Car Gossip
Requirements Specification

Final version

Revision History

Date	Version	Description	Author
2012-10-30	0.1	Initial Draft	Achille Penna
2012-10-31	0.2	Added sections: 1, 3, 4, 5	Achille Penna
2012-11-02	0.2	Added section: 2	David Raypke
2012-11-18	1.0	Fixed Use Case Model, Added new requirements, minor fixes	Achille Penna
2012-11-25	1.1	First revision	Nikola Vranešić
2012-01-17	Final version draft	Final version proposal, Added new Use Vase Models, completion of unfinished sections.	Achille Penna
2012-01-20	Final version	Final version revision	Nikola Vranešić

Table of Contents

1. Introduction

1.1 Purpose of this document

This document describes in detail all the requirements for the project "Car Gossip" and provides an overview of the functionalities that will be implemented.

Since the project will be developed in iterative steps, this document will be reviewed after every release cycle.

1.2 Intended Audience

The Requirements Definition document is primarily used by the project members. It will be used (and modified if needed) throughout the whole development process.

USER	USE
Project Manager	To keep track of whether the requirements support the development plan
Other project members	A guideline on which requirements to implement.
Supervisor and other stakeholders	Overview of the product to be developed

1.3 Scope

This document covers the requirements specification, related use cases and is describes the basic project development needs. Details will be shown with the help of explanatory diagrams.

Since there are no specific customer demands, there is a certain amount of freedom redefining the functionalities whenever needed during the development.

With the functional requirements we define the functionalities to be provided by the final product, while through the non-functional requirements we specify the quality attributes to be achieved by the product. Use case models will be employed as support to the requirements definition.

1.4 Definitions and acronyms

1.4.1 Acronyms and abbreviations

Acronym or abbreviation	Definitions
ETHZ	Federal Institute of Technology Zurich
NS-2	Network simulation tool in the version 2
DB	Database
ODS	Open Data Source
TS	Traffic Simulator
ICS	Internal Car Sensors
ICD	Internal Car Device
DSRC	Dedicated Short Range Communication
GPS	Global Positioning System
API	Application Programming Interface
GUI	Graphical User Interface
HCI	Human-Computer Interaction
App	Application
AApp	Android Application
WS	Web Server
WApp	Web Application

1.4.2 Definitions

Keyword	Definitions
Swiss Database	ETHZ Open Database is a database used that provides 24 hours of traffic information in Zurich.
Gossip	A message containing information about the cars position and movement.
Alert	A default message sent by drivers to inform others of the traffic status.

1.5 **References**

Swiss Database, http://www.lst.inf.ethz.ch/research/ad-hoc/car-traces/
"NS-2 movement format files", Kai Nagel, Bryan Raney and Hinnerk Spindler, ETH Zurich
DSRC, http://en.wikipedia.org/wiki/Dedicated_short-range_communications
GPS, http://en.wikipedia.org/wiki/Global_Positioning_System

2. Overall Description

The main goal of this project is to simulate a message-based information exchange for vehicular traffic. The aim is to enrich the information that the driver has at their disposal regarding traffic conditions, hinders and generally their surroundings.

Such information can be provided through different means, mainly by visualizing other cars in the driver's surroundings on a map using the information from Gossips (car ID, GPS coordinates, velocity, direction) received via DSRC.

The system will also enable the driver to send and receive Alerts regarding traffic-related events (car crash, street closed for repair etc.). The driver will be able to report those events and distribute them via DSRC to other vehicles within the DSRC range.

Moreover, the messages (Gossips and Alerts) are stored in a database which is connected to a web service that is able to exploit them in order to provide predictions regarding traffic-related events. Such predictions will be distributed in the same way as for custom messages in the surrounding area via DSRC.

2.1 Product Perspective

The product is self-contained, but since we don't have the real hardware components e.g. cars or ICDs, ICSs, we provide a simulation of them as separated components that can be iteratively replaced by real hardware devices.

2.2 Product Functions

The product functions are provided by a set of components interacting through interfaces:

1. The Traffic Simulator provides a parameterized output of traffic movement in Zurich and its communication (DSRC). Both are visualized on a map.
2. Each simulated ICD is able to connect to an Android device running the Car Gossip App and exchange Gossips and Alerts via Bluetooth.
3. The Android App provides a GUI to view processed traffic information useful to the driver and upload it, if possible, to the Web Server.
4. The Web Server stores all data received from the Android App and offers them through Web Services.
5. The Web App provides dynamic analysis of the stored traffic information.

3. Requirements Description

3.1 Introduction

Since the main actor in the project is the driver, the requirements must provide functionalities that result in driver awareness of their surroundings.

