|  |
| --- |
| **E:\My Documents\Desktop\Logo_FPT_University_doc.jpg**  **MINISTRY OF EDUCATION AND TRAINING** |
| Capstone Project Documents |
| Smart Traffic |
|  |
|  |
| |  |  | | --- | --- | | *Group Members:* | Lam Hai Vu - SE05148  Vu Van Cuong - SE04725  Nguyen Duy Tri - SE04770  Le Hong Dung - SE04562 | | *Supervisor:* | Phan Duy Hung | | *Capstone Project Code:* | STF |   **Hanoi, July 2019** |

Table of Contents

[Definitions and Acronyms 6](#_Toc17907569)

[ABSTRACT 6](#_Toc17907570)

[1. Introduction 7](#_Toc17907571)

[1.1. Problem and motivation 7](#_Toc17907572)

[1.2. Related Work 7](#_Toc17907573)

[1.3. Project objectives 8](#_Toc17907574)

[2. Project Management 10](#_Toc17907575)

[2.1. Project Organization 10](#_Toc17907576)

[2.1.1. Software Process Model 10](#_Toc17907577)

[2.1.2. Organization Structure 11](#_Toc17907578)

[2.1.3. Roles and Responsibility 11](#_Toc17907579)

[2.2. Project management plan 13](#_Toc17907580)

[2.2.1. Project Milestones and Deliverables 13](#_Toc17907581)

[2.2.2. Boundaries of the project 14](#_Toc17907582)

[2.2.3. Project Schedule 14](#_Toc17907583)

[2.3. Quality Management 16](#_Toc17907584)

[2.3.1. Quality control activities 16](#_Toc17907585)

[2.3.2. Document Standard 16](#_Toc17907586)

[2.4. Communication Management 17](#_Toc17907587)

[2.5. Risk Management 18](#_Toc17907588)

[2.5.1. Risks 18](#_Toc17907589)

[2.5.2. Risk control activities 19](#_Toc17907590)

[3.1. User Requirement Specification 21](#_Toc17907592)

[3.2. System Requirement Specification (Specific Requirements) 21](#_Toc17907593)

[3.2.1. External Interface Requirements 22](#_Toc17907594)

[3.2.2. System Features 23](#_Toc17907595)

[3.3. Other material (if any) 38](#_Toc17907596)

[4.1. Proposed system architecture 39](#_Toc17907598)

[4.2. Analysis and selection of tools, devices 40](#_Toc17907599)

[4.2.1. Server 40](#_Toc17907600)

[4.2.2. Database 41](#_Toc17907601)

[4.2.3. Smart Traffic Road Collection application 41](#_Toc17907602)

[4.2.4. Smart Traffic Map application 41](#_Toc17907603)

[4.3. Introduction/Resume a theory problem 42](#_Toc17907604)

[4.3.1. Use case of Smart Traffic Map application 42](#_Toc17907605)

[4.4. Algorithm and Solution Analysis 43](#_Toc17907606)

[4.4.1. Overview existing algorithms and solutions 43](#_Toc17907607)

[4.4.2. Implementation algorithm in our system 44](#_Toc17907608)

[4.5. Interface design 48](#_Toc17907609)

[4.5.1. Smart Traffic Map interface design 48](#_Toc17907610)

[4.5.2. Smart Traffic Road Collection interface design 51](#_Toc17907611)

[4.6. Database design 54](#_Toc17907612)

[4.6.1. List of tables 54](#_Toc17907613)

[4.6.2. Table admin 54](#_Toc17907614)

[4.6.3. Table road 55](#_Toc17907615)

[4.6.4. Table Points 55](#_Toc17907616)

[4.7. Protocol design 56](#_Toc17907617)

[4.8. Testing (Co-Design in Embedded System). 56](#_Toc17907618)

[4.8.1. Introduction 56](#_Toc17907619)

[4.8.2. Test Plan 56](#_Toc17907620)

[4.8.3. Testing level 58](#_Toc17907621)

[4.8.4. Testing Tools and Environments 58](#_Toc17907622)

[4.8.5. Test strategy 59](#_Toc17907623)

[4.8.6. Test schedule 60](#_Toc17907624)

[4.8.7. Unit Test cases 61](#_Toc17907625)

[4.8.8. Acceptance Test case 61](#_Toc17907626)

[4.8.9. Test Report 62](#_Toc17907627)

[5.1. Risk or limitation of system 65](#_Toc17907629)

[5.2. Strategies or solution for reducing limitations 65](#_Toc17907630)

[7.1. Installation Guide 68](#_Toc17907633)

[7.1.1. Android (Operating System) 68](#_Toc17907634)

[7.1.2. System Requirements 68](#_Toc17907635)

[7.1.3. Preparing for Installation 68](#_Toc17907636)

[7.2. User’s Guide 68](#_Toc17907637)

[7.3. Other 73](#_Toc17907638)

[References 73](#_Toc17907639)

Part 1

Introduction

## Definitions and Acronyms

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
| Dev | Developer |
| Dps | Degree Per Second |
| STF | Smart Traffic |
| G | Acceleration of Gravity |
| GSM | Global System for Mobile |
| PC | Personal Computer |
| PCB | Printed circuit board |
| PM | Project Manager |
| QA | Quality Assurance |
| QC | Quality Control |
| SMS | Short Message Service |
| SRS | Software Requirement Specification |
| STOT | Sum of total accelerations |
| SOP | Shocking point |

## ABSTRACT

One of the major problems in developing countries maintenance of roads. Well maintained roads contribute a major portion to the country’s economy. Identiﬁcation of pavement distress such as potholes and humps not only helps drivers avoid accidents or vehicle damages but also helps authorities to maintain roads. This project discusses previous pothole detection methods that have been developed and proposes a cost-effective solution to identify the potholes and humps on roads and provide timely alerts to drivers to avoid accidents or vehicle damages. We develop an android application to alert drivers if there is a bad road ahead by building 2 software. It includes 1 software to collect data with cassava acceleration sensor on the lack of android data that will be stored and processed on the server before sending the scene to users.

# Introduction

## Problem and motivation

According to statistics, the bad road surface is the cause of 1/3 of traffic accidents every year in Europe. In the UK, potholes cause a loss of £ 432 ($ 537) for every 17 minutes. In 2011, up to 20 million ‘potholes’ were recorded across Europe [2]. In Vietnam, we can't even figure that out. At present, there are many sections of roads from highways, national highways, provincial roads, and inter-village roads.



Figure 1. Highway surface Da Nang- Quang Ngai [1]

## Related Work

Potholes and Humps on Roads detection is an intriguing topic of research and researchers have been working on pothole detection techniques. This section gives a brief description of the existing solutions for detecting potholes and humps on roads.

Strutu et al. [2] have proposed a method for detecting defects on the road surface using accelerometers. It also makes use of the GPS systems to identify the exact location of the defects. Pothole detection algorithm runs on a mobile platform (moving vehicles), which is installed with an accelerometer, GPS, local computer and a wireless router. The sensed data is communicated to the central database using primary access points and secondary access points which can be used for future processing. However, installing a wireless router and local computer on all mobile platforms and setting up access points turns out to be quite expensive.

Murthy and Varaprasad et al. [3], have proposed a system that detects potholes based on a vision-based approach. The pictures of the road surface are captured using a properly mounted camera. The images are then processed using MATLAB to detect the occurrence of potholes. It is a 2D vision-based solution and works only under uniform lighting conditions and also the system does not involve any kind of warning system. The above solutions are limited only to the identiﬁcation of a pothole. These solutions do not provide any aid to the driver to avoid accidents due to potholes and humps.

Rode et al. [4], have proposed a system in which, Wi-Fi equipped vehicles collect information about the road surface and pass it to the Wi-Fi access point. The access point then broadcasts this information to other vehicles in the vicinity in the form of warnings. However, the system turns out to be an expensive one as all vehicles should be installed with Wi-Fi stations and number of access points have to be set up.

Hegde et al. [5], have proposed an intelligent transport system to detect potholes. It makes use of ultrasonic sensors to detect the presence of potholes. This system also sends warning messages to all the vehicles in the range of 100 meters using the Zigbee module. However, the system provides warnings after detecting the potholes which do not effectively help drivers to avoid potential accidents.

## Project objectives

The idea of the project is to build a driver assistance system and improve driver safety. With this system, not only it is the bad road warning function, but in the future, there may be other functions such as black traffic point warning, stuck in traffic warning, etc.

Because of the high workload and complexity of developing the device, a 4-man team may need about least 6 months to complete a commercial product of Smart Traffic, including other functions to maximize usability.

Therefore, in the scope of this Capstone Project, we aim to build a prototype device accelerometer, which could provide 3 basic functions:

• Complete system structure

• Automatically identify bad roads

• Complete user interaction software

The main purpose of this phase is to study, measure and evaluate possible methods to find an optimal solution for the system. The project focuses on software development for cars. Data were collected on the highway, which is only for cars. Interactive software users are preferentially incorporated in the car. Many aspects and functions that can be optimized for environmental or commercial purposes may not be included due to short development time. The device will be tested in laboratory that is different from actual environment.

Part 2

Project Management

# Project Management

## Project Organization

### Software Process Model

The STF is a project which requires studying, implementing and measuring an existing algorithm to optimize it for our purposes. In the progress, we intend to divide the project into 2 cycles of designing, implementing, testing and measuring the usability. Therefore, it is suitable with Interactive Software Process.



