

## **Design Specification**

### **I. Project Overview**

Car is the one of the main ways of transporting in US. One of the problems for car drivers is checking for cars approaching from blind spot area. Some modern cars, especially self-driving car, have already fixed this problem by integrating the detection system into the car itself. However, many old models or cheaper models still don't have this system yet. The goal of this project is to create a system for detecting approaching car in blind spot area and alert the users with sound for smartphone with emphasis on reliability and usability.

### **II. User Requirements**

Sponsor requirements:

The sponsor requires for that the system should be able to detect the objects when they get into detection zone and be able to warn the user through sound and no visualization. The sponsor also requires that the system will be a mobile app and connects with the sensor through wireless technology. The sponsor also states that the system should emphasis on accuracy, reliability, usability. The app should have a good UI and a mechanism to auto start-up after setting up. The sensor also needs to be low power consumption in order to be convenience for the user.

User Action:

For this system, there will be only one type of user, which is the normal user. In order to use the system, the user needs to mount the sensor to a place on the car (The detail of how to set up is not yet decided). The user will need to do a connection setup

with the sensor and enable the app. After the app is enabled, it will start up automatically whenever the user enter the car. When it is started up, it will constantly check for objects moving into its detection area in the background and warns the user through sound.

### **III. Development Environment**

#### **Hardware**

An Android mobile device, a proximity sensor that have wireless capability (this might be a sensor that includes processor and wireless features or a sensor that is connected to a board). The type of sensor is now narrowed down to ultrasonic or RADAR but is not yet finalized.

#### **Software**

Android studio for developing Android native app and a software to program the processor that is connected to the sensor.

#### **Explanation for Environment Choice**

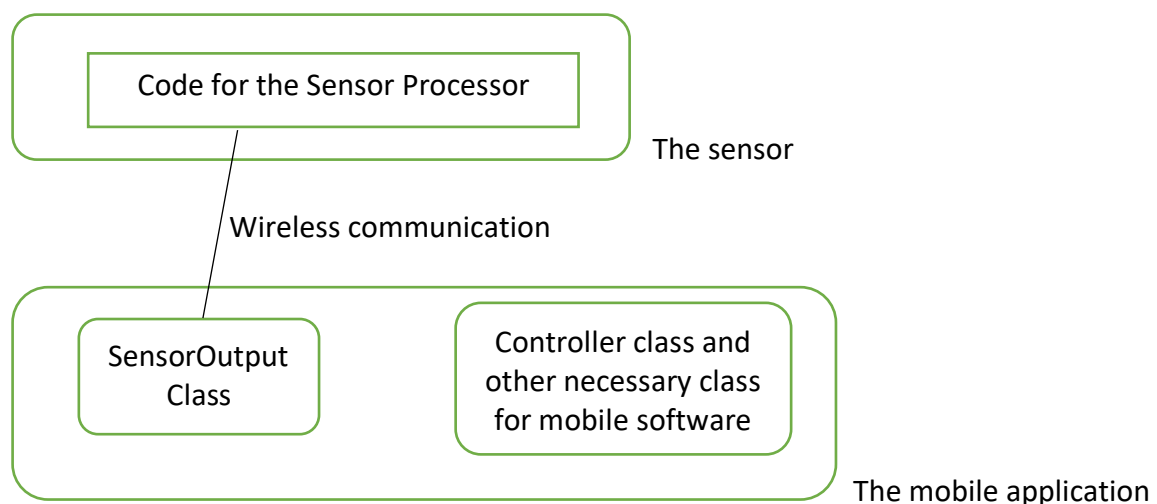
I decide that I would like to make this app a native application instead of hybrid or web app. There are 2 reasons for this decision. First, the sponsor requires a real-time reliable system for detection blind spot for car. This means that the system will need to run very fast in order to detect the car in time. Second, the sponsor wants a wireless connection to the sensors. Native app runs faster than the other options and have more choice for native functionality than can provide the wireless connection.

In the automotive field, there are several popular proximity sensors: LiDAR, RADAR, ultrasonic, and camera. Since the project required wireless connection, camera would not fit because camera would use too much power and will be slower too transmitted to the mobile device. Keeping reliability of the system i. mind, since car might still be used during dark and during rain, fog, and other weather, both LiDAR and camera are not a good choice. So, it is either ultrasonic or RADAR.

#### IV. Deployment Environment

The deployment Environment is supposed to be very similar to the development environment. The possible change lies in the sensor. The sensor might be changed to improve reliability and accuracy, or the sensor might be package together with the board and battery into a smaller unit that is more convenience to mount onto the car. If the sensor is changed, the software that is used to write the program for the sensor's processor will also have to be change.

#### V. Architecture



The SensorOutput Class would be responsible for establishing the connection, reading in data from sensor, constant check for update from sensor, and alerting the user with sound.

The ViewController and other file are responsible for the UI and flows of the app, how the app moves from one View to another.

## **VI. Implementation Strategies**

## **VII. User Interfaces**

User stories:

- When the user opens the app, the user will be shown a Home Page.
- The Home Page will show the status of the sensor, whether it is connected or not, a enable button and a set up connection button.
- When the start application button is clicked/tapped, the app will start scanning for objects. When an object gets into its detection zone, it should alarm the user with sound.
- When the set up connection button is clicked, The app will allow the user to set up the connection with the sensors (how this is implemented is currently undecided).

## **VIII. Test and Integration plan**

There are several tests that we need to do for the system. First, the wireless connection needs to be tested to make sure the connection is good and the device should be able to connect to the sensor (after first time set up) without any action from user. Second, we need to test the accuracy of the system to see how well it detects. Third, we need to

test to see the response time of the system to measure the reliability of the system.

Fourth, if it is required to implement the algorithm to detect objects using the sensor's raw data, then it is necessary to test that algorithm using simulation data as well as real data.

#### **IX. Project timeline**

Sep 21 – Deadline for deciding on the sensor a

Sep 30 – Deadline for getting the sensor and done with the basic UI

Oct 7 – Deadline for connect the sensor and the mobile device through wireless

Oct 14

Oct 21

Oct 28 – Deadline for object detection

Nov 4 – Initial Presentation and various documents submit

Nov 11

Nov 18 – Deadline for perfecting the system.

Nov 26 – Final presentation

#### **X. MVP**

The MVP of the app is application that is able to detect objects and warns the user through sounds. The app should connect with the sensor through wireless connection. There are focus on hardware in this project that might take time from developing a better software.

#### **XI. Bibliography**

- Referencing a lot of online website for comparing different type of sensors.
- Referencing a lot of Blind Spot Detection System on Amazon to survey existing products.

- Refencing a lot of Sensor company website while looking for industrial sensor and also referencing some of the knowledge in programming sensor
- Will referencing website to choose the sensor and reference their documents to program the sensor if needed
- Will referencing stackoverflow, google and other sites to complete the Android Mobile App