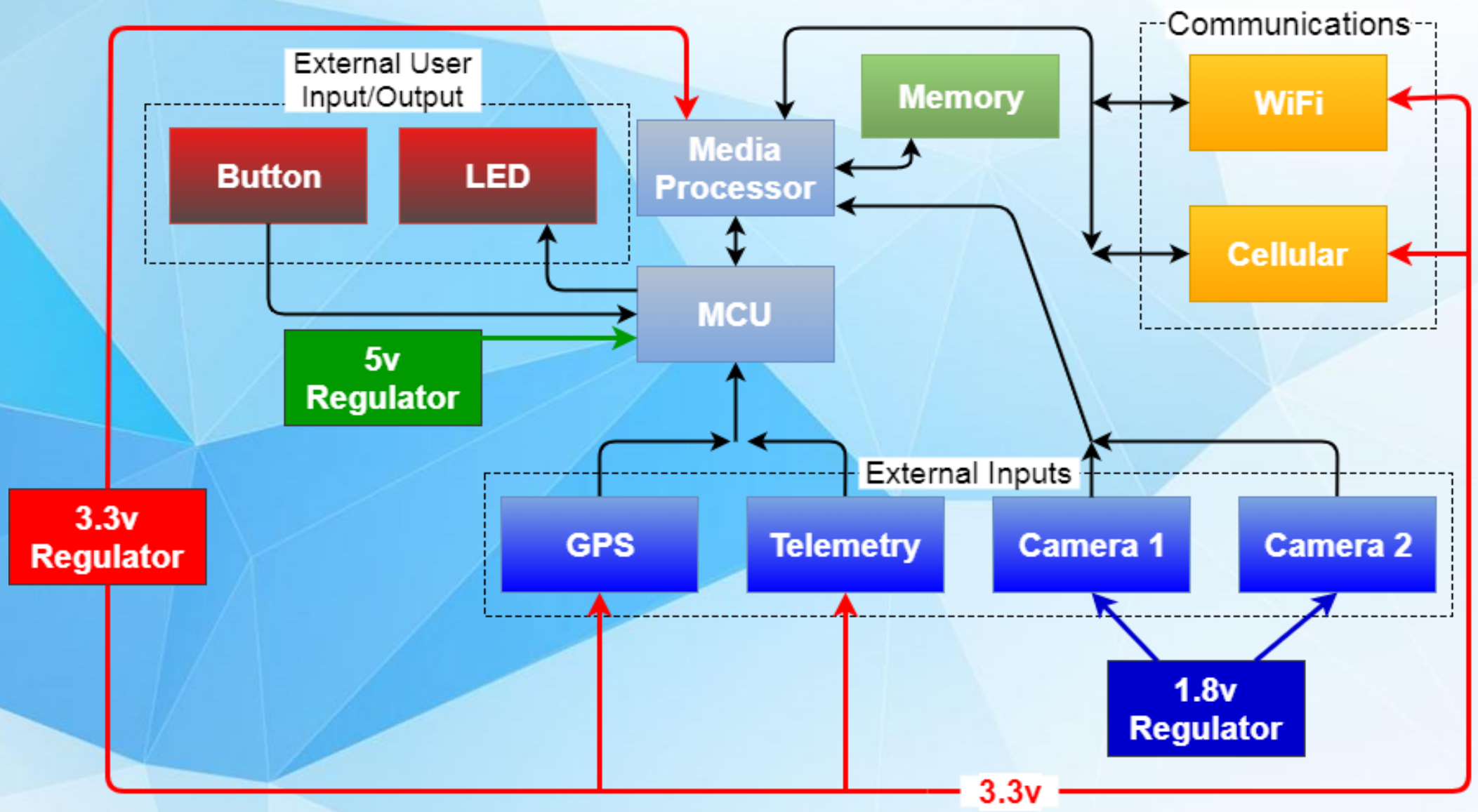
Texas Instruments Video Processor

Atmel MCU

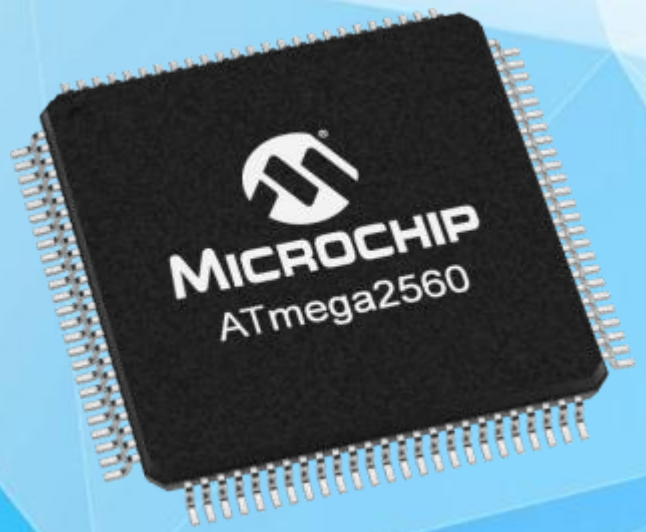
Texas Instruments WiFi Module

Sierra Wireless Cellular Chip

Hardware Block Diagram



Microcontroller (MCU)

