SENIOR PROJECT

SAFAY – TH3

(Safe Way for Drivers)

Submitted to

School of Information, Computer and Communication Technology
Sirindhorn International Institute of Technology
Thammasat University

May 2017

by

Preecha Wanaraksakul 5622792216
Thanyanit Kaewka 5622792323

Advisor: Dr Teerayut Horanont

Abstract

Smartphone applications have become one of the most convenient ways to receive information or services. Likewise, a developer can also retrieve large scale crowd sourced data from many users' smartphones such as position or sensor data while driving.

To monitor road conditions in such a setting, this project explores utilizing a mobile phone with a tri-axial accelerometer, GPS sensors to collect required data. The data are analyzed to detect road anomalies and classified into road conditions.

Resulting in the ability to report drivers awareness of the poor road surface on both mobile phone and web application which is available for the users to review the reporting results in any places on the map.

Acknowledgements

Thanks to all the contributors, senior project advisor, master degree students and supporters of this study, from idea discussion and data collection to algorithm implementation and finally senior project preparation. We appreciate everyone’s concern and help at difficult times. Again thanks in advance.

Contents

Abbreviations

The technical terms listed below will be expressed by their abbreviations in this report.
Other abbreviations will be defined at their first occurrence.

Accelerometer	an instrument for measuring acceleration typically that of an automobile, ship, aircraft or that involved in the vibration of a machine, building ,or other structure
Geolocation	The identification or estimation of the real-world geographic location of an object, such as mobile phone or Internet-connected computer terminal
GPS	Global Positioning System
Gyroscope	Device for measuring or maintaining orientation and direction

Chapter 1
Project Concept

1.1 Summary

Road traffic injuries are a multi-dimensional problem that require a comprehensive view when examining the determinants, consequences and solutions. This project is aim to monitor road surface anomalies, such as potholes, speed bumps etc.

To monitor road conditions in such a setting, this project explores utilizing a mobile phone with a tri-axial accelerometer, GPS sensors to collect required data. The data are analyzed to detect road anomalies and classified into road conditions.

Resulting in the ability to report drivers awareness of the poor road surface on both mobile phone and web application which is available for the users to review the reporting results in any places on the map.

1.2 Motivation

The Smartphone has become a common tool for most people so that they can communicate or access to many services easily. Smartphone applications have become one of the most convenient ways to receive information or services. Likewise, a developer can also retrieve large scale crowd sourced data from many users' smartphones such as position, sensor data or even velocity while driving cars.

Road networks connecting buildings, villages, cities, and countries are the most important transportation infrastructure in modern life. The surface condition of sealed roads is directly related not only to driving experience and traveling comfort, but more importantly to driving safety. For a responsible government, it is important to maintain the good condition of road networks. Awareness of road surface anomalies in the earliest manner possible is the first step toward prompt repair. By cooperating with mobile sensing technologies, it is possible to develop a vehicle-based crowdsourcing system that automatically and pervasively monitors the existence of road surface anomalies.

1.3 Users and Benefits

This system has been proposed and built to leverage computing and communication technology for various purposes:

- traffic management
- routing planning
- safety of vehicles and roadways
- emergency services

The benefits to all the users include: abilities to track the road surface condition with a draw line mapping while driving vehicles and report information back to a server for aggregation and analysis. To leverage mobile phones carried by multi-users as traffic probes. As well as to provide the awareness to users with an alert function on the mobile application about speed control and obstacles.

1.4 Typical Usage

The system works on both cars and motorcycles. A road surface condition detection and classification system was implemented and proposed based on a smartphone's sensors. The sensor's data was collected by a mobile phone and sent directly to a laptop for analysis. The wireless communication interfaces of smartphones e.g. Wi-Fi, 3G, or 4G are used to report events to or obtain information from the cloud servers. Three classes of road condition were defined including good road, bad road and road bump.

While driving vehicles, mobile phones are used to detect the road surface and the real time position where the users are. The device does not need to be at the same linear along the road as it is automatically set to be the same as the vehicle forward direction. The road condition and the draw line mapping information will be computed and transferred between the cloud servers and the clients. Any bad roads or road bumps detected nearby will be alerted by the mobile application.

1.5 Expected Challenges

Data usage: In a crowdsourcing system, the data source is contributed by the crowds. Since the communication interface for the vehicle on the road surface is mainly via mobile networks such as 3G/LTE, the data usage will be considerable if excessive unnecessary data are reported, which may discourage the user to contribute in the crowdsourcing process.

Power consumption: Since the smartphone battery is the most valuable resource to keep applications alive, the running detection should be less power consumed.

User Interface: Signs, symbols or any illustrations would be considered carefully to grasp the users' attentions as well as to be simple and easy to understand.

Chapter 2

Requirements Specification

2.1 System Description

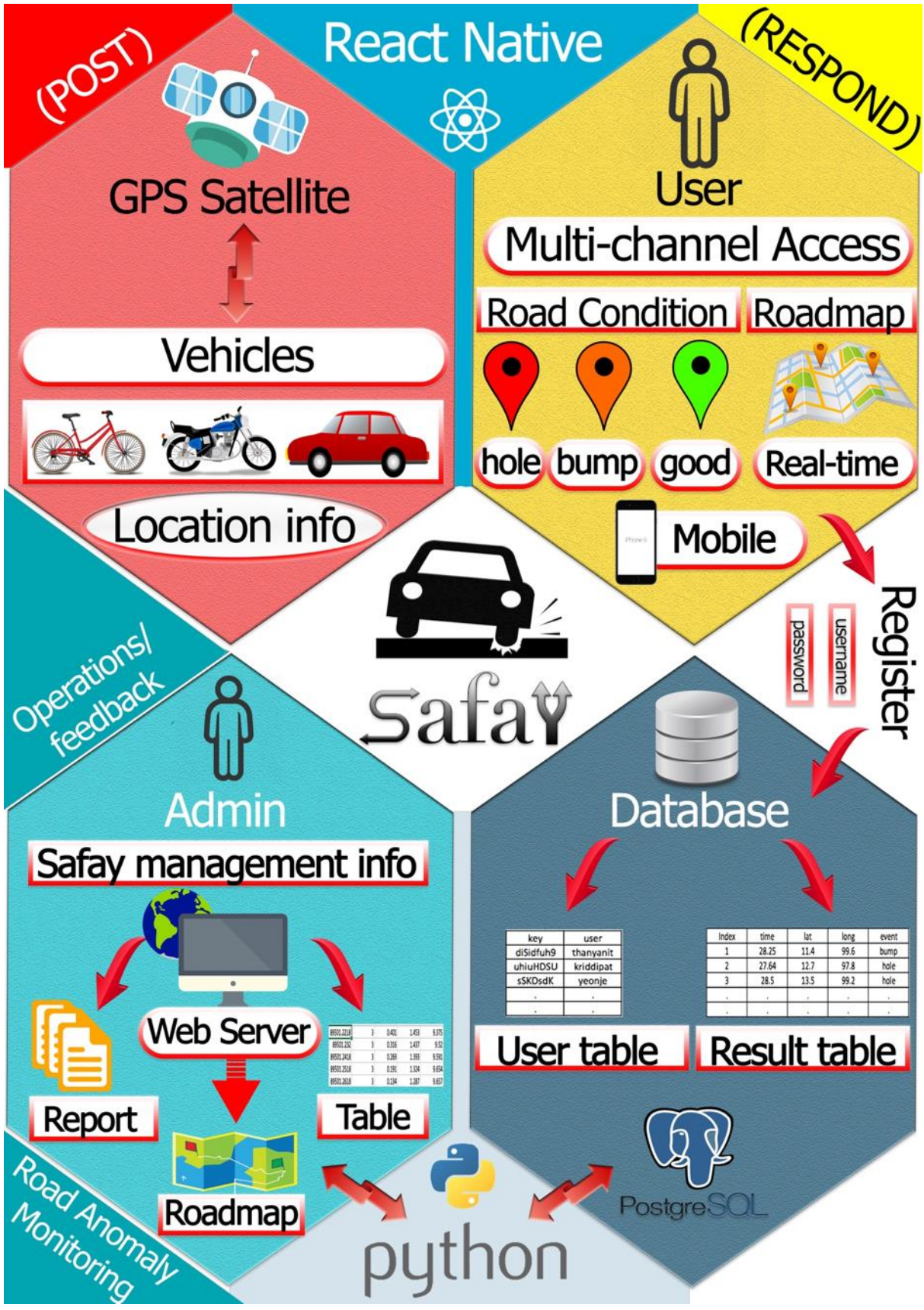
Safe way for drivers (Safay) is an application that report users the up-to-date road qualities using google map for reference location in the real world. Our application is aim to monitor road surface anomalies, such as potholes, speed bumps etc. by using sensors data from a mobile phone.

