RASD

Introduction

SafeStreets is a crowd-sourced application whose intention is to notify the authorities when traffic violations occur. The system should provide the possibility for users to give all needed information for authorities to take actions against the violations and improve the service provided by authorities taking care of violations notification from the user notifying it until the violation is resolved.

The sources of notifications are the citizens which takes photos of violations and sends them to the authorities through the application.

The information provided by users are integrated with other suitable information and are stored by the service.

The system also runs an algorithm to read the license plate of the vehicle in the photos. All collected data can be mined by citizens and authorities to find which streets are the safest. Users can have different levels of visibility authorities must be able to know the license plates of vehicles in the photos while normal users can only see data in the form of statistics.

Moreover, data is sent to the municipal district so that important information can be extracted through statistics in order to make decisions to improve the safety of the area.

Ultimately, the system will have to be easy to use, reliable and highly scalable to fit perfectly with the mutable context in which it will end to be used.

Current System

SafeStreets aims to become the smartest and quickest way to report violations in Italy. The current systems are phone calls which can take lot of time, take up the phone lines for more critical events and lacs evidence of the violation or other system like the site www.poliziamunicipale-online.it which is similar to phone calls , is more strict , the interface is not user friendly and it lacs possibility to give evidence of violations too. The idea of our service is not unique there are already similar services in other countries, like in India they have “Public Eye. OFFICIAL BTP APP”

GOALS

USER:

G1) Notify authorities about traffic violations

G1-1) Send picture of violation

G1-2) Send Position of the violation

G2) Allow authorities to reserve an assignment

G3) allow authorities to report a finished assignment

SafeStreets:

G4) Allow a User to join the system registering him/herself to ensure reliability of the information provided by them

G5) Store information provided by users:

G5-1) Complete it with metadata

G5-2) Mine information

G6) Identify potentially unsafe areas:

G6-1) Suggest possible interventions

G7) Allow municipality to register Authorities to the service

Security Goals:

S1) Offer different levels of visibility to different type of users

S2) Personal data of users are stored respecting current security standards

SCOPE (TODO order to be fixed)

World phenomena:

1)Violation

2)Intervention of authorities

3)Municipality put into effect interventions to improve safety

Machine phenomena:

1)Shortest path calculation for authority’s intervention

2)the creation of an object of type violation

3)run algorithm to identify the license plate/s in the photos

4)database queries

5)schedule most efficient path to look up the notified violations

6)periodically run algorithm to suggest possible interventions to municipality

Shared phenomena:

1)user notify the system about violation (observed by the system controlled by the world)

2)send notification to authorities (controlled by system observed by world)

Definitions

-Violation: parking violations which can be notified by citizens to authorities

-Mapping System: external software that provides maps and directions to reach the position of a violation

-Licence plate Recognition Algorithm:

-Spam:

ACRONYMS

RASD: Requirement Analysis and Specification Document.  
API: Application Programming Interface  
GPS: Global positioning system

ABBREVIATIONS

• [Gn]: n-th goal.

• [Dn]: n-th domain assumption.

• [Rn]: n-th requirement.

1.4. References

• Specification Document: “Assignments AA 2016-2017.pdf”.

• GPS Performances: “http://www.gps.gov/systems/gps/performance/accuracy/".

• Alloy Dynamic Model example: “<http://homepage.cs.uiowa.edu/~tinelli/classes/181/Spring10/Notes/09-dynamic-models.pdf> "

• IEEE Std 830-1993 - IEEE Guide to Software Requirements Specifications.

• IEEE Std 830-1998 - IEEE Recommended Practice for Software Requirements Specifications.

•<https://milano.corriere.it/notizie/cronaca/18_dicembre_12/milano-allarme-sosta-selvaggia-ogni-giorno-divieto-centomila-auto-solo-3percento-sanzioni-abe397ce-fe44-11e8-89a1-ceb28fd9db2c.shtml?refresh_ce-cp>

OVERVIEW //TODO when everything else is done

2 Overall Description

2.1 Product perspective

The system will be developed from scratch using external elements such as Mapping systems and algorithm to recognise licence plates. We choose to take those external elements to decouple mapping problems from our project implementation and to use already tested algorithms to recognise licence plates.

2.2 Product functions

In this section we provide a list of functionalities offered by our system. We will describe those functionalities and in later section we will better analyse interactions of users with those functions.

2.2.1 Mapping System

An external mapping system will be used to guarantee better performances than a system to implement from scratch.

This won’t ensure always the correctness of information given by that system. Some issues about mapping systems like accidents occurred and blocked streets can’t be addressed in real time. Those kinds of problems may need to rely on authority’s knowledge of the area to be overcome.

