POLITECNICO
MILANO 1863

Computer science and engineering
Software engineering 2 - Project 2019/2020

SafeStreets

Requirement Analysis and Specification Document

Deliverable:	RASD
Title:	Requirement Analysis and Verification Document
Authors:	Maldini Pietro , Paone Angelo
Version:	1.0
Date:	10-November-2019
Download page:	https://github.com/pm390/MaldiniPaone
Copyright:	Copyright © 2019, Maldini Pietro, Paone Angelo – All rights reserved

Contents

1 Introduction

1.1 Purpose

This document represents the Requirement Analysis and Specification Document (RASD). Goals of this document are to completely describe the system-to-be in terms of functional and non-functional requirements, analyse the real needs of the users in order to model the system, show the constraints and the limit of the software and indicate the typical use cases that will occur after the release. This document is addressed to the developers who have to implement the requirements and could be used as a contractual basis.

1.1.1 Goals

- USER:

  G1 Notify authorities about traffic violations

  G1-1 Send picture of violation

  G1-2 Send Position of the violation

  G2 Authorities must be able to take an assignment if they are free

  G3 allow authorities to report a finished assignment

  G4 allow all actors to visualize update statistics

  G5 Allow the system manager to register Municipality to the service

- SafeStreets:

  G6 Allow a Visitor to join the system registering him/herself to ensure reliability of the information provided by them

  G7 Store information about violations provided by users:

  G7-1 Complete it with metadata

  G7-2 Mine information

  G8 Identify potentially unsafe areas:

  G8-1 Suggest possible interventions

  G9 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

1.2 Scope

1.2.1 Description of the given problem

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.

1.2.2 Phenomena

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

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

- Violation: parking violations which can be notified by Citizens to authorities
- Report: Notification sent by Citizens to the system
- Mapping System: external software that provides maps and directions to reach the position of a violation
- Licence plate Recognition Algorithm: calculation process that identify the alphanumeric number on license plate
- Spam: a series of message that are undesired
- App: application software
- Blocked: means that the account is banned for a given period
- Metadata: data about a violation. Position , date, time and the username of Citizen.
- Assignment : Work Request for authorities generated upon the receiving of a notification made by Citizens.

1.3.2 Acronyms

- RASD: Requirement Analysis and Specification Document.
- API: Application Programming Interface
- GPS: Global positioning system
- HTTP: HyperText Transfer Protocol
- HTTPS: HyperText Transfer Protocol over Secure Socket Layers
- UML: Unified Modeling Language

1.3.3 Abbreviations

Gn : n-th goal.

Dn : n-th domain assumption.

Rn : n-th requirement.

1.4 Revision History

- RASDv1.0 delivered on 10/11/2019

1.5 Reference Documents

- Specification Document: “Assignments AA 2019-2020.pdf”.
- Alloy Dynamic Model example
- IEEE Std 830-1993 - IEEE Guide to Software Requirements Specifications.
- IEEE Std 830-1998 - IEEE Recommended Practice for Software Requirements Specifications.

1.6 Document Structure

This chapter debates about contents and structure of RASD, indeed this document, based on standards IEEE, is divided in six different sections:

1. The Introduction provides a general appearance of the systems defining which are the goals to reach, describing the problem and introducing the world and shared phenomena.
2. The Overall Description provides the description of the relevant components that system needs, the involved actors’ characteristics, and the assumptions to better clarify the needs and the boundaries of the system.
3. Specific Requirements contains the goals, the functional and non-functional requirements of the system are presented. In addition, a mock-up representation of the application expslain and gives a feeling on how the app will look like and show the main actions it can perform.
4. Scenarios describe the usefulness of SafeStreet and its features in some situations that could happen.

5. UML Modelling contains the diagrams that are referred to the functionality of the system, they explain the workflow of some scenarios, the actions and the structure of the actors and the state that system assumes. These features are represented by Use Case diagram, Sequence diagram, Class diagram and Activity diagram.

6. Alloy Modelling allows to explain the world models through Alloy model of system. The mocks-up generated by the Alloy modelling grants that given requirements and domain assumption the goals are satisfied.

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.

2.1.1 Further analysis on shared phenomena

2.1.2 Class Diagram

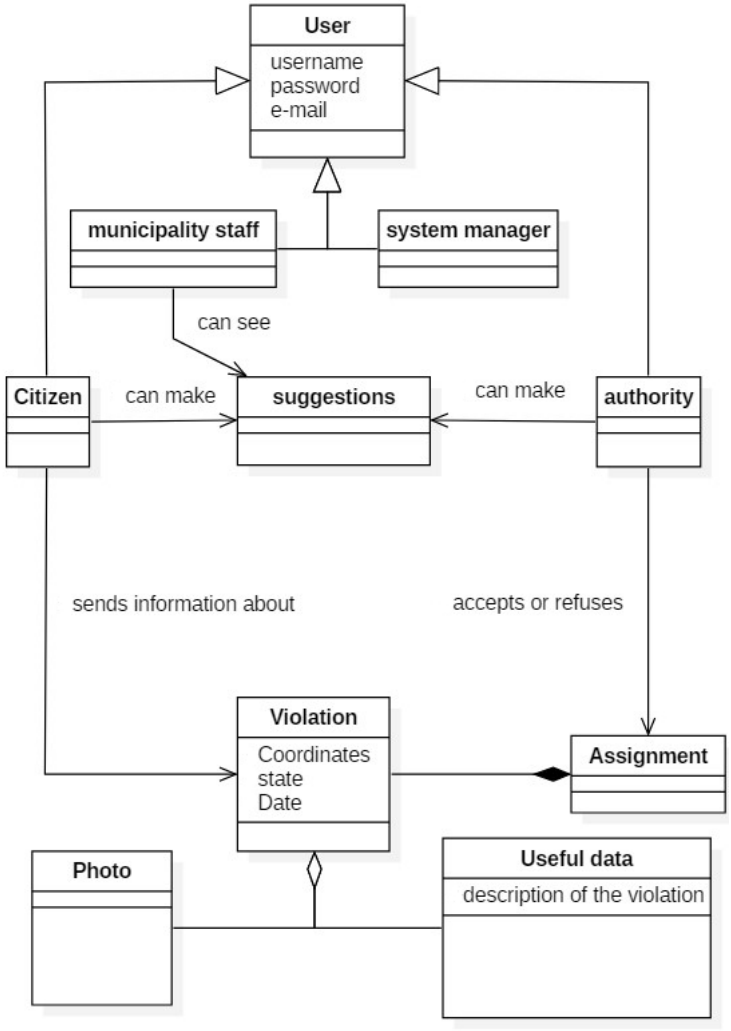

Figure 1: Class Diagram

2.1.3 Statechart report

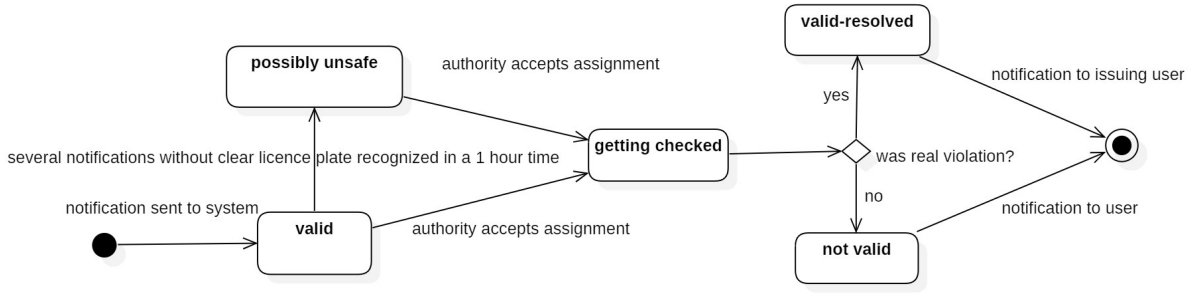

Figure 2: Statechart report

2.1.4 Statechart assignment

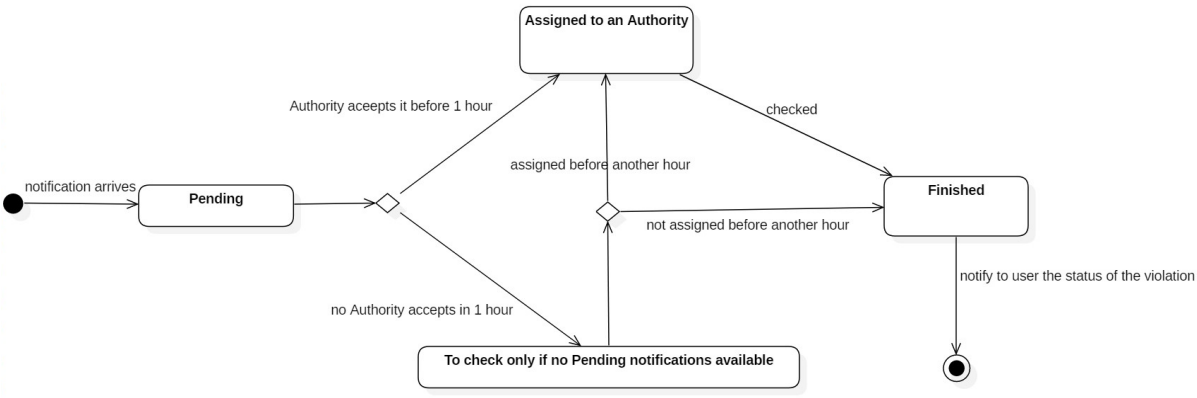

Figure 3: StateChart assignment

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 Actors characteristics

- 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.
- System Manager: User who is responsible of the system, therefore he/she has all the possible privileges to manage the system.For example he/she can register a new municipality

2.4 Assumptions, dependencies and 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 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.

