

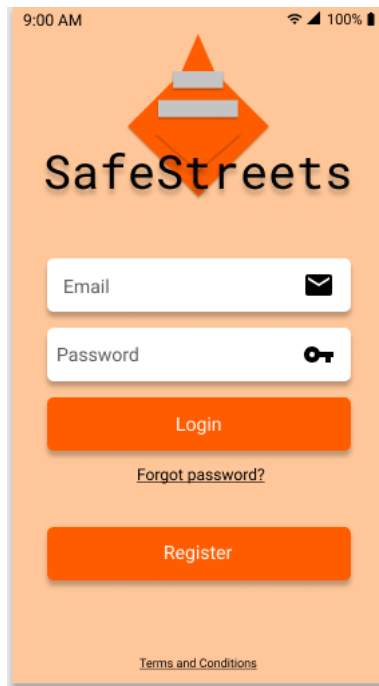
SafeStreet
RASD document

Dario Miceli Pranio
Pierriccardo Olivieri

Academic year: 2019 - 2020



POLITECNICO
MILANO 1863



Contents

1	Introduction	3
1.1	Purpose	3
1.2	Scope	3
1.2.1	World Phenomena	4
1.2.2	Shared Phenomena	4
1.2.3	Machine Phenomena	4
1.3	Definitions, acronyms, abbreviations	4
1.3.1	Definitions	4
1.3.2	Acronyms	5
1.3.3	Abbreviations	5
1.4	Revision History	5
1.5	Reference documents	5
1.6	Document Structure	5
2	Overall Description	5
2.1	Product perspective	5
2.1.1	World Phenomena	5
2.1.2	Machine Phenomena	5
2.1.3	Shared Phenomena	6
2.1.4	Class Diagram	6
2.1.5	State Chart	6
2.2	Product functions	6
2.3	User characteristics	8
2.4	Assumption and Dependencies Constraints	8
2.4.1	Assumption	8
2.4.2	Dependencies	8

2.4.3	Constraints	8
3	Specific Requirements	9
3.1	External Interface Requirements	9
3.1.1	User Interfaces	9
3.1.2	Hardware Interfaces	9
3.1.3	Software Interfaces	9
3.1.4	Communication Interfaces	9
3.2	Functional Requirements	9
3.2.1	Use Case Diagrams	9
3.2.2	Scenarios	11
3.2.3	Use Cases	11
3.2.4	Sequence Diagrams	11
3.2.5	Goal Mapping on Requirements	11
3.3	Performance Requirements	11
3.4	Design Constraints	11
3.4.1	Standards compliance	11
3.4.2	Hardware limitations	12
3.5	Software System Attributes	12
3.5.1	Reliability	12
3.5.2	Availability	12
3.5.3	Security	12
3.5.4	Maintainability	12
3.5.5	Portability	12
4	Formal Analysis with Alloy	12
5	Efforts	12

1 Introduction

This is the RASD document for *SafeStreet*, it provides a general view about key aspects of the project. The purpose of this document is to formalize a description of the system's requirements both functional and non-functional. In the following pages will be covered goals of the application with respect to phenomena. This document is addressed to developers as a guideline to implement the requirements that follows.

1.1 Purpose

SafeStreets is a service that aims to provide users with the possibility to notify authorities when traffic violations occur, and in particular parking violations. The application's goal is achieved by allowing users to share photo, position, date, time and type of violation and by enabling *Authorities* to request them.

Safestreets requires the users to create an account to access its services, the functionalities unlocked after registration depend on the type of account created.

If a user creates an account as *Citizen*, he/she must provide information about ID card on order to prove that he/she is a real person. Furthermore, he must provide an email with which he will be uniquely identified and a password. Once the account has been activated, user can finally start to report parking violation. The users can also see a summary of the streets with the highest frequency of violations.

On the other hand, an officer will create an account as *Authority* and he will need to provide his name, surname, work's Matricola, a password and as for *Citizen*, will be uniquely identified by an email. Once the Matricola has been verified and the account has been activated, the officer can retrieve the potential parking violations sent by *Citizen* that have not been taken into account yet by other officers, analyze them and, if it is the right case, generate traffic tickets. *Authorities* can also see a summary of the vehicles' license plate that commit the most violations.

From this brief description of the functionalities we may extract the following goals for SafeStreets:

- [G1]: Allow users to be identified as a *Citizen* or as *Authority*;
- [G2]: Allow *Citizens* to report parking violations;
- [G3]: *Citizen* has to be able to input information about the violation that he has reported;
- [G4]: Must provide a visualization of the areas with high frequency of violations;
- [G5]: *Authority* can retrieve vehicles that commit the most violations;

Safestreets offers also some advanced functions in addition to the basic version.

