RASD

Carlo Dell'Acqua, Adriana Ferrari, Angelica Sofia Valeriani

October 21, 2019

Contents

Chapter 1

Introduction

1.1 Purpose

1.1.1 General Purpose

SafeStreets is a service that aims to improve the safety of the general traffic. This is achieved by creating a community of users who are able to report any violation they see while the system manages all the aspects of data validation and statistical analysis. Different services contribute to this purpose:

- The first service offered by the front end application is the report service. Any registered user can submit violation reports and SafeStreets will validate them as described in the following sections with the help of the community.

- The second service offered by the front end application is the the Unsafe Areas Map. SafeStreets will provide statistics about areas that have a higher risk of violations based on the reports it receives and the danger of the infractions. Data can also be collected from public services if available to increase accuracy.

- The third service is the ticket generation. Traffic policemen will have access to a dedicated section of the application where SafeStreets will collect validated reports. This will enable any registered policeman to take actions against those violations.

1.1.2 Goals

The following goals describe the key points of our system:

- G1: the system will provide an easy interface for violation reports

- G2: the system will provide aggregate anonymous data to show unsafe areas

- G3: the system will show violation reports to registered authorities

- G4: the system will integrate a validation procedure that will be followed strictly for each submitted report

- G5: the system will provide secure authentication methods

- G6: user data will only be available to the user itself, authorities and SafeStreets authorized employees

1.2   Scope

SafeStreets is a service that is available to any citizen having a valid ID number. The service creates a direct channel to communicate with authorities and with other users about the observed traffic violations, promoting street safety and helping others who might be in need, for example by reporting a vehicle which is blocking an access for disabled people. Registered users can submit reports that are going to be validated by other end-users, creating a network of trust. The reports they file provide information about the violation, the location and the license plate of the subject. This makes reports detailed enough for approval by public officers, who are able to review them and take proper action to discourage further infractions.

Chapter 2

Overall description

2.1   Product Perspective

The core of SafeStreets is the opportunity to create a community that cooperates to reduce traffic violations: for this to be possible, it is necessary to integrate third party services with our software. Our software works with the APIs provided by Google Maps to best help the users identify the location of the violation they want to report. We also work with different municipalities that help us provide better statistics by sharing with us their data about traffic violations, so our software needs to interact with their APIs and reduce the differences in communication standards as much as possible.

2.2   Product Functions

The following section contains the main product functions of SafeStreets. Some of them are available to everyone, some only to registered users and others only to authorities.

2.2.1   Show safe and unsafe areas

Who has access:
    Anyone

Description:
    SafeStreets helps people understand which parts of their town, or any customizable region, are more dangerous because of traffic violations. A map highlights areas differently according to the frequency and severity of violations that occur. The data used to build these statistics is both entered by SafeStreets users and, whenever possible, integrated by data provided by the municipality.

2.2.2 Submit and confirm violations

Who has access:
Registered users

Description:
SafeStreets core functionality is allowing users to submit documentation about traffic violations they see. Only registered users have access to this functionality because it is necessary to be able to connect a report to the person who filed it, as intentionally filing fake reports may have legal consequences. In order to keep fake reports to a minimum, all violations need to be approved by an established amount of SafeStreets users and only then are they shown to the authorities. The ratio of approved versus non approved reports of a user, as well as the actual amount of violations they reported, is also taken into account when computing the reliability of a user.

2.2.3 Allow authorities to take action

Who has access:
Registered authorities

Description:
Authorities have access to another section of SafeStreets, which allows them to see more data about who submitted and committed violations. They can check the latest reports and either send a patrol in the area or, if the violations respects certain requirements, even directly send tickets to the culprit.

2.3 User Characteristics

- **Visitor**: a visitor is a person who visits SafeStreets without being registered. He can only proceed to registration in order to have complete access to system services, otherwise he can't have access to any service or data.

- **User**: a registered user, called simply user, is a person who registered to SafeStreets, providing his personal data. He can login to the system through provided credentials to exploit full services.

- **Authorities**: authorities are special users, they are persons that registered themselves as authorities to SafeStreets, demonstrating before completing registration their effective identity as authority, and have access to more advanced functions of the system.

- **Municipality**: municipality is the local authority, it generates traffic tickets using verified data provided by SafeStreets.