2.4.3 Domain assumptions

- D1) For each notification data and metadata about the violation are correct
- D2) Each Username is unique
- D3) Authorities always intervene in case of a notified violation
- D4) Information about authorities' location are always available through GPS

- D5) Only agents close to the violation area are notified
- D6) Citizen sends only clear photos (if it is not clear he/she would retake the photo)
- D7) System Manager, Municipality and authorities respects their duty of care
- D8) When an authority is sent an email to register this will surely be received
- D9) Information provided by authority are correct and no false report is ignored (always reported by authority as false).
- D10) Citizens' Location are retrieved by GPS or manual input and are correct
- D11) The Agent must be contactable from municipality
- D12) Information of authority are known by municipality
- D13) Information provided from Users must be correct

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

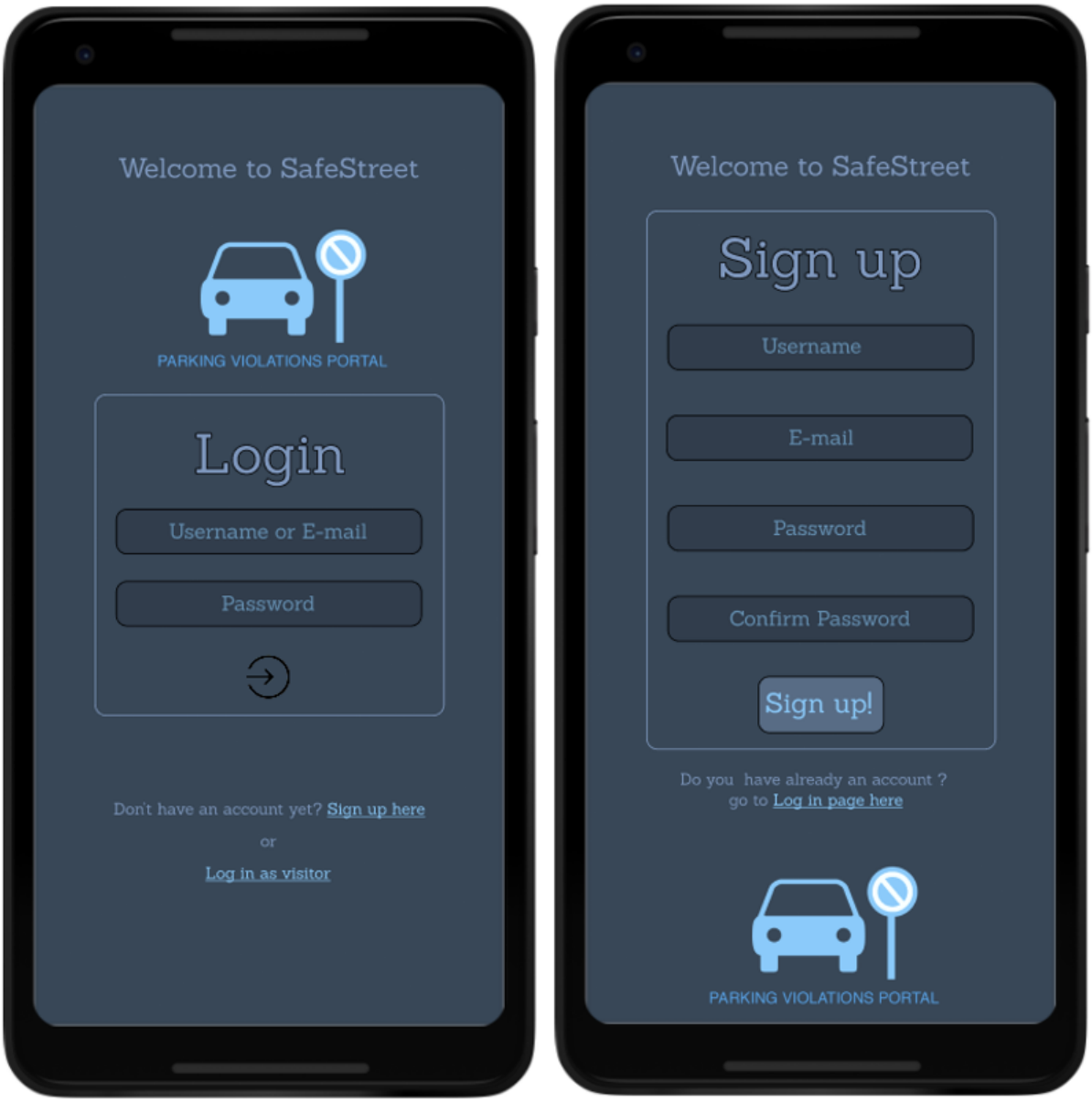

Figure 4: Login and Sign up

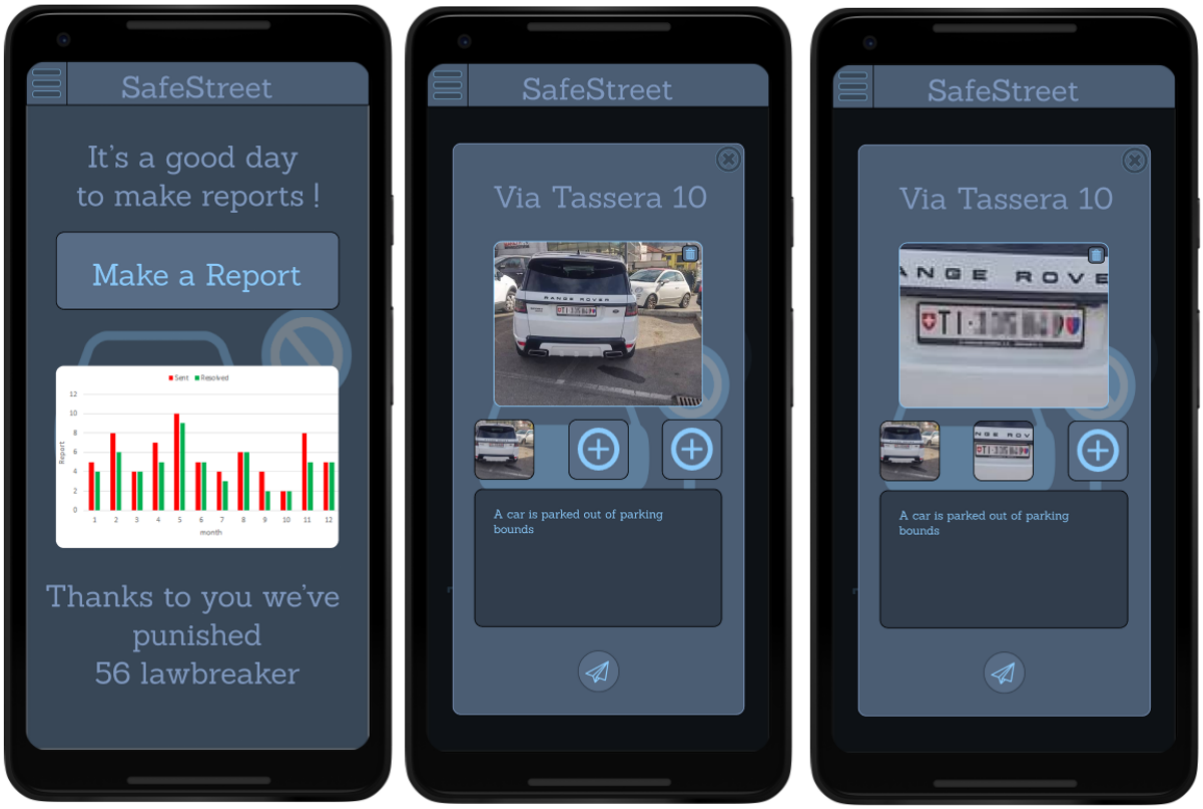

Figure 5: Citizen Interfaces

Figure 6: Authority Interfaces

Figure 7: Common Interfaces

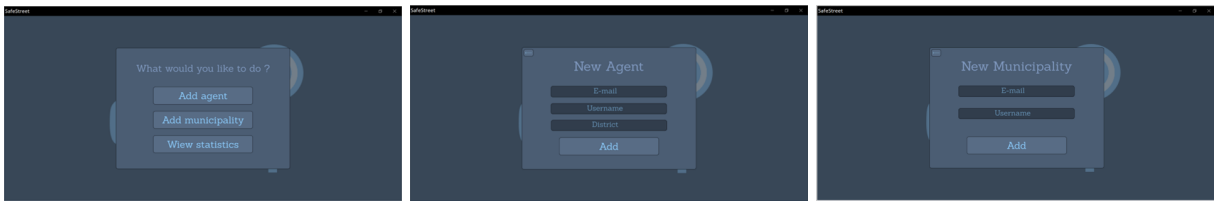

Figure 8: Municipality Interfaces

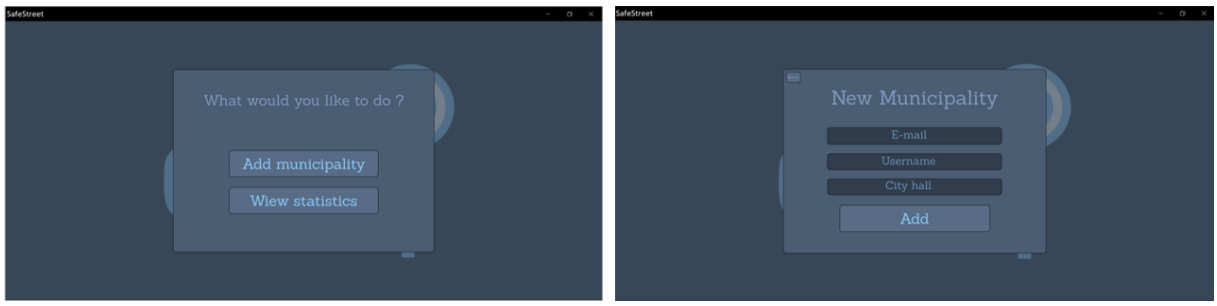

Figure 9: Sytem Manager Interfaces

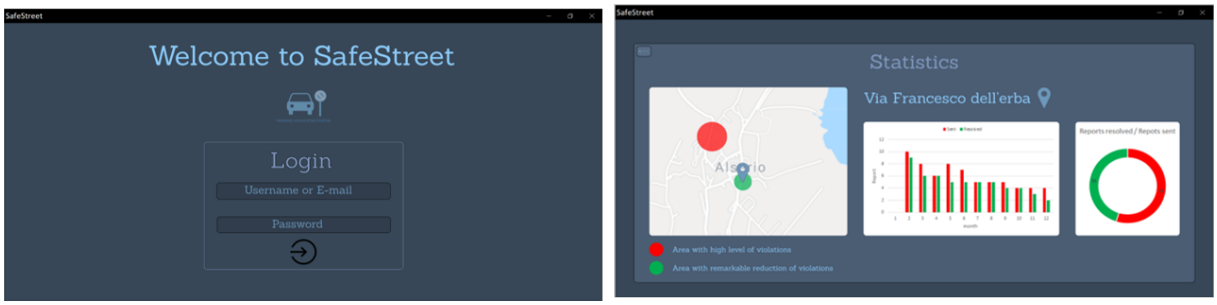

Figure 10: Common web Interfaces

3.1.2 Hardware Interfaces

There are no hardware interfaces to be implemented

3.1.3 Software Interfaces

Software interfaces to communicate with mapping systems, municipality databases and the algorithm to read Lisence plates are needed to make S2B communicate with external systems.

3.1.4 Communication Interfaces

To communicate the various internal parts of the S2B use HTTP and HTTPS protocol. No specific communication interface must be implemented.

3.2 Functional Requirements

3.2. Functional requirements

- R1) Authority's location must be known by the system when they are in service
- R2) Data relative to the violation sent by the user must be clear
- R3) The right authorities are notified about violations
- R4) Authority must be able to provide the system how the assignment finished: resolved, no intervention needed when arrived, false report.
- R5) The system must make data available when asked.
- R6)Data and statistics are always updated when an event happens.
- R7) System Manager must fill correctly the form with Municipality's data
- R8) A visitor must be able to begin sign up process in the SafeStreets App.
- R9) When the creation is successful the system must notify the Visitor
- R10) A citizen must be able to correctly report violations details
- R11) Authority must be able to correctly finish their assignments and specify how the violation was resolved and the correctness or non-correctness of the report

3.2.1 Scenrarios

- Scenario 1:
  Dimitri has an important appointment, but in front of his underground garage there is a car parked that doesn't allow him to go out. He doesn't know who the car owner is, so he can't call him to move it. So, Dimitri decides to use Safestreet application: he takes a photo of the car, he adds the notes and he insert manually the position, because the GPS can't take correctly the position, in order to send the violation to the authorities. After 10 minutes the public security agent, who has received the notification of Dimitri's alert, arrives and means of removal take the car away and finally Dimitri can go to the appointment.

- Scenario 2:
  Angelo is the town councilman of Alserio. In the last period some citizen has alerted him that there are some people who park their car in the disabled parking near the Enigma café. They also have complained about the fact that there is not a public security agent to control the area. So, Angelo asks the mayor to use SafeSteets application in order to keep the violations under control. The mayor, enthusiast of his suggestion asks the system manager to add him into the service.After that, the mayor adds the other municipality employees, so they can add the authorities too.Therefore , thanks to the app , the authorities receive warnings about infractions in various part of the town and they can intervene. Moreover, thanks to the statistics supplied by the application, the mayor manages to start a policy of prevention of the violation.

- Scenario 3
  Manuel asks his friend Fred if he wants to join him for dinner this evening at his place. Unfortunately, Fred's car is blocked by a vehicle parked in front of his garage. He has noticed that car has been there several days in the past months. Since Fred works near his house, he goes to work by bike, but his friend lives far from his house, so he can't get there without his car. So, Fred decides to download SafeStreets app on his smartphone. He provides all the information required to the system in order to sign up; then he logs in and sends a notification to the authorities. One hour before leaving he notices that the car has been removed. From that day on the car owner stopped parking there.

- Scenario 4:
  Today in Milan there are lot of violations and the authorities can't check every one of them. After some time, the violations not assigned lose their priority and go down in the assignment list. Agent Carlo usually checks the areas between Milano Piola and Milano Lambrate. Lots of violations are notified by users in those areas. However, by the time he finishes checking some violations, other become less visible due to the great amount of notifications. When different users notify a violation concerning a license plate in the same area, this one becomes more visible in the assignment list. Thanks to this feature, Carlo is able to first address issues affecting more people.

- Scenario 5:
  Pietro is an agent who has just been employed in Alserio police; his method of street controlling is "old school" and a little bit disorganized, but his colleague Sonia recommends him SafeStreet application. So, Pietro goes to the town hall to register himself into the Safestreet database. The employee checks his credential and his work district, and the system sends him the transient credential. Then, Pietro installs the app and uses the transient credentials in order to log into his personal page and change the credentials to confirm the account. Thanks to SafeStreet application, his job is now organized better because he can go to exact violations place.

- Negative Use Scenario 6:
  Fernand got his car fined after having parked in a forbidden area. He has been doing it for years, but now he notices that people in that area started using SafeStreets app and thinks that he was fined because of it instead of his behaviour. He decides to "take revenge" subscribing to the app and sending several false notifications. Since he has sent several random photos without any visible licence plate, the system signs the assignments given by Fernand account as "possibly unsafe". After that some agents have checked that no violation is actually in any of the photos sent to the system, his account gets blocked and Fernand can no longer interfere with the normal activities of authorities using the app. After being blocked, Fernand stops complaining about the app and a week later he finds a parking area to replace his former parking spot.

