seGuRo Drive Mobile Web Application

Supervised By: Dr. Yugyung Lee Team:

Sivakumar Pothala Karthik Repala Surendra koripella

# Project Plan

## Introduction

Advancements in the today's mobile technology created a huge scope to develop applications that are beneficial to the users. To provide best solution for common problems like when a user forget parked vehicle location, for efficient fuel usage, and when the driver won't be in a position to call emergency services and give all his details. So this application stores the required information and provide optimum solutions at all the times.

## Project Goal and Objectives

* **Overall goal**

The objective of this project is to develop a mobile web application to deliver travel related services, which includes Navigation, Parking, Fuel monitoring, Weather, and Emergency Help services. This will be developed using HTML5, JQuery, Java script and CSS to make it platform independent, thus it works on Android, IOS and other mobile platforms.

* **Specific objectives**

Currently we have different applications for individual services. Here our goal is to build a one stop solution for all these travel needs so that user get the better service by combining data from various services. And also the user inputs the voice commands rather than using traditional texting method to initiate any process in this application.

* **Significance**

Seguro Drive Application provides safety and confidence to the Driver. This Application is developed by using mobile web technologies, so it is compatible with all mobile and web platforms. It alerts Driver with sudden weather changes, fuel level and driving tips to save fuel. User can get emergency services help at all the times by just inputting two voice commands.

## Project Background and Related Work

1. Fuel Monitor

This App is useful to monitor the fuel level in the vehicle. If the fuel level goes down, it will remind user to refill the tank. For this user has to enter the Odometer reading and amount of fuel filled in the tank.

1. Find My Car

This app useful to remember the GPS position of a car, hotel or any other location. It shows current position of a user and the car position on the map. So User can find the car easily.

1. Navigation

User can Explore new places, discover local attractions, and navigate with Google Maps.

1. Weather Services

User can see two-day forecasts with severe weather warnings for current location, and Push notifications for severe weather alerts in the United States.

## Proposed System

1. Requirement Specification

* Functional:
* User has to input vehicle make, year, model, odometer reading, fuel level, fuel price and trip distance to get fuel performance charts
* User has to input source and destination to get Navigation directions and also can specify to get weather information
* User has to mark the location while parking thus user can find the parked car by hitting find my car button from current location
* User has to input voice commands to get Emergency Help by saying Emergency Help and Reason for help.
* Non-functional:

All geo location services are always up and available

* Technical/business Requirements:

Front End: HTTPS Client, JQuery, Ajax

Back End: C#.NET

Project Management Tool: Scrum Do

Version Control: GIT Hub

Server: IBM Cloud

* Business Process/Workflow analysis

1. Tracking work using Scrum Do
2. Technical meetings with Professor to clarify any questions
3. Email communications among team members
4. Implementing each service and testing compatibility with the application

* Technological and Architectural requirements

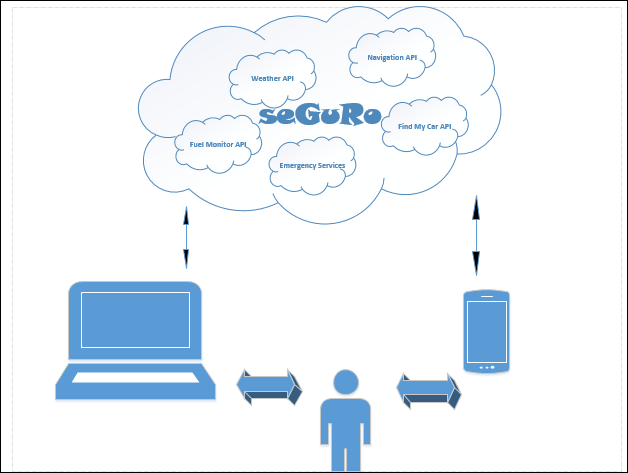
Google Maps API, Navigation services API, Find My Car API, Fuel Monitor API, HTTPS Client, Jquery, C#.NET and SQL Server.

1. Framework Specification

* Assumptions and Principles

Users accent should be clear to match with voice recognition service. Users should able to access this application with smart phones or any other devices with web connection.

