**Software Requirements**

**Specification**

**for**

**CTguard**

**Version 1.0 approved**

**Prepared by**

**Franz Catuncan**

**Franz De Mesa**

**Dwight Pilares**

**Lorina Maraya**

**Dan Agustin**

**Jacques Barriga**

**Albert Mojica**

**CTguard**

**11/5/2016**

***Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.***

***Software Requirements Specification for CTguard Page ii***

**Table of Contents**

**Table of Contents .......................................................................................................................... ii**

**Revision History ............................................................................................................................ ii**

**1. Introduction..............................................................................................................................1**

1.1 Purpose ............................................................................................................................................ 1

1.2 Document Conventions.................................................................................................................... 1

1.3 Intended Audience and Reading Suggestions.................................................................................. 1

1.4 Product Scope .................................................................................................................................. 1

1.5 References........................................................................................................................................ 1

**2. Overall Description ..................................................................................................................2**

2.1 Product Perspective ......................................................................................................................... 2

2.2 Product Functions ............................................................................................................................ 2

2.3 User Classes and Characteristics ..................................................................................................... 2

2.4 Operating Environment.................................................................................................................... 2

2.5 Design and Implementation Constraints .......................................................................................... 2

2.6 User Documentation ........................................................................................................................ 2

2.7 Assumptions and Dependencies ...................................................................................................... 3

**3. External Interface Requirements ...........................................................................................3**

3.1 User Interfaces ................................................................................................................................. 3

3.2 Hardware Interfaces ......................................................................................................................... 3

3.3 Software Interfaces .......................................................................................................................... 3

3.4 Communications Interfaces ............................................................................................................. 3

**4. System Features........................................................................................................................4**

4.1 System Feature 1.............................................................................................................................. 4

4.2 System Feature 2 (and so on)........................................................................................................... 4

**5. Other Nonfunctional Requirements .......................................................................................4**

5.1 Performance Requirements.............................................................................................................. 4

5.2 Safety Requirements ........................................................................................................................ 5

5.3 Security Requirements ..................................................................................................................... 5

5.4 Software Quality Attributes ............................................................................................................. 5

5.5 Business Rules ................................................................................................................................. 5

**6. Other Requirements ................................................................................................................5**

**Appendix A: Glossary....................................................................................................................5**

**Appendix B: Analysis Models .......................................................................................................5**

**Appendix C: To Be Determined List............................................................................................6**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

**1. Introduction**

**1.1 Purpose**

This document specifies the software requirements of the Ligtas Sakay mobile application and covers the following aspects:

* Google Maps API
* Distress Function
* Report Function
* End Trip Function
* Destination Search
* Contacts Function
* User Log-in
* QR Code Scanning Function
* Access Levels (user and admin accounts)

**1.2 Document Conventions**

This document was created using Arial font size 11 with regular spacing.

**1.3 Intended Audience and Reading Suggestions**

This document is intended for use by development personnel as well as quality assurance teams, specifically testers of the mobile application.

The rest of the document contains the scope, the overall description of the application, system features, external interface and other requirements that are essential to the project.

**1.4 Product Scope**

**Scope of Initial Release**

One of the primary features our group plans to include in the application’s initial release would be the following:

* QR Code Scanner – to ensure the safety and sense of security of our commuters, the application will be equipped with a QR code scanning function to fetch identifying details of the vehicle the commuter is in.
* Automatic loading of local contacts – with regards to convenience, the application will do the task of fetching the commuter’s contacts for him/her.
* Google Maps Navigation – the application will help the commuter pick a destination as well as know what routes to be taken so as to avoid being taken to unfamiliar places by potentially abusive drivers.
* Distress Feature – with the simple push of a button, the application will contact 911 as well as the commuter’s selected contacts and will state that he/she is in danger accompanied by his/her current coordinates that constantly update
* Complaint function – should the commuter’s trip involve anything unpleasant with regards to the driver’s conduct or that of another encountered along the way, he/she can file a complaint with a single push that will be sent directly to the proper authorities (LTFRB)

**Scope of Subsequent Releases**

Our group has already anticipated future releases of the project with the following features to be added in the process:

* Community-based response – the application would be able to notify other users of the application regarding nearby distress calls from fellow users
* Location-based contacting of local law enforcement – the application, via data analytics, will be capable of contacting the nearest police station based on the user’s current location and/or destination point

**Limitations and Exclusions**

As of this time, our group has not considered any possible features anticipated by the users that we would not include in both the present and subsequent releases of the application.

**1.5 References**

**2. Overall Description**

**2.1 Product Perspective**

Considering the attachment most, if not all, individuals have with their smartphones most especially with regards to social media and entertainment our group believes we will be able to bring them an effective solution with regards to commuting security. Although there are already mobile applications using technologies such as live tracking, QR code scanning and route analysis, we have realized that these apps are built separately from each other. This is why the Team plans to develop an application that integrates all these features in a single app and will enable commuters to effectively identify their mode of transport, where they are and where they are planning to go, and informing their close companions of their status so as to avoid any untoward incident during their trip.