Figure 2 Interactive Software Model (Wikipedia)

Using this process provides the following advantages:

* Allow us to divide project into cycles
* Easy to change requirement and modify design
* Suitable for inexperience team

### Organization Structure

Figure 3 Organization Structure

### Roles and Responsibility

The table below shows the roles and responsibility of each team member.

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Role | Responsibilities |
| 1 | Vu Van Cuong | PM, Developer | Manage project plan  Delegate task, manage schedule and deadlines  Manage communication, report to the instructor  Manage code  Study solutions  Implement shocking detecting algorithm  Test and measuring accuracy of algorithm  Code server  Fix bugs |
| 2 | Le Hong Dung | QA, Developer | Study solutions  Code Road’s Data Collection application  Fix bugs  Control document quality  Test and measuring accuracy of algorithm  Create document |
| 3 | Lam Hai Vu | Developer, Tester | Study solutions  Code Smart Trafficapplication  Fix bugs  Create documents  Review document  Test and measuring accuracy of algorithm |
| 4 | Nguyen Duy Tri | Developer, Tester | Study solutions  Fix bugs  Create document  Collect data |

## Project management plan

### Project Milestones and Deliverables

|  |  |  |  |
| --- | --- | --- | --- |
| No | Stage | Deliverables | Committed |
| 1 | Initiating | Report 1: Project Overview | 11-May-2019 |
| 2 | Planning | Report 2: Project Management Plan | 20-May-2019 |
| 3 | Executing | Report 3 : SRS | 06-June-2019 |
| 4 | Executing | Select technology and offer solutions for server and client | 12-June-2019 |
| 5 | Executing | Build and improve the system | 22-June-2019 |
| 6 | Executing | Shock detection algorithm evaluation report | 15-July-2019 |
| 7 | Executing | Complete the entire system | 06-August-2019 |
| 8 | Executing | Report 4,5: Implementation and Conclusion | 09- August -2019 |
| 9 | Closing | Report 6: Summary and Final Report | 15- August -2019 |

### 

### Boundaries of the project

The system under development of this Capstone Project will include:

|  |  |  |
| --- | --- | --- |
| No | Item | Description |
| 1 | Documents | Final report includes SRS, Project Management plan, algorithm study report, implementation documents, test documents. |
| 2 | Source code | Source codes in Java, NodeJs and Matlab |

Table 1 Boundary of project

### Project Schedule

The full plan of STF project is attached in the same folder with this document.



Table 2 Project Schedule

## Quality Management

### Quality control activities

|  |  |  |
| --- | --- | --- |
| Deliverables | Quality control activity | Quality Standard |
| Report 1: Project Overview | Group review | Document standard |
| Report 2: Project Management Plan | Group review | Document standard |
| Report 3 : SRS | Group review | Document standard |
| Select technology and offer solutions for server and client | Group review | Document standard |
| Build and improve the system | Unit test | 100% passed |
| Shock detection algorithm evaluation report | Group review | Document standard |
| Complete the entire system | Unit test, Acceptance test | 100% Unit test cases are passed  90% Acceptance test cases are passed |
| Report 4,5: Implementation and Conclusion | Group review | Document standard |
| Report 6: Summary and Final Report | Group review | Document standard |

Table 3 Quality Control Activities

### Document Standard

* Top margin: 2.5 cm, bottom margin: 2.5 cm, left margin 3.5 cm, right margin 2 cm
* Text font: Calibri 12
* Content is reviewed carefully and based on outline of the university

## Communication Management

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Communication | Methods | Schedule | Information | Participants |
| Internal Communication | | | | |
| Project Meetings | Direct Meeting | Daily | Project status, problems, risks, changed requirements | PM, Project Team |
| Sharing of project data | Sharing directly or via mail FPT University | When available | All project documentation, development tools and reports | Project Manager Project Team Members |
| Milestone Meetings | Direct Meeting or via email | Before milestones releasing date | Project progress | PM, Project Team Members, Instructor |
| Final Project Meeting | Direct Meeting | End of Project | Wrap-up  Experiences | PM, Project Team |
| External Communication and Reporting | | | | |
| Project Report | MS Word, PMP file | Follow Schedule | Report No1 to No6 | Project Team Members |
| Weekly Report | MS Word | Weekly | Project progress | PM, Instructor |

Table 4 QC Activities

## Risk Management

### Risks

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Risk Description | Probability | Effect | Status | Action |
| 1 | Technical skill:  Team member lack of experience in matlab, nodejs | Very high | Serious | Occurred | Extend server build time to choose the right technology, review carefully. |
| 2 | Lack of verifiable sample data may affect the ability of the primary external stakeholder to validate end product. | High | Serious | Potential | Spend more time collecting standard data. |
| 3 | Project Management skill:  Project Manager lacks experience in project management. | Moderate | Serious | Occurred | Maintain communication carefully among members. |
| 4 | Human resource:  One of team members may have many days off during developing times. | Moderate | Tolerable | Potential | Given due date clearly. Review and criticize if overdue. |
| 5 | Algorithm:  It may be hard to implement algorithm | Moderate | Serious | Occurred | Carefully compare many algorithms. Review carefully and thanks to the advice of teachers. |
| 6 | Data collection is difficult in terms of time and transportation. | High | Serious | Potential | Borrow your family car, and try to do data collection as quickly as possible. |

Table 5 Risk management

### Risk control activities

* Validating mitigation strategies and alternatives.
* Closing settled risks and identify new risks.
* Taking corrective action when actual events occur
* Assessing impact on the project of actions taken (money, time, resource).
* Identifying new risks resulting from risk mitigation action.
* Ensuring the Project Plan (including the Risk Management Plan) is maintained.
* Ensuring change control addresses risks associated with the propose change.
* Revising the Risk Assessment Questionnaire and other risks management documents to capture results of mitigation actions.
* Revising Risk Response Plan.
* Communications.

Part 3  
Requirement



## User Requirement Specification

Customer (the person using application) requirement:

* Easy to use the application
* Choose destination on map.
* Search for routes to destination.
* View the location of vehicle in map.
* Warning alert by sound and message before passing the shock point.
* Show all shock points in the current road

Administrator (the person manage roads and shock points) requirement:

* Login with administrator account.
* Show, add, delete, update road.
* Show, add and delete shock points.

## System Requirement Specification (Specific Requirements)

Client application requirement:

* Send data about the road to server.
* Receive data about the shock points on current road from server.
* Save data from accelerometer sensor.
* Store data of shock points, send data to server at the end of the journey.
* Process data from server to detect upcoming shock points in order to alert user.

Server requirement:

* Get data of the road from database and send to client application.
* Receive data from client application, process and update to database.

### External Interface Requirements

##### User Interfaces

An easy to use graphic mobile interface. System uses English and Vietnamese language. All function buttons and content must display at appropriate position. Messages will show at a good-looking location combined with notification sound.

##### Hardware Interfaces

The system hardware require:

* A computer connected to the internet to monitor the system.
* A mobile phone with built-in accelerometer sensor and GPS receiver.

##### Software Interfaces

The system software require:

* A computer running Linux OS with MySQL and Matlab installed
* Require an Android mobile version 4.2 or higher.

##### Communications Protocol

System uses Web-Socket protocol to communicate between client and server through socket.io. Besides, client also uses http request and GPS signal to get location data from Google Maps API.

### System Features

##### System Feature Find Way

###### Use Case Diagram

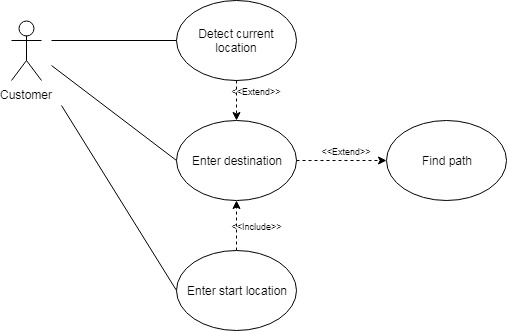


Figure 4 Use cases of direction function

###### Use Case Specification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE 2 SPECIFICATION** | | | | |
| **Use-case No.** | UC001 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Find way | | | |
| **Author** | DungLH | | | |
| **Date** | 05/06/2019 | **Priority** | Medium | |
| **Actor:**  *System and user*  **Summary:**  *User can use this feature to search road and find direction*  **Goal:**  *Provide a good-looking and easy to use display of roads and navigation*  **Triggers:**  *User want to find direction and calculate distance*  **Preconditions:**  *User have logged in the application*  *GPS signal have been turned on*  **Post Conditions:**  *Find and display the most suitable way on the screen*  **Main Success Scenario:**   * *Step 1: System display main screen.* * *Step 2: User enter start location.* * *Step 3: System display suggestions and move the maps to the chosen location.* * *Step 4: User enter destination.* * *Step 5: System display suggestions and move the maps to the chosen destination.* * *Step 6: User click find path button.* * *Step 7: System analysis and display the best fit way*   **Alternative Scenario:**   * *If user don’t enter start location, system will use GPS to indicate start location.*   **Exceptions:**   * *Google API not working* * *GPS signal not working*   **Relationships:**  *This use case is not necessary for all other use cases*    **Business Rules:**  *N/A* | | | | |