* **System Architecture Diagram**



1. **System Specification: Identify Primary Services**

**Existing Services**

1. Navigation Service

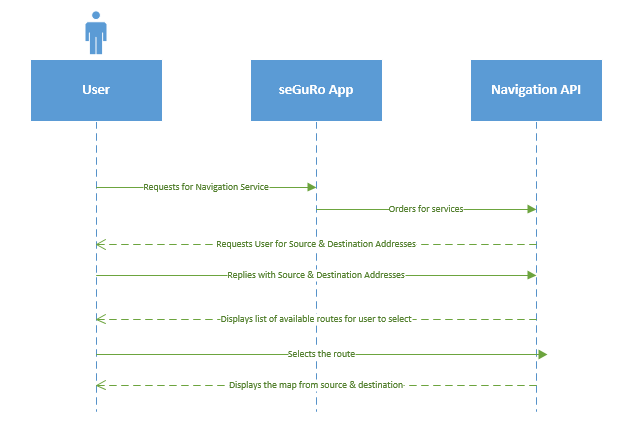
URL: <https://play.google.com/store/apps/details?id=com.google.android.apps.maps&hl=en>

1. Weather Service

URL: <https://play.google.com/store/apps/details?id=com.accuweather.android>

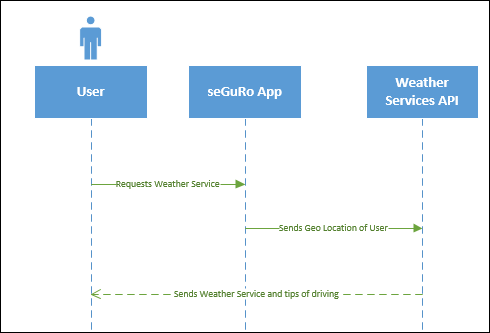
**Navigation Service**

**Sequence Diagram**

****

**Weather Services Service**

**Sequence Diagram**

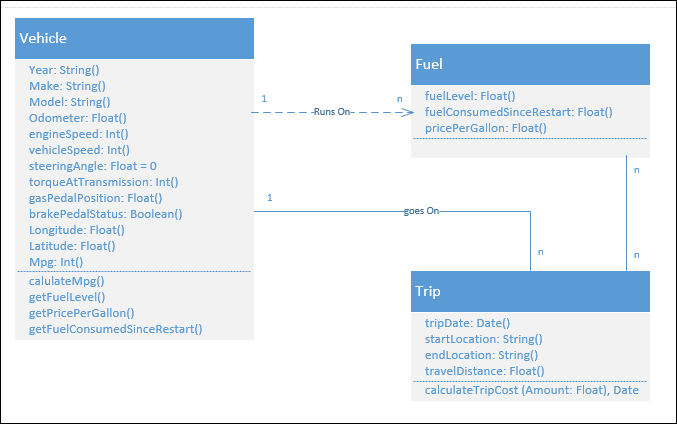
****

**New Services to be built**

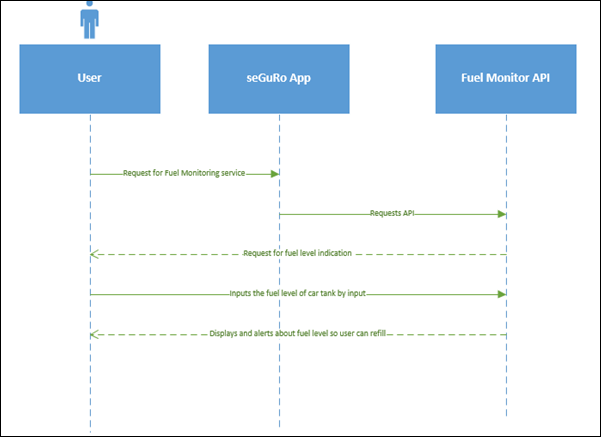
1. Fuel Monitoring Service
2. Parking Service
3. Emergency Help

