



POLITECNICO DI MILANO
DIPARTIMENTO DI ENERGIA,
INFORMAZIONE E BIOINGEGNERIA
Computer science and engineering
Software engineering 2

SafeStreets - DD

Professor:

Prof. Elisabetta Di Nitto

Candidates:

Nicolò Albergoni - 939589

Luca Loria - 944679

Academic Year 2019/2020

Milano - 09/12/2019

Contents

1	Introduction	3
1.1	Purpose	3
1.1.1	Goals	3
1.2	Scope	4
1.2.1	Description of the given problem	4
1.2.2	Current system	5
1.2.3	Shared Phenomena	5
1.3	Definitions, acronyms and abbreviations	7
1.3.1	Definitions	7
1.3.2	Acronyms	7
1.3.3	Abbreviations	7
1.4	Revision History	8
1.5	Reference Documents	8
1.6	Document Structure	8
2	Architectural Design	9
2.1	Overview: High-level components and their interaction	9
2.2	Component view	9
2.3	Deployment view	9
2.4	Runtime view	9
2.5	Component interfaces	9
2.6	Selected architectural styles and patterns	9
2.7	Other design decisions	9
3	User interface design	10
4	Requirements traceability	11
5	Implementation, integration and test plan	12

6	Effort Spent	13
6.1	Luca Loria	13
6.2	Nicolò Albergoni	14
7	References	15

Introduction

1.1 Purpose

The purpose of this Requirement Analysis and Specification Document (RASD) is to provide a detailed description about SafeStreets. In particular, this document is focused on important aspects that are useful during the design of the software architecture like: scope, functional and non-functional requirements, use cases and scenarios, constraints and assumption, UML diagrams, limitation and interfaces with other software. Overall the document is a useful guide for the developers that will have to follow and implement all the necessary requirements; nevertheless it is also a document that can be given to potential customers to get them an idea of what the software will be like.

The software features have been identified in the following list of goals.

1.1.1 Goals

- [G1]: Allow the user to send a violation report, consisting in a picture and some metadata;
- [G2]: Allow the user to watch the history of his reports and their status;
- [G3]: Allow the user to watch (on a map) areas and streets with an high frequency of violations:
 - [G3.1]: The user gets the latest violations near its current location;
 - [G3.2]: The user gets the latest violations close to the location that he specifies;
- [G4]: Allow the local system administrator of the police station to create accounts for the police technicians;

- [G5]: Allow the authorities to visualize, schedule and change the status of reports submitted by the users;
- [G6]: Allow the authorities to mine data to make analysis and retrieve statistics;
- [G7]: If the municipality offers a service that provides data about car accidents, the system must be able to cross this information with its own data in order to identify possible unsafe areas;
 - [G.7.1]: The system must be able to suggest possible interventions;
 - [G.7.2]: The users must be able watch this unsafe areas;
 - [G.7.3]: The authorities must be able to consult the crossed data.

1.2 Scope

1.2.1 Description of the given problem

SafeStreets is a service that aims to improve the safety of the streets via the help of the users. They can notify violations or any illegal behaviour related to street parking to authorities. In particular, users can interact with the service via an application that can be used to send the violation reports; the latters mainly consist in a picture of the vehicle responsible for the violation. Moreover users can send, along with the picture, location, date and additional information related to the infringement. The system also provides a Web interface that can be used by authorities in order to check the violations received. It is important to note that the picture sent by the user must contain the car plate in order to let police officers know which is the real vehicle that committed the violation. The application also offers the possibility to see areas/streets with the highest violations rates thanks to the data collected over time. As an advanced functionality the application can interact with services offered by the municipality; in particular if a service offers data related to car accidents, SafeStreets can cross this information with its owns, in order to get a better idea of the potentially unsafe areas and therefore suggest some possible interventions. Finally the application will have to be scalable and easy to use in order to provide a fast and efficient utilization for users that detect a violation and want to report it immediately.

1.2.2 Current system

SafeStreets is a new service that it is entering the market. There are not legacy systems that need to be integrated to the application, with the exception of the third party services that offers some functionality required by the application (like Maps and plate recognition). Those services will be better explained in the following paragraphs.

1.2.3 Shared Phenomena

Phenomenon	Shared	Who controls it
The user detects and takes a picture of a violation	N	W
The user sends a violation report	Y	W
The system performs the plate recognition from the report image	Y	M
The system stores the report and marks it as a pending violation	N	M
The user wants to see his report history	N	W
The user asks the system to list his latest reports	Y	W
The system filters the current user reports	N	M
The system shows the list of current user reports	Y	M
The user wants to see the most dangerous streets in a certain location	N	W
The system filters the specified location in order to find violations	N	M
The system determines the most dangerous streets close to the specified location	N	M
The system shows the list of the most dangerous streets in the specified location	Y	M
The police corporate wants to register a newly hired technician	N	W
The police corporate submit the information to register a new technician account	Y	W