The system works on both cars and motorcycles. A road surface condition detection and classification system was implemented and proposed based on a smartphone's sensors. The sensor's data was collected by a mobile phone and sent directly to a laptop for analysis. The wireless communication interfaces of smartphones e.g. Wi-Fi, 3G, or 4G are used to report events to or obtain information from the cloud servers.

While driving vehicles, mobile phones are used to detect the road surface and the real time position where the users are. The device does not need to be at the same linear along the road as it is automatically set to be the same as the vehicle forward direction. The road condition and the draw line mapping information will be computed and transferred between the cloud servers and the clients. Any bad roads or road bumps detected nearby will be alerted by the application.

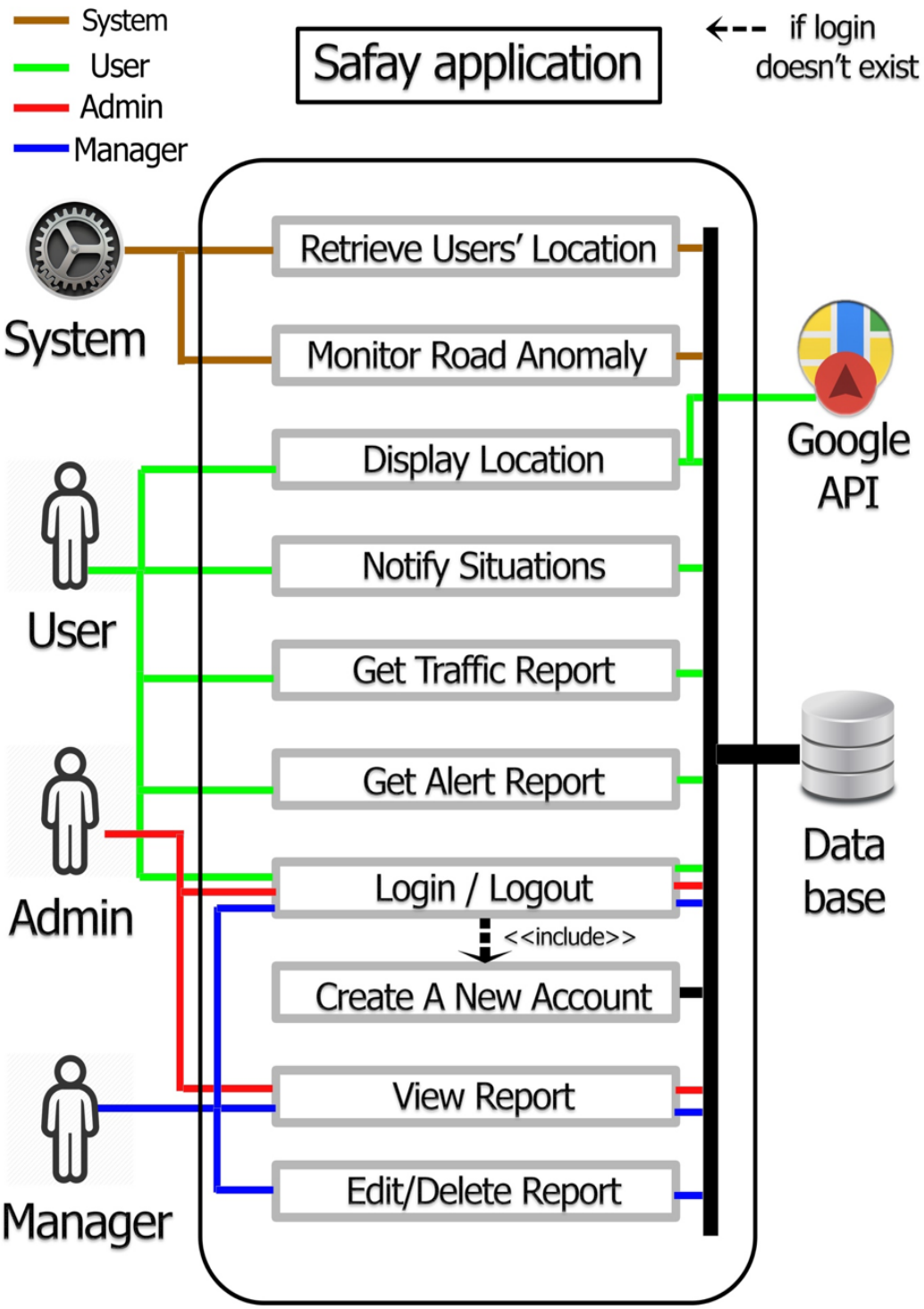
2.1.1 Perspective

The relationship between Safay and other systems in shown in Figure 1


(POST)
React Native
(RESPOND)
GPS Satellite
User
Multi-channel Access
Road Condition
Roadmap
Vehicles
hole
bump
good
Real-time
Location info
Mobile
Register
password
username
Operations/
feedback
Safay
Admin
Database
Safay management info
key
user
diSidfuh9
thanyanit
uhiuHDSU
kriddipat
sSKDsdK
yeonje
Index
time
lat
long
event
1
28.25
11.4
99.6
bump
2
27.64
12.7
97.8
hole
3
28.5
13.5
99.2
hole
Web Server
User table
Result table
Report
Table
Roadmap
Road Anomaly
Monitoring
python
PostgreSQL

2.1.2 Function

The functions of Safay Application are shown in Figure 2.



2.2 Requirements

The functions are summarized as:

Use Case Name:	**Login / Logout**
Actors:	- User - Admin - Manager
Precondition:	Actors are identified.
Stakeholder & their Interests:	*Drivers:* - want just easy process to start the app - do not want to fill up the login process every time
Main Success Scenario:	1. Insert username and password. 2. System returns visibility of related functions based on each actor.
Extension:	*If login does not exist:* 1. Create a new account 2. Data stored into database
External Requirement:	Access Token

Use Case Name:	**Display Location**
Actor:	User
Precondition:	User is identified.
Stakeholder & their Interests:	*Drivers:* - want to see the clear image on the app which can tell where they are
Main Success Scenario:	1. Login process is underway 2. System returns user's location on the map 3. System retrieves user's location 4. System monitors road anomalies
Extension:	*If the location are not shown:* 1. Reset the mobile network 2. Login
External Requirement:	- Mobile Network - GPS Tracking

Use Case Name:	**Retrieve User's Location**
Actor:	System
Precondition:	User is identified.
Stakeholder & their Interests:	*Admin:* - want to have accurate data stored to the database
Main Success Scenario:	1. Login process is underway 2. System returns user's location on the map 3. Location data is stored into database.
Extension:	-
External Requirement:	Database