##### System Feature Collect Data

###### Login Use Case Diagram

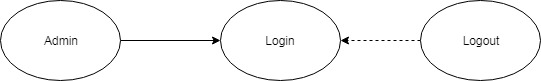


Figure 5 Use cases of administrator sign in function

###### Login Use Case Specification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE 4 SPECIFICATION** | | | | |
| **Use-case No.** | <UC002> | **Use-case Version** | | <1.0> |
| **Use-case Name** | Use cases of administrator sign in function | | | |
| **Author** | VuLH | | | |
| **Date** | 06/05/2019 | **Priority** | <High> | |
| **Actor:** *System and user*  **Summary:** *The Administrator manage all users in system*  **Goal:**  *Admin log in success to manage all functions in the system*  **Triggers:**  *Input email and password correctly in the right textbox then click*  *“Sign in” button*  **Preconditions:**  *The Administrator have logged in the application*  **Post Conditions:**  *Admin have permission to use every function in admin application*  **Main Success Scenario:**   * *Step 1: System display login screen* * *Step 2: Admin enter username and password* * *Step 3: Click on “Sign in” button* * *Step 4: System display main screen with logged in account*   **Alternative Scenario:**   * *At step 2, if user leave username blank or password is too short (<5 character):*   + *Step 2.1: System disable “Sign in” button* * *At step 3, if user enter wrong username or password:* * *Step 3.1: System display alert wrong username and password message*   **Exceptions:**  *Server is not working*  **Relationships:**  *This use case is necessary for all other features*  **Business Rules:** *N/A* | | | | |

###### Manage Road Use Case Administrator Diagram

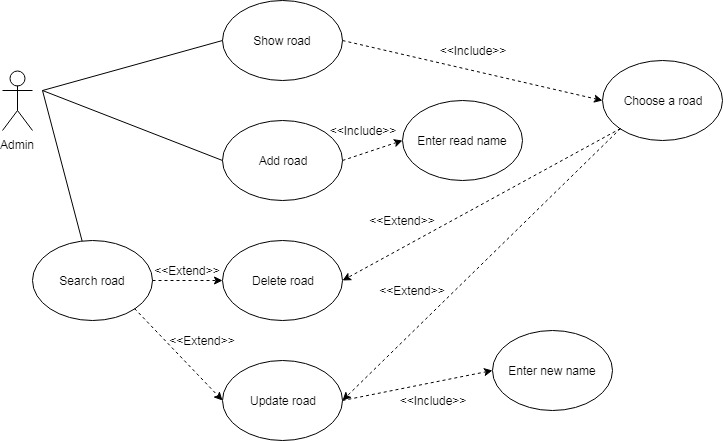


Figure 6 Use cases of road management function

###### Manage Road Use Case Specification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE 3 SPECIFICATION** | | | | |
| **Use-case No.** | <UC003> | **Use-case Version** | | <1.0> |
| **Use-case Name** | Show road | | | |
| **Author** | VuLH | | | |
| **Date** | 06/05/2019 | **Priority** | <High> | |
| **Actor:** *System and user*  **Summary:** *Administrator manage roads in system*  **Goal:** *Get list of all roads in system data*  **Triggers:** *User go to management screen and click “Road List” button*  **Preconditions:** *Administrator have logged in the application*  **Post Conditions:** *Administrator can excuse delete and edit road feature*  **Main Success Scenario:**   * *Step 1: User move to manage screen* * *Step 2: User click on “Road List” button* * *Step 3: System display all roads exits*   **Alternative Scenario:**   * *At step 2, if server contains no road:*   + *Step 2.1: System show message no road available.*   **Exceptions:** *Server is not working*  **Relationships:** *This use case is necessary for all other use cases*  **Business Rules:**  *N/A* | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE 4 SPECIFICATION** | | | | |
| **Use-case No.** | <UC004> | **Use-case Version** | | <1.0> |
| **Use-case Name** | Edit road | | | |
| **Author** | VuLH | | | |
| **Date** | 06/05/2019 | **Priority** | <High> | |
| **Actor:** *System and user*  **Summary:** *The Administrator manage roads in system*  **Goal:** *Change road name of an existing road in system data*  **Triggers:** *User go to management screen and click “Road Tools” button*  **Preconditions:** *The Administrator have chosen the road to edit*  **Post Conditions:** *N/A*  **Main Success Scenario:**   * *Step 1: User choose a road in road list* * *Step 2: User click on “Road Tools” button* * *Step 3: System display information of current road in a text box* * *Step 4: User click on “Edit Name” button* * *Step 5: System allow user to edit road name in the text box* * *Step 6: User enter new name for the road* * *Step 7: User click “Save” button* * *Step 8: System display message change road success*   **Alternative Scenario:**   * *At step 8, if user enter duplicate name with an exist road or leave it blank:*   + *Step 8.1: System show message cannot update road name.*   **Exceptions:** *Server is not working*  **Relationships:** *This use case is not necessary for all other use cases*  **Business Rules:**  *N/A* | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE 5 SPECIFICATION** | | | | |
| **Use-case No.** | <UC005> | **Use-case Version** | | <1.0> |
| **Use-case Name** | Delete road | | | |
| **Author** | VuLH | | | |
| **Date** | 06/05/2019 | **Priority** | <High> | |
| **Actor:** *System and user*  **Summary:** *The Administrator manage roads in system*  **Goal:** *Delete an existing road in system data*  **Triggers:** *User go to management screen and click “Road Tools” button*  **Preconditions:** *The Administrator have chosen the road to delete*  **Post Conditions:** *N/A*  **Main Success Scenario:**   * *Step 1: User choose a road in road list* * *Step 2: User click on “Road Tools” button* * *Step 3: System display information of current road in a text box* * *Step 4: User click on “Delete Road” button* * *Step 5: System display message delete road success*   **Alternative Scenario:** *N/A*  **Exceptions:** *Server is not working*  **Relationships:** *This use case is not necessary for all other use cases*  **Business Rules:**  *N/A* | | | | |

###### Shock Point Management Use Case Diagram

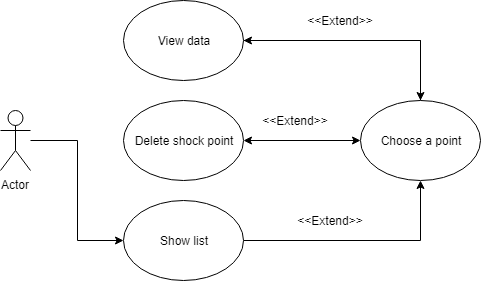


Figure 7 Use cases of shock point management function

###### Shock Point Management Use Case Specification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE 6 SPECIFICATION** | | | | |
| **Use-case No.** | <UC006> | **Use-case Version** | | <1.0> |
| **Use-case Name** | Show list of shock points in the road | | | |
| **Author** | VuLH | | | |
| **Date** | 06/05/2019 | **Priority** | <High> | |
| **Actor:** *System and user*  **Summary:** *The Administrator view list of all shocking points in the chosen road*  **Goal:** *Get list of shock points available in the chosen road*  **Triggers:** *Administrator go to management screen and choose a road*  **Preconditions:** *The Administrator have logged in the application*  **Post Conditions:** *Admin have permission to shock points*  **Main Success Scenario:**   * *Step 1: User choose a road in road list* * *Step 2: System display all shock points available in the road*   **Alternative Scenario:**   * *At step 2, if the road don’t have any shock point:*   + *Step 2.1: System show message no shock point available*   **Exceptions:**  *Server is not working*  **Relationships:**  *This use case is necessary for delete shock point use cases*  **Business Rules:** *N/A* | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE 7 SPECIFICATION** | | | | |
| **Use-case No.** | <UC007> | **Use-case Version** | | <1.0> |
| **Use-case Name** | Delete shock points in the road | | | |
| **Author** | VuLH | | | |
| **Date** | 06/05/2019 | **Priority** | <High> | |
| **Actor:** *System and user*  **Summary:** *The Administrator delete a shock point*  **Goal:** *The selected shock point is deleted in the system*  **Triggers:** *Administrator go to management screen and choose a road*  **Preconditions:** *Administrator have logged in the application*  **Post Conditions:** *N/A*  **Main Success Scenario:**   * *Step 1: User click on a shock point icon displayed in the map* * *Step 2: System display information of the selected shock point* * *Step 3: User click on “Delete” button* * *Step 4: System display message delete shock point success*   **Alternative Scenario: N/A**  **Exceptions:**  *Server is not working*  **Relationships:**  *This use case is not necessary for other use cases*  **Business Rules:** *N/A* | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE 8 SPECIFICATION** | | | | |
| **Use-case No.** | <UC008> | **Use-case Version** | | <1.0> |
| **Use-case Name** | View data of shock points in the road | | | |
| **Author** | VuLH | | | |
| **Date** | 06/05/2019 | **Priority** | <High> | |
| **Actor:** *System and user*  **Summary:** *Administrator view data of a shock point*  **Goal:** *Data of the selected shock point is displayed*  **Triggers:** *Administrator go to management screen and choose a road*  **Preconditions:** *Administrator have logged in the application*  **Post Conditions:** *N/A*  **Main Success Scenario:**   * *Step 1: User click on a shock point icon displayed in the map* * *Step 2: System display information of the selected shock point* * *Step 3: User click on “View Data” button* * *Step 4: System display accelerometer data of the selected shock point*   **Alternative Scenario: N/A**  **Exceptions:**  *Server is not working*  **Relationships:**  *This use case is not necessary for other use cases*  **Business Rules:** *N/A* | | | | |

Part 4  
Implementation



## Proposed system architecture

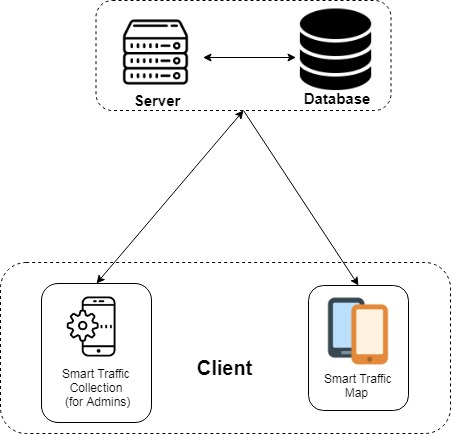


Figure 8 Design Overview diagram

## Analysis and selection of tools, devices

After researching the possibilities of components for this project, the team has decided to use the below components to make the Smart Traffic System:

* Server: base on Node JS and working in Google Cloud Platform.
* Database: base on MySQL server and storage in Google Cloud Platform.
* Smart Traffic Collection application: base on Android. The application need to access internet, location, GPS and accelerometer sensor permission.
* Smart Traffic Map application: base on Android and need to access internet, location, GPS permission.