**Fuel Monitor**

**Class Diagram**

****

**Sequence Diagram**

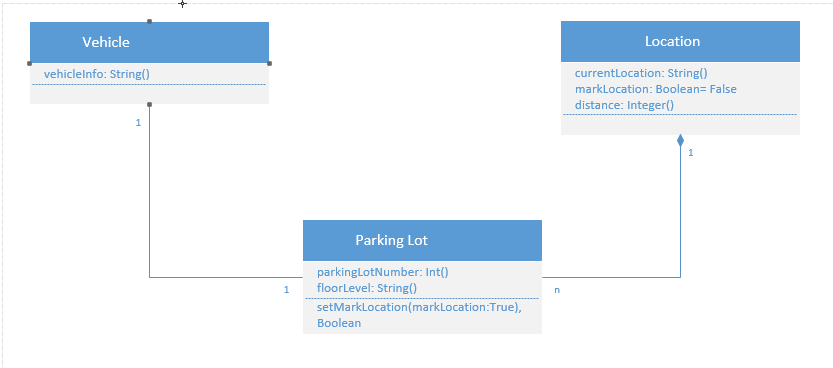
****

**Service Specification**

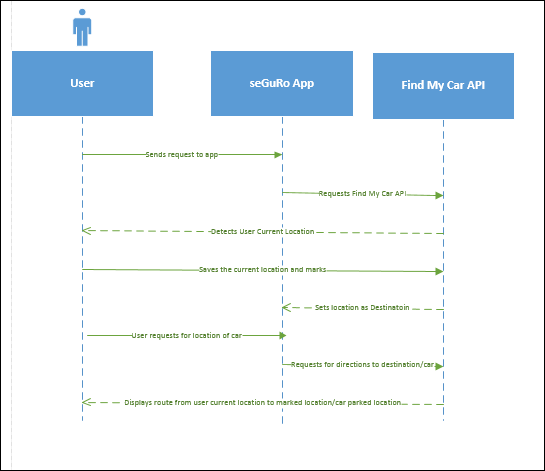
* Operational description: User inputs fuel fills, trip destination, fuel cost while starting the trip and gets fuel performance for daily trips.
* Input/output for services: Input is Fuel and Vehicle information and Output is Pictorial representation of fuel consumption
* Constraints/exceptions: User has to initiate his trip every time manually user wants get service
* Service flow/alternative flow:Client Browser interacts with Fuel Monitor API to get data and process this information in backend and displays on client browser with suitable charts daily.
* Priorities: IBM Cloud instance should be active and Fuel API should provide accurate data for every vehicle type available in the market

**Find My Car**

**Class Diagram**

****

**Sequence Diagram**

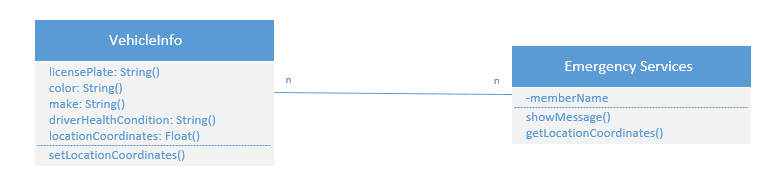
****

**Service Specification**

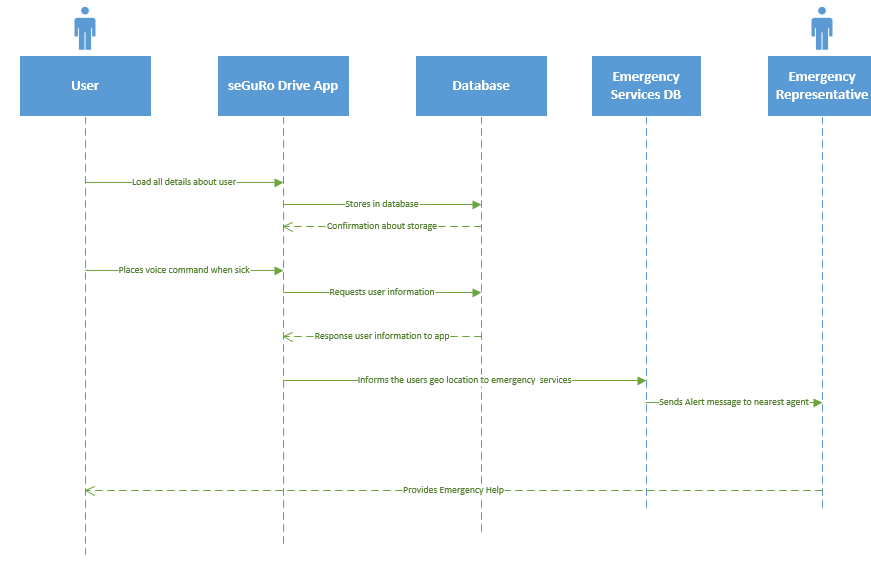
* Operational description: User marks location when parked a vehicle and asks mobile web application to give directions from current location of user.
* Input/output for services: Input is user clicks on client web application and Output is Navigation map with directions to find the car
* Constraints/exceptions: Internet connection and Maps service available
* Service flow/alternative flow: Client Browser interacts with Find my Car API through secured HTTP connection over IBM cloud
* Priorities: IBM Cloud instance is active, Testing Car parking different locations.