3.2.2 Use Case Diagrams

Figure 11: Visitor Use Case

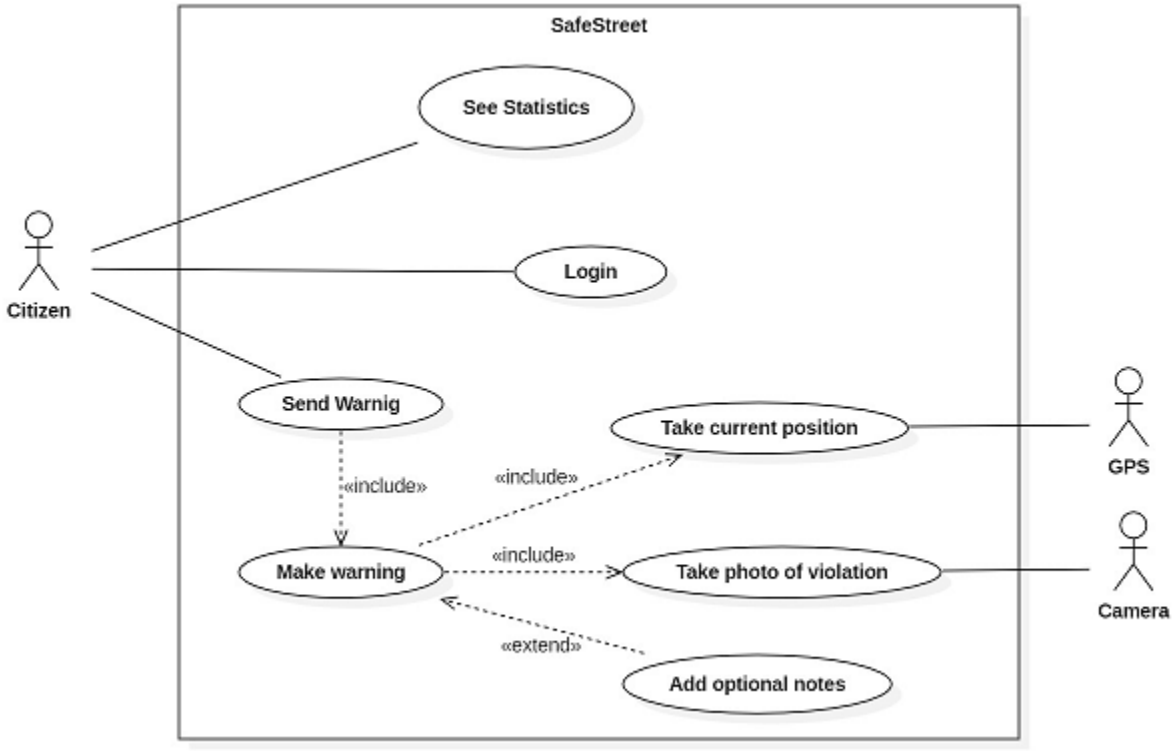

Figure 12: Citizen Use Case

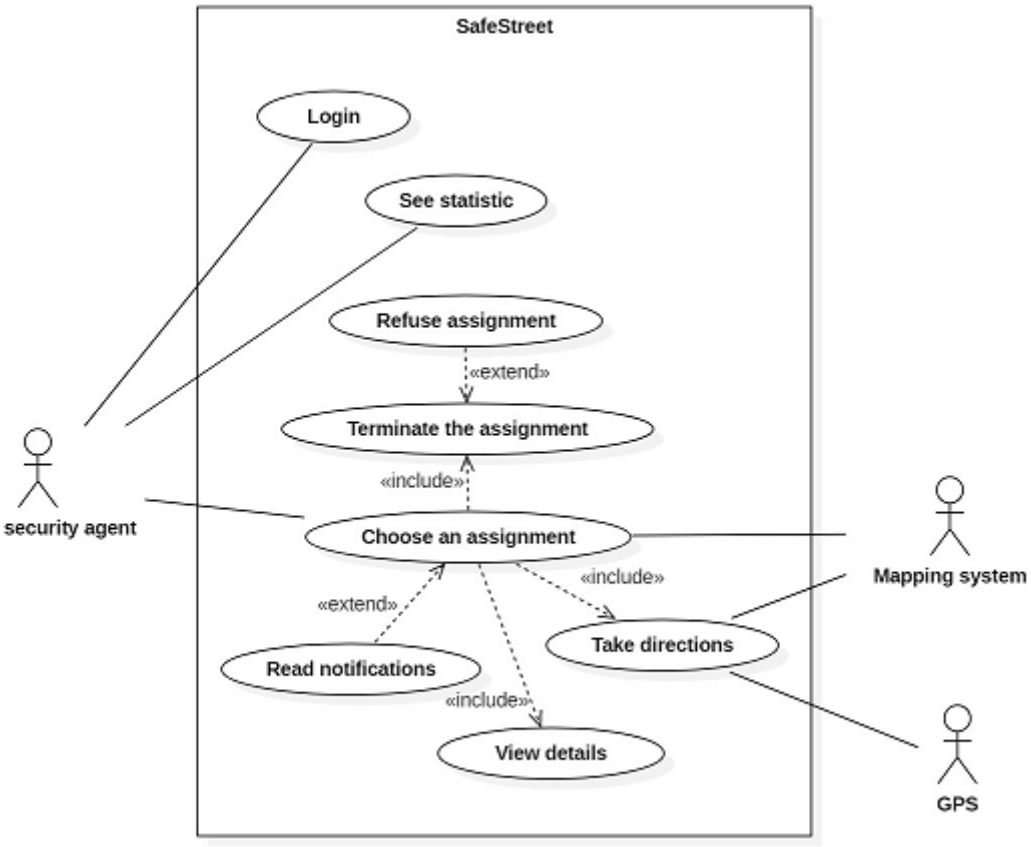

Figure 13: Authority Use Case

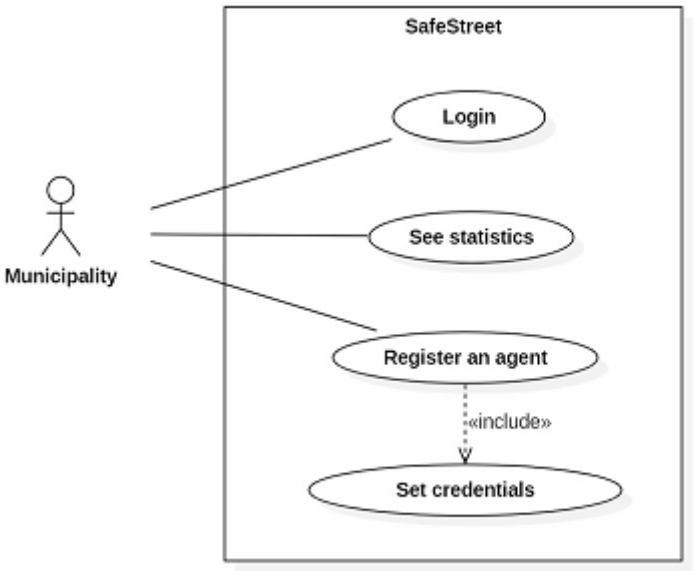

Figure 14: Municipality Use Case

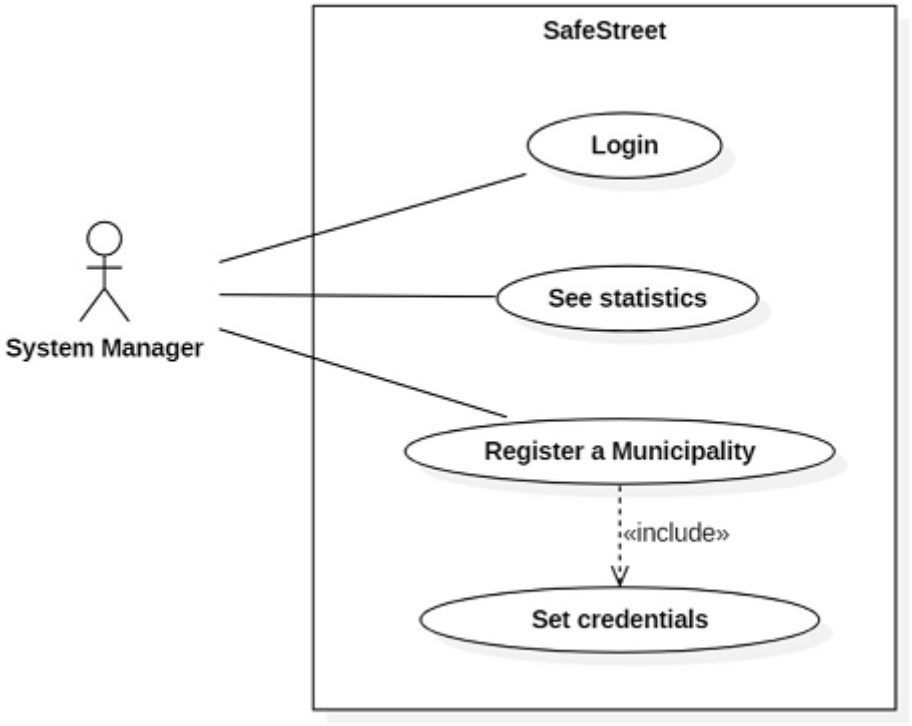

Figure 15: System Manager Use Case

3.2.3 Use Case Descriptions

ACTORS	Visitor
GOALS	[G6]
INPUT CONDITIONS	There aren't entry conditions
EVENTS FLOW	1. The visitor on the login page clicks on the signup link to start the. registration process. 2. The visitor fills the registration form. 3. The visitor clicks on the signup button to send the form. 4. The system checks the data. 5. The system saves the data. 6. The system shows the confirm of the user's registration.
OUTPUT CONDITONS	The visitor successfully ends the registration process and become a new User. From this moment using his/her credential he/she can log in and report violations in addition to seeing statistics
EXCEPTIONS	1. The Visitor provides not valid information in the form. 2. The Visitor chooses a username belonging to another User. 3. The Visitor chooses an email that has been associated with another User. When an exception occurs, the visitor is notified, and the flow is taken back to point 2.