**2.2 Product Functions**

* QR Code Scanner – to ensure the safety and sense of security of our commuters, the application will be equipped with a QR code scanning function to fetch identifying details of the vehicle the commuter is in.
* Automatic loading of local contacts – with regards to convenience, the application will do the task of fetching the commuter’s contacts for him/her.
* Google Maps Navigation – the application will help the commuter pick a destination as well as know what routes to be taken so as to avoid being taken to unfamiliar places by potentially abusive drivers.
* Distress Feature – with the simple push of a button, the application will contact 911 as well as the commuter’s selected contacts and will state that he/she is in danger accompanied by his/her current coordinates that constantly update
* Complaint function – should the commuter’s trip involve anything unpleasant with regards to the driver’s conduct or that of another encountered along the way, he/she can file a complaint with a single push that will be sent directly to the proper authorities (LTFRB)

**2.3 User Classes and Characteristics**

*<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>*

**2.4 Operating Environment**

The performance of the application will be contingent on the following assets:

* Internet connection must be stable, if not excellent
* Any user using a public utility vehicle (i.e., taxi) will be using the application at any given time or place
* Data scanned from the QR Codes of the vehicles will be thrown to the application’s web service and fetched to the requesting smartphone; the server will be configured to handle multiple requests at any given time
* The application will be published on both Android and iOS platforms
* Service interruptions must be eliminated in situations of distress features
* Data protection provisions will be dependent on the extent to which the government will allow specific data to be disseminated to the public

**2.5 Design and Implementation Constraints**

**Constraints**

* Downtime will not be controlled by the team as the services of the application are processed by the cloud service provider
* GPS Accuracy is dependent on the user’s location and available satellites that the device can detect

**2.6 User Documentation**

The application itself will provide quick tutorials for new users.

**2.7 Assumptions and Dependencies**

AS-1: The user must have a decent internet connection; decent being defined as being able to connect to the internet without undue delays and/or disconnections for the duration of a trip

AS-2: The smartphone must have the following features: a camera, GPS capabilities

AS-3: There must be a mandate (i.e., executive order or a similar implementation) from the government to require all public utility vehicles (in this case, taxis) to have QR codes

**3. External Interface Requirements**

**3.1 User Interfaces**

(Show prototype)

**3.2 Hardware Interfaces**

* Smartphone Camera – responsible for scanning the QR code and providing the application with the vehicle plate number
* GPS module – will be responsible for calculating the current location of the user

**3.3 Software Interfaces**

* Android operating systems and iOS (for further development)
* Cloud Services

**3.4 Communications Interfaces**

**4. System Features**

SOFTWARE REQUIREMENTS SPECIFICATION CHECKLIST

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ASSIGNED TO | YES | NO | REMARKS |
| **Output**  ✓ The Google Maps API must correctly display the coordinates of a user starting from the scanning of the vehicle’s QR Code to the conclusion of his/her trip  ✓ The Report Button, once pressed, must display correct data from the application and prompt the user to input a certain field (seen below);  APPLICATION-BASED – vehicle’s plate number and its operator, date USER-BASED – complaint type data  ✓ The distress button, once pressed, must correctly send details to the primary law enforcement entity (911) and to selected contacts of the user;  TO CONTACTS – current coordinates and a description that he/she is in danger  TO LAW ENFORCEMENT – launch a phone call  ✓ The End Trip button, once pressed, must conclude the trip of the user by terminating all live tracking activities and returning to the main page  ✓ Upon selecting a destination point on the Google Maps canvas via touch, the application must notify the user’s selected contacts that he/she is travelling from her current position to the selected point and must enter the “begin trip” state  ✓ Upon pressing the Contacts button, the application must display all local contacts of the user |  |  |  |  |
| **Input**  ✓User must register in the application.  ✓User will select desired contacts for the application.  ✓ Each input form must include Qr code that includes information about location, contact number of the driver and the user.  ✓ Qr code must be scan or must be entered in the application. |  |  |  |  |
| **Process**  ✓ User logins to the application  ✓ The application should not allow any new trips from a device while the another device with the same account is in active trip.  ✓ User indicates a destination point  ✓ User scans the QR Code inside the vehicle  ✓ Users sends information (destination point and vehicle information) to chosen contacts  ✓ If in danger, user can press the distress call button. Therefore, calling 911.  ✓ If user has a complaint, user presses the report button, application auto-fill information such as Vehicle’s plate number and its operator and user fills out further details the report.  ✓ User ends trip by either arriving to the destination or tapping the end trip button. |  |  |  |  |
| **Performance**  ✓ The system can support hundreds of users online simultaneously. The system must support 25 users online simultaneously.  ✓ Our system has the capability to operate 24/7 , 7 times a week, 365 days a year.  ✓ The System aims to Minimize Application Launch Response Time  ✓Decrease the chance of Application Failure  ✓Maximize the QR code Scanner Efficiency  ✓Maximize the Compatibility of Android API  ✓Lessen the chance of having failures when it comes to Distress Module Features  **Control**  ✓ The system must provide logon security at the application level.  ✓ User account and its details must be added or changed only by the user and can only be deleted by the administrator.  ✓ The system must maintain separate levels of security for users and the system administrator.  ✓The user can only be tracked by whom he/she appoints.  ✓ The application must create an error log file that includes the error type, description, and time. |  |  |  |  |

**5. Other Nonfunctional Requirements**

**5.1 Performance Requirements**

|  |  |
| --- | --- |
| **SPECIFICS** | **DESCRIPTION** |
| Minimum API | Android 4.4 (Kit Kat) |
| Device Functionalities | * GPS Enabled * Data enabled * Camera |

**5.2 Safety Requirements**

* Misuse (i.e., improper use of 911 distress call) will be grounds for legal action against the user

**5.3 Security Requirements**

* The application must maintain the data privacy of the user (i.e., location sharing) by providing it ONLY to the user’s selected contacts.
* The application’s main functions may only be accessed once the user successfully logs in

**5.4 Software Quality Attributes**

*<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>*

**5.5 Business Rules**

* User logs in
* User scans QR code or manually inputs data
* User can search and set destination
* User selects one or multiple contacts to share vehicle information, location, and destination.
* User can end trip
* User can call for distress
* User can file a report
* User logs out

*<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>*

**6. Other Requirements**

*<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>*

**Appendix A: Glossary**

*<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>*

**Appendix B: Analysis Models**

*<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams*.>

**Appendix C: To Be Determined List**

*<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>*