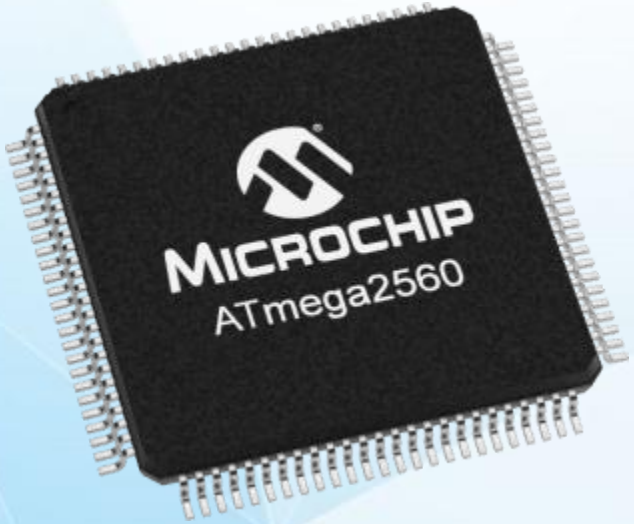


CPU Clock	16 MHz
RAM	8 KB
Program Memory	256 KB
EEPROM	4 KB

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

Media Processing Unit (MPU)

- Low power mode
- Video processing/encoding
- Server communication
- Wireless
- Cellular
- TI-RTOS



Schematics

Digital media processor- media processor to process the data of the two cameras using two 6GB DDR RAM chips.

MCU- processor that controls interrupts, and sends stops commands to cameras

WiFi- talks through uart to wirelessly communicate with app

CDMA Cell Chip- cheaper than GSM, lots of info readily available

Camera

Requirements: 1080p 3Mp, 60FPS, near 180degree view, h.246

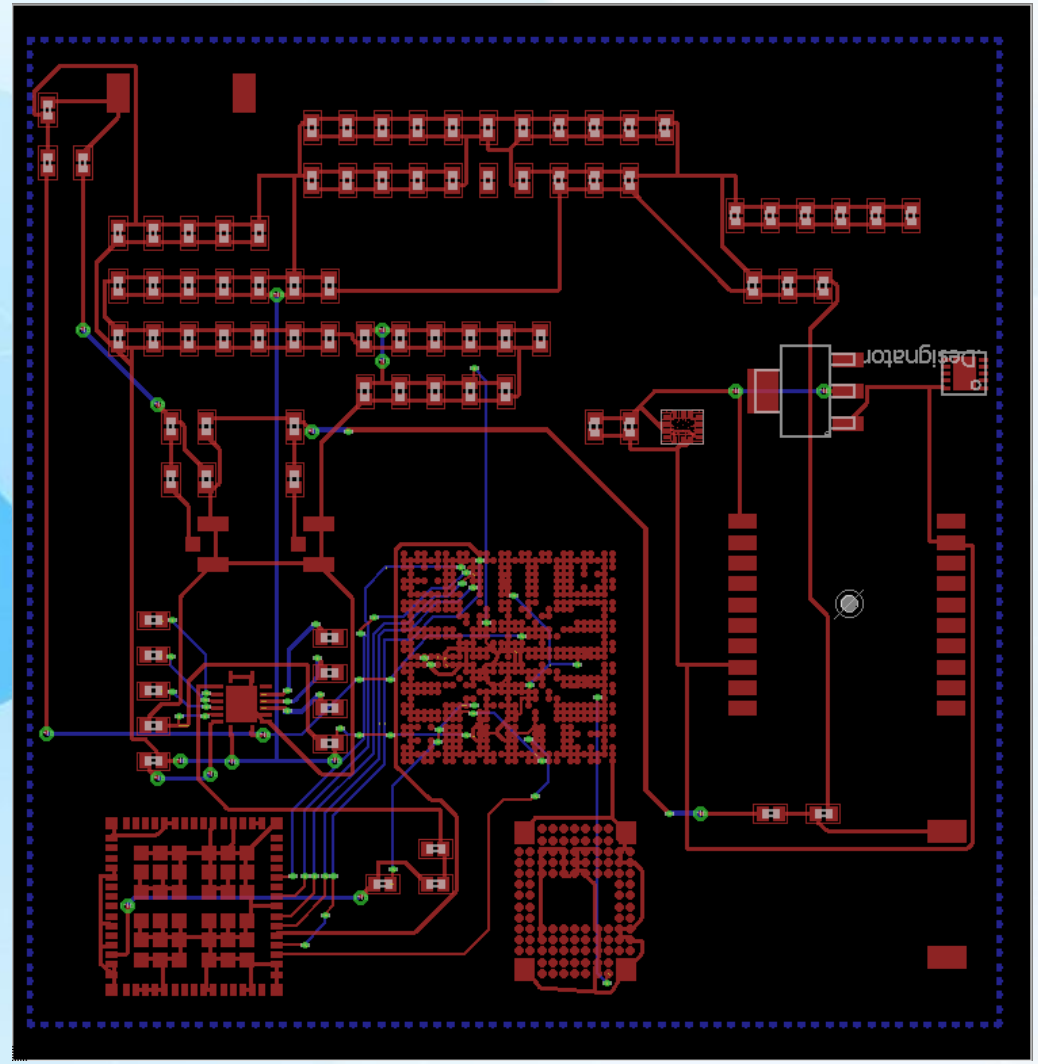
calculation of Mbps:

C(compression variable) x **#(cameras)** x **Mp(Megapixels)** x FPS

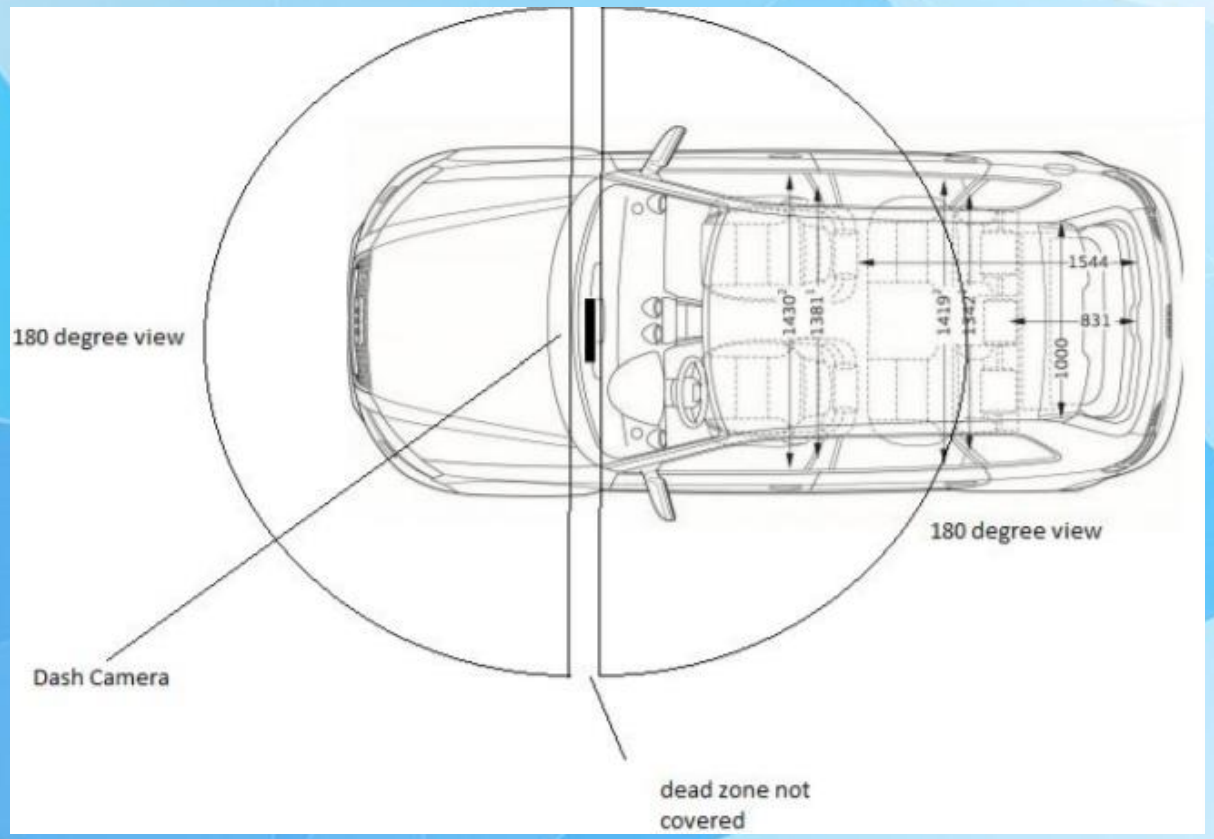
= 90Mbps

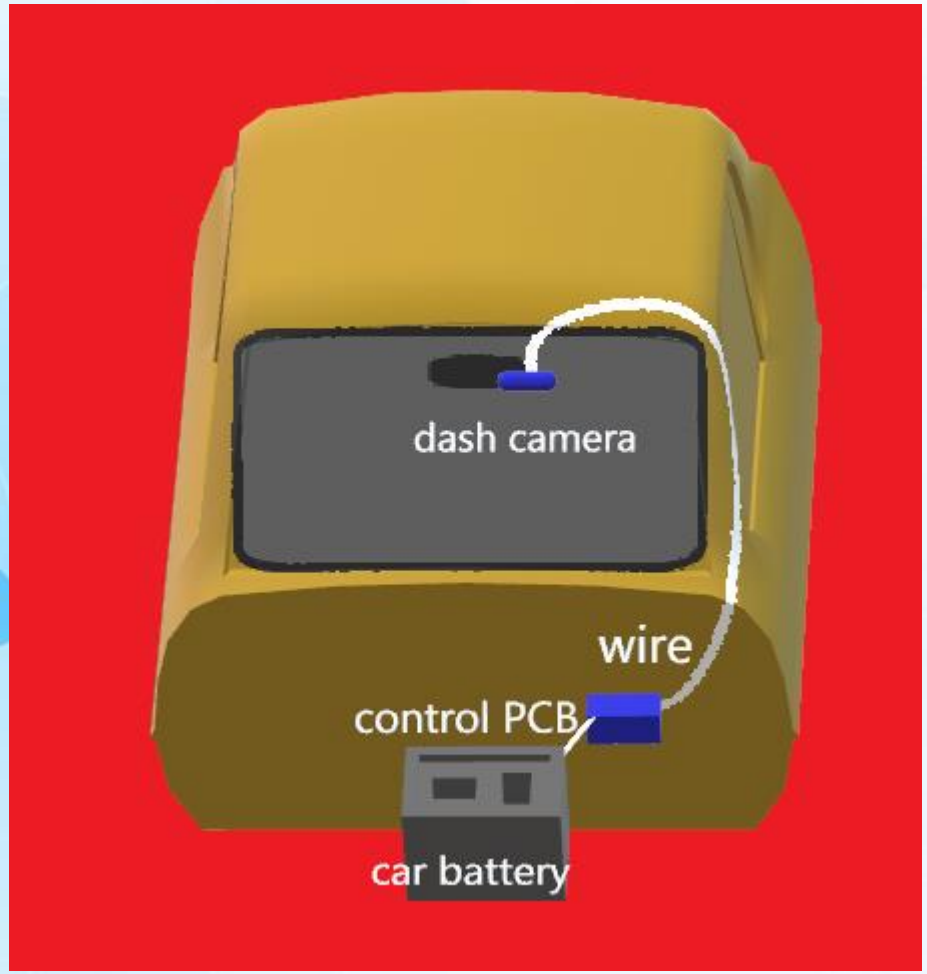
PCB Design

- PCB design is very intensive
- PCB design includes a Cat 4 modem connection, 80-100 Mbps WiFi chip, and TI MCU, GPS, and accelerometer chip