Figure 16: Visitor Use Case Description

ACTORS	Citizen
GOALS	[G1]
INPUT CONDITIONS	The citizen is logged in and clicks on “Make a Report” button
EVENTS FLOW	1. The citizen takes a photo of the violation 2. The citizen checks if the recognized licence plate is correct 3. Position of the citizen is taken by GPS or directly inputted by user 4. The citizen may add a note the report form 5. The citizen sends the report
OUTPUT CONDITONS	The violation is successfully notified to the system which will send it to authorities in the area
EXCEPTIONS	1. Internet connection fails 2. No licence plate recognized The first exception can be solved by notifying the citizen that connection failed and ask him to try to resend the report in another place. The second exception can be solved by asking the citizen to highlight the licence plate in the photo or re-taking the photo. If it is not recognized anyway the report will be sent alerting citizen that false reports may get their account blocked.

Figure 17: Citizen Use Case Description

ACTORS	Authority
GOALS	[G2], [G3]
INPUT CONDITIONS	A violation occurs close to the area controlled by the authority and a Citizen reports it
EVENTS FLOW	1. The Agent receives the notification of violation nearby. 2. The Agent reads the detail of violation from the notification. 3. The Agent clicks the start button to take the assignment. 4. The agent can read the direction to violation place. 5. The agent clicks stop assignment button to conclude the job. 6. The Agent clicks the terminate assignment button to send and to specify at server how he finished his work. 7. The Agent’s interface returns to the assignment schedule.
OUTPUT CONDITONS	The violation is removed from the assignment list and the statistics are updated
EXCEPTIONS	1.Internet connection fails. In case of internet connection failure, the authority is asked to retry the action checking a place with working internet connection.

Figure 18: Authority Use Case Description 1

ACTORS	Authority
GOALS	[G6]
INPUT CONDITIONS	An E-mail that contains the transient credential of the Agent
EVENTS FLOW	1. The Agent receives the e-mail. 2. The Agent fills the Log in form. 3. The Agent clicks the log in button to send the form. 4. The server checks the data. 5. The server allows the agent to access his reserved data. 6. The Agent fill the change password form. 7. The Agent clicks the change button to send the new password. 8. The system saves the data. 9. The system shows the confirm of the user's registration.
OUTPUT CONDITONS	The server stores the new Agent's account and the Agent successfully logs in
EXCEPTIONS	1.Internet connection fails. In case of internet connection failure, the authority is asked to retry the action checking a place with working internet connection. 2.Authority mistakes credentials filling in the form In this case authority is notified of the mistake and is asked to refill the form

Figure 19: Authority Use Case Description 2

ACTORS	Municipality
GOALS	[G10]
INPUT CONDITIONS	There aren't entry conditions
EVENTS FLOW	1. The employee clicks the add municipality button. 2. The employee fills the form with municipality data. 3. The employee clicks the add button to send the form. 4. The server checks the data. 5. The server generates a transient password of new municipality's account. 6. The server stores the data. 7. The server sends email with credentials to the municipality.
OUTPUT CONDITONS	The server stores the transient municipality's account
EXCEPTIONS	5. The employee provides not valid information in the form. 6. The municipality chooses an email that has been associated with another municipality. When an exception occurs, the employee is notified, and the flow is taken back to point 2.

Figure 20: Municipality Use Case Description 1

ACTORS	Municipality
GOALS	[G10]
INPUT CONDITIONS	There aren't entry conditions
EVENTS FLOW	1. The employee clicks the add municipality button. 2. The employee fills the form with municipality data. 3. The employee clicks the add button to send the form. 4. The server checks the data. 5. The server generates a transient password of new municipality's account. 6. The server stores the data. 7. The server sends email with credentials to the municipality.
OUTPUT CONDITONS	The server stores the transient municipality's account
EXCEPTIONS	5. The employee provides not valid information in the form. 6. The municipality chooses an email that has been associated with another municipality. When an exception occurs, the employee is notified, and the flow is taken back to point 2.

Figure 21: Municipality Use Case Description 2

ACTORS	System Manager
GOALS	[G5]
INPUT CONDITIONS	There aren't entry conditions
EVENTS FLOW	8. The manager clicks the add municipality button. 9. The manager fills the form with municipality data. 10. The manager clicks the add button to send the form. 11. The server checks the data. 12. The server generates a transient password of new municipality's account. 13. The server stores the data. 14. The server sends email with credentials to the municipality.
OUTPUT CONDITONS	The server stores the transient Municipality's account
EXCEPTIONS	3. The manager provides not valid information in the form. 4. The municipality chooses an email that has been associated with another municipality. When an exception occurs, the manager is notified, and the flow is taken back to point 2.

Figure 22: System Manager Use Case Description

ACTORS	Visitor, Citizen, Authority and municipality
GOALS	[G4]
INPUT CONDITIONS	There aren't entry conditions
EVENTS FLOW	1. The actor clicks the statistic button. 2. The server provides the general data of the map 3. The actor views the area with relevant data. 4. The actor selects a place of the map by sending the request to visualize the information about it. 5. The server sends the statistics about the place in question. 6. The actor can view the statistics
OUTPUT CONDITONS	There aren't output conditions
EXCEPTIONS	1. Internet connection fails. In case of internet connection failure, the actor is asked to retry the action checking a place with working internet connection.

Figure 23: Common Use Case Description 1

ACTORS	All Users
GOALS	[G6]
INPUT CONDITIONS	There aren't entry conditions
EVENTS FLOW	1. The User fills the Log in form. 2. The User clicks the log in button to send the form. 3. The server checks the data. 4. The server allows the User to access his reserved data.
OUTPUT CONDITONS	The user successfully logs in
EXCEPTIONS	1.User mistakes credentials filling in the form In this case User is notified of the mistake and is asked to refill the form