2.2.2 Licence Plate Recognition Algorithm

An external Licence Plate recognition algorithm will be adopted in our system. There are a lot of services provided online and some open-source solutions. Those systems are used by several people and companies and are tested so most kind of issues that may occur has already been noticed and fixed. Considering our system is to be launched in Italy we will consider a solution better suited to recognize European licence plates.

2.2.3 Municipality Servers maintenance

Our system can’t address problem of municipality server’s unavailability issues. If a server is unavailable and needs maintenance statics for the area covered by that server may become unavailable for an indefinite amount of time. To solve this problem SafeStreets may use the email address used to register the municipality to contact it and to point out the issues.

2.3 User Characteristics

2.3.1 Actors

-Visitor: a person using SafeStreets without being registered. He/she can only see statics, register or sign-in to be recognized as a User

-Registered User/ User: term used to identify any person which uses our application and has registered to our service:

- Citizen: Is a User who provide the system information and are the main contributors to the service. They provide information about violations with photos and possibly some notes. They can access data gathered by the system in form of statistics.

-Municipality: Users managing local systems in each area. Those users should be able to take decisions to change unsafe areas thanks to their status.

-Authority: police agents. They are invited to use the service by municipality users who can ask creation of their account. They can reserve assignments of violations to be addressed. They can also refuse the assignment, mark it as spam or send it to another authority.

2.4 Constraints

2.4.1 Regulatory policies

The system will ask user for minimal information to recognize them. The visitor should give the system only his /her email address and provide the system a username and a password to create an account. Email addresses won’t be used for commercial uses and will be stored only to give the possibility to recover an account in case the user loses his/her credentials.

2.4.2. Hardware limitations

• Mobile App

- Android smartphone

- 2G/3G/4G connection

- GPS

• Web App

- Modern browser able to retrieve user's location

2.4.3 Interfaces to other applications

In the first release no public interfaces will be opened and SafeStreets will only communicate with municipality servers to retrieve useful information about accidents.

2.5. Assumptions and dependencies

2.5.1 Text assumptions

2.5.2 Domain assumptions

D1) For each notification data and metadata about the violation are correct

D2) Accounts are personal and login credentials never gets stolen

D3) Each fiscal code number is unique

D4) Authorities always intervene in case of a notified violation

D5) Information about authorities’ location are always available through GPS

D6) Only agents close to the violation area are notified //TODO check this

D7) Citizen sends only clear photos (if it is not clear he/she would retake the photo)

D8) Municipality and authorities respects their duty of care

3. Specific Requirements

3.1. External interface requirements

3.1.1. User Interfaces

3.1.1.1. Statistic

3.1.1.2. Login

3.1.1.2.1 Citizen Interface

3.1.1.2.2 Municipality Interface

3.1.1.2.3 Authority Interface

//TODO add photos

3.2. Functional requirements //TODO add requirements

G1) Notify authorities about traffic violations

R1)

G2) Allow authorities to reserve an assignment

G3) allow authorities to report a finished assignment

SafeStreets:

G4) Allow a User to join the system registering him/herself to ensure reliability of the information provided by them

G5) Store information provided by users:

G5-1) Complete it with metadata

G5-2) Mine information

G6) Identify potentially unsafe areas:

G6-1) Suggest possible interventions

G7) Allow municipality to register Authorities to the service

3.3. Non-functional requirements

3.3.1. Performance

The system should be able to respond to a possibly great number of simultaneous requests. Based on data about Milan there are over 100.000 parking violations a day, the system should be able to keep track of at least 100 times that number of notifications a day. The number of violations that could be taken care using the app could grow eventually to cover even special cases like when there are strikes of public transports when the load of violation considerably increases.

3.3.2. Reliability

The system should be available 99,99% of time. Considering only one state also gives a time range in which notifications will be considerably reduced (the night-time where citizen will less likely report a violation) and so reliability constraint for night-time could be reduced to 99% time also reducing resources allocated to the system.

3.3.3 Security

Users credentials will be stored. Security of the data and of the communications user-system is a primary concern.

3.3.4 Scalability

This system is designed to be optimized in Italy. It is possible later to expand it choosing a more suitable algorithm to recognize licence plates and dividing computation of different states on different servers so that reliability analysis made for Italy are still true for every single state. Information about boundaries of states should be replicated in both states making the transition from one server to the other smoother.

3.3.5. Accuracy

Accuracy of the position of authorities and violations has to be the best possible. All the sensors used must provide positions' data with an error lower than 20 meters. We can consider a larger bound of accuracy because authorities work on a given area so from a well taken photo, they should be able to recognize the place of the violation even if the position given by the GPS is not too accurate.

4. Scenarios

//TODO add scenarios here

5. UML modeling

5.1. Use case descriptions

5.2. Use case diagrams

5.3. Class diagram

5.4. Statechart diagrams

5.5. Sequence diagrams

6. Alloy modeling