2.4 Assumptions, dependencies and constraints

2.4.1 Domain assumption

- Personal information that a visitor has to provide to become a registered user are name, surname, email, ID number and password

- Emails are unique and a user can only be associated to one email address

- After registration the system sends an email to the user containing a recap of the information they provided and an activation link

- Once the activation link is opened in a browser the email address of the user is verified and the user is automatically logged in

- Two-factors authentication can be enabled by the user by providing a phone number (SMS authentication) or by scanning a QR Code with an authenticator app (Token authentication)

- Once logged in a user can notify traffic violations

- Pictures with a resolution of less than 2MP are automatically rejected

- ————————————————————— ALT1

- An unsafe area, represented as a circle, is described by the coordinate and the radius as the maximum acceptable distance from the center of the area

- Different unsafe areas do not overlap between them

- Every area that is not unsafe is considered safe

- Unsafe areas are determined by statistics made by the system, according to the areas where the maximum number of violations occur

- ————————————————————— ALT2

- Users can view a custom map that is updated periodically and shows unsafe streets by highlighting them with a gradient of colors from yellow to red which indicates the level of danger

2.4.2 Dependencies

- The authorities will be in charge of every effective measure of security taken to make streets safer

Chapter 3

Specific requirements

3.1   External Interface Requirements

3.1.1   User Interface

The following mockups represent a basic idea of how the Mobile Application and the Web Interface are supposed to look like. Users can access to complete SafeStreets functionalities through the smart-phone application, and they will notify violations through it.

On the other side, SafeStreets provides a Web Interface for users. As well as on the mobile phone, they can exploit all the functionalities provided by the application, such as the registration or the notification of a violation.

3.1.2   Hardware Interfaces

Since the application must run over the Internet, all the hardware shall re- quire to connect network will be an hardware interface for the system, both server and client side.

- Server-side: e.g. Modem, WAN - LAN, Ethernet Cross-Cable.

- Client-side: e.g. Wi-Fi 802.11ac, 3G/4G.

3.1.3   Software Interface

SafeStreet provide an API, besides the Web Application Interface. In a more detailed way, it will provide an API that allows third parties to access the entire set of functionalities provided by the system. In this way third party companies and applications can embed in their system the functionalities concerning with the individual access requests and sampling requests.

3.1.4 Communication Interface

SafeStreets must rely on the newer TLS version 1.3, released in 2018, to guarantee the best security possible. This version, in comparison to the previous one (TLS v. 1.2) speeds up encrypted connections; in fact the handshake phase is executed with only one round-trip, which cuts the encryption latency in half. The use of TLS version 1.3 is guaranteed at least during the HTTPS connections involving messages carrying credentials or other sensible data.

3.2 Functional Requirements

3.2.1 R1: Allow any citizen, as a visitor, to register on the mobile phone app or the web app and to become a user by providing a document

- The system must allow the visitor to provide credentials and personal data

- The system must verify the correspondence between the ID number provided by the visitor and their personal information

- The system must allow the visitor to verify the account with an e-mail or SMS verification code

- The system must send a recap of registration to email of the visitor

- The system must verify that there are no other registered users or authorities with the same e-mail or ID number

- The user must accept users data privacy conditions to successfully register to the system

- A citizen can register to SafeStreets by providing their ID number

- A citizen can register to SafeStreets by providing their ID number

- The system must allow a user to register with an email and a password, that will be asked every time they log in

3.2.1.1 Scenario 1

Mario Rossi saw the advertisement of SafeStreets and decided to download the mobile application in order to be allowed to notify violations and improve safety of streets in his city. After opening the new app, he is asked to

fill a form with all his personal information, full name, email address, mobile number, ID card, and phone number. To proceed with his registration, after having filled all the form, including the users data privacy conditions, Mario clicks on the "Create account" button. The system verify the correspondence between inserted personal data and the ID card information, and if the verification is positive, Mario is asked whether he prefers to confirm his registration

by phone number or by email. He chooses the alternative he prefers, then he immediately receives the verification code on the account he chose. He can confirm his registration by clicking on the verification code. The system then sends a recap to notify the positive outcome of registration. Mario is now a Registered User of SafeStreets; he can login to notify violations on the streets and is enabled to the use of all the functions for a basic user (not an authority) that the system provides.