The system checks wether there is already an account registered to the new technician	N	M
The system rejects the new registration	Y	M
The system registers the new technician	Y	M
The authority wants to change the status of a violation	N	W
The authority specify the new status and sends the request to the system	Y	W
The system checks wether the violation occurred in the jurisdiction of the authority	N	M
The system rejects the new request	Y	M
The system updates the status of the violation	Y	M
The authority wants to schedule a violation to a set of police technicians	N	W
The authority submits the violation and the list of PT to the system	Y	W
The system checks wether the PT belongs to the asking authority and if the number of allocated technician is lower than six	N	M
The system rejects the schedule request	Y	M
The system updates the agenda of the PTs and the list of allocated PTs on the violation	Y	M
The authority wants to mine information about a set of violations	N	W
The authority specifies the list of filters and queries the system	Y	W
The system retrieves the list of violations that fit into the specified filters	N	M
The system shows the filtered list of violation reports	Y	M
The authority wants to receive suggestions about possible interventions in its jurisdiction	N	W
The authority asks the system for suggested intervention	Y	W

The system retrieves the list of accidents occurred in the authority's jurisdiction from the municipality database	Y	M
The system computes a list of suggested violations	N	M
The system shows the list of suggested violations	Y	M

1.3 Definitions, acronyms and abbreviations

1.3.1 Definitions

- *Most Dangerous Streets*: Streets with the highest frequency of violation reports.

1.3.2 Acronyms

- EU: European Union;
- CET: Central European Timezone;
- API: Application Programming Interface;
- HTTP: Hyper Text Transfer Protocol;
- GPS: Global Positioning System;
- GDPR: General Data Protection Regulation.

1.3.3 Abbreviations

- [Gn]: n-th goal;
- [Dn]: n-th domain assumption;
- [Rn]: n-th functional requirements;
- MDS: Most Dangerous Street;
- LSA: Local System Administrator;
- PT: Police Technician.

1.4 Revision History

- Version 1.0:
 - First release.

1.5 Reference Documents

- SafeStreets assignment document;
- Previous years RASDs of the Software engineering 2 project;
- IEEE Std 830--1998 IEEE Recommended Practice for Software Requirements Specifications;
- Slides from the course "Software engineering 2".

1.6 Document Structure

1. In the first part an overall vision of the system is given. We provide a rather summarized description of the purpose of the application and its scope as a list of fundamental goals that the application must accomplish;
2. In the second part the general architecture of the software and its principal functions are described. Moreover in this section are also defined the classes of utilizators;
3. The third part is the core of the document. It defines: the requirements and the domain assumptions for each goal, a set of scenarios that describes real uses cases of the application as well as different types of UML diagrams related to different aspects of the service, like use cases diagrams and sequence diagrams. Also, a list of assumptions and design constraint and an analysis of the non-functional attributes of the system is proposed;
4. The fourth part propose the Alloy formalization of the system composed by its fundamental functionalities;
5. The last part states the hours of work division and the tools used to create all the part of this RASD document.

Architectural Design

- 2.1 Overview: High-level components and their interaction
- 2.2 Component view
- 2.3 Deployment view
- 2.4 Runtime view
- 2.5 Component interfaces
- 2.6 Selected architectural styles and patterns
- 2.7 Other design decisions

User interface design

Requirements traceability

Implementation, integration and test plan

Effort Spent

6.1 Luca Loria

Day	Hours	Topic
20/10/2019	1.5	Text assumptions
22/10/2019	1	Domain assumptions
23/10/2019	2	Design constraints
24/10/2019	2	Revision ch 1 and 2
26/10/2019	3	Software system attributes
28/10/2019	0.5	Performance requirements
29/10/2019	2	External interface req
30/10/2019	5	Mockups creation
31/10/2019	1.5	Revision ch 3
01/11/2019	1	Alloy signatures
04/11/2019	1.5	Alloy facts part 1
06/11/2019	3	Alloy facts part 2 and world predicates
07/11/2019	4	Finish alloy, fix class diagram and start impaginating
08/11/2019	3.5	Reviewing requirements and sequence diagrams. Create references and Revision. Create alloy world section in RASD. Start impaginating correctly
09/11/2019	2	Shared phenomena matrix, frontpage and pagination fixes
10/11/2019	2	Final Revision

6.2 Nicolò Albergoni

Day	Hours	Topic
22/10/2019	2.5	Purpose
23/10/2019	3	Scope, Current system
24/10/2019	1.75	Goals, Overview
26/10/2019	3	Product perspective, Product function
27/10/2019	3	Product function, Revision ch 1 and 2
29/10/2019	2.5	Scenarios
30/10/2019	4	Use cases
31/10/2019	3	Use cases, Revision ch 3
01/11/2019	1	Finish use cases
02/11/2019	2.75	Requirements
05/11/2019	2.25	Finish requirements
06/11/2019	2.5	Sequence diagrams
07/11/2019	3.25	Sequence diagrams
08/11/2019	1.75	Finish and reviewing of sequence diagrams
09/11/2019	2	Use case diagrams, pagination fixes
10/11/2019	2	Traceability matrix, final revision

References

- LaTeX Workshop extension for Visual Studio Code:
<https://github.com/James-Yu/LaTeX-Workshop/>
- LaTeX compiler:
<https://www.latex-project.org/>
- StarUML for UML diagrams:
<http://staruml.io/>
- Moqups for creating Mockups:
<https://app.moqups.com/>
- Alloy extension for Visual Studio Code:
<https://github.com/s-arash/org.alloytools.alloy/tree/ls>
- Alloy compiler and visualizer:
<http://alloy.lcs.mit.edu/alloy/>