Use Case Name:	**Monitor Road Anomalies**
Actor:	System
Precondition:	User is identified.
Stakeholder & their Interests:	*Admin:* - want to receive accurate responses compare to the actual road anomalies
Main Success Scenario:	1. Login process is underway. 2. System returns user location on the map. 3. Location data is stored into database. 4. System monitors the road anomalies around the user's location by the system defensing area.
Extension:	-
External Requirement:	Database

Use Case Name:	**Get Alert Report**
Actor:	User
Precondition:	User is identified.
Stakeholder & their Interests:	*Drivers:* - want to really get alert when the anomalies are close to them *Admin:* - want to make sure the main function in alerting the anomalies which data retrieved from the database are really work
Main Success Scenario:	1. Login process is underway. 2. System returns user location on the map. 3. Location data is stored into database. 4. System monitors the road anomalies around the user's location by the system defensing area. 5. The color signs are shown on the map

	according to the road conditions while the user can get notice by the app feature.
Extension:	*If no data collected on that road path before:* 1. System returns notifications on having no data along the path. 2. The data will be stored and getting alert to the user next time. 3. Other users along the same path behind get alert as the data are stored. *If the internet are out of control:* 1. System continues monitoring the road anomalies. 2. The data will be updated on the map after the internet problems are corrected.
External Requirement:	- Database - The stored anomalies data retrieve from the database

Use Case Name:	**Get Traffic Report**
Actor:	User
Precondition:	-
Stakeholder & their Interests:	*Drivers:* - want to get some additional functions on the app eg. traffic report
Main Success Scenario:	1. Login as guest / User Login 2. System returns the location on the map. 3. System displays reports retrieving from the database showing other users' routes on the map. 4. The color lines are shown on the map according to the traffic conditions.
Extension:	-

External Requirement:	- GPS Tracking - The routing data retrieved from the database

Use Case Name:	**Notify Situations**
Actor:	User
Precondition:	-
Stakeholder & their Interests:	*Drivers:* - want to get some additional functions on the app eg. traffic report
Main Success Scenario:	1. Login as guest / User Login 2. System displays related contact or useful information eg. Mass Transit Authority 3. The user can contact any of these authorities given on the app, in case that accident occurs.
Extension:	-
External Requirement:	Stored and update the related contacts and information.

Use Case Name:	**View/ Edit/ Update Reports**
Actors:	- Admin - Manager
Precondition:	Actors are identified
Stakeholder & their Interests:	Admin: - can only view reports Manager: - can view, edit and update reports
Main Success Scenario:	1. Login with username and password. 2. System returns visibility of related functions based on each actor. 3. Admin gains a permission in order to monitor the system and to view reports 3.1 Manager gains an authority in order to monitor the system as well as to view, edit and update reports.
Extension:	-
External Requirement:	System Accessing Authentication

2.2.1 System Architecture

SA1: The system **must** interface with the Authentication System

SA2. Database:
The system **must** be able to retrieve data from and update data to the database accurately.

SA3. Self-learning capability:
The system **should** have the capability to learn the system parameters for road anomaly detection and indexing without user-involved training. Otherwise, it would discourage users from participating in the crowdsourcing process

SA4. Data usage:
In a crowdsourcing system, the data source is contributed by the crowds. Since the communication interface for the vehicle traffic is mainly via mobile networks such as 4G/LTE, the data usage will be considerable if excessive unnecessary data are reported, which **may** discourage the user to contribute in the crowdsourcing process. Therefore, it is important to extract key features that represent the severity of road anomalies.

SA5. Power consumption:
Since the smartphone battery is the most valuable resource to keep applications alive, the running detection **should** be less power consumed.

SA6. Performance:
The server **must** calculate and report back the road conditions to the users as a batch file in not more than 5 seconds at a time as well as a close to real time tracking on GPS.

2.2.2 Smartphone Capabilities

SC1. Computing:
CPU, operating system, and storage that provides a programmable computing platform

SC2. Sensing:
– Audio: microphone.
– Localization: GPS receiver.
– Motion: accelerometer sometimes included for functions such as gesture recognition.

2.2.3 Authentication

AU1: The system **must** allow users, admin, and manager to access to the functions of the system as they supposed to be.

AU2: The system **should** verify what type of user logged on to the system.

AU3: The system **must** show restrained user-type information.

AU3: The system **must** notify when the user inputs an incorrect username or password or both.

Chapter 3

Design Specification

3.1 System Architecture

Safay is a mobile application that report users the up-to-date road qualities using google map API for reference location in the real world. Our application is aim to monitor road surface anomalies, such as potholes and speed bumps by using sensors from mobile devices which are GPS, accelerometer, gyroscope, and magnetometer.

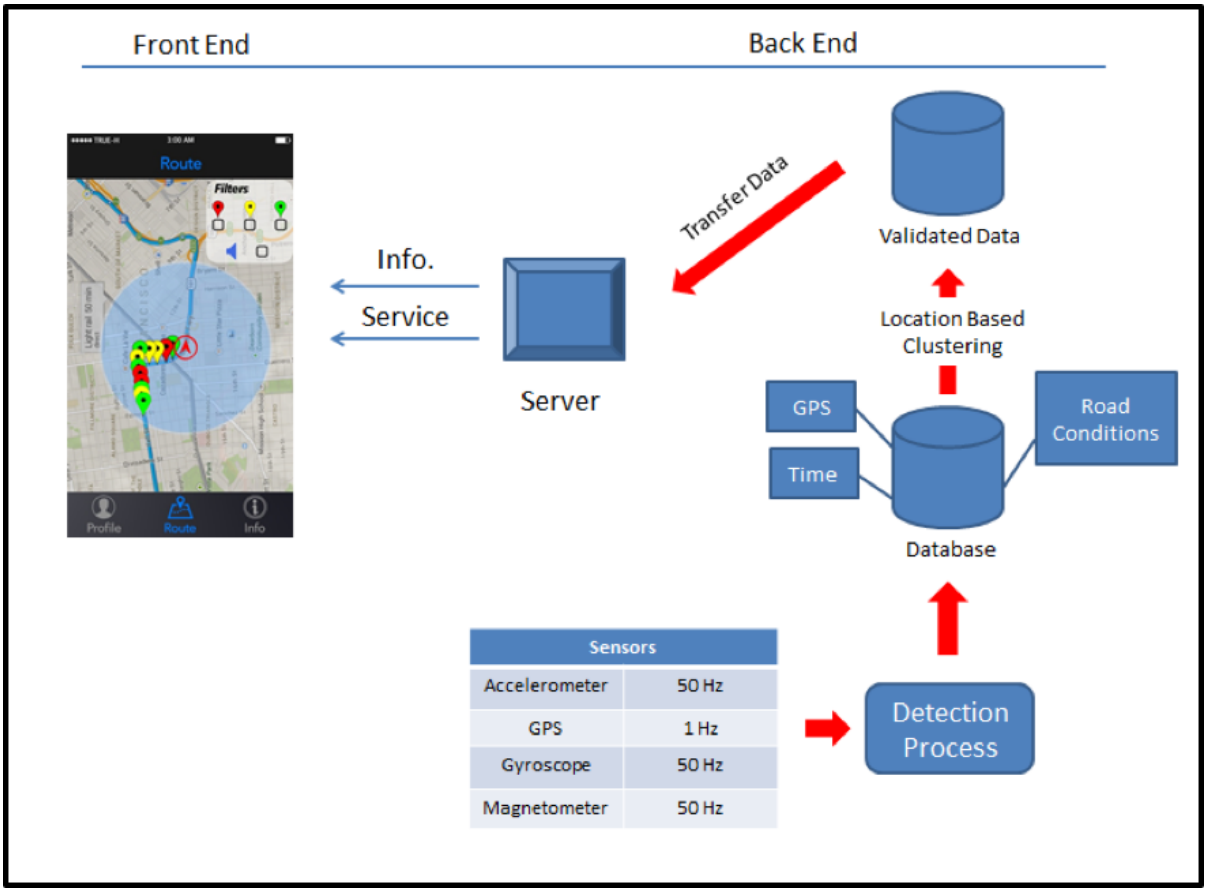


Figure 3 : System Architecture

3.1.1 Server-side Functionality (Back End)

- **Road Condition Detection and Classification System**

  Three types of road conditions are defined, good road, bad road and road bump. The good road is a road that has a flat and smooth surface. It have a low vibration in both the vertical axis and lateral axis of the vehicle. In contrast, a bad road is a road that has a rough surface and full of potholes.