3.2.1.2 Use case: Visitor registration on mobile phone

Visitor Registration

Name	Registration to SafeStreets
Actor	Visitor
Goals	R1
Entry conditions	The visitor must have downloaded the mobile phone application or the visitor must have acceded the web interface
Event flow	1. The Visitor must fill all mandatory fields in the registration form 2. The Visitor must click on the "Create account" button 3. The system validate the provided data 4. The Visitor chooses how to confirm his registration (phone number/email) 5. The system sends the code on the visitor's chosen mode, mobile phone or email address 6. The Visitor clicks on the confirmation code 7. The system sends a recap of registration, as a positive outcome 8. The system saves the Visitor's data
Exit conditions	The Visitor has become a registered used
Exceptions	- The Visitor is already registered - The email is already registered - The ID card is already registered - Visitor's data don't match with ID card data - Some mandatory fields are not filled

3.2.2 R2: Allow a user to send pictures about street and parking violations

- A user can submit a picture to the system whenever they see a street violation

- SafeStreets analyses the image to read the license plait of the vehicle

- The user can manually attach the license plait to the image, to help the system with its validation

- The user must provide the type of violation, either selecting it from a list or manually typing it if it is not already in the system

- The user can help localizing the violation by inserting the name of the street where it occurred

3.2.2.1 Scenario 2

Mario Rossi is a user of SafeStreets, he has just finished working, and he wants to go back home, but he is prevented by a car, that is in double-row parking with his own. He waits for ten minutes to see if the owner of the car is coming, but then decide to notify the violation. In order to notify the violation he must have logged into the mobile application, inserting his data, so he does; then he makes two pictures representing the violation, then he clicks on the button to notify a violation, he inserts time (5.55 p.m.), position (Via Botticelli, Viale Romagna, 2, 20133 Milano MI), a brief description of the violation, and the two pictures. He is asked to click the category of the notified violation (Double-Row Parking), or to add it, in the field "other" if the type of violation is not still present in the system. Then, he is asked to insert the license plate of the vehicle involved in the violation in the mobile application, he does, then the app verifies that it corresponds to the license plate identified with the photograph.

3.2.2.2 Use case: User sends pictures on mobile phone to notify a parking violation

User sends pictures to notify violation

Name	Notification of violation by a user
Actor	User
Goals	R2
Entry conditions	The user must be registered to the application, and he must have logged
Event flow	1. The user makes pictures of the violation 2. The user must click on the "Notify a violation" button 3. The user inserts data of the violation 4. The user includes the two pictures 5. The user checks whether the category of notified violation is present or not 6. The user chooses "Double-Row Parking", then submits 7. The system ask the user to insert the license plate (notified violation includes a vehicle) 8. The user inserts license plate and submits 9. The system validates the license plate
Exit conditions	The violation has been notified to the system
Exceptions	- The user is not registered - The user is not logged - The chosen category of the violation includes a vehicle but the license plate is not manually inserted by the user - The license plate notified by the user doesn't correspond to the license plate identified by the pictures sent

3.2.3 R3: Validation of reported violations

- Whenever a violation is reported by a user, it is sent to a random group of k SafeStreets users

- Users who receive the notification about this approval process are then asked to approve or reject the report or declare neutral if they are not able to verify

- The report is validated if and only if the algebraic sum of approvals (+1), neutrality (0) and rejections (-1) is at least v

12

- Every user is supposed to respond, the system expects to receive k answers

- If the answers of the users for validation, after a settled timeout, are not all received, the answer is automatically considered neutral

3.2.3.1   Scenario 3

SafeStreets has received the notification of Mario Rossi regarding a violation of double-row parking. To verify the violation, it sends a request to a number k of users. The number of users is randomly chosen, and depends on the number of users logged in the application at the time the system received the report of Mario Rossi and the users that were logged just a few time before, to increase probability they will be available to answer the report. K users receive the request for validation, and they start answering the report. The system receives reports and makes the algebraic sum of approvals and rejections. The calculated sum is equal to m and comparing the values, the result of the inequation $m > v$ is positive; as a consequence, the system validates the report and registers the violation.