### Server

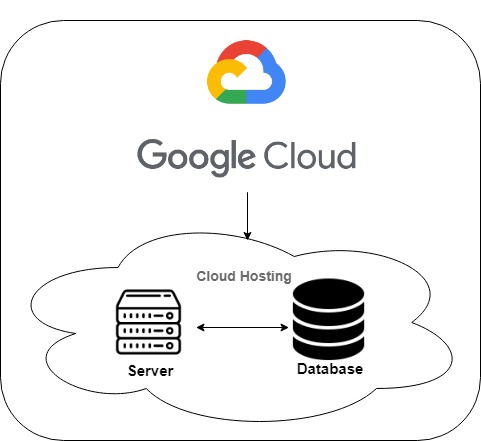


Figure 9 Server Overview diagram

Server has the missions for implementation as below:

* Receive the data from Smart Traffic Collection application.
* Storage data that be receive to database.
* Implement detection shock point
* Storage the information of accounts.

### Database

Database storage all data from client and server.

\*[Refer to 4.6 of content design.](#_Database_design_1)

### Smart Traffic Road Collection application

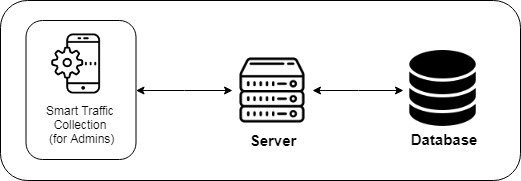


Figure 10 Smart Traffic Road Collection application overview structure diagram

Smart Traffic Collection application has the missions for implementation as below:

* Allow sign in with administrator account.
* Collection data of journey with accelerometer sensor on device.
* Show graph of accelerometer sensor.
* Manage shock points with map interface.

### Smart Traffic Map application

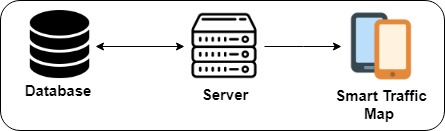


Figure 11 : Smart Traffic Map application overview structure diagram

Smart Traffic Map application has the missions for implementation as below:

* Alert user when their ahead are a shock point. Alert is shown before points from three to five second.
* Find the direction with the start and destination.
* Auto show all shock points on the road that user drive on that road.
* User can pin marker. After that can get direction from or to that marker.
* Show all shock points, where that marker is pinned.

## Introduction/Resume a theory problem

### Use case of Smart Traffic Map application

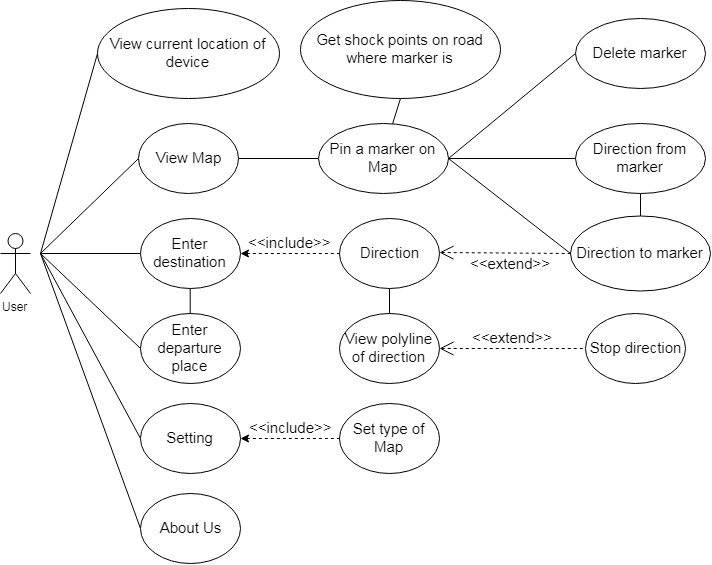


Figure 12 Use case of Smart Traffic Map application

## Algorithm and Solution Analysis

### Overview existing algorithms and solutions

##### Shocking Detection using tri-axial accelerometer.

The device will receive data from accelerometer as Ax, Ay, Az which are the acceleration of the x, y, z axis respectively.

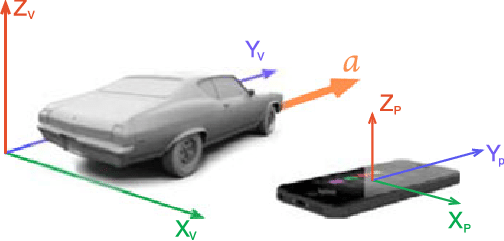


Figure 13 Illustration of the vehicle's coordinate system and the smartphone's coordinate system [6]

When a vehicle moves on the road there are 5 basic states. It is acceleration, deceleration, lane change, shocking and normal status.

The Total Sum of Acceleration Vector (STOT):

When the vehicle moves on the road, in normal state, STOT will change with small amplitude. When speeding, deceleration, lane change or shocking are performed, the STOT change amplitude will change and the amplitude depends on the status. Based on the amplitude, we can determine the threshold to change the amplitude of the shock point.

This is a simple method to implement, however, to determine the threshold of the shocking point depends on the sensitivity of the accelerometer sensor of each device, each car will have a different amplitude of STOT.

##### Pattern detect using HMM/SVM (Machine learning)

Another method is to use pattern recognition with HMM / SVM.

The process can be divided into two main processes:

• Training

* Carry out the collection of multiple data samples of many vehicles in shock and no shock state
* Then, they will be vectorized using the Hidden Markov Model (HMM) and the Supporting Vector Machine (SVM).
* These samples will be trained for the machine. Samples will be divided into two categories: positive and negative.

• Recognition

* Any subsequent data from the sensor will be vectorized and calculated to match these two classes.
* From the match rate, the program will deduct whether this action is positive or negative.

This method requires a good set of samples, and is large enough both positive and negative, so it is very demanding in collecting samples and samples from many different vehicles in different cases. Therefore using this method is quite complicated for a student project.

##### Conclusion

In conclusion, we chose to use the first shock detection method using tri-axial accelerometer.

### Implementation algorithm in our system

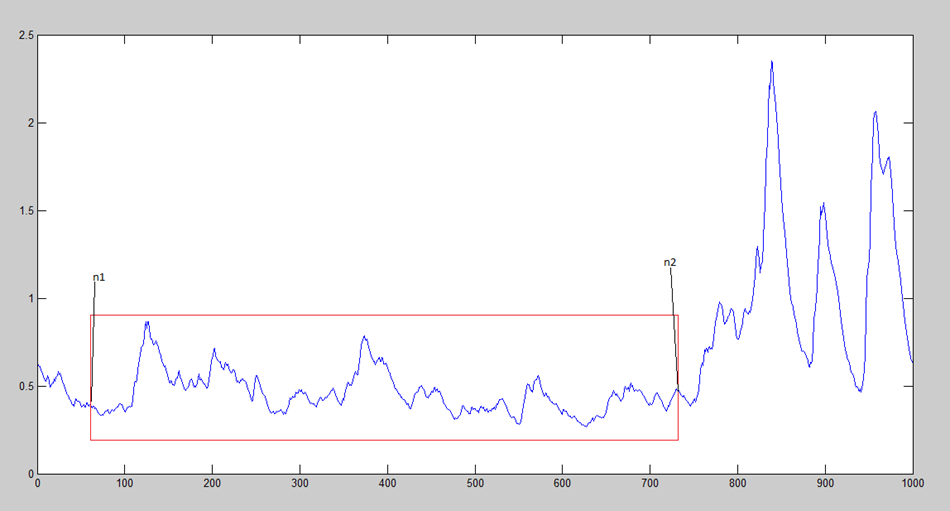
As stated above, we will implement the shocking detecting method 1 (4.7.1.1) in our system.

This algorithm uses only data from Accelerometer. This algorithm only uses data from Accelerometer. As mentioned above, this algorithm depends on the sensitivity of the accelerometer sensor on the machine and the amplitude of STOT will depend on the vehicle. To solve the above problem. We use two values: VASU (Vehicle acceleration sensor unit) and AAV (Average acceleration value).

* VASU is the average value of STOT when the vehicle is running in a normal state on a sampling unit.

Figure 14 Range data from n1 to n2 example

n1 to n2 is the data range when the vehicle is running in normal state.



Each vehicle, each speed range will have a unique VASU value.

* AAV is the value of STOT divided by VASU

We will use this value to find the threshold of the shocking point. Since this is a dimensionless value, it solves the sensitivity of each device and the smoothness of each vehicle.