- **Data Collection**

  The acceleration, orientation and position from the smartphone's GPS sensor was collected. The smartphone was placed on the vehicle floor on the passenger side and the y-axis acceleration was set to be the same as the vehicle forward direction.

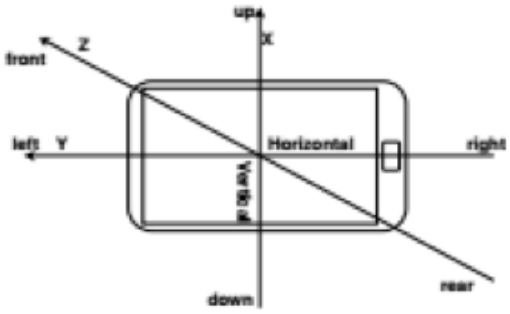


Figure 4 : Smart phone orientation inside the car.

- **Event Retrieving Reports**

  Once inserted into the database the reports are then made available for retrieval. An interface exists between the database and web server that allows for the retrieval of reports in readable format. It is used by the client devices to display road event reports on the map fragment, and it is used by the web server user interface to display the data in a map format as well as in a data table format.

Figure 5 : Detected data used to display road event reports on the map.

3.1.2 Client-side Functionality (Front End)

- **Authentication**

  The *multi-user authentication* allows you to create different profiles for different users or groups of users. The profiles define the access rights to different resources on the Web server. The users which are allowed to access the Web server are in the user database.

- **Information Service**

  Current position of each users will be displayed on the map as well as the road qualities alerting system along the paths.

3.2 Detailed Design

3.2.1 Database designs

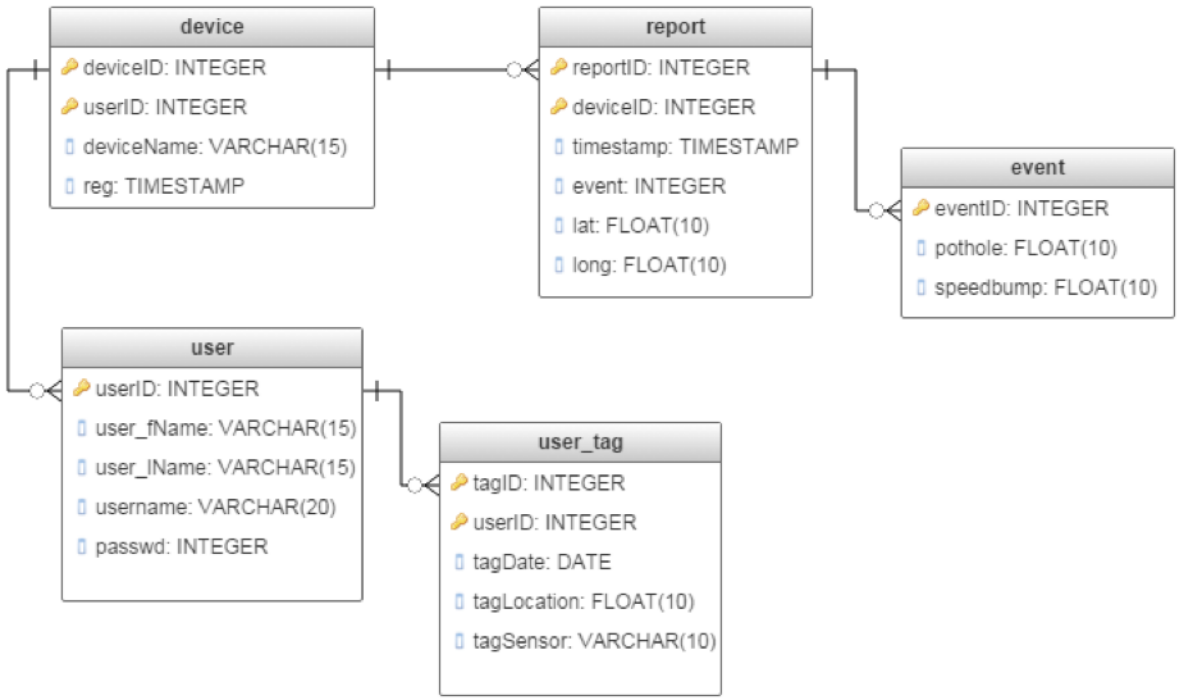


Figure 6 : Database ER Diagram

For data storage, a postgres_SQL database is used. The schema of the database is illustrated in the form of an entity-relationship diagram in Figure 4. The database consists of five tables, and two main table are considered: Device and Report. Device stores a unique identifier for a client as well as a timestamp of the first report received from that device. The report table will hold records of pothole reports. Each report identifies the client, the location the pothole was reported from, the time of the report, and the sensors of the action on the phone that triggered the pothole and speedbump detection.