3.2.3.2   Use case: The system validates a reported violation

User sends pictures to notify violation

Name	Validation of violation
Actor	k users
Goals	R3
Entry conditions	A violation must have been notified by a user
Event flow	1. The system sends a request to a number k of users 2. Every user, that receives the request and is able to answer, does 3. The system counts the answers and makes the algebraic sum, considering that: - +1 is a confirmation of the violation - 0 is an absent response or an agnostic report of the violation - -1 is a refusal of the violation 4. The system compares the calculated sum with the memorized acceptance threshold value 5. The system validates the report 6. The system registers the violation
Exit conditions	The system has validated and registered the violation
Exceptions	• The calculated sum is smaller then the memorized acceptance threshold value • If no requested user answers the notification is automatically discarded

3.2.4   R4: Allow a user to see the areas where a violation is more likely to happen

- SafeStreets can aggregate data inputted by users to show relevant statistics about the frequency of street violations

- Users can see how many violations usually happen in their neighborhood, in their current location or in an area of their choice

- A user has access to the type and amount of violations that happened

- Users can insert a specific address to check

- The system answers showing the near unsafe area, if the address is located in a safe area

- The system answers showing more detailed data about frequency of car accidents and violations, if the address is in an unsafe area

3.2.4.1 Scenario 4

Mario Rossi searches for a language school to improve English for his daughter, he founds different schools in his city on the internet, but he wants an area not too far from his house where the little girl can securely go also on foot. He has a list of five schools to decide. He is a user of SafeStreets, so for every school in his list, he looks on the web interface of the application for the statistics about car accidents in all the five zones. He inserts the address of the school, and the system responds with a map, showing the near unsafe area if the school is in a safe area, or, if the school is in an unsafe area, showing more detailed data about frequency of car accidents and violations. This operation is repeated five times.

3.2.4.2 Use case: User looks for statistics for unsafe areas

User looks for statistics for unsafe areas

Name	Viewing of statistics
Actor	User
Goals	R4
Entry conditions	The user must be registered to the application, and he must have logged on the web interface
Event flow	1. The user must click on the "View statistics" button 2. The user must insert the address he looks for 3. The system shows a map with the area surrounding the inserted address 4. The user repeats this operation five times
Exit conditions	The user has seen the statistics he required
Exceptions	- The user is not logged - Some fields of the address are wrongly inserted - Some fields of the address are not filled

3.2.5 R5: Register the violation

- The reported violation must have been validated by the system

- The system checks the type of violation

- In case the violation is already supported by the system, the number of violations of that type is increased, the license plate of the involved vehicle is added to the list of offenders, and in a report that SafeStreets will deliver to authorities

- In case the violation is not supported by the system, the system updates the list of supported violations, then proceed as in the previous case

3.2.5.1 Scenario 5

Luca Bianchi is in his lunch break, with his colleagues, and he is waiting for crossing the road on his way to go back to the office. While stopped at the semaphore, he sees a cyclist crossing the road with earphones. The intersection is very busy, and this behavior is dangerous either for the cyclist himself or for those arounds. Luca Bianchi is a user of SafeStreets, so he immediately takes a picture to catch the moment. He logs into the mobile application, inserting his personal data, then he clicks on the button to notify a violation, he inserts time (1.40 p.m.), position (Viale Zara, 100, 20125 Milano MI), a brief description of the violation, and the two pictures. He is asked to click the category of the notified violation; as he doesn't manage to find the correct category, he clicks on the field "other", in order to add the type of violation as it is not still present in the system. He inserts a name for the type of violation "Guide with earphones on the bike", clicks on the field that says that the violation doesn't include a vehicle (in the system a violation of type "Guide with earphones in the car" already exists, but it requires also the specification of the license plate as the violation includes a vehicle). The system checks the inserted category doesn't overlap with the existing ones, adds it to the system, then proceeds as in scenario 3 to validate the notification. If the notification is validated and a policeman is working in that area, when he receives the notification of "New infraction", he can verify himself and generate a traffic ticket.