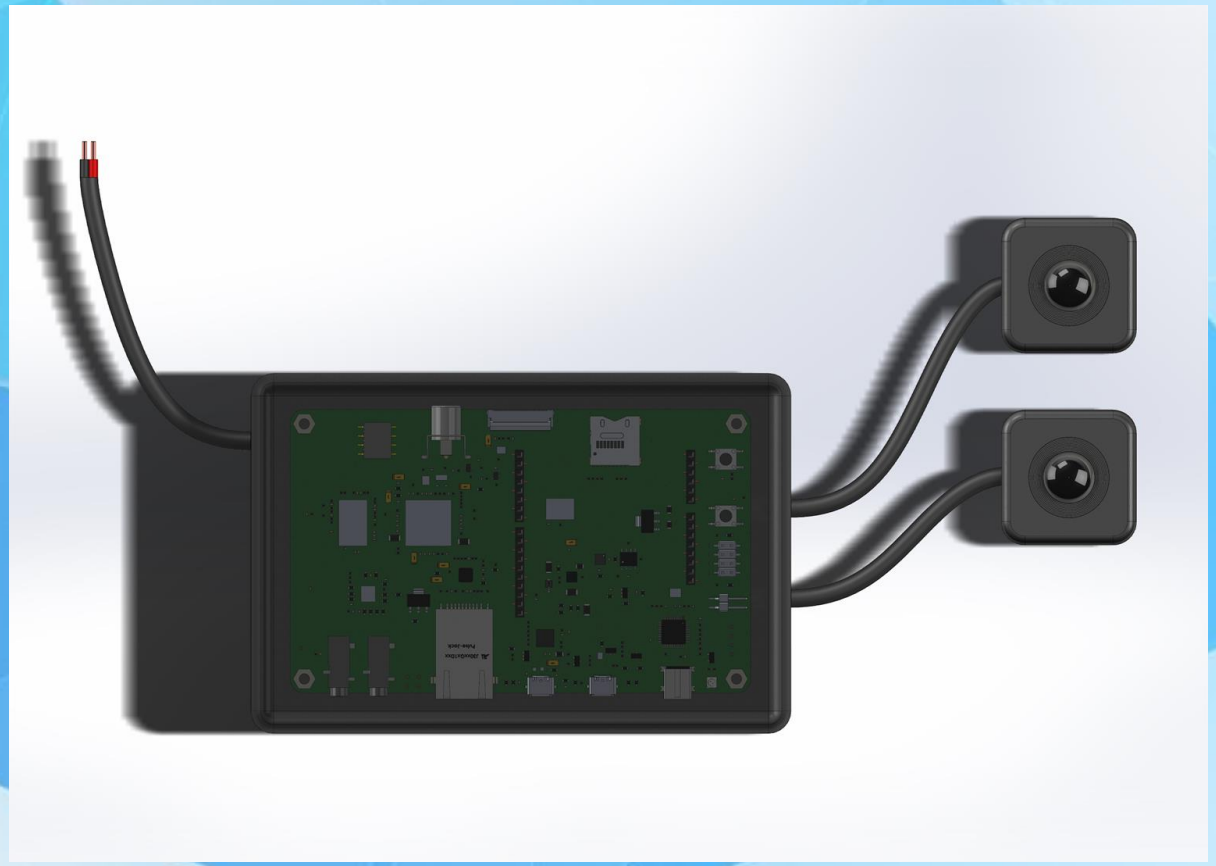
Design Diagrams





PCB and Camera Enclosures

Front View

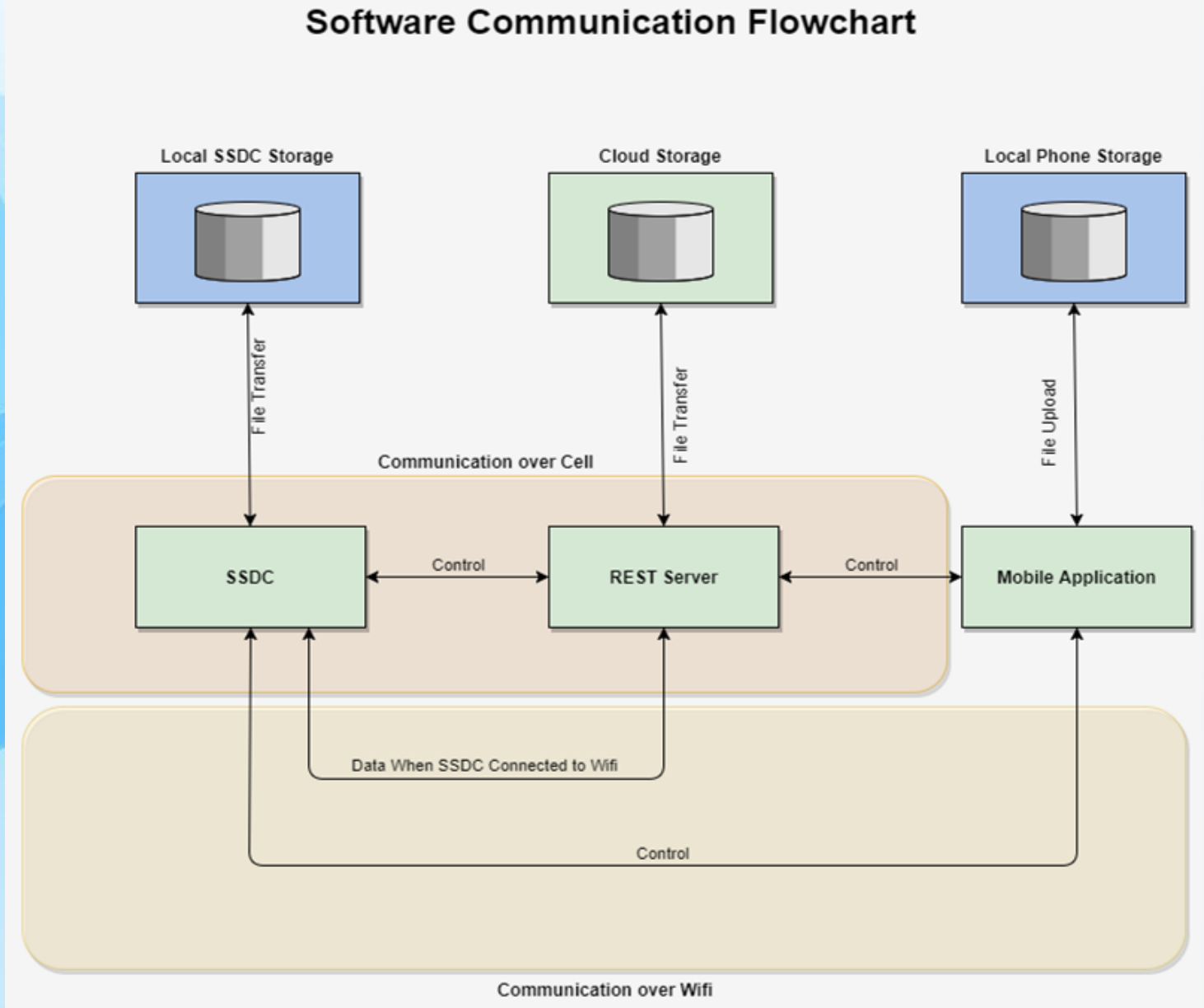


Side View

Server