3.2.2 User Interface design

Login page

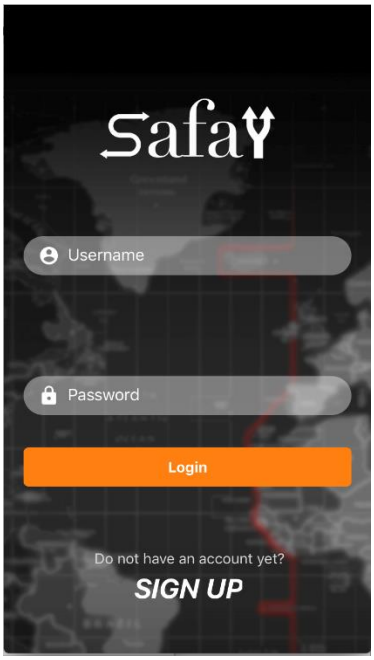


FIgure 7 : Login page

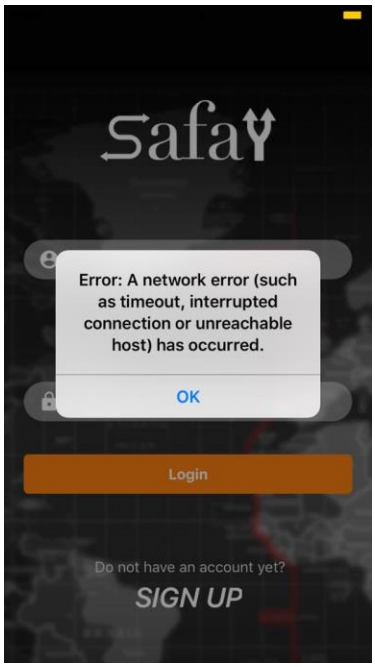


FIgure 8 : Network error

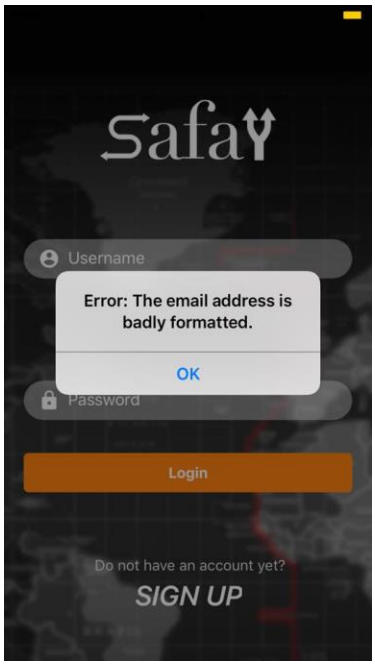
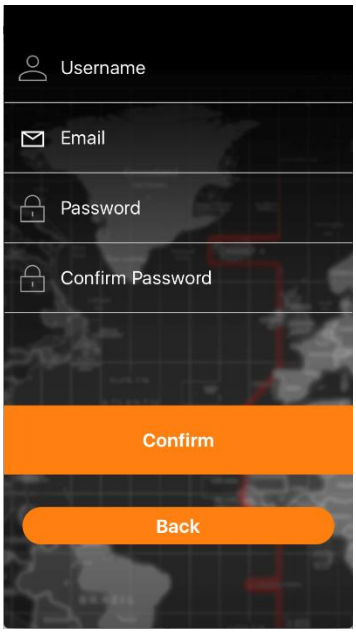
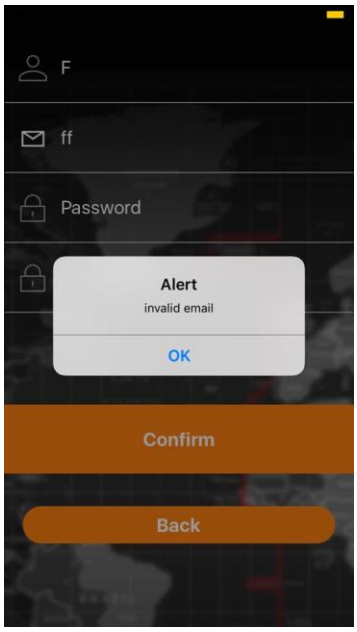


Figure 9 : Invalid ID or pass

This is the first page of the application that user can login. Just enter your username and password and hit the login button. If user don't have an account user can click signup at the bottom of screen. If user has not connect to wifi or internet will be appear network error. If user have wrong type their id or password will be appear badly formatted.

Sign up page





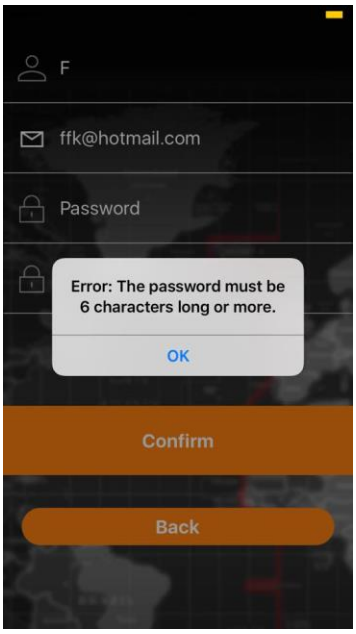


Figure 10 : Signup page

Figure 11 : Invalid email

Figure 12 : at least 6 password

User must fill all of personal information. If user has some blank. An error message will be displayed. When user fill all blank and hit the sign up button will be display successfully. And keep personal information to database. User can hit button back or go to login page. If user has type not correct the format will be appear invalid email. User have to type correctly e-mail; Example : youremail@hotmail.com. And user have to type a password at least 6 characters.

Route page

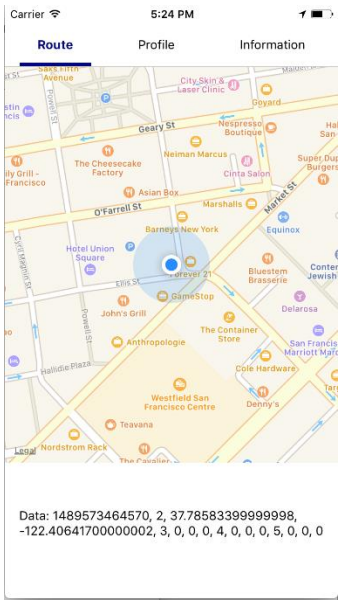


Figure 13 : Route page

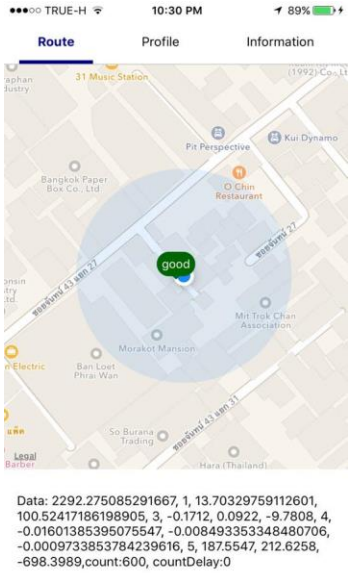


Figure 14 : Good road

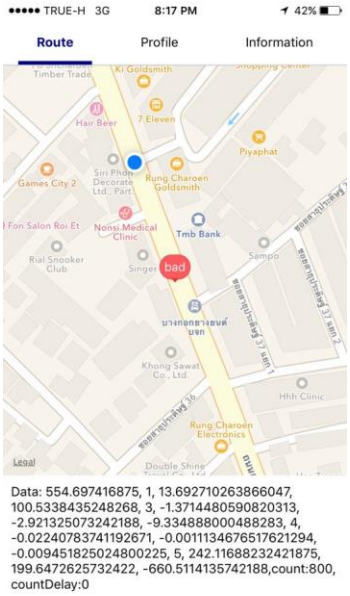


Figure 15 : Bump road

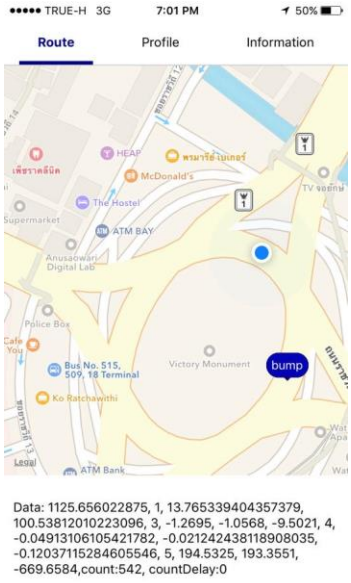


Figure 16 : Bad road

When user login will appear route page. User can filter to show road condition that got on the cloud. Red is pothole. Yellow is speedbump. And Green is good. Mobile must detect 4 Sensor. Accelerometer, Gyro Sensor, Magnetometer and GPS.

Profile page

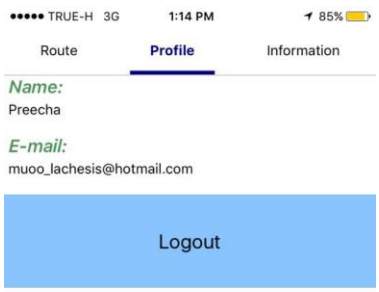
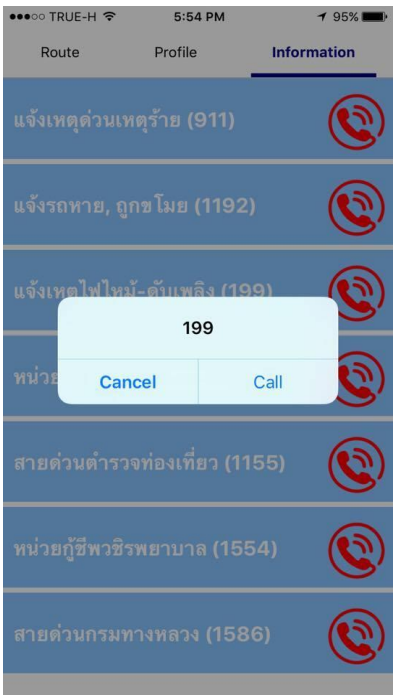


Figure 17 : Profile page

This page is personal data. User can view in my profile and must read and agree term of service, and logout if user want to disconnect to the Safay systems.

