Title of the project

Matteo Secco, Mohammad Rahbari

release date: October 27, 2019
version 0: skeleton

Contents

1 Introduction

1.1 Purpose

here we include the goals of the project

The required system, called SafeStreets, is a distributed system to allow the citizens to signal parking violations to the competent autorities.

The system must allow the citizen to submit pictures of the violation, attaching data such as date, time and position. The user will have to specify the type of the violation when sending these data.

When reciving such data, the system must store them, toghether with the plate of the car that performed the violation (elictated from the picture the citizen sent), the informations about the violation itself (in particular the type of violation), and the name of the street where the violation occourred (which can be retrieved by the positioning information the user sent).

In addition, the application must allow both autorities and citizens to analyze the stored data, for example highlighting streets or plates with most violations registered. Different levels of security can be offered.

Finally, the application must be enable authorities to automatically generate traffic tickets using its data for people commiting violations, if and only if it can be verified that the data concerning it has not been altered. In this case, SafeStreets could use this informations to build additional statistics.

Goal list

`G.1` Allow citizens to notify parking violations in real time

`G.2` Allow citizens to provide all the needed data about violation, in particular infraction type, picture, date, time and position

`G.3` Prevent the autorities to have to manually address parking tickets

`G.4` Ensure no notification is saved if the notification's data has been modified somehow

`G.5` Ensure no notifications are stored if the plate of the car that committed the infringment owns a permission for that infringiment

`G.6` Every notification not covered by `G.4` or `G.5` will always be saved

`G.7` Allow both citizens and authorities retrive informations about previous violations and released tickets, possibly in an aggregated form

1.2 Scope

here we include an analysis of the world and of the shared phenomena
The world where the system must fit is an everday city, with focus on the traffic of moterized veichles.
The events the system aims to influence are the parking of motorized veichles, in particular the ones considered infractions.
In the context of the system, when any user notices an illegal parking, he/she may notify the system and provide any needed informations to the competent authorities. In particular, the notification is composed by a picture of the infraction, a timestamp (date and time), the geographical location of the infraction and the type of infraction wich is to be notified. Some of these informations can be gathered automatically from the user's device. Notice that, for legal reasons, SafeStreets will just make the data available to the auth, and will not generate tickets itself.
In addiction, the user may interrogate the system to gather aggregated informations about the locations with more violation incidence, and the cars which committed more violations.

Table 1: Phenomena

World phenomena	**Shared Phenomena**
A person parks its car somewhere	A person inserts the data about the infraction on its device
A person finds out a traffic violation	An user accesses aggregated data about previous infractions
Auth generates a ticket for some violation	Auth access data about infractions and produces tickets
	Auth accesses aggregated data on violations

1.3 Definitions, Acronyms,Abbreviations

Person: A person in the real world. Every Citizen is a person, generally an Authorithy is not

User: A person, an organization or a system which somehow uses SafeStreets

Citizen (cit): This term will be used to denote every user not owning particular privileges or permissions. A citizen is only allowed to notify violations and see some aggregated data

Authority (auth): This term will denote every user (phisical or digital) having privileged access to the stored data. An example of Authority is the Local Police.

Notification: Represents a set of data submitted by any user composed by:

- A picture of a parking infraction
- A timestamp of when the notification occourred, containing date and time (may be gathered automatically by the citizen's device)
- A geographical position of where the infraction occurred (may be gathered automatically by the citizen's device)
- The type of infraction notified

Car: The word car will be used to issue every mothorized vehicle

Plate: Identifies a car

Permisison: A document released by a verified authority, granting to a car the permission to park in a set of reserved parkings (ex: permission to park on parking reserved for disabled people).

2 Overall descriprion

2.1 Product perspective

here we include further details on the shared phenomena and a domain model (class diagrams and statecharts)

2.2 Product functions

here we include anything that is relevant to clarify their needs

2.3 Assumptions, dependencies and constraints

Assumptions:

`A.1` Different cars always have different plates

`A.2` Each car exactly has 1 plate

`A.3` No car is owned by more than 1 person

`A.4` If an authority is enabled to give tickets, that authorty will have access to every permission ever released by any authority

In particular, if the auth is evaluating a ticket for plate p, it will be able to check every permission granted to p

Constraints:

C.1 The authorities will not be able to register automatically to the service. For authentication, they'll be verified and added by an administrator of the system

C.2 Another constraint

3 Specific Requirements

Here we include more details on all the aspects in Section 2 if they can be used for the development team