Software Communication

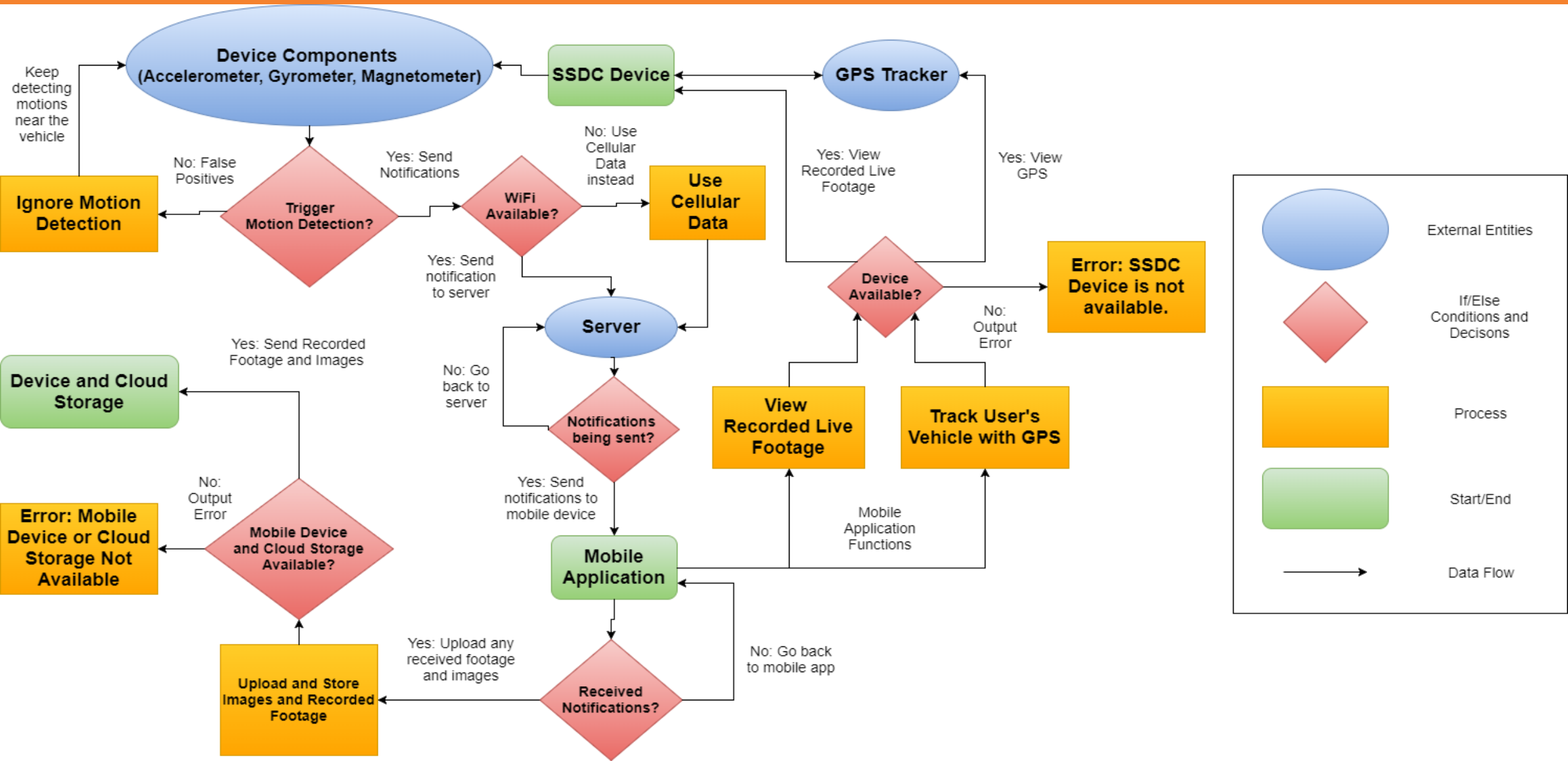
- Django Rest Framework
- AWS Hosting
- Dev Environment: Local Server
- Python 3
- Java Mobile App