The signals x, y, z and AVV of the above states are shown as follows:

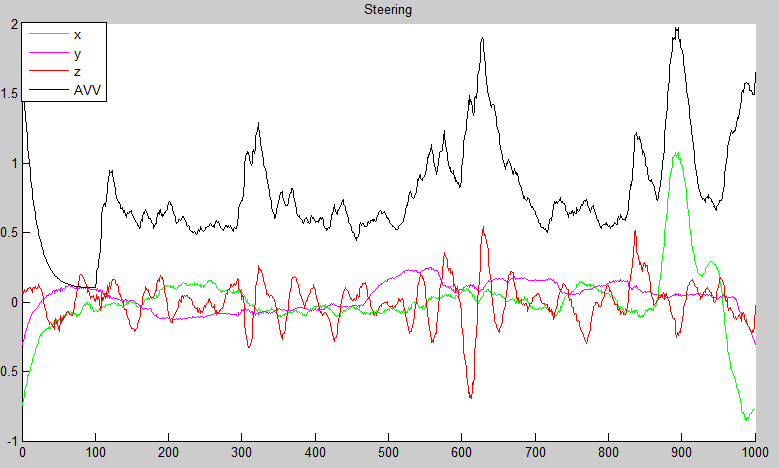


Figure 15 Distribution of x,y,z and AVV in lane change event

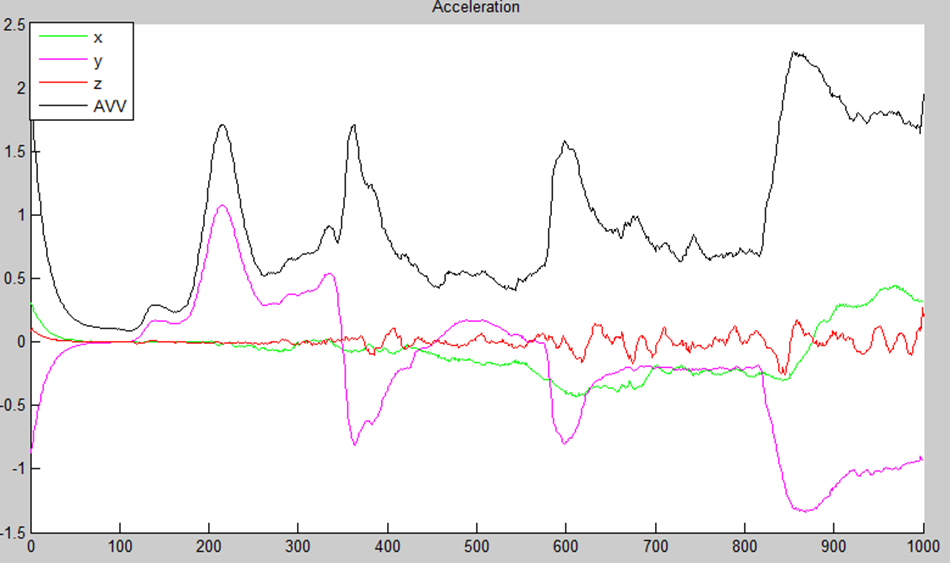


Figure 16 Distribution of x,y,z and AVV in acceleration event

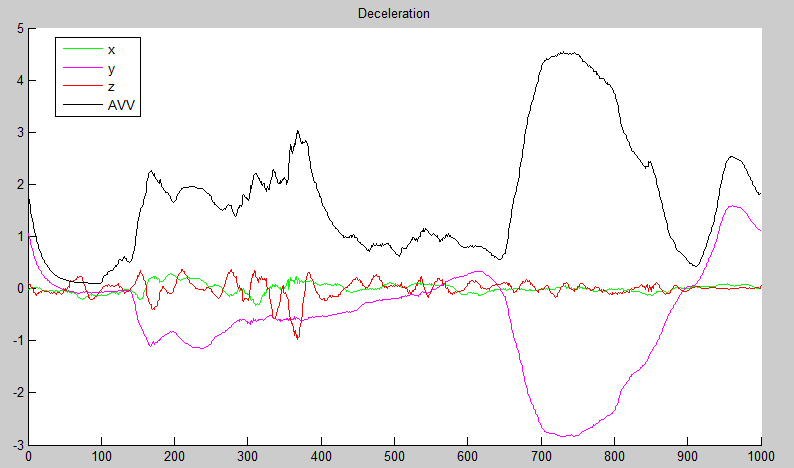


Figure 17 Distribution of x,y,z and AVV in deceleration event

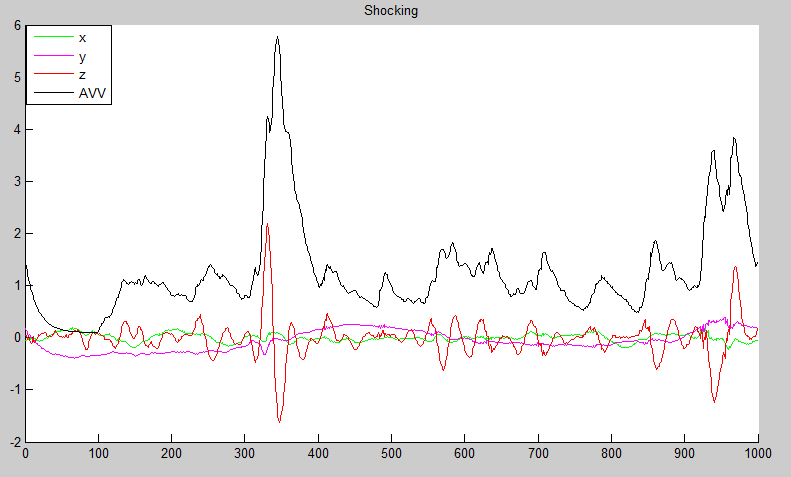


Figure 18 Distribution of x,y,z and AVV in shocking event

After taking sample data of states then perform low pass filter at frequency of 5 hz. Compare the x, y, z, AAV data of each status and perform a review of more than 40 samples. We found that the car was in a slump when the value of AVV exceeded 5.5.

## Interface design

### Smart Traffic Map interface design

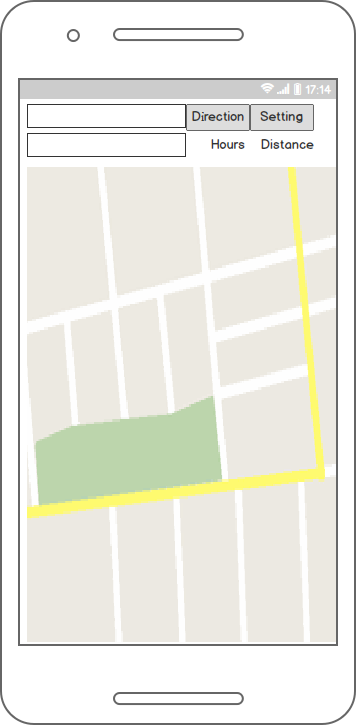


Figure 16 Home screen design

* Edit text : User enter place to start and destination
* Direction button: start direction
* Setting button: go to setting screen
* Fragment: show the map

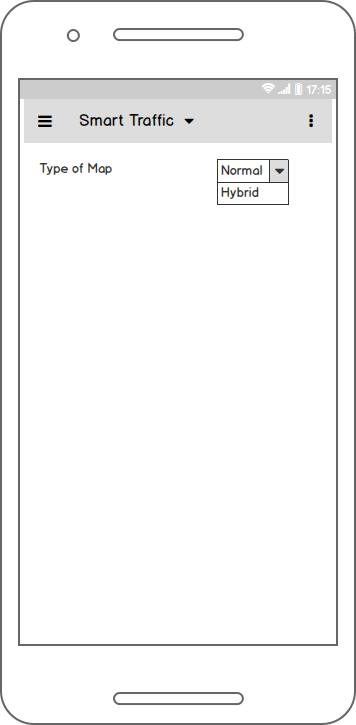


Figure 17 Setting map screen design

* Type of map: User can choose a type of Map

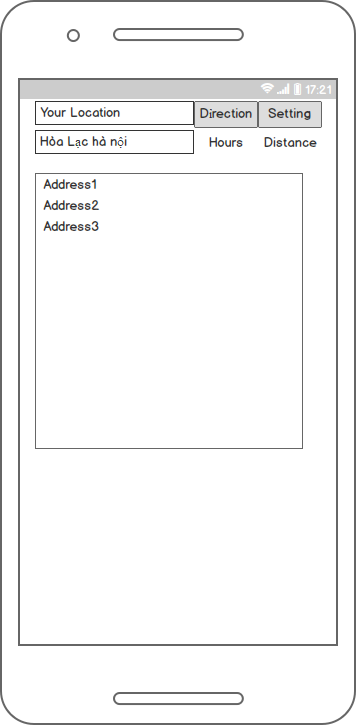


Figure 18 Search list of places screen design

* Enter to edit text: When user press key to search a place, a list of suggestion places is shown. User can choose one item of list. That item auto fill to edit text.