3.2.5.2 Use case: A violation is registered

User notify a violation, that is registered

Name	Registration of a notified violation by a user
Actor	User
Goals	R5
Entry conditions	The user must be registered to the application, and he must have logged
Event flow	1. The user makes pictures of the violation 2. The user must click on the "Notify a violation" button 3. The user inserts data of the violation 4. The user includes the two pictures 5. The user checks whether the category of notified violation is present or not 6. The user clicks on the button "Other" to add a violation 7. The user writes "Guide with earphones on the bike" 8. The user clicks on the button "No vehicle involved", to say the new type violation doesn't include a vehicle 9. The user submits 10. The system checks the new category doesn't overlap with existing ones 11. The user inserts license plate and submits 12. Validation procedure follows structure of use case associated to Scenario 3
Exit conditions	The user has registered a new type of violation
Exceptions	- The user is not registered - The user is not logged - The added category (or similar) already exists

3.2.6 R6: Allow authorities to see data about violations and who committed them

- Authorities can access all data about violations that a standard user can access

- Authorities also have access to more specific data about who committed a violation, like the license plait of the car or how many infractions have been associated to a specific car

- In case the validated report includes a vehicle and the license plate is correctly identified, a randomly selected authority approves the forwarding of the identity of the offender to the municipality, in order to allow them generating a traffic ticket

3.2.6.1 Scenario 6

A notification of a violation including a vehicle has just been made by a registered user and has been validated. The type of violation is "Stop in front of driveway", and the offender car, that has been identified with its license plate, was already present in the list of offenders that SafeStreets uses to keep track of most frequently guilty vehicles. The system updates the number of infractions related to that car; besides, an authority is randomly chosen. This authority checks the correct update and then enables the system forwarding data of offender to the municipality in order to allow local police generating a traffic ticket. The selected authority is also enabled to see the history of violations regarding the offender and attaches this information to the data sent to the municipality.

3.2.6.2 Use case: Viewing of data about violations and offenders by authorities

Viewing of data about violations and offenders by authorities

Name	An authority sees offenders profile and data
Actor	Authority
Goals	R6
Entry conditions	A notification, including a vehicle, must have been made, validated and registered
Event flow	1. The system has identified the license plate of the offender 2. The system update number of violations associated to that vehicle 3. The system randomly chooses an authority 4. The authority checks the correct update of data 5. The authority enables the system forwarding data of the offender to the municipality 6. The authority checks the history of violation related to the offender 7. The authority enables the system sending this information to the municipality
Exit conditions	The municipality has been forwarded data of an offender validated by SafeStreets
Exceptions	- Data regarding the number of violations of the offender are not correctly updated - The license plate of the offender is not identified - There are no available authorities

3.2.7 R7: Allow authorities to register with a special user profile

- Authorities must first register as citizens

- A standard user profile can be upgraded to authority profile if it is verified by the system

- To obtain privileged access, the user must provide a valid document that proves their authority status

3.2.7.1 Scenario 7

Paolo Brambilla is a policeman, he wants to register to SafeStreets, so by his tablet he accedes the web interface, inserts his data for standard user registration (see Scenario 1), then he is a registered user. He wants to update his profile to be enabled, as a member of the local police, to accede the most advanced functions of the system. He must click on the button "Update profile", then he is asked to write his police ID badge number and his actual district, then clicks on the button "Update to authority". The system verifies his ID badge number, sending it to the municipality that confirms or discard the identification of the policeman, checking his personal data, his area of competence and his badge. If the authority identity is validated, the profile is correctly updated. An email of confirmation of updating is sent to Paolo Brambilla's address. Now he can log as an authority to SafeStreets and is enabled to all the advances functions of the system. If the authority identity is rejected, the system doesn't change anything in the user profile.

3.2.7.2   Use case: User registers as an authority

Registration of an authority

Name	Registration of an authority
Actor	Visitor/User/Authority
Goals	R7
Entry conditions	The user must have accessed the web interface
Event flow	1. The Visitor registers as in use case of scenario 1 2. The user must click on the "Update profile" button 3. The system ask to insert data for proving authority role 4. The user inserts the police ID badge number and his district 5. The user must click on the "Update to authority" button 6. The system sends data to municipality 7. The municipality validates data 8. The system updates the user profile to authority level 7. The system sends an email of confirmation 8. The system saves the authority's data
Exit conditions	The user has become an authority
Exceptions	- The inserted district is not valid - The police ID badge is already registered - The police ID badge is not valid - User's data don't match with data of municipality to confirm authority role - Some mandatory fields are not filled

3.2.8   R8: Unsafe Areas Identification

- SafeStreets can cross information from different sources to identify potentially unsafe areas

- SafeStreets may be integrated with a public service, offered by the municipality, that provides such information

- Once identified, SafeStreets can suggest possible interventions to prevent accidents

- The system checks whether the selected areas (from data for new statistics) are considered safe or unsafe