Information page



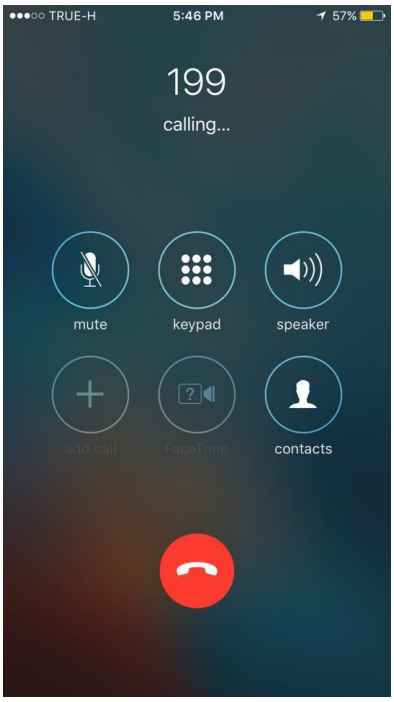


Figure 18 : information

Figure 19 : press number

Figure 20 : calling

This page is all of telephone number emergency. User can search specific emergency by name. When user hit telephone number emergency will call to emergency directly.

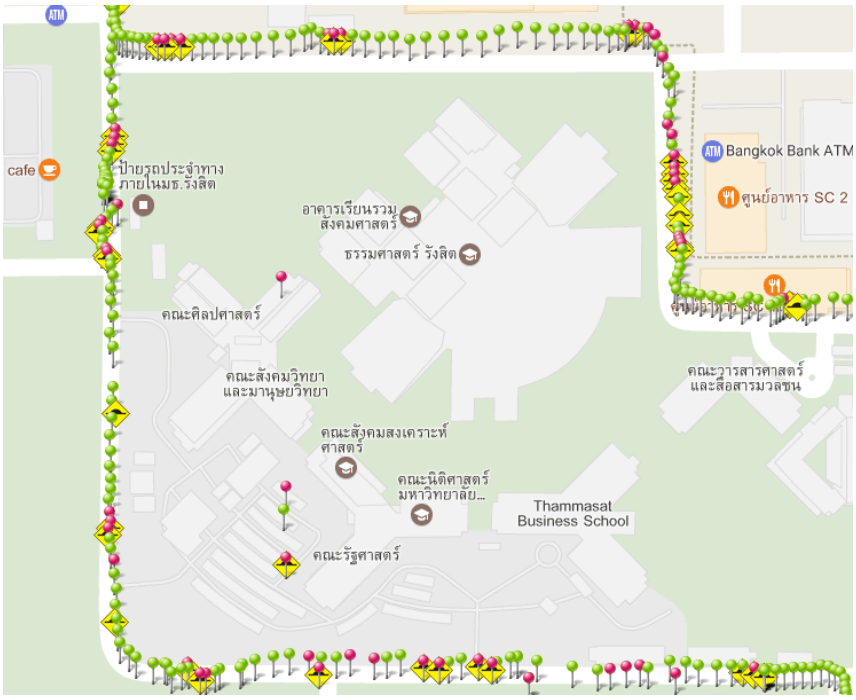


Figure 21 : Road Conditions Report

Users can search for all the result on the web. Markers can be filtered out by their types of qualities and date/time.