3.1 External Interface Requirements

These requirements come from the needing to communicate to the local police to submit notifications and gather ticket statistics

3.1.1 User Interfaces

1. A graphical user interface should be provided to the auths to allow them to retrive dat manually

3.1.2 Software Interfaces

1. An api should be produced to allow auth's servers to access data programmatically

3.2 Functional Requirements

Definition of use case diagrams, usa cases and associated sequence/activity diagrams, and mapping on requirements

3.2.1 Scenarios

S.1 Mario is walking down the street when he comes to a cross. He would like to **cross the cross** staying on the pedestrian lines, but a monstertruck decided to park over there. So Mario needs to walk around the truck and through the street (cars are driving fast), but at least he can take a little revenge by taking out his mobile phone, opening the SafeStreets application, and in a couple passages notifying the police and knowing the crazy-minded driver will pay what he deserves! (like some fines)

S.2 **Prevent is better than healing**Luigi is late for work. He cannot find any parking, and is thinking about parking on the bycicle line. He's about to do that, when he suddently remembers that nowaday anyone could signal his infractions with SafeStreets. So he decides to keep searching for a proper parking, and granny Maria can take her daily ride on her bycicle!

3.2.2 Requirements

`R.1` Requires the authorities to give SafeStreets access to the tickets they emitted using SafeStreets' data

`R.2` Automatically extract the plate number of the car from the photo, ignoring cars in the background if present

`R.3` Ensure no data is altered from the insertion to the eventual storage

`R.4` For each data to insert, if the user's device has a sensor to collect thatkind of data, that sensor should be used instead of manual insertion (example: GPS)

`R.5` The system should notify the user when his notification has been processed correctly, or ask for more datailed data (example: a more focused picture) if needed.

3.3 Performance requirements

1. The notification should be sent to the server as soon as possible. It should be recived and elaborated as soon as internet and the server's speed allows.
2. The quality of the photo should be at most XYZ (to reduce overload on network and server) but at least PQR (to enable automatic recognition.

3.4 Design Constraints

3.4.1 Hardware limitations

The aut's hardware only needs to have access to the internet. The cit's device must be a smartphone so to ensure the photo is taken inside the application and no modification happens

3.4.2 Any other constraint

If possible, it should be checked that the cit's device has not been modified anyhow

3.5 Software system attributes

3.5.1 Reliability

The only critical part of the system is the one concerning data mining, as people could exploit it to decide about it's daily routine. For this reason, it's recommended to make the result of the mining (old ones if needed) more available than the rest of the system. In addiction, the system must avoid to generate fake notifications because of system faults to prevent innocent people from getting an unfair ticket.

3.5.2 Availability

As the system does not concern life-critical events, we think an availability of 99% of the time should be enought to ensure a good experience for both cits and auths.

3.5.3 Security

1. Transmit data encrypted to prevent alteration in the network
2. Only allow auths to access the license plate numbers stored

3.5.4 Maintainability

Maintainability is not a crucial issue as no updates are planned for the system.

3.5.5 Portability

Portability is a crucial aspect of the system, at least on the citizen's side: it is needed to allow them to submit a notification as they spot a violation. For this reason, the cit's client should be avaiable on as most mobile devices as possible.

3.6 Traceability matrix

Table 2: Traceability matrix

Goal ID	Req ID	Usecase ID
G.1	2.2	S.1

4 Formal analysis using alloy

This section should include a brief presentation of the main objectives driving the formal modeling activity, as well as a description of the model itself, what can be proved with it, and why what is proved is important given the problem at hand. To show the soundness and correctness of the model, this section can show some worlds obtained by running it, and/or the results of the checks performed on meaningful assertions

The signatures are omitted in the document as they reflect the entities desctibes in section 1.3, plus some "enumerations". The goals we were able to define in Alloy are G.4,G.5 and G.6. There was no need to check G.7 and G.2 as they can be satisfied by an equal requirement

The assumptions we were able to define are `A.1`, `A.2`, `A.3`, `A.4`. Please note not all of them has been expressed by facts as some was already expressed by multiplicity in the signatures.

5 Effort spent

In this section you will include information about the number of hours each group member has worked for this document

Matteo Secco

- October 15: created the skeleton. 1h
- October 16: Introduction and Scope. 30m
- October 16: Definitions. 15m.
- October 23: Scenarios and goals. toghether. 1h
- October 24: Update goals, assumptions, alloy skeleton. 2h
- October 24: Teamwork. 2h

Rahbari

- October 23: Scenarios and goals. toghether. 1h
- October 24: Teamwork. 2h

6 References