### Smart Traffic Road Collection interface design

##### Login Screen

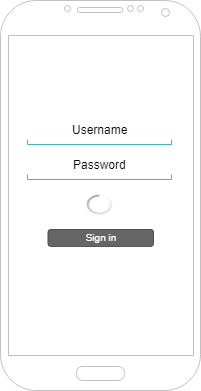
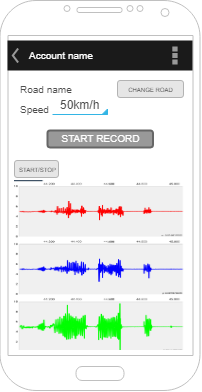


Figure 19 Login screen design

* Username edit text: user enter username
* Password edit text: user enter password
* Progress bar: show when user click log in
* Sign in button: user click to login

##### Main display



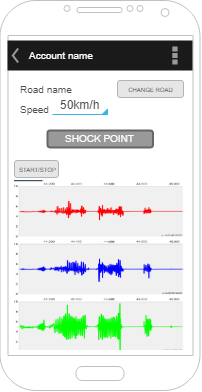


Figure 20 Manual and auto mode shock point detection

* Account name text field: system show name of user logged in.
* Menu selection: user use other functions.
* Road name text field: system show name of current road.
* Speed spinner: user choose the speed.
* Start record button: start record session.
* Stop record button: stop record session.
* Shock point button: user detect shock point manually.
* Start/stop button: start/stop sensor data
* Graph display: show accelerometer data change in graph view

##### Manage road display

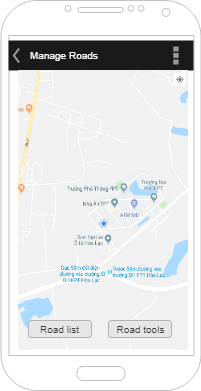


Figure 21 Manage roads screen display

* Action bar tittle: show name of the chosen road.
* Map fragment: show maps.
* Road list button: show list of roads available in server.
* Road tools button: user can edit/delete the chosen road.

## Database design

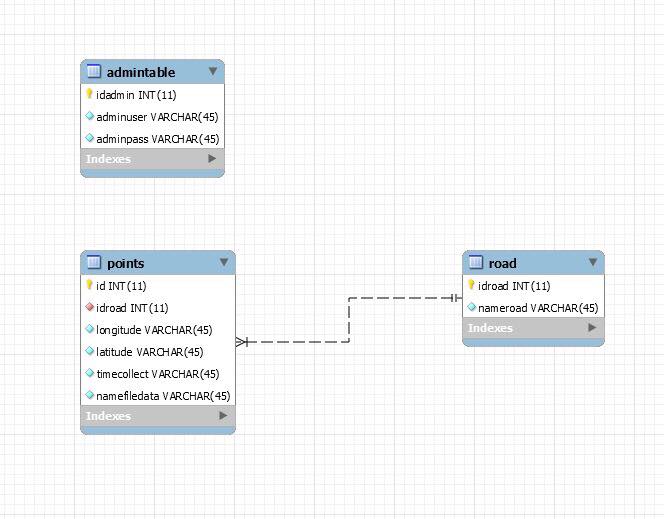


Figure 22 Database diagram

### List of tables

|  |  |  |
| --- | --- | --- |
| No. | Table Name | Description |
|  | admintable | Store username and password of admin, who can collect data and manage roads, manage shock points. |
|  | road | Store road |
|  | Points | Store details information about a shock point |

### Table admin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Attribute | Type | Constraints | Description |
|  | idadmin | Integer | PK, not null | Identify number of admin |
|  | adminuser | String | Not null | Login username of admin |
|  | adminpass | String | Not null | Login password of admin |

### Table road

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Attribute | Type | Constraints | Description |
|  | idroad | Integer | PK, not null | Identify number of road |
|  | nameroad | String | Not null | Road name |

### Table Points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Attribute | Type | Constraints | Description |
|  | id | Integer | PK, not null | Identify number of shock point |
|  | idroad | Integer | FK, not null | Identify number of road, reference to table Road(idroad) |
|  | longitude | String | Not null | Longitude value of shock point’s location |
|  | latitude | String | Not null | Latitude value of shock point’s location |
|  | timecollect | String | Not null | Time in seconds of collect data |
|  | namefiledata | String | Not null | Name of the data file, use to view or review data |

## Protocol design

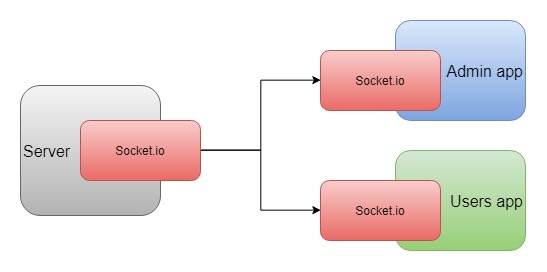


Figure 23 Smart Traffic protocol design

## Testing (Co-Design in Embedded System).

### Introduction

##### System Overview

The goal of this phase is to guarantee all the features of Smart Traffic software are working properly and minimize the defects when used.

##### Test Approach

This document serves as a test report for this project and to ensure project quality. It describes the testing scope, testing strategy, test plan to validate the quality of this product prior to release. It also provides test-cases which are used to cover all situations arising bugs.

### Test Plan

##### General Information

###### Project information

|  |  |
| --- | --- |
| Project short name | SM |
| Project full name | Smart Traffic |
| Start Date | 13-May-2019 |
| Duration (span days) | 108 |
| End date | 29-August-2019 |
| QA engineers | DungLH, CuongVV, TriND, VuLH |

###### Revision history

|  |  |  |
| --- | --- | --- |
| Date | Version | Description |
| 01-June-2019 | 1.0 | Create Test Plan |
| 17-August-2019 | 2.0 | Update Test Plan |

###### Table of content

|  |  |
| --- | --- |
| Sheet name | Status |
| General information | Complete |
| Testing Scope | Complete |
| Testing Level | Complete |
| Testing Tools and Environments | Complete |
| Test Strategy | Complete |
| Test Schedule | Complete |
| Acceptance test case | Complete |
| Test report | Complete |

##### Testing scope

The following features will be tested:

|  |  |  |
| --- | --- | --- |
| Module | Features | Description |
| Data management application | Login/Logout | User use username and password was supplied to use the application |
| Detect shock points manually | Admin can create a shock point manually |
| Detect shock points automatically | The detect algorithm is implemented |
| Send data to server | Send accelerometer and location data to server |
| View/review data in graph form | View accelerometer data in graph form |
| Manage shock points | Delete unavailable shock points |
| Manage road | Add, edit, delete a road |
| Shock point warning application | Show map | Show map |
| Display shock points | Display shock points available in map |
| Alert shock points | Alert when there is a shock point ahead |

### Testing level

The testing for the Smart Traffic system will focus on Unit test (Hardware and Software) and Acceptance test.

UNIT TEST will be performed by the developers only.

ACCEPTANCE TEST will be performed under user’s point of view with the assistance of the tester and developers. Program will enter Acceptance test after all the critical and major defects have been corrected.

### Testing Tools and Environments



Figure 24 Testing tools and environments

##### Hardware

|  |  |  |
| --- | --- | --- |
| No | Equipment | Description |
| 1 | Car | A car to go to road get data and test application |
| 2 | Smartphone | A smart phone installed Smart-Traffic application |

##### Software

|  |  |  |
| --- | --- | --- |
| No | Software | Description |
| 3 | Matlab | Data analyzing |
| 4 | Microsoft Windows 10 32/64 bits | Operating System |

### Test strategy

In this phase we focus on Acceptance test and Algorithm Unit test (which will be presented below).

Software Unit test will be executed by the developers.

|  |  |  |
| --- | --- | --- |
| Test method | Test cycle# | |
| 1 | 2 |
| Test level | | |
| Unit test | X |  |
| Acceptance test |  | X |
| Test approach | | |
| Blackbox testing | X | X |
| Test types | | |
| Functional testing | X | X |
| GUI testing |  | X |
| Testing tools | | |
| Matlab |  | X |
| Operating System Environment | | |
| Windows 10 Education 32bits |  | X |
| Windows 10 Education 64bits |  | X |
| Hardware Environment | | |
| Smart phone | X | X |

### Test schedule

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Functionality | Tester | Status | Planned Start date | Planned end date |
|  | Create test plan | DungLH | Finish | 01/06/2019 | 02/06/2019 |
|  | Review test plan | CuongVV | Finish | 03/06/2019 | 03/06/2019 |
|  | Update test plan | DungLH | Finish | 04/06/2019 | 04/06/2019 |
|  | Create unit test cases | DungLH | Finish | 15/06/2019 | 16/06/2019 |
|  | Execute unit test cases | TriND | Finish | 20/06/2019 | 21/06/2019 |
|  | Review test cases | DungLH | Finish | 22/06/2019 | 22/06/2019 |
|  | Update test cases | DungLH | Finish | 22/06/2019 | 23/06/2019 |
|  | Experimental result | VuLH | Finish | 29/07/2019 | 29/07/2019 |
|  | Create acceptance test cases | DungLH | Finish | 01/08/2019 | 02/08/2019 |
|  | Execute acceptance test cases | TriND | Finish | 09/08/2019 | 09/08/2019 |
|  | Review test cases | DungLH | Finish | 10/08/2019 | 10/08/2019 |
|  | Fix bugs | DungLH, TriND, CuongVV, VuLH | Finish | 11/08/2019 | 20/08/2019 |
|  | Write test report | DungLH | Finish | 21/08/2019 | 21/08/2019 |

### Unit Test cases

### Acceptance Test case

##### Admin Application

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Test Case Description | Test Procedure | Expected Outputs |
|  | Test login | Enter valid username and password and click login | Show message login success and go to main screen |
|  | Test login | Enter invalid username and password and click login | Show message login fail and not go to main screen |
|  | Test detect shock points manually | Click “Shock point” button in main screen manage application | Show message add shock point success, create a file save all information of the shock point |
|  | Test detect shock points automatically | Put the smartphone in a fixed position. Drive the car passing a shock point | Show message add shock point success, create a file save all information of the shock point |
|  | Test save data to server | Create some shock points and click “Save” button | All shock points created is saved to server |
|  | Test view/review data | Click “Open” button and choose a shock point file to view data | Accelerometer data is displayed accurately in graph view |
|  | Test delete shock points | Go to manage screen, choose an available point and click “Delete” button | Shock point is removed in maps and in server data |
|  | Test edit road name | Go to manage screen, choose an available road and change road name to a new valid road name | Road name changed in server data |
|  | Test edit road name | Go to manage screen, choose an available road and change road name to an invalid road name | Road name not changed |
|  | Test delete road | Go to manage screen, choose an available road and click “Delete” button | The road together with all shock points in that road is deleted in maps and server data |