3.2 General requirements

The General requirements are *what* must be delivered to provide value. Products, systems, software, and processes are the ways how to deliver, satisfy, or meet the general requirements *whats*.

http://en.wikipedia.org/wiki/Business_requirements

1. Open Data Source
    a. Swiss DB
    b. Parser
    c. Parsed DB
2. Desktop App
    a. Traffic Simulator
        i. GUI
        ii. Vehicular Scenario Creator
        iii. DSRC cloud
    b. ICD
        i. ICS
        ii. Priority Filter
        iii. DSRC device
        iv. Bluetooth device
3. An Android device running the Car Gossip App
    a. GUI
4. Web Server
    a. DB
    b. Alert Validator
5. Web App
    a. GUI

3.3 Functional requirements

In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs

http://en.wikipedia.org/wiki/Functional_requirement

There are three different types of users that can interact with this product with different applications. Each application is responsible for its own requirements.

3.3.1 Simulate cars (ICDs)

An operator is a user that uses the Desktop App. Firstly he needs to connect with the Open Data Source and import a mov file to get the needed traffic data. Using a parser, those data are extracted and parsed into data that is easier to process. Using the parsed traffic data, all the cars are represented on a map. The Operator selects any number of cars that he wishes to simulate and defines a timeframe which defines their behavior and its length. That is referred to as creating a scenario. They can be saved and loaded for future uses. Once a scenario is selected and the simulation started the Desktop App simulates the cars movements and visualizes them on the map. Each car is represented by an ICD which is simulated within the Desktop App. The ICD collects its data from simulated sensors and creates gossips depending on their output. All this data is simulated using the parsed traffic data.

3.3.2 Simulate DSRC (gossips and alerts)

To simulate a communication between the ICDs the Desktop App uses a DSRC cloud that forwards broadcasted messages to other ICDs that are within its range. The operator can select any car on the map and see which cars receive his DSRC messages.

3.3.3 ICD-Android communication

The selected car can also be connected to an Android App via Bluetooth. The second user is the driver. The driver uses the Android App which can connect to an ICD. In this case the ICD is simulated and the Android App connects to the selected simulated car. The driver views a map which visualizes traffic and alerts on it. The cars and alerts on the map are results of message processing. The messages are received via the Bluetooth connection with the ICD. Every driver is able to publish an alert by pressing in the Android App.

3.3.4 Android-Webserver communication

The Android App was planned to connect to the internet and forward all messages by uploading them to the webserver where they would be stored in a database. The third user is the Analyst who uses the Web App. The Web App was supposed to connect to the webservers database and offer the analyst dynamic creation of analysis queries by supplying parameters needed for the analysis.

3.4 **Not-Functional requirements**

3.4.1 Accessibility

The Open Data Source is only accessible through the Desktop Application. Its traffic data is parsed and used for traffic simulation.

The Android App decreases accessibility since only android devices are able to connect to the ICD and give the information to the driver. This is easily correctable by creating apps for different platforms.

The Web Server's stored traffic data will be publicly accessible as open data via the project Web App and specific APIs.

3.4.2 Availability

Since the cars communicate non-stop, their data is always available via an Android device running the Car Gossip App.

3.4.3 Performance

The software response time is being optimized by quality software development to make the HCI better.
The Desktop App uses a non-relational database for faster traffic data processing to produce traffic simulations as quick as possible. The Android App refreshes the traffic data every 5 seconds to minimize overflow errors and to maximize data flow speeds. The Web App uses a relational database for easier analysis.

3.4.4 Privacy

Each time a message is leaving a physical component it is encrypted to ensure the public's privacy and decrypted as it is received.

3.4.5 Portability

The portability will be limited to a certain set of devices.

3.4.6 Security

Our Web Server will be secured from basic malicious software and hacking attempts.