- [G6]: Must ensure the chain of custody of the information sent by *Citizens*;
- [G7]: *Authority* can retrieve traffic violations' data inserted by *Citizens*;
- [G8]: Build statistics with the information about issued tickets;

1.2 Scope

Here we will describe all the relevant phenomena that may occur.

1.2.1 World Phenomena

Those are the events that may occur in the real world and are not affected by the Machine.

We identify:

- *Citizen* sees a parking violation and wants to report it;
- *Authorities* want to know about some violations that have been occurred;
- A *parking violation* occurs;

1.2.2 Shared Phenomena

Shared phenomena are the events based on the link between World Phenomena and Machine Phenomena. We can distinguish them in two types:

Controlled by the world observed by the machine:

- A Citizen take a photo of a violation;
- Users can enter data for registration/login;
- Users can request data;

Controlled by the machine observed by the world:

- track position of the violation;
- mark areas with an high rate of violations;
- System can fulfill data requests;

1.2.3 Machine Phenomena

The Machine Phenomena are the events that occur inside the machine and are not affected by the real world.

We identify:

- storing permanently collected data;
- encryption of sensitive data;
- retrieving data for a request;

1.3 Definitions, acronyms, abbreviations

1.3.1 Definitions

- user: can be either citizen or authorities

traffic violation: generic violation that can occur in a street parking violation: a violation caused by a bad parking violation: general violation, identity both traffic or parking violation unsafe areas: areas with an high rate of violations

1.3.2 Acronyms

Table with all acronyms used in document.

ACRONYM	COMPLETE NAME
RASD	Requirements Analysis and Specification Document
GPS	global positioning systems
ID	Identity document
S2B	Software to be
GDPR	General Data Protection Regulation

1.3.3 Abbreviations

1.4 Revision History

1.5 Reference documents

1.6 Document Structure

- **Chapter 2:**
- **Chapter 3:**
- **Chapter 4:**

2 Overall Description

2.1 Product perspective

This section aims to explain in more detail the World, Machine and Shared Phenomena described in the previous Chapter.

2.1.1 World Phenomena

- **Citizen sees a parking violation and wants to report it:** While the Citizen is quietly walking, he sees a parking violations like a double parking or a car parked in the middle of bike lane so he logs in, takes a picture, inserts the type of violations and sends it to the *System*.
- **Authorities want to know about some violations that have been occurred:** An Authority logs in and retrieves all the violations that have been send by the Citizen
- **Parking violations occurs:** Someone in the city decides to not follow parking rules and does not park his car in a proper way

2.1.2 Machine Phenomena

- **Storing permanently collected data:** The system needs to store, in a secure way, all the data submitted. In order to achive this purpose and guarantee the best service the system needs to use a DBMS.
- **Encryption of sensitive data:** Personal user's data and all the data relative to the violations that can only be seen by authorities need to be encrypted in order to proctect it from non-allowed third parties.

- **Retrieving data for a request:** System have to fullfill the data request from the users. Data requests can be of two types, a Citizen request who want to see statistics of a certain city area or data request by Authorities who want to receive the violation reports collected by SafeStreet

2.1.3 Shared Phenomena

Controlled by the World observed by the Machine

- **A Citizen take a photo of a violation:** Situation in which a Citizen spots a generic violation and wants to report it through the application. Using the phone camera he can take the photo of the violation.
- **User can enter data for registration/login:** A user decide to use the application and provides his personal in order to register if it's the first time he use the app, or to identify himself.
- **Users can request data:** In this phenomena we make a distinction between Citizen and Authorities. A Citizen or may want to see violation statistics of a certain area, or in the case of Authorities they can, request violation statistics and additional informations

Controlled by the Machine observed by the World

- **Track position of the violation** The *System* can retrieve the position where the violation occurred by fetching it from GPS service
- **Mark areas with an high rate of violations** Once lots of violations occured, the *System* mines the information that it has in order to highlight the areas with the highest frequency of violations
- **System can fullfill data requests** After processing a request, the *System* will show to the user the result of the DBMS query in a proper way

2.1.4 Class Diagram

2.1.5 State Chart

2.2 Product functions

In this section are explained the functions associated to User.