##### Client Application

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Test Case Description | Test Procedure | Expected Outputs |
|  | Test show map with GPS turned on | Start application | A map is displayed, navigate to user’s location |
|  | Test show map with GPS turned off | Start application | A dialog displayed to alert user turn on GPS |
|  | Test display shock points | Choose a road have shock points in the map, and click “Get Shock Points” button | All shock points in the road is displayed in map. |
|  | Test display shock points | Drive the car to a road have shock points | All shock points in the road is displayed in map automatically |
|  | Test alert shock points | Drive the car coming to a shock point ahead in the maps | An alert message with sound notification is displayed. |
|  | Test alert shock points | Drive the car coming to a shock point in other side of the road | No alert message displayed. |

### Test Report

##### Admin application test result

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Test Case Description | Test Time | Passing percentage |
|  | Test login | 5 | 100% |
|  | Test login | 5 | 100% |
|  | Test detect shock points manually | 20 | 100% |
|  | Test detect shock points automatically | 20 | 100% |
|  | Test save data to server | 5 | 100% |
|  | Test view/review data | 10 | 100% |
|  | Test delete shock points | 10 | 100% |
|  | Test edit road name | 5 | 100% |
|  | Test edit road name | 5 | 100% |
|  | Test delete road | 5 | 100% |

##### Client application test result

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Test Case Description | Test Time | Passing percentage |
|  | Test show map with GPS turned on | 5 | 100% |
|  | Test show map with GPS turned off | 5 | 100% |
|  | Test display shock points | 20 | 100% |
|  | Test display shock points | 20 | 100% |
|  | Test alert shock points | 20 | 100% |
|  | Test alert shock points | 20 | 100% |

Part 5

Results



## Risk or limitation of system

The Smart Traffic system may have these following risks and limitations:

Limitations:

* Shocks point are not being updated usually. Its depends on data collectors
* Accelerometer sensor data is different in different phone. Therefore it is different to find the exact threshold to automatically detect shock points.
* Only be implemented in highways or one way street.
* The study of Smart Traffic systems is not implemented in IOS or non-Android smart phone.

Risks:

* Network error may cause the application not show alert message.
* GPS signal in some low-cost device sometimes not correct, make the application detect wrong road and therefore no alert message or alert wrong points.

## Strategies or solution for reducing limitations

In the next phase, the project can be improved as below:

* Merge 2 separate application into one. The clients using Smart Traffic application can detect and update shock-points.
* Using just one accelerometer and GPS type to implement inside the car.
* Improve detect road algorithm to make it easier to detect shock points in other road types.
* Implement in IOS smartphones to gain more users.

Part 6

Conclusions

The project was successful to develop a system of real-time easy-to-go Smart Traffic which can detect shock points ahead and alert to users whenever there is an incoming shock point.

The system is implement is implemented in Android phone without any hardware devices or accessories. Therefore, this system has high-mobility alongside with convenient and high applicability. In addition, the system is very easy to use, users just have to open the application and put the phone in an easy listening position and it will work. This project can reduce uncomfortable feelings when driving and make it safer.

Although the system has several limitations in usability, the detection is confirmed to be evidently effective. It is feasible to build a complete device in future.

Part 7

Software User’s Manual



## Installation Guide

### Android (Operating System)

Android is a mobile operating system developed by Google. It is based on a modified version of the Linux kernel and other open source software, and is designed primarily for touchscreen mobile devices such as smartphones and tablets.

### System Requirements

Android API 21+

### Preparing for Installation

Step 1: Go to CHPlay, search by name: “Smart Traffic Map”.

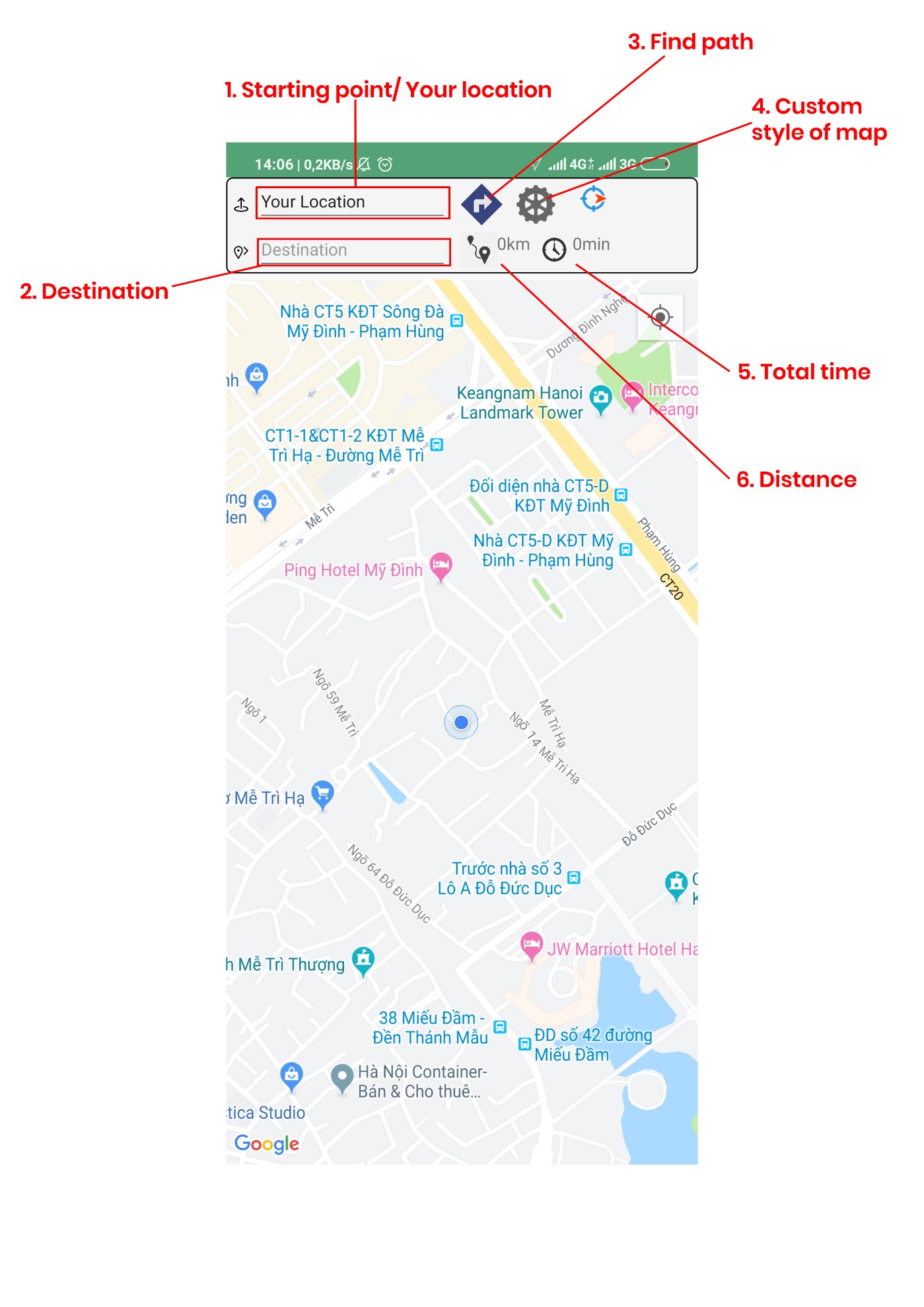
Step 2: Install the application.

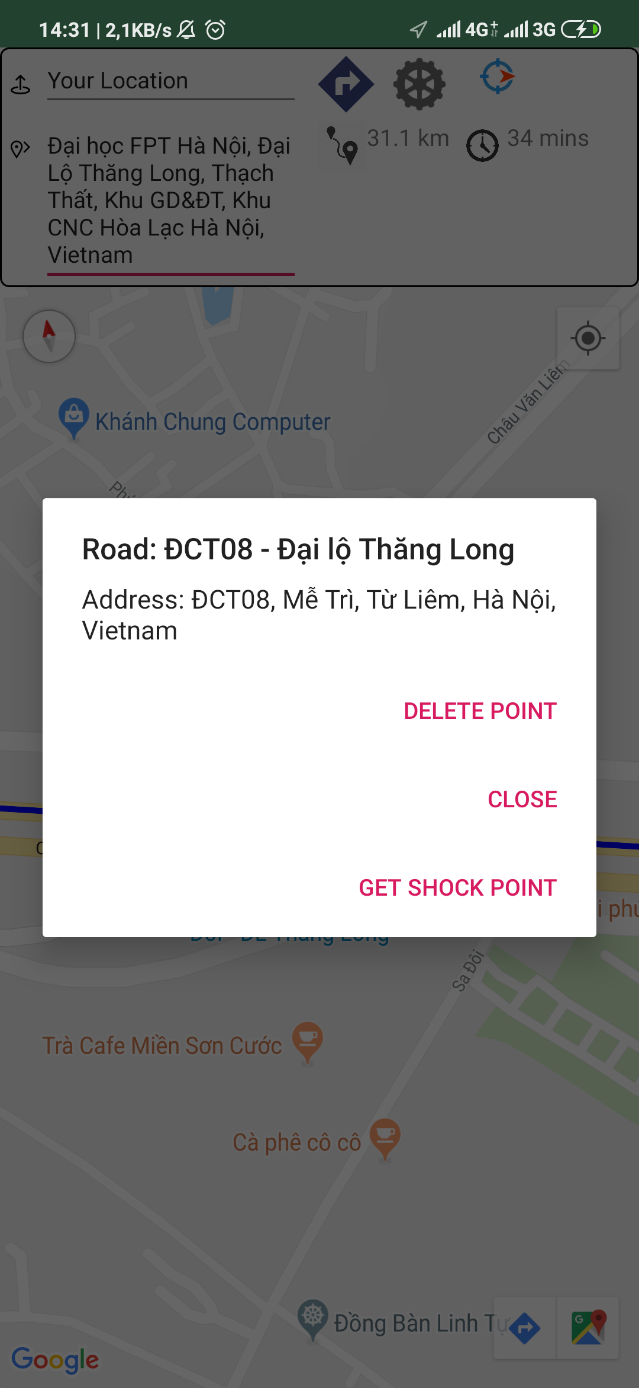
Step 3: Open application on your device.