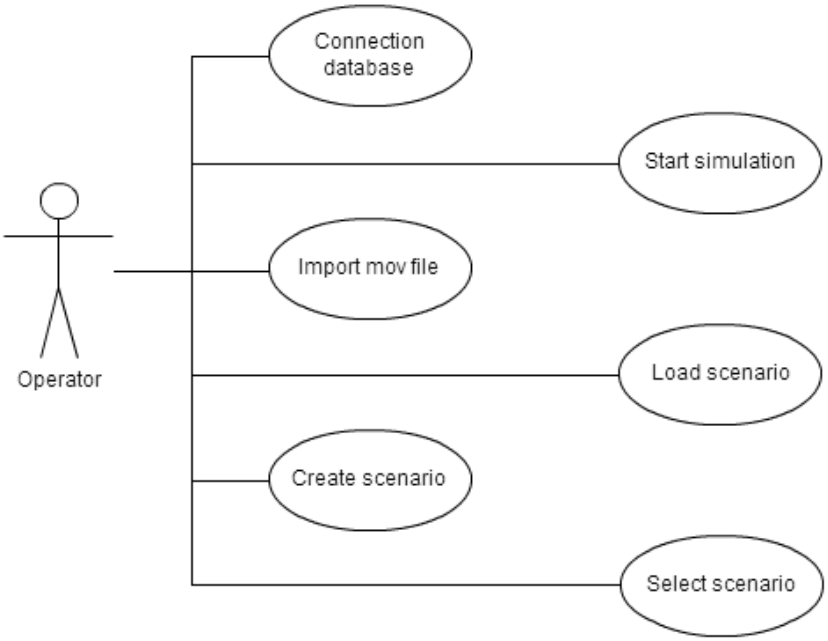
3.4.7 Usability

Since this software will be mainly used in a car, the environment has to be taken into account. The interaction with the Android App is minimized so it does not interfere with the driving. The GUI will avoid small fonts and icons to make the App very intuitive for understanding and usage. A few elements that should be addressed are the position, color and size of buttons – as well as contrast between traffic map and Alerts.

4. Use Cases

Actors	Definitions
Operator	A person using the Desktop App to control the traffic simulation.
Driver	A person driving a car and using the Android App.
Analyst	A person using the Web App to analyze traffic data.

4.1 **Operator**



4.1.1 Connection Database

Use case ID	O1
Name	Connection Database
Goal	To connect the database with dabase.
Participating actors	Operator
Precondition	MongoDB is running
Main scenario	1. The operator clicks “Connect to Database” button
Extensions	

4.1.2 Import mov file

Use case ID	O2
Name	Import mov file
Goal	To import data from mov file
Participating actors	Operator
Precondition	MongoDB is running
Main scenario	1. The operator clicks “Import mov file” button 2. The browser appears to choose a file to import 3. If the operator chooses a mov file the simulator drops the database

	And store the data. 4. Else the operator chooses a file different from mov the simulator drops the database but there are data to start the simulator
Extensions	

4.1.3 Create Scenario

Use case ID	O 4
Name	Create Scenario
Goal	To create the scenario for the simulator
Participating actors	Operator
Precondition	1. MongoDB is running 2. The operator opened the connection between simulator and database
Main scenario	5. The operator selects some cars on the map 6. The operator decides the long time for the simulation 7. The operator clicks “Create Scenario” buttons
Extensions	

4.1.4 Load Scenario

Use case ID	O 4
Name	Load Scenario
Goal	To take the scenario already created from database
Participating actors	Operator
Precondition	1. MongoDB is running 2. The operator opened the connection between simulator and database
Main scenario	1. The operator stores the scenario already created
Extensions	

4.1.5 Select Scenario

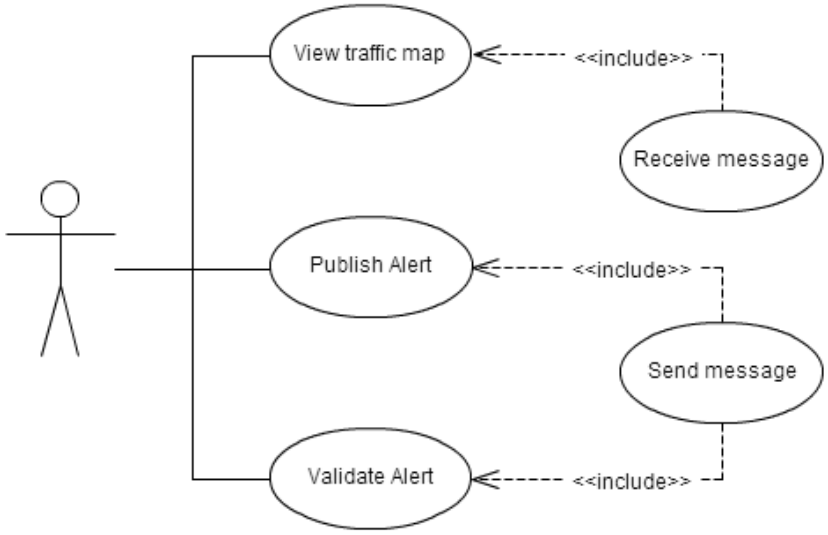
Use case ID	O 5
Name	Select Scenario
Goal	To select the scenario for the simulation
Participating actors	Operator
Precondition	1. MongoDB is running 2. The operator opened the connection between simulator and database 3. The scenarios are created already
Main scenario	1. The operator chooses a scenario
Extensions	1. The operator can delete the scenario 2. The operator can start the simulation