- If the area is considered unsafe, SafeStreets runs an algorithm to add received data and generate new percentages, combining new data with the already existing ones

- If the area is considered safe, the information received is simply overwritten and the area is marked as unsafe

3.2.8.1   Scenario 8

Paolo Brambilla and his colleagues are authorities registered on SafeStreets, they work for local police. This month a huge number of car accidents has been registered by the municipality, the 65% of them in the area of San Basilio, Roma RO, Italia. As both members of municipality and authorities of SafeStreets they are appointed by municipality to collect all data regarding the unsafe identified area. Then they log into SafeStreets by web interface and they upload information about types of accidents, percentage of injuries, percentage of frequency on SafeStreets. The system receives data for new statistics, checks whether the selected areas are considered safe or unsafe. The area is considered unsafe, so SafeStreets runs an algorithm to add received data and generate new percentages, combining new data with the already existing ones. Then saves results.

3.2.8.2 Use case: Authorities provide unsafe areas identification

Unsafe areas identification

Name	Unsafe areas identification
Actor	Authorities
Goals	R8
Entry conditions	The authorities must have been registered to SafeStreets
Event flow	1. Members of municipality and authorities collect data about types of accidents and percentages 2. The authorities log on the web interface 3. The authorities upload the information collected 4. The system receives new data 5. The system checks whether the selected area is considered safe or unsafe 6. The system finds the area is unsafe, so it runs an algorithm to add received data 7. The system's algorithm generates new percentages, combining new data with the already existing ones 8. The system saves results
Exit conditions	New statistics have been saved by the system
Exceptions	 • The authorities are not registered • Data inserted are not correctly uploaded • Data of violations and accidents are not correctly inserted in fields • The algorithm doesn't find a suitable solution, combining data

3.2.9 R9: Traffic Ticket Generation

- SafeStreets can generate tickets for traffic violations reported from registered users

- The information is kept safe and intact through the chain of custody, from the end user to the local police officer issuing the tickets

3.2.9.1   Scenario 9

It continues from scenario 6. The system enables just one authority to keep the chain of custody of information. As soon as the system validates the violation, the information regarding the license plate is immediately locked under the supervision of an authority, and the general statistics are updated. Only the approval of the authority will enable the forwarding towards municipality on a safe channel, reserved to authorities.

3.2.9.2   Use case: An authority provides municipality of information and traffic ticket is generated

A traffic ticket is generated

Name	A traffic ticket is generated
Actor	Authority
Goals	R9
Entry conditions	The authority must be registered to the application, and he must have logged
	A violation must have been notified
Event flow	1. The system enables an authority to keep chain of custody
	2. The system validates the violation and updates statistics
	3. The information is locked under the supervision of the authority
	4. The authority checks the system's correct update of statistics
	5. The authority approves the system forwarding offender's license plate towards municipality on a safe channel
Exit conditions	Traffic ticket is generated
Exceptions	- The chain of custody is broken (encryption algorithm fails) - The system doesn't validate the violation - The information is not correctly locked

3.3   Performance Requirements

3.3.1   Performance features

- 90% of the API calls should be completed within 4.5 seconds

- The response times during the testing phase will be measured using HP LoadRunner (or similar tools) located behind the firewall and in front of the web servers

- Real response times will be measured using the application server log files

- The system will be deployed to a machine able to serve a huge number of users in parallel

3.3.2   Evolution of the system

- At the beginning the system should be able to handle up to 10'000 users

- As the number of users will grow in unpredictable ways, at first the system will be deployed on cloud infrastructure that enables automatic up and down scaling (e.g. Azure Autoscale or AWS Elastic)

- Due to the cost of such infrastructures, as the number of users will become more stable the system will be migrated to a fixed resource machine

3.3.3   Performance testing

- Automatic monthly tests to check performance will be executed during system idle time which is determined statistically

- The performance tests should not exceed an execution time of 15 minutes

3.4   Design Constraints

3.4.1   Standards compliance

- To guarantee the compatibility with the potential municipality accident information service, our backend software should communicate and be able to read data in a portable format like JSON or XML

- To transfer this data, the software should expose an HTTP REST API

3.4.2   Hardware limitations

- The frontend application should be as lightweight as possible to support the diversity of mobile device hardware

- Devices with a camera resolution of less than 2MP should be marked as incompatible with SafeStreets due to the poor quality of the pictures that are taken with them