- **Citizen functions: Report a violation**
When a Citizen sees a parking violation occurs, he takes a picture of the vehicle paying attention to focus on the license plate, inputs the type of the violation and sends it. The System will provide to add the position retrieving from GPS, to add the right time and date and to add the license plate obtained through the algorithm. **Retrieve statistics about unsafe areas**
Safestreets enables Citizen to visualize statistics about unsafe areas. SafeStreets mines the informations it has and let the Citizen retrieves the result through a clear interface containing significant plots, tables and charts.
- **Authority functions:**
Retrieve statistics about unsafe areas
Safestreets enables Authority to visualize statistics about unsafe areas. SafeStreets mines the informations it has and let the Authority retrieves the result through

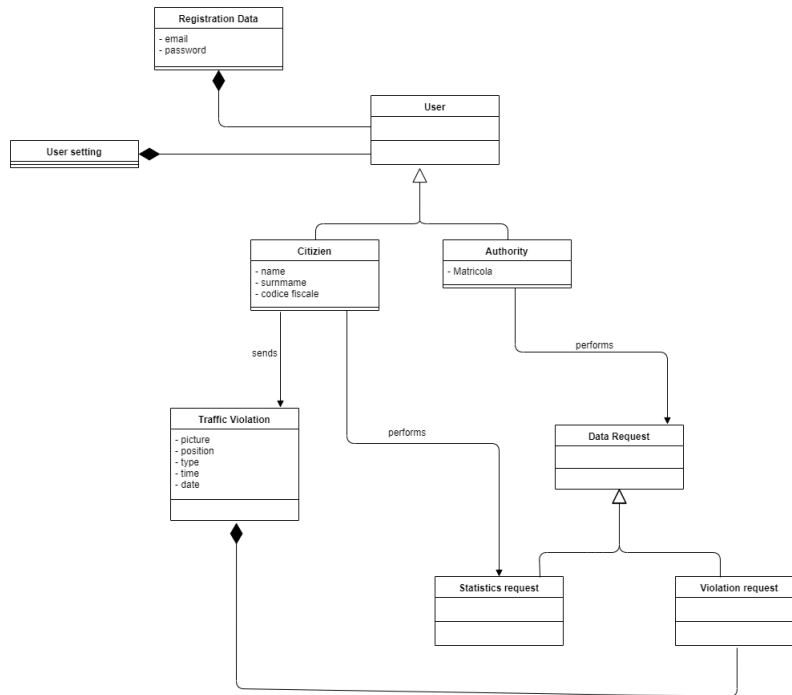


Figure 1: Safestreets' Class diagram

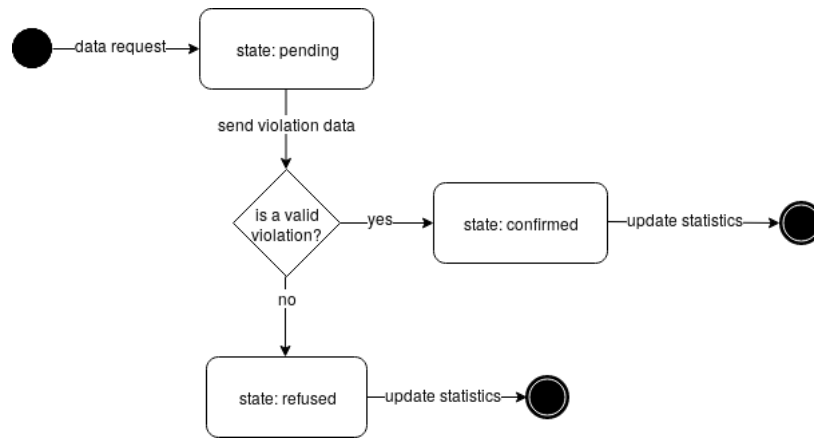


Figure 2: Authority request state chart

a clear interface containing significant plots, tables and charts. For instance Authority can understand which are the vehicles that commit the most violations or who are the most egregious offenders.

Request violations data for traffic tickets SafeStreets enables Authority to retrieve all the parking violations sent by Citizens. For each parking violation Authority can accept it or declines it. In the first case he can generate traffic ticket, in the second case he discards the information about the parking violations. In both cases SafeStreets records response in order to build statistics.

2.3 User characteristics

Below we describe the convention used to identify the user of the application and the function that those users are allow to perform.

- **Guest:** A user that have donwload the application but is not registered yet. This type of user is not allowed to access the application functionalities.
- **Citizen:** is a generic user app not related to authorities, a common Citizen that want to use the application. After the registration process and the validation of the ID card provided He can log in the application and use the functionalities:
 - report a violation
 - request informations about the statistics of a certain area.
- **Authorities:** This user is associated to the local municipal police district, any traffic warden, once registered with is matricola number and logged in have access to those functionalities:
 - request informations about the statistics of a certain area.
 - request all the violations reported from Citizens.
- **User:** can be both a Citizen or Authority type, in this document this name is used when it's not necessary make a distinction between the two.