**Emergency Services**

**Class Diagram**

****

**Sequence Diagram**



**Service Specification**

* Operational description: User gives audio inputs on Client Browser to get help from emergency services
* Input/output for services: Input is Audio commands and Output is Physical Help from Emer services
* Constraints/exceptions: Users voice should sync with application interface
* Service flow/alternative flow: Mobile Web Browser to Emer Services Database through IBM Cloud server
* Priorities: Highest priority is integrating Voice Recognition Service with the Application

**Design of Mobile Client**

* Features: HTML5
* Styles: Cascading Style Sheet (CSS3)
* Technologies: Java Script, HTML, JQuery

## Plan by Services (using ScrumDo)

A new project named seGuRO Drive has been created with the three team members.

A global story has been created under the seGuRo Drive project. Four new iterations have been created named Increment 1, Increment2, Increment3 and Increment4 with the respective deadlines. For each iteration a story has been written.

http://www.scrumdo.com/projects/project/seguro-drive/

GIT Hub project also created and posted in spread sheet.

## Risk management

1. Testing this application for Emergency Help service is a risk in communication with realistic emergency services like Police, Ambulance and Fire departments.
2. Integrating all services into one application makes this application complex if we miss projected plan
3. Risk with voice recognition service as it is build on US English, App may not receive international people accent

* Technological and Architectural Requirements

Need tools like NUnit, GIT Hub, JsFiddle

## Posting Web Sites

1. Done Hosting Client web on IBM cloud site
2. Done creating ScrumDo Project and link posted in Google spread sheet
3. Done creating GIT Hub repository and project and link posted in Google spread sheet

## Bibliography

**Fuel Monitor**

<https://play.google.com/store/apps/details?id=com.mad.app.fuelmonitor&hl=en>

**Find my Car**

<https://play.google.com/store/apps/details?id=com.elibera.android.findmycar>

**Navigation**

<https://play.google.com/store/apps/details?id=com.google.android.apps.maps&hl=en>

**Weather Services**

<https://play.google.com/store/apps/details?id=com.accuweather.android>

**Google Text to Speech**

[http://weston.ruter.net/2009/12/12/google-tts/](https://bluprd0113.outlook.com/owa/redir.aspx?C=rk3EPbpPvE2Q70HWAdnybOeSh4hjiNAIKdDwa2ysa6Gxmk3PXUlDiZutd264Yyp1v9wv902hTZs.&URL=http%3a%2f%2fweston.ruter.net%2f2009%2f12%2f12%2fgoogle-tts%2f)

**Speech Recognition**

[http://developer.android.com/reference/android/speech/RecognizerIntent.html](https://bluprd0113.outlook.com/owa/redir.aspx?C=rk3EPbpPvE2Q70HWAdnybOeSh4hjiNAIKdDwa2ysa6Gxmk3PXUlDiZutd264Yyp1v9wv902hTZs.&URL=http%3a%2f%2fdeveloper.android.com%2freference%2fandroid%2fspeech%2fRecognizerIntent.html)

**Note:** Current Project Plan is estimated plan and is subject change up on implementation progress and requirement changes.