3.4.3   Other constraints

- The language used to develop the backend application will be chosen from the ones that are most supported by the main cloud infrastructure providers, in order to have a wider choice of hosting plans

3.5   Software System Attributes

3.5.1   Reliability

3.5.1.1   Index MTBF

- Index MTBF must consider as failures those out of design conditions which place the system out of service, for example by an overload of user's requests

- MTBF is of 5,000 hours

3.5.1.2   Index MTTR

- Index MTTR must consider time of testing and solution of bugs

- MTTR must be less than two hours

3.5.1.3   The position should be verifiable

- The system expects to receive one or two pictures

- The received pictures must be of at least 2MP

- The first picture must represent a general overview of the violation, to identify the street

- The second picture must allow the system to identify correctly the license plate, if the violation includes a vehicle

3.5.2   Availability

- The system must be available 99.9%

- The system must be available when performing the standard routine tests for maintenance

- The system can bear slowdowns of the service in case of extraordinary overload of requests

- The system can become unavailable for some seconds due to extraordinary maintenance or major updates

3.5.3   Security

3.5.3.1   Secrecy of data received and of users' information

- The system should never allow both non-registered users and registered users to see the identity of the person that notifies the violation

- User personal data must be encrypted and safely stored

- Users' passwords must be salted and hashed before being stored on any persistent medium

- Every user can log in using their email and password, but they can optionally enable Two-Factor authentication to improve security

- Violation reports must be encrypted before being sent to the server to preserve the secrecy and integrity in the chain of custody

3.5.3.2   Integrity of data

- The information sent by users (picture, date, time and position of the violation) has to be encrypted in order to keep the integrity and to avoid manipulation of the data

3.5.3.3   Measures of security according to danger level

- Public statistics to identify the most dangerous areas are provided for SafeStreets

- SafeStreets makes a ranking of dangerousness (minimum, medium, high)

- The authorities can decide to improve security by highlighting streets in areas at minimum risk

- The authorities can decide to improve security by adding cameras in the areas at medium risk

- The authorities can decide to improve security by both adding cameras and doubling patrol shifts in the areas at large risk

3.5.4   Maintainability

3.5.4.1   Testing overview

- The system is checked in its correct functioning by software testing

- Unit tests, Integration tests and System tests are performed before and after every update

3.5.4.2 The system is controlled and monitored

- The system is equipped with condition-monitoring algorithms

- The condition-monitoring algorithms identify the functions that cause alarm

- A unit test is made as soon as a function is identified as potentially at risk

- An integration test is performed after the unit test of the potentially at risk function

- A system test is performed after the integration and unit test of the potentially-at-risk function

3.5.4.3 The code must be clean and easy to understand

- Code must follow common best practices

- It is advisable to follow the design patterns to provide a standard terminology and to make the software more adaptable to future extensions and improvements

- Complete and detailed documentation, even for low level functions, is mandatory in order to keep the maintainability on the highest level on the whole system

3.5.5 Portability

- The software shall run on Windows, macOS, Linux, Android and iOS through a modern web browser supporting at least ECMAScript 2015

- To guarantee the portability of the front end application the software shall be developed using a web framework such as React

- This approach for the front end application will enable code reuse among native mobile platforms (iOS and Android) and the Web Application

- The back end software shall be developed in a language that can be compiled to run on virtual runtime such as Java, JavaScript or C# which can target the JVM, Node.js and dotnet core respectively

- The software can be developed using the standard API's that span different types of operating systems

- The software will then be able to be transferred with no modifications on other destination machines

- The architecture must be flexible

Chapter 4

Formal using Alloy

Chapter 5

Effort spent

Carlo Dell'Acqua

Task	Time spent (hours)
Project setup	2
Functional requirements	0.5
Design constraints	0.5
General Review	1
Introduction	1
App Images	2.5

Adriana Ferrari

Task	Time spent (hours)
Project setup	2
Functional requirements	1
Design constraints	0.5
Product perspective	0.5
Product functions	0.5
General Review	1.5

Angelica Sofia Valeriani

Task	Time spent (hours)
Software System Attributes	2
Performance Requirements	1
Domain Assumptions	1
External Interface Requirements	1
User Characteristics	1
Scenarios	3
Use cases	3
General review	1

Bibliography