2.4 Assumption and Dependencies Constraints

2.4.1 Assumption

The following list present all the domain assumption made.

- [D1]: Users can't make more than one account.
- [D2]: The Citizen assumes all responsibility for misrepresentation.
- [D3]: Citizens who use the application are evenly distributed in any city area.
- [D4]: The ID card present by the Citizen during the registration is valid.
- [D5]: The S2B allows to take photos from the application.

2.4.2 Dependencies

This list below represent all the dependencies that S2B need in order to work properly.

- A internet connection.
- A Photocamera (with a minimum precision of 3Mp ?(non so se va messo, algoritmo può non riconoscere low quality foto))
- A GPS.
- A Trusted external Storage for violations data

2.4.3 Constraints

- The S2B must guarantee the European data protection GDPR for user's sensitive data.
- The S2B will be used only in Italy due to personal data type like (fiscal code and police matricola).

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

3.1.2 Hardware Interfaces

The System does not offer any Hardware Interfaces

3.1.3 Software Interfaces

As mobile applications, the main software interfaces are:

- iOS
- Android

3.1.4 Communication Interfaces

HTTPS protocol: to safely communicate through the internet

3.2 Functional Requirements

3.2.1 Use Case Diagrams

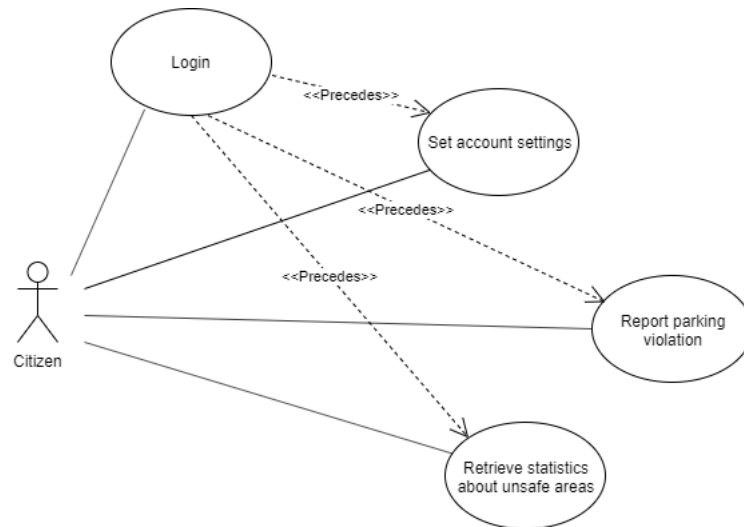


Figure 3: Citizen Use Case Diagram

ID	UC1
Description	A <i>Guest</i> creates a Citizen account
Actors	<i>Guest</i>
Precondition	<i>Guest</i> 's smartphone satisfies hardware limitations <i>Guest</i> has downloaded the app from the store <i>Guest</i> has not an account
Flow of events	<ol style="list-style-type: none"> 1. <i>Guest</i> opens the app 2. <i>Guest</i> clicks the registration button 3. <i>System</i> asks which type of account he wants to create 4. <i>Guest</i> selects Citizen account 5. <i>System</i> shows the form 6. <i>Guest</i> fills the form with his personal data plus mail and password 7. <i>System</i> checks the validity of the data inserted 8. <i>System</i> sends confirmation email 9. <i>Guest</i> receives the email and clicks the URL to complete the registration
Postconditions	<i>System</i> has stored a new Citizen account <i>Guest</i> can login as Citizen
Exceptions	<i>Guest</i> inserts an email that has been used by another account <i>Guest</i> inserts a FC that has been inserted by another account In these case <i>System</i> shows user an error message and the flow of events restart from point 5

Table 1: *Guest* creates a Citizen account

ID	UC2
Description	A <i>Guest</i> creates a Authority account
Actors	<i>Guest</i>
Precondition	<i>Authority</i> 's smartphone satisfies hardware limitations <i>Guest</i> has downloaded the app from the store <i>Guest</i> has not an account
Flow of events	<ol style="list-style-type: none"> 1. <i>Guest</i> opens the app 2. <i>Guest</i> clicks the registration button 3. <i>System</i> asks which type of account he wants to create 4. <i>Guest</i> selects Authority account 5. <i>System</i> shows the form 6. <i>Guest</i> fills the form with his personal data plus P.IVA mail and password 7. <i>System</i> checks the validity of the data inserted 8. <i>System</i> sends confirmation email 9. <i>Guest</i> receives the email and clicks the URL to complete the registration
Postconditions	<i>System</i> has stored a new Authority account <i>Guest</i> can login as Authority
Exceptions	<i>Guest</i> inserts an email that has been used by another account <i>Guest</i> inserts a P.IVA that has been inserted by another account In these case <i>System</i> shows user an error message and the flow of events restart from point 5