3.2.3 Algorithm

- **Flow Chart**

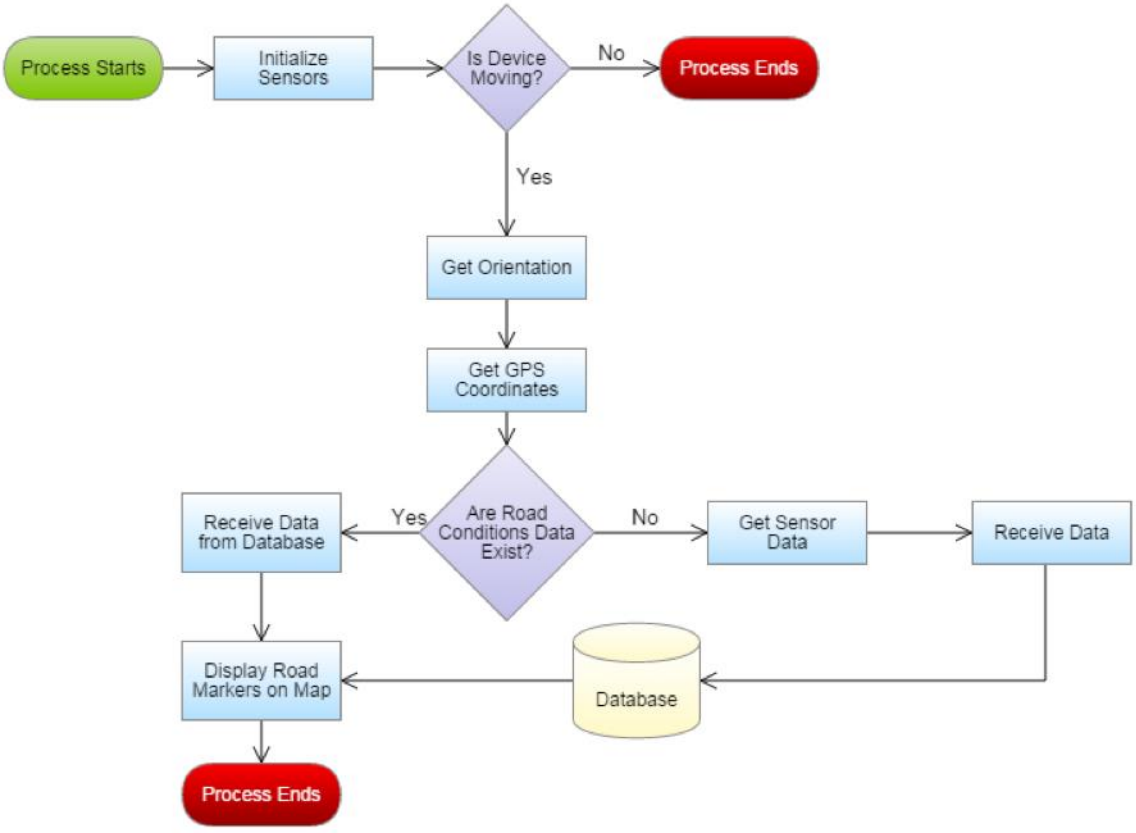


Figure 22 : Sensing, Road Condition Detection, Visualization Activities Diagram

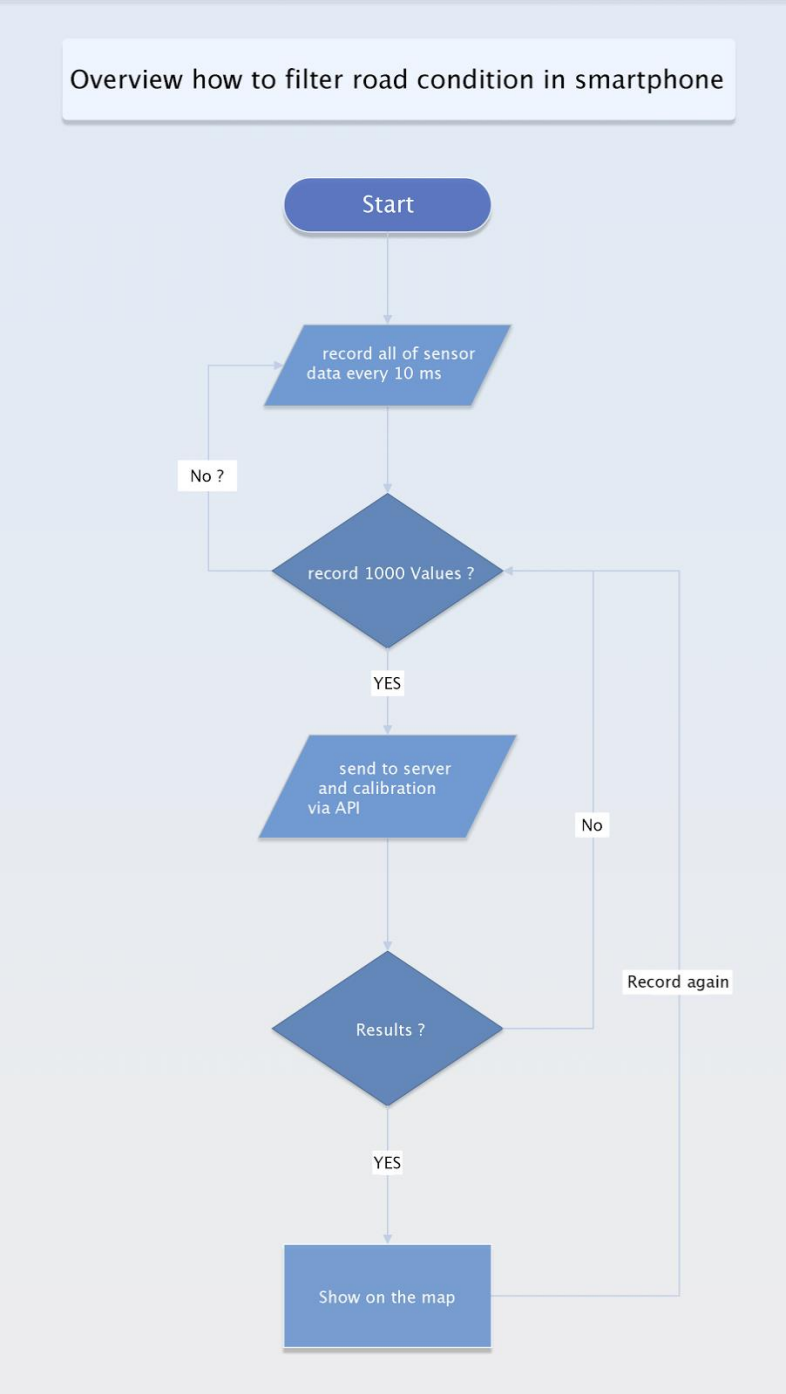


Figure 23 : Overview road condition in smartphone diagram

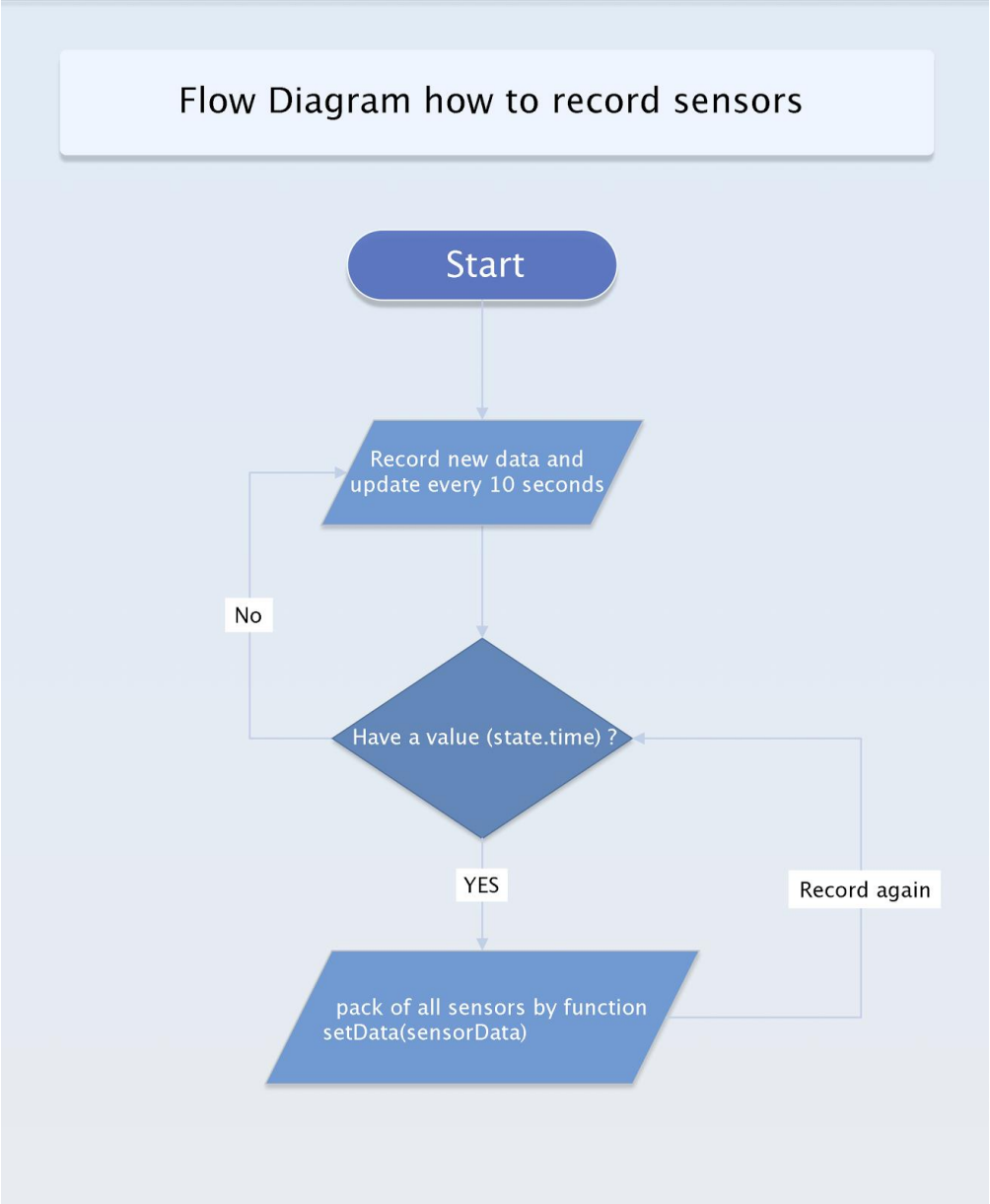


Figure 24 : record sensors diagram

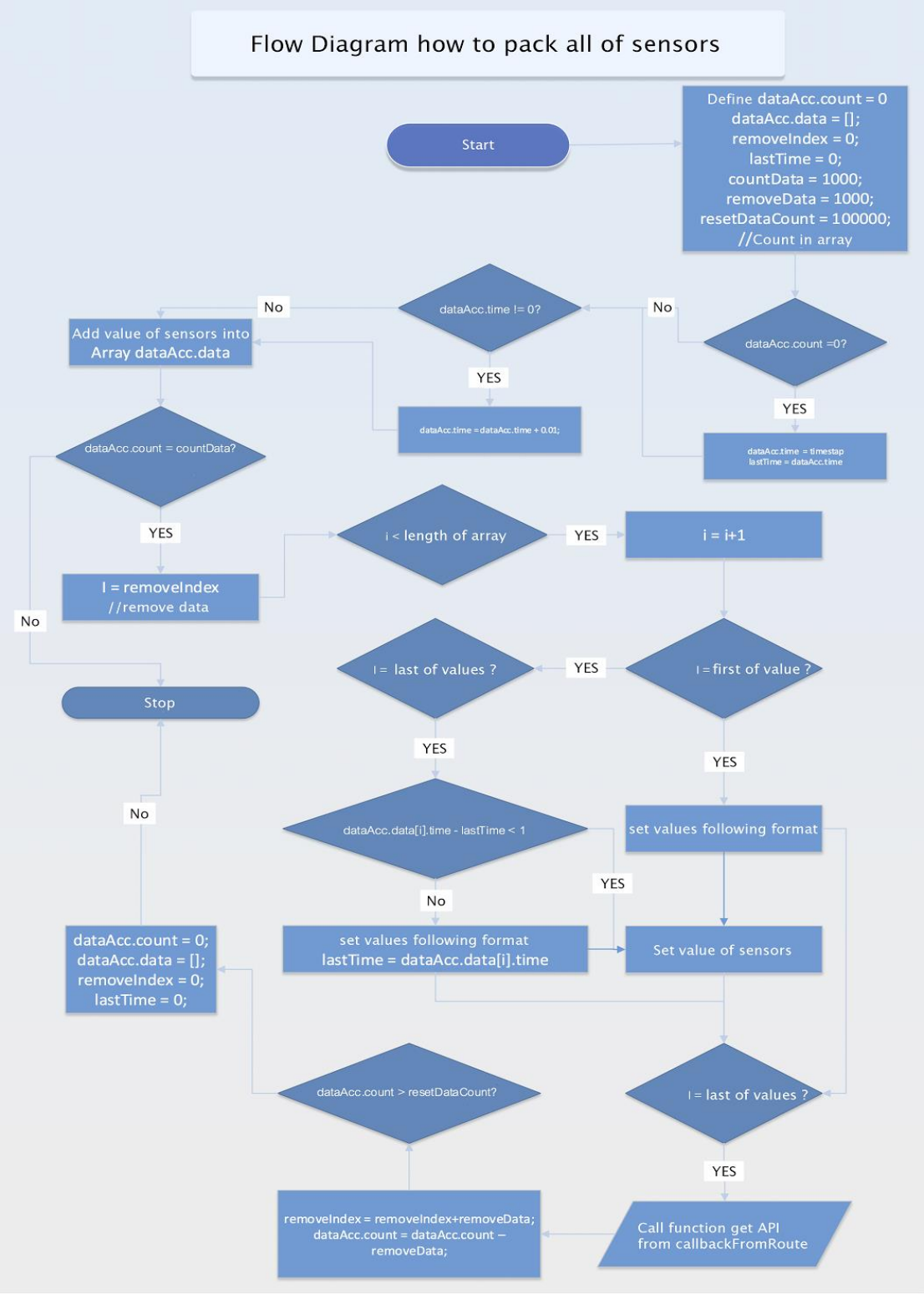


Figure 25 : pack all of sensors diagram

Chapter4
System Manual

Safay project we have 2 systems 1. Mobile application 2.Web application
In Mobile Application we use react-native for mobile platform which can run on cross platform that means we can use the same source code for develop in both platform (ios, android) and we try on ios platform (ios can run on mac only).

4.1 Installation and Usage

1. Install PgAdmin database
- Connect the database with python server.

2. Import tested data into database.

3. Query data
Server URL:
http://gisddns.hopto.org:5555/json?fetchby=location&lat=18.798285&lon=99.035038&rad=10
Port: 5555 or 5556

How to fetch:

1. Fetch by location
   Follow by lat(latitude) , lon(longitude), rad(radius)

2. Fetch by date/time
   start_d/end d = “(yyyy-mm-dd)”
   start_t/end_t = “(hh:mm:ss)”

4. install react-native

5. install react-native module and link

6. open xcode and sign your development (99 $) and set permission module

7. setting in xcode allow arbitrary loads to “Yes”

8. sync to your real device and run application

If you got have some error you should solve following error code and if something is not response you have to close it all and reopen it again.

4.2 Implementation Overview

React-native is a new language to experience but problem can occur easily if you don’t really understand its structure. You have to check function on hardware which some are not support. When you run on a real device for a long time the device will slow down to run and high temperature. Some errors can occur on a real device.

Software

React-native v.0.42.3

Firebase v.3.7.3

react-native-communications v.2.2.1

react-native-maps v.0.13.0

react-native-motion-manager v.0.42

Hardware

Iphone 6s

Macbook pro 2016

4.3 Feature List

1. Authentication

- Identifies users individually through the registration process and users login via mobile devices. Then allows the system to access based on their identities and make updates to the data.

Feature status: ✓ Complete ☐ Partial ☐ Initial

2. Geolocation

- Defines an interface to location information associated with the device hosting the implementation, such as latitude and longitude.

Feature status: ✓ Complete ☐ Partial ☐ Initial

3. Mobile Detecting Sensors