Figure 24: Common Use Case Description 2

3.2.4 Sequence Diagrams

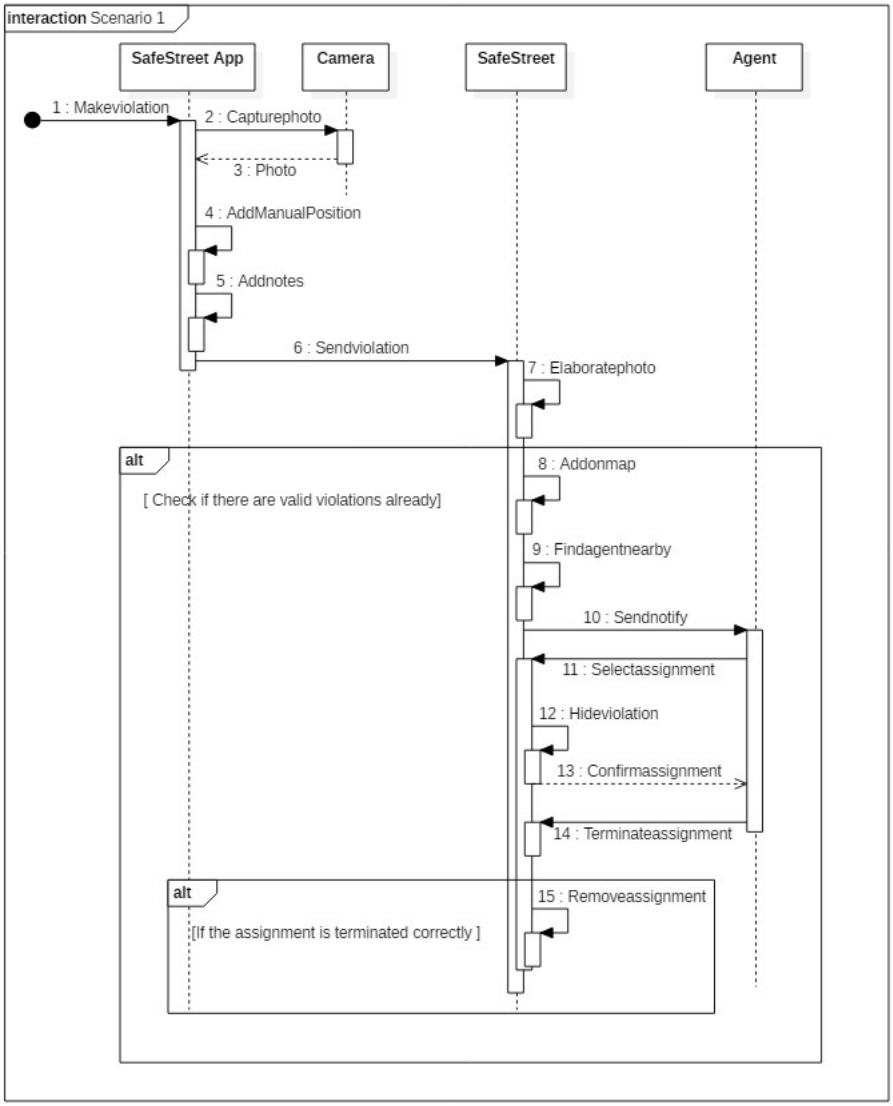

Figure 25: Sequence Diagram Scenario 1

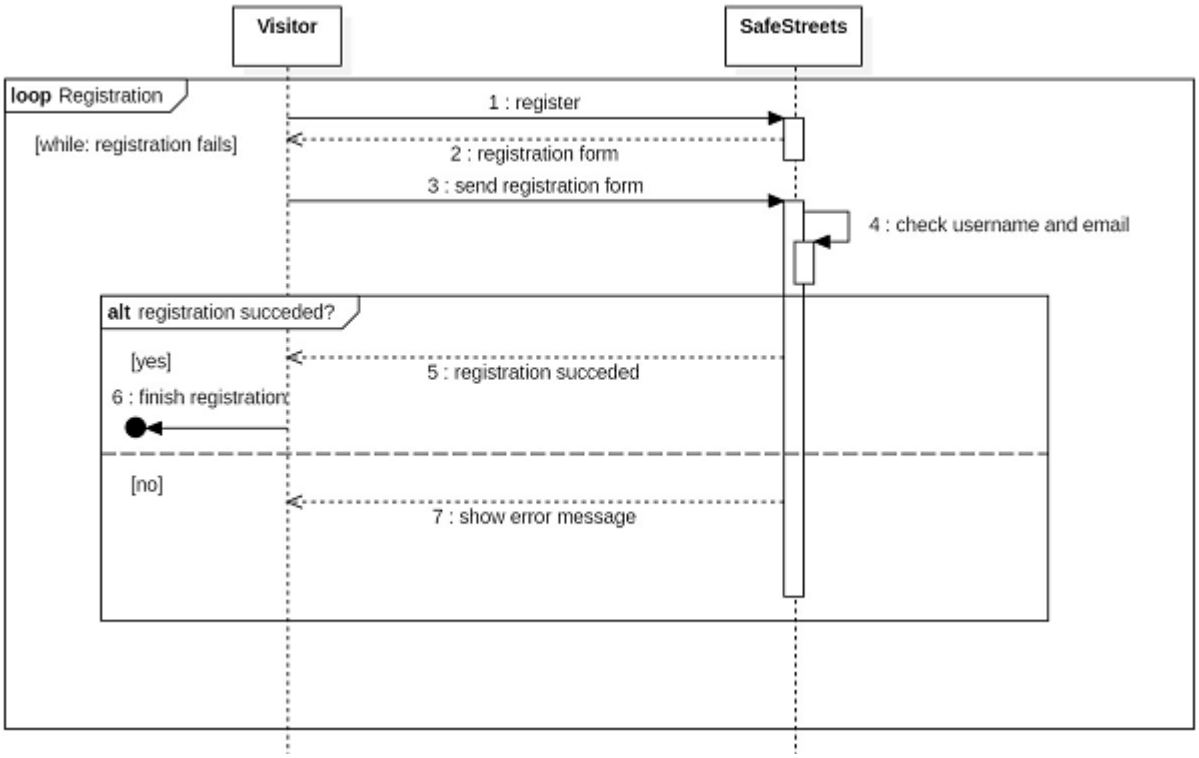

Figure 26: Sequence Diagram Registration

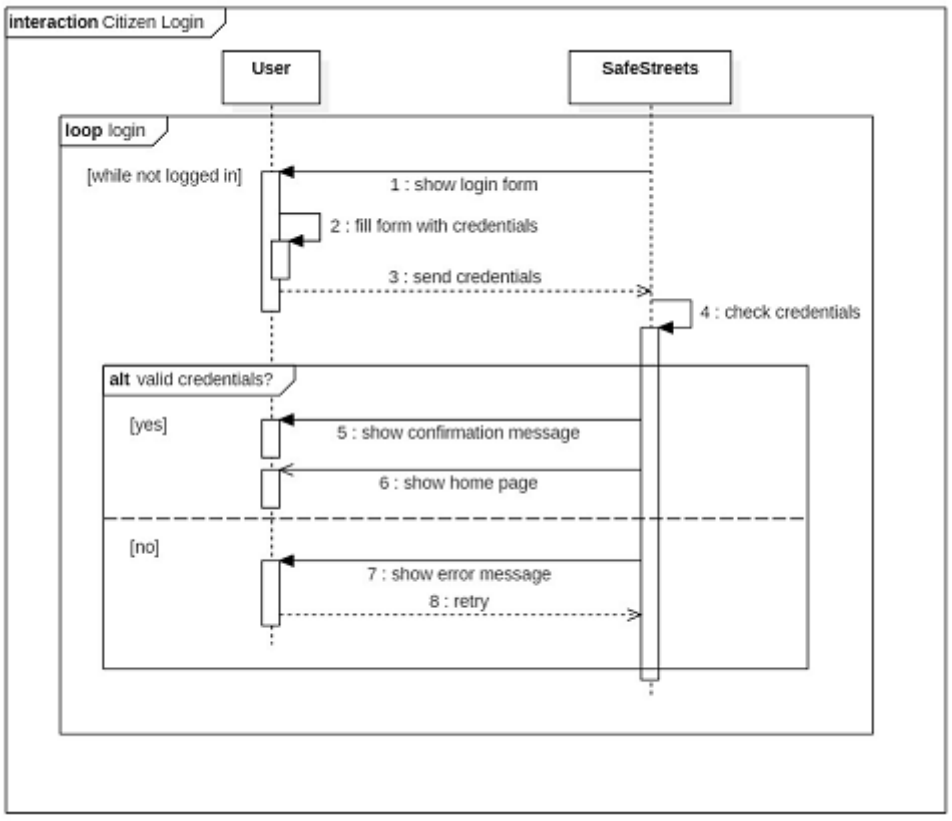

Figure 27: Sequence Diagram Login

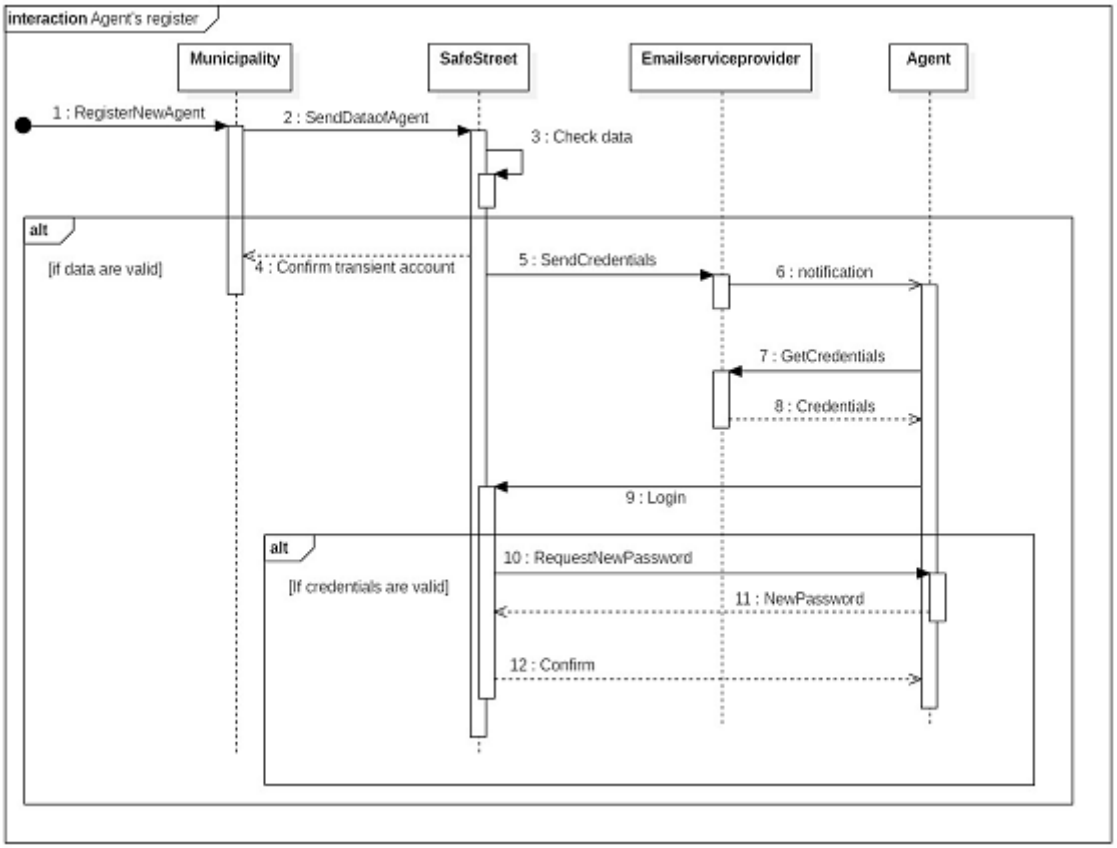

Figure 28: Sequence Diagram Agent Registrtion

3.2.5 Mapping of Requirements and Domain Assumptions to their relative goal

- G1) Notify authorities about traffic violations
  - R1) Authority's location must be known by the system when they are in service
  - R2) Data relative to the violation sent by the user must be clear
  - D1) For each notification data and metadata about the violation are correct
  - D4) Information about authorities' location are always available through GPS
  - D6) Citizen sends only clear photos (if it is not clear he/she would retake the photo)
  - D10) Citizens' Location are retrieved by GPS or manual input and are correct
- G2) Authorities must take an assignment if they are free
  - R3) The right authorities are notified about violations
  - D4) Information about authorities' location are always available through GPS
  - D7) System Manager, Municipality and authorities respects their duty of care
- G3) allow authorities to report a finished assignment
  - R4) Authority must be able to provide the system how the assignment finished: resolved, no intervention needed when arrived, false report.
  - D7) System Manager, Municipality and authorities respects their duty of care
  - D9) Information provided by authority are correct and no false report is ignored.
- G4) allow all actors to visualize updated statistics
  - R5) The system must make data available when asked.
  - R6) Data and statistics are always updated when an event happens.
  - D9) Information provided by authority are correct and no false report is ignored.
- G5) Allow the system manager to register Municipality to the service
  - R7) System Manager must fill correctly the form with Municipality's data
  - D7) System Manager, Municipality and authorities respects their duty of care
- G6) Allow a Visitor to join the system registering him/herself to ensure reliability of the information provided by them
  - R8) A visitor must be able to begin sign up process in the SafeStreets App.
  - R9) When the creation is successful the system must notify the Visitor
  - D2) Each Username is unique
  - D8) When an authority is sent an email to register this will surely be received
- G7) Store information about violations provided by users:
  - R10) A citizen must be able to correctly report violations details
  - R11) Authority must be able to correctly finish their assignments and specify how the violation was resolved and the correctness or non-correctness of the report
  - D9) Information provided by authority are correct and no false report is ignored.
  - D10) Citizens' Location are retrieved by GPS or manual input and are correct

- G8) Identify potentially unsafe areas:
  - D9) Information provided by authority are correct and no false report is ignored.
  - D13) Information provided from Users must be correct
- G9) Allow municipality to register Authorities to the service
  - R12) Municipality must fill correctly the form with Agent's data
  - D7) System Manager, Municipality and authorities respects their duty of care
  - D11) The Agent must be contactable from municipality
  - D12) Information of authority are known by municipality

3.3 Performance Requirements

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 10 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.

3.4 Design Constraints

3.4.1 Standards compliance

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"

3.4.2 Hardware limitations

- Mobile App
  - Android smartphone
  - 2G/3G/4G connection
  - GPS
- Web App
  - Modern browser able to retrieve user's location

3.5 Software System Attributes

3.5.1 Availability

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.

3.5.2 Reliability

There are no strict costraints of Reliability if Availability costraints are respected

3.5.3 Security

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

3.5.4 Maintainability

The system must be developed in a modular way so that new feature can be added. Using a modular approach it is also possible to test the functionality of the single parts making it simpler to maintain. Documentation must be also always updated in case of modifications of the system to always keep documentation aligned to the system.

3.5.5 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.

3.5.6 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 Formal Analysis Using Alloy

4.1 Alloy Model Description