Security

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

Data Flowchart



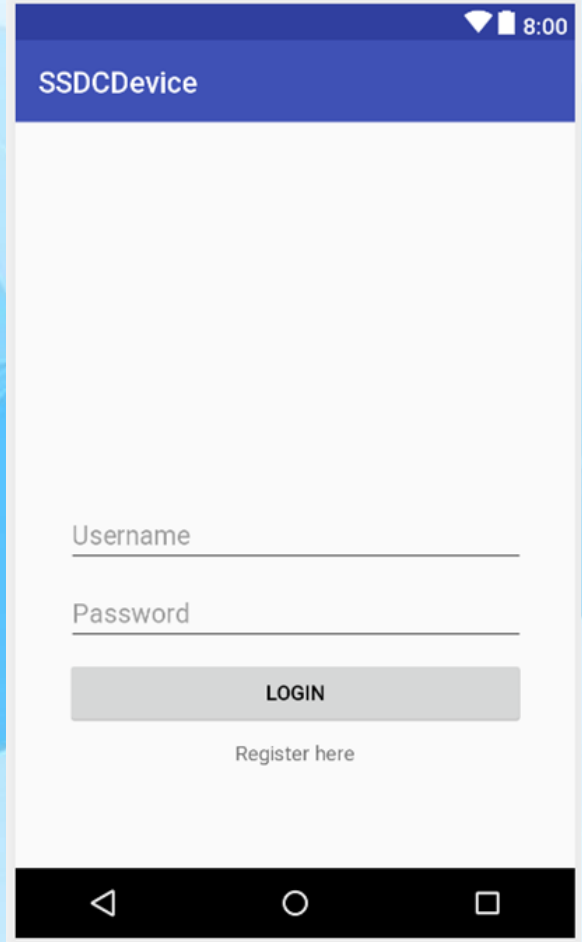
Mobile Application

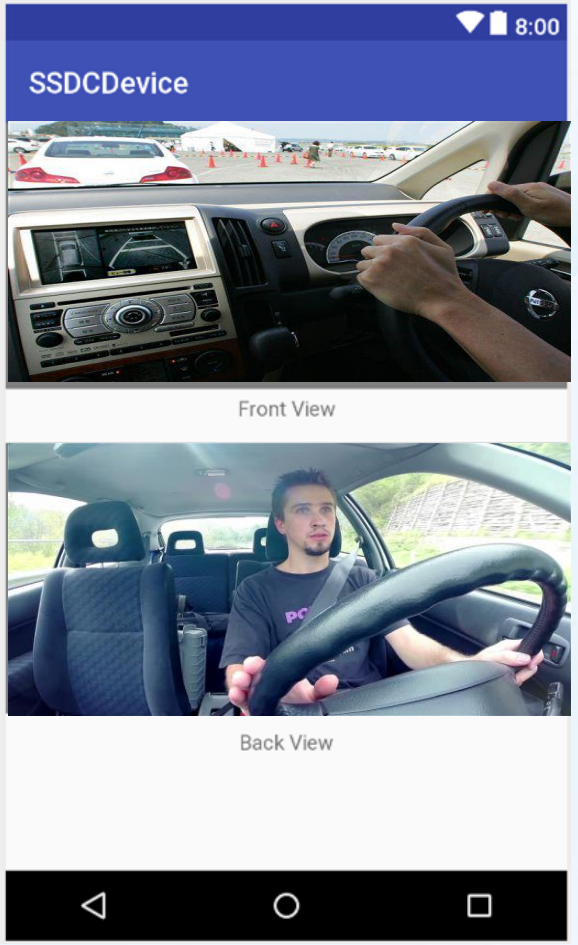
Android Studio