- The mobile device is equipped with GPS and mobile sensors to collect road conditions data and location information in order to report the road surface anomalies results to the users.

Feature status: ✓ Complete ☐ Partial ☐ Initial

4. Responsive Web Application

- Besides mobile app, users can go through the web application for more features.

Feature status: ✓ Complete ☐ Partial ☐ Initial

5. Road Condition Report

- Road condition results are available to retrieve from the server and installed on to the map.

Feature status: ✓ Complete ☐ Partial ☐ Initial

6. Filter

- Filter the road conditions markers by date/time and their colors.

Feature status: ✓ Complete ☐ Partial ☐ Initial

7. News and Events

- Users can be able to read, report daily situations and catching up with the news feed.

Feature status: ✓ Complete ☐ Partial ☐ Initial

8. Service

- Provides related emergency numbers and be able to make a phone call as well as to connect to the emergency online service on their websites.

Feature status: ✓ Complete ☐ Partial ☐ Initial

4.4 Test Results

We need to make sure that our road condition results are accurate enough.

To do so:

1. record the video of the road surface while driving and using the mobile application together.
2. compare the results from the real road surface with the result on the map (the accurate is around 85-90%)

Appendix A

Motion sensor.js

A.1 Accelerometer

```
Accelerometer.setAccelerometerUpdateInterval(0.1); // in seconds
DeviceEventEmitter.addListener('AccelerationData', function (data) {
  /**
  * data.acceleration.x
  * data.acceleration.y
  * data.acceleration.z
  **/
});
Accelerometer.startAccelerometerUpdates(); // you'll start getting AccelerationData events above
Accelerometer.stopAccelerometerUpdates();
```

A.2 Gyroscope

```
Gyroscope.setGyroUpdateInterval(0.1); // in seconds
DeviceEventEmitter.addListener('GyroData', function (data) {
  /**
  * data.rotationRate.x
  * data.rotationRate.y
  * data.rotationRate.z
  **/
});
Gyroscope.startGyroUpdates(); // you'll start getting GyroscopicData events above
Gyroscope.stopGyroUpdates();
```

A.3 Magnetometer

```
Magnetometer.setMagnetometerUpdateInterval(0.1); // in seconds
DeviceEventEmitter.addListener('MagnetometerData', function (data) {
  /**
  * data.magneticField.x
  * data.magneticField.y
  * data.magneticField.z
  **/
});
Magnetometer.startMagnetometerUpdates(); // you'll start getting MagnetomerData events above
Magnetometer.stopMagnetometerUpdates();
```

Bibliography

[1] Github, “pwmckenna/source code listings,” 2014, [Online; accessed 14-September-2016]. [Online]. Available: https://github.com/pwmckenna/react-native-motion-manager