The main goal of SafeStreets is to increase safety on the strets , for reaching this goal it is importante that every report made by Citizens is taken care as fast as possible by authorities. We have modeled the goal stating that every agent must be working on an assignment if he/she is not already dealing with another. In the model there are only the entities that are relevant to this goal and semplification about users' location are made to make the model more understandable. In addition to checking that our model with domain assumptions and requirements makes it possible to reach the goal we also checked how different worlds may be generated and that some borderline cases aren't ignored.

4.2 Alloy Model

```
//Signatures
abstract sig  User{
        credentials: one Credentials
}
sig Username{}
sig Password{}
sig Credentials
{
        username: one Username,
        password: one Password
}

sig Statistic
{
        assignment: set Assignment
}
{
no a:assignment|a.state≠Ended
all disj a1,a2:assignment| a1.city=a2.city
assignment≠none //no void static
}

sig City{}//for simplicity we consider city instead of area of duty

sig Suggestion{}

sig Citizen extends User
{
report: set Report,
gps: lone City,
manualInput: lone City
}
{
gps=manualInput or gps=none or manualInput=none
}

sig Authority extends User
{
        city: one City, //D4
        notifications: set Report,
        municipality : one Municipality
}
sig Municipality extends User
{
        city: one City,
        suggestions: set Suggestion
}

sig Report
{
        city: one City,
        assignment : one Assignment
}
```



```
abstract sig Status{}
one sig Pending extends Status{}
one sig Ended extends Status{}
one sig Accepted extends Status{}
sig Assignment
{
        city: one City,
        authority : lone Authority,
        state: one Status
}
//predicates and functions
pred IsWorking[a:Authority]
{
one as1:Assignment| as1.authority=a and as1.state=Accepted
}
fun FindCitizen [c:Citizen] : one City
{
c.gps+c.manualInput
}
//General Facts
fact StateMeaning{
all as1:Assignment|     ((as1.authority=none and as1.state=Pending) or
(as1.authority≠none and as1.state≠Pending and as1.city=as1.authority.city))
}

fact AllReportAreAssociatedToACitizen
{
Report=Citizen.report
}

fact AuthorityWorkingOnAtMostOneAssignmentAtTheSameTIme
{
all au:Authority | no disj as1,as2:Assignment | as1.authority=au and
 as2.authority=au and as1.state=Accepted and as2.state=Accepted
}

fact AllAssignmentAreRelatedToAtLeastOneReport
{
all as1:Assignment| some re:Report | re.assignment=as1
}

fact ReportAndAssignmentInSameCity
{
        all r:Report| r.city=r.assignment.city
}

fact AllCityAssociatedToAnEntity
{
        all c:City| c in Municipality.city or c in Report.city
}

fact AllSuggestionsAreSentToAMunicipality
{
all su:Suggestion| some mu:Municipality| su in mu.suggestions
}


fact NoAuthorityOutOfBounds
{
all au:Authority| au.city=au.municipality.city
}

fact SuggestionsOnlyWhenThereAreAssignments
{
all m:Municipality| (m.suggestions≠none implies (some a:Assignment|a.city=m.city))
}

fact NoSameReportByASingleUser
{
```

```
all ci:Citizen| no disj re1,re2:Report| re1 in ci.report and re2 in ci.report and re1.
    ↪ assignment=re2.assignment
}


fact NoLooseCredentials
{
Password=Credentials.password and #(Credentials)=#(User) and #(Username)=#(User)
}

fact CredentialsArePersonal
{
no disj u1,u2:User | u1.credentials=u2.credentials
}



//Domain Assumption
fact UsernameAreUnique //D2
{
no disj c1,c2:Credentials| c1.username=c2.username
}

fact RespectDutyOfCare //D7 if there are no Pending Assignments authority can't work on
    ↪ an assignment
{
all au:Authority| !IsWorking[au] implies (no re:Report| re in au.notifications and
 (some as1:Assignment| re.assignment=as1 and as1.state=Pending))
}

fact AllCitizenLocationAreAvailableAndCorrect//D10
{
all c:Citizen| FindCitizen[c]≠none and (all re:c.report | re.city=FindCitizen[c])
}

//Requirements
fact EveryRightAuthoritiesAreNotified //R3
{
all re:Report,au:Authority| (re.city=au.city) iff re in au.notifications
}

fact StatisticsAreAlwaysUpdated //R6
{
all as1:Assignment| (as1.state=Ended) iff (some s:Statistic| as1 in s.assignment)
}

//Goals
assert AuthorityMustTakeAssignments
{
all a:Authority|  (IsWorking[a]) or (no as1:Assignment| as1.city=a.city and as1.state=
    ↪ Pending)
}

//Analyzed worlds
pred world1() //Credentials are unique and every User is associated with a City(which
    ↪ reprents his/her position)
{
#(Assignment)=0
#(Report)=0
#(Citizen)>0
#(Authority)>0
#(Municipality)>0
}

pred world2() //same assignment can be derived from reports from different Citizens
{
#(Assignment)=2
#(Statistic)=0
#(Report)=4
#(Authority)=1
#(City)=1
```



```
some as1:Assignment |as1.state=Pending
}

pred world3() //a more generic world with all possible states of assignments
{
#(Assignment)=4
#(Authority)=2
some as1:Assignment |as1.state=Ended
some as1:Assignment |as1.state=Pending
some as1:Assignment |as1.state=Accepted
}

check AuthorityMustTakeAssignments for 10
run world1 for 5
run world2 for 8
run world3 for 5
```