Table 2: *Guest* creates a Authority account

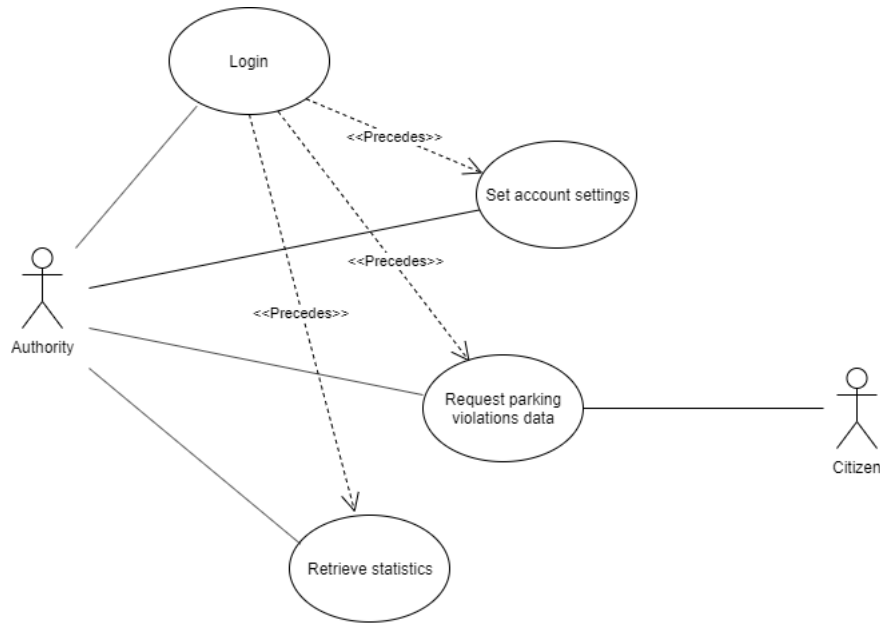


Figure 4: Authority Use Case Diagram

3.2.2 Scenarios

3.2.3 Use Cases

3.2.4 Sequence Diagrams

3.2.5 Goal Mapping on Requirements

3.3 Performance Requirements

In this section we discuss requirements for what regards performance. The System must be able to support up to 1 million of registered users. This limitation is not posed by the front-end of the System, but rather by the back-end part, specifically the DB. For the same reasons it must be able to handle up to 1 million of parking violations sent by the *Citizen*. In order to avoid any kind of saturation, every parking violation that has not been taken into account by any *Authority* for 30 days, must be automatically discarded. This operation does not update the information about statistics.

Requests about statistics shall be processed in less than 5 seconds. Requests about parking violations, instead, shall be processed in less than 1 second.

3.4 Design Constraints

3.4.1 Standards compliance

The S2B will use certain measures as:

- Standard longitude and latitude measures for the position

For what concerns the privacy, the S2B is subject to GDPR, a regulation in EU law on data protection and privacy for all individual citizens of UE

3.4.2 Hardware limitations

In order to work properly the application must rely on hardwares that have certain requirements such as:

- GPS
- internet connection (4G/3G/2G)
- Photocamera with a minimum precision of 5Mp

3.5 Software System Attributes

3.5.1 Reliability

The system must be able to run continuously without any interruptions. In order to do that, it must be ensured that the system is fault tolerant. To prevent downtime, one of the main goals of architecture design must be ensuring graceful degradation of the System

3.5.2 Availability

SafeStreets does not present any critical functions so 99% availability with 3.65 days/year as downtime should be good.

3.5.3 Security

Security is a key aspect of SafeStreets because it is very important that the informations are never altered. The S2B must:

1. use HTTPS to safely communicate with the Server and DBMS
2. Hash the passwords so that they are not stored in clear in the DB
3. Encrypt sensitive data before storing it
4. digital sign the parking violation sent by Citizen and then hash it

3.5.4 Maintainability

In order to achieve maintainability some good practices must be followed to reduce coupling and avoid code duplication

3.5.5 Portability

S2B, as it stated previously, will work both in Andorid and iOS and this ensures itself portability. For the back-end part, it should be OS independent

4 Formal Analysis with Alloy

5 Efforts