- 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

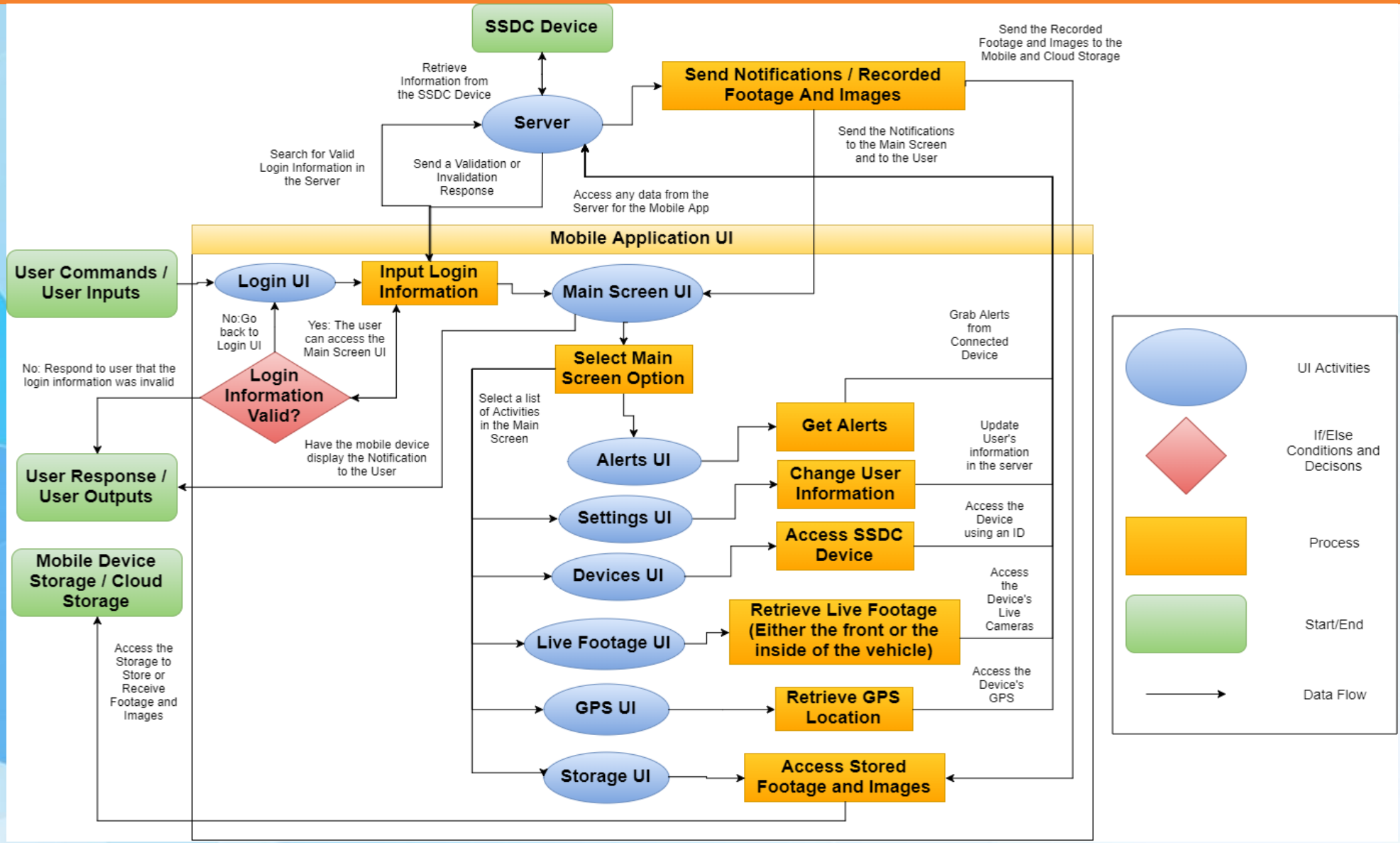
GUIs

- Login/Register
- Alerts
- View Live Footage
- GPS (Vehicle Location)
- View Stored Footage and Images (Cloud Storage)
- Manage Devices
- Settings