4.1.6 Start Simulation

Use case ID	O6
Name	Start Simulation
Goal	To see the cars is moving and they send alert message and spread the gossip message
Participating actors	Operator

Precondition	1. MongoDB is running 2. The operator opened the connection between simulator and database 3. The scenarios are created already 4. The operator chose the scenario
Main scenario	1. The simulator works
Extensions	

4.2 **Driver**



4.2.1 View Traffic Map

Use case ID	D1
Name	ViewMap
Goal	To get a traffic overview of his surroundings (car positions and traffic alerts)
Participating actors	Driver
Main scenario	1. The Android App visualizes cars surrounding you and alerts near you on a map

4.2.2 Receive a Message

Use case ID	D2
Name	Receive Message
Goal	To receive messages containing information about the cars surroundings
Participating actors	Driver
Precondition	1. The drivers car must be within DSRC range of a car that has broadcasted a message 2. The Android App must be connected via BT to the ICD
Main scenario	1. The drivers ICD receives a message via DSRC 2. The ICD sends the message to the Android app via Bluetooth 3. The Android app receives the message from the ICD 4. The Android app processes the message and visualizes the result on a map

4.2.3 Publish alert

Use case ID	D3
Name	Publish alert
Goal	To help make other drivers aware of specific problems in traffic
Participating actors	Driver
Precondition	The Android App must be connected via BT to the ICD and/or have an internet connection
Main scenario	1. The driver presses one of the alert icon located on the screen 2. The alert is processed to be send via Blutooth and Internet, and is visualized on the map

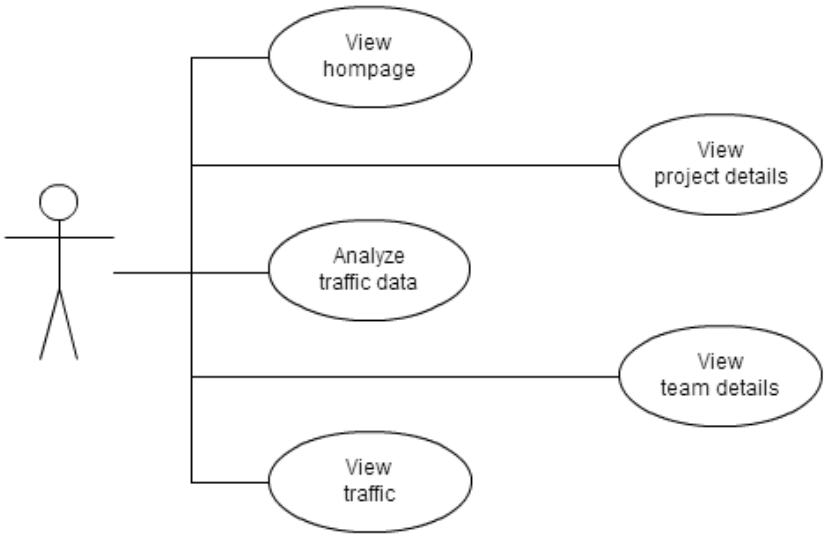
4.2.4 Send Message

Use case ID	D4
Name	Send Message
Goal	To send a message to the ICD and/or Web Server
Participating actors	Driver
Precondition	The driver either publishes or verifies an alert
Main scenario	1. Android app sends the alert message to web server 2. If it is not a fake the Web Server sends a confirm message to Android app 3. The Android app receives the confirm message 4. The Android app sends the warning message to ICD via Bluetooth 5. The ICD sends the warning message to others Cars

4.2.5 Verify alert

Use case ID	D5
Name	Verify alert
Goal	To prevent spamming
Participating actors	Driver
Precondition	The alert message is on the map
Main scenario	1. The driver can click on the warning message to remove it 2. Android app sends the warning message to web server and ICD via Bluetooth 3. ICD sends the message to others ICD

4.3 **Analyst**



4.3.1 View homepage

Use case ID	A1
Name	View homepage
Goal	To get access to other Web App content and functionality
Participating actors	Analyst
Precondition	Internet connection
Main scenario	1. Open page cargosip.webapp.com* in web browser or press the CarGossip logo located in the page header
Extensions	