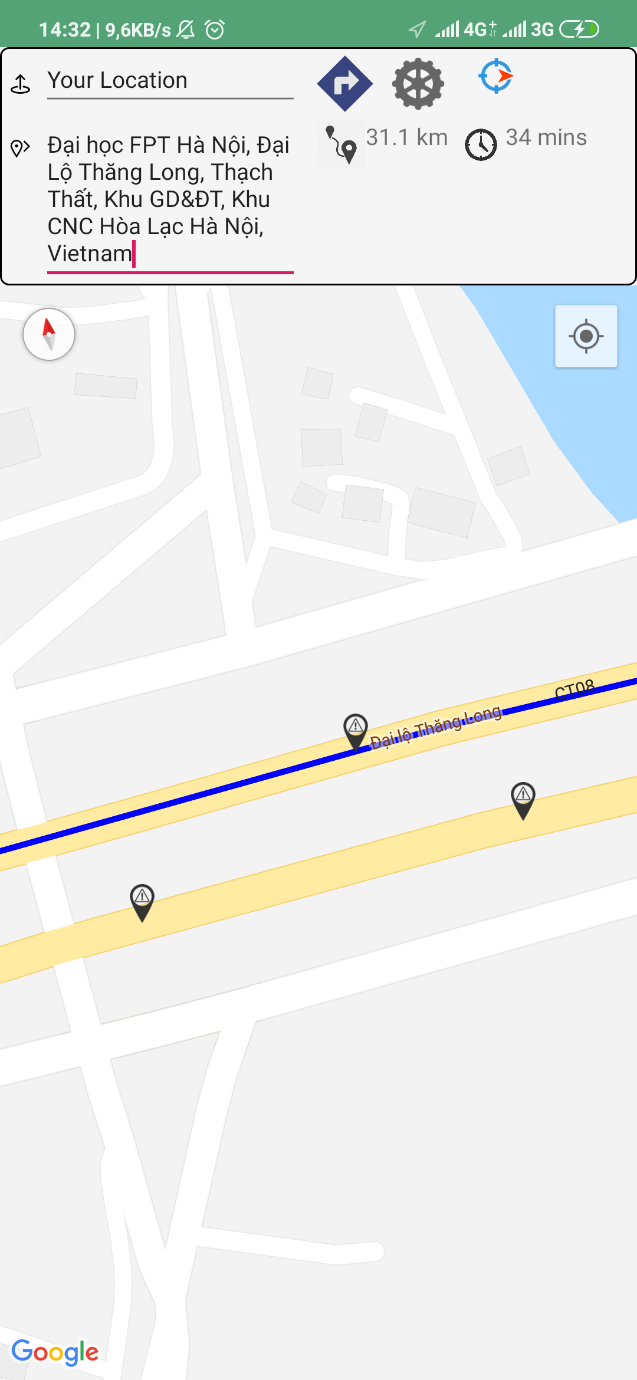
## User’s Guide

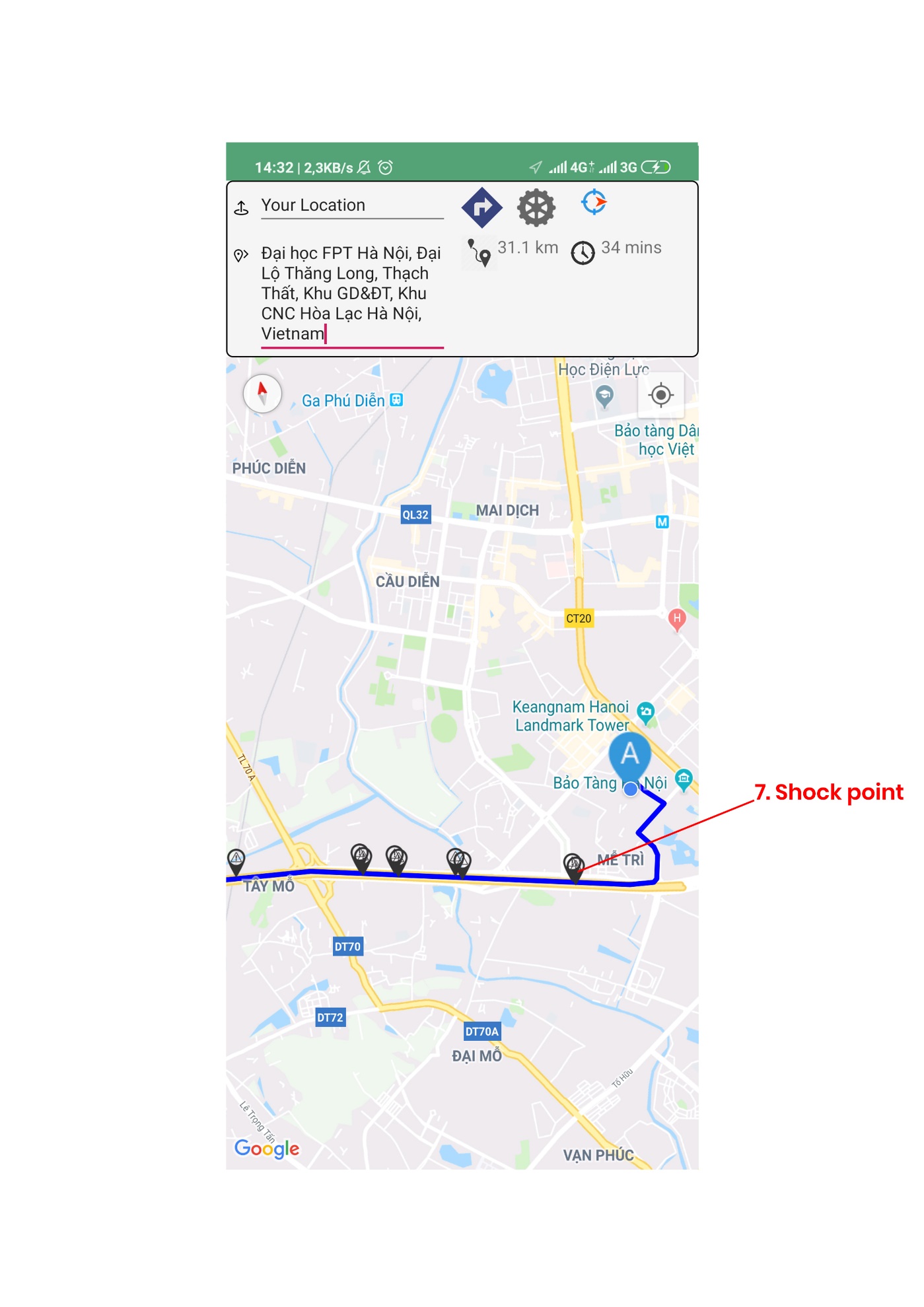
Just start this application, it’s very simple to use. You can find the path from your location (default)/ tap to custom starting point to your destination (tap to custom destination). Then, you can tap to show all shocking points in your road (if it’s high way).

If you don’t use find path function, the list of shocking points will show when you in high way (auto detect) and notify (with speaker) when distance from your car to shocking point equal about 300 meters.









## Other

# References

**[1]** <https://news.zing.vn/vec-noi-gi-khi-cao-toc-34500-ty-lai-xuat-hien-chi-chit-o-ga-post899829.html>

**[2]** M. Strutu, G. Stamatescu, and D. Popescu, “A mobile sensor network based road surface monitoring system,” in Proc. 17th Int. Conf. Syst. Theory, Control Comput. (ICSTCC), Oct. 2013, pp. 630–634.

**[3]** S. B. S. Murthy and G. Varaprasad, “Detection of potholes in autonomous vehicle,” IET Intell. Transp. Syst., vol. 8, no. 6, pp. 543–549, Sep. 2013.

**[4]** S.S.Rode,S.Vijay,P.Goyal,P.Kulkarni, and K. Arya, “Pothole detection and warning system: Infrastructure support and system design,” in Proc. Int. Conf. Electron. Comput. Technol., Feb. 2009, pp. 286–290.

**[5]** S. Hegde, H. V. Mekali, and G. Varaprasad, “Pothole detection and inter vehicular communication” in Proc. IEEE Int. Conf. Vehicular Electron. Safety (ICVES), 2014, pp. 84–87.

**[6]**https://www.researchgate.net/profile/Hongzi\_Zhu/publication/305527771/figure/fig1/AS:613872440471569@1523370003650/Illustration-of-the-vehicles-coordinate-system-and-the-smartphones-coordinate-system.png