UI Diagram


SSDC Device
Retrieve Information from the SSDC Device
Server
Send Notifications / Recorded Footage And Images
Send the Recorded Footage and Images to the Mobile and Cloud Storage
Search for Valid Login Information in the Server
Send a Validation or Invalidation Response
Send the Notifications to the Main Screen and to the User
Access any data from the Server for the Mobile App
Mobile Application UI
User Commands / User Inputs
Login UI
Input Login Information
Main Screen UI
No:Go back to Login UI
Yes: The user can access the Main Screen UI
Login Information Valid?
No: Respond to user that the login information was invalid
Have the mobile device display the Notification to the User
User Response / User Outputs
Select Main Screen Option
Select a list of Activities in the Main Screen
Grab Alerts from Connected Device
Alerts UI
Get Alerts
Update User's information in the server
Change User Information
Settings UI
Access the Device using an ID
Access SSDC Device
Devices UI
Access the Device's Live Cameras
Retrieve Live Footage (Either the front or the inside of the vehicle)
Live Footage UI
Access the Device's GPS
Retrieve GPS Location
GPS UI
Storage UI
Access Stored Footage and Images
Mobile Device Storage / Cloud Storage
Access the Storage to Store or Receive Footage and Images
UI Activities
If/Else Conditions and Decisons
Process
Start/End
Data Flow

Adminstrative Content

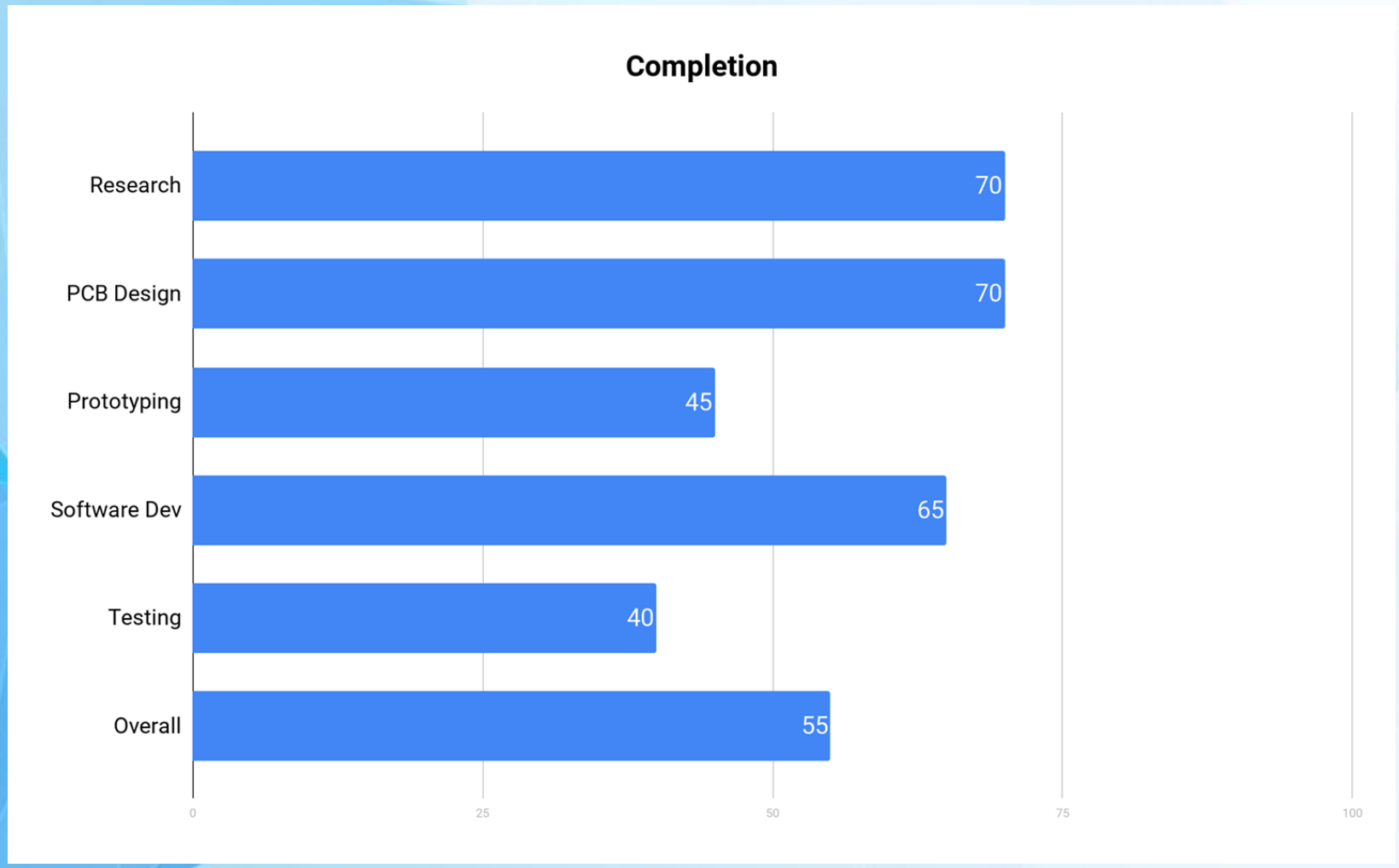
Design Difficulties

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

Budget and Financing

Development Costs				
Item	**Quantity**	**Cost**		
Atmel MCU Chip	1	$12.71		
TI Video Processor Dev board	1	$412.00		
Arduino	1	$44.99		
Rental of Sony Sensors	1	$150.00		
PCB	2	$800.00		
Cameras	2	$200.00		
Cell Dev Plan	1	$100.00		
Sensors	2	Free		
Sierra Wireless modem	1	$70.00		
Total	**12**	**$1729.70**		

Current Progress



Questions?

We’ve got answers