4.3.2 Analyze traffic data

Use case ID	A2
Name	Analyze traffic data
Goal	To receive analyzed traffic data based on previously sent parameters defined by the analyst
Participating actors	Analyst
Precondition	Internet connection
Main scenario	1. Open page cargosip.webapp.com/stats* in web browser or press the Stats button located in the page header 2. Input parameters to define the analysis 3. Press the Submit button to send the parameters. 4. Receive and view the analysis results
Extensions	

4.3.3 View traffic

Use case ID	A3
Name	View current traffic
Goal	To view the currently simulated traffic in certain areas on Google Maps.
Participating actors	Analyst
Precondition	Internet connection

Main scenario	1. Open page cargosip.webapp.com/traffic* in web browser or press the Traffic button located in the page header
Extensions	

4.3.4 View project details

Use case ID	A4
Name	View project details
Goal	To view the details about the CarGossip project.
Participating actors	Analyst
Precondition	Internet connection
Main scenario	1. Open page cargosip.webapp.com/project* in web browser or press the Project button located in the page footer
Extensions	

4.3.5 View team details

Use case ID	A1
Name	View team details
Goal	To view the details about members of the CarGossip team.
Participating actors	Analyst
Precondition	Internet connection
Main scenario	2. Open page cargosip.webapp.com/team* in web browser or press the Team button located in the page footer
Extensions	

*since the Web App was never deployed to a webserver, these URLs are not valid.

5. Requirements Definition

5.1 Requirement Group Definitions

Identification	Requirement Group	Rem.
ODS	Open Data Source	
TS	Traffic Simulator	
ICD	Internal Car Device	
AApp	Android App	
WS	Web Server	
WApp	Web App	

5.2 Requirement Sources

.

Source	Description	Rem.
ODS	Open Data Source	
OPRT	Operator	
TS	Traffic Simulator	
ICD	Internal Car Device	
AD	Android Device	
DRV	Driver	
WS	Web Server	
DB	Data Base	
WApp	Web App	
ANLZ	Analyzer	

5.3 **Requirement Definitions**

Identity	Status	Priority	Reference	Description	Source
				Open Data Source	
ODS-1	I	1		Parses data from NS-2 movement format files downloaded into CSV formatted data with new calculated data as well	ODS
				Traffic Simulator	
TS-1	I	1		Simulates vehicular movement (traffic behavior) based on input parameters from the Operator and retrieved data from the Open Data Source.	OPRT, ODS
TS-2	I	1		Simulates a DSRC cloud (gossip spreading) by forwarding messages to other cars that are in range of the message sender.	ICD
TS-3	I	2		Visualizes the traffic behavior on a map as a real time event.	TS
TS-4	I	2		Visualizes the gossip spreading on a map as a real time event.	
				ICD	
ICD-1	I	1		Creates Gossips out of data retrieved from the ODS simulating ICS's supplying the data.	TS
ICD-2	I	1		Sends messages to the DSRC cloud simulating broadcasting messages via DSRC.	ICD
ICD-3	I	1		Receives messages from the DSRC cloud simulating reception from other cars via DSRC.	TS
ICD-4	I	2		Sends messages to an connected Android device running the Car Gossip App via BT.	ICD
ICD- 5	I	2		Receives messages from an connected Android device running the Car Gossip App via BT.	AApp
ICD-6	A	3		Filters received messages using a priority queue.	ICD
				Android (Car Gossip) App	
AApp-1	I	2		Receives messages from a connected ICD via BT.	ICD

AApp-2	I	3		Visualizes traffic movements on a map.	AApp
AApp-3	A	3		Enables the Driver to send Alerts by pressing an icon button on the screen.	AApp
AApp-4	I	2		Sends messages to a connected ICD via BT.	DRV
AApp-5	I	4		Sends messages to Web Server via HTTP(S).	AApp
AApp-6	I	4		Receives messages from Web Server via HTTP(S).	WS
				Web Server	
WS-1	I	4		Receives messages from an Android device running the Car Gossip App via HTTP(S).	AApp
WS-2	I	4		Stores the received messages into a DB which can be accessed via a public API.	DB
WS-3	A	5		Validates Alerts to prevent spamming.	WS
WS-4	I	4		Pushes messages to all Android devices running the Car Gossip App via HTTP(S).	WS
WS-5	A	5		Responds to Web App requests using Web services.	WApp
				Web App	
WApp-1	A	5		Sends requests to Web Server to access and process stored traffic data.	WS

Requirement status:
I = initial (this requirement has been identified at the beginning of the project),
D = dropped (this requirement has been deleted from the requirement definitions),
H = on hold (decision to be implemented or dropped will be made later),
A = additional (this requirement was introduced during the project course).

Requirement priority:
1-very high priority; 2-high priority; 3-medium priority; 4-low priority; 5-very low priority