4.3 Results

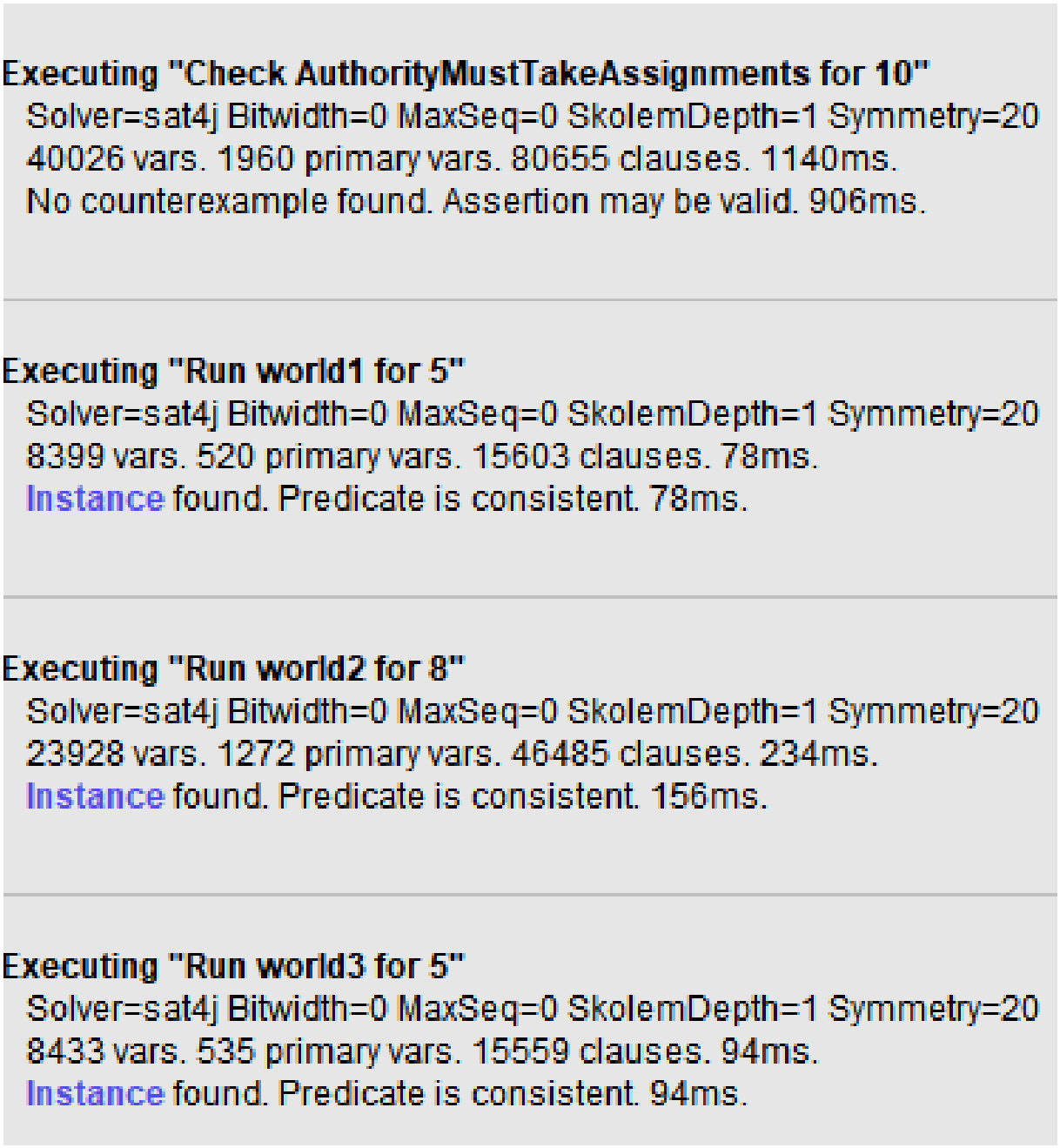

Figure 29: Result

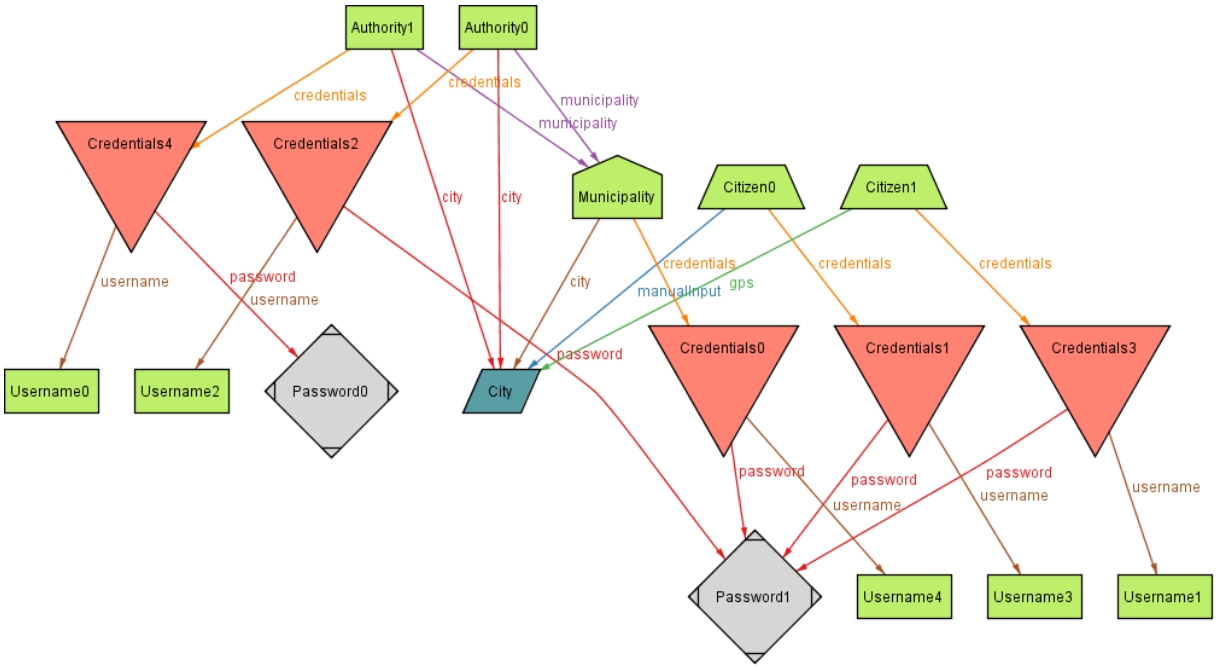

Figure 30: First world

This world represents the borderline case in which no notification are present and so no assignment is associated to authorities , there are neighter statistics nor suggestions

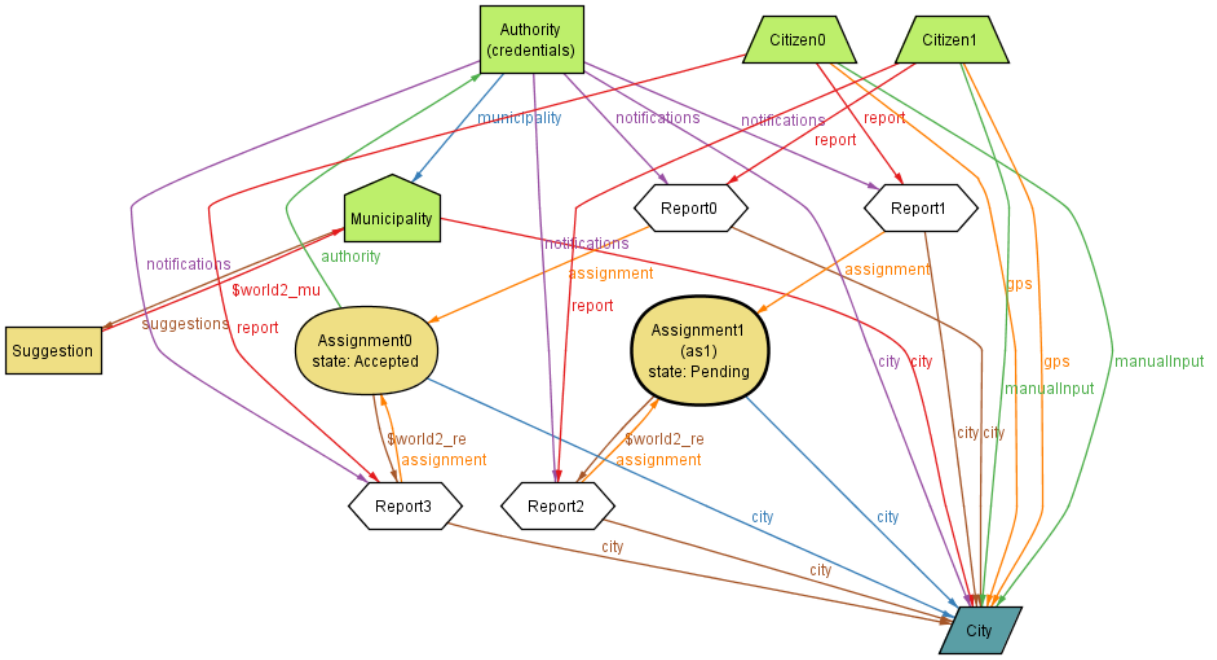

Figure 31: Second world

This world represent the case in which no assignment is completed and so no statistics are available Credentials are projected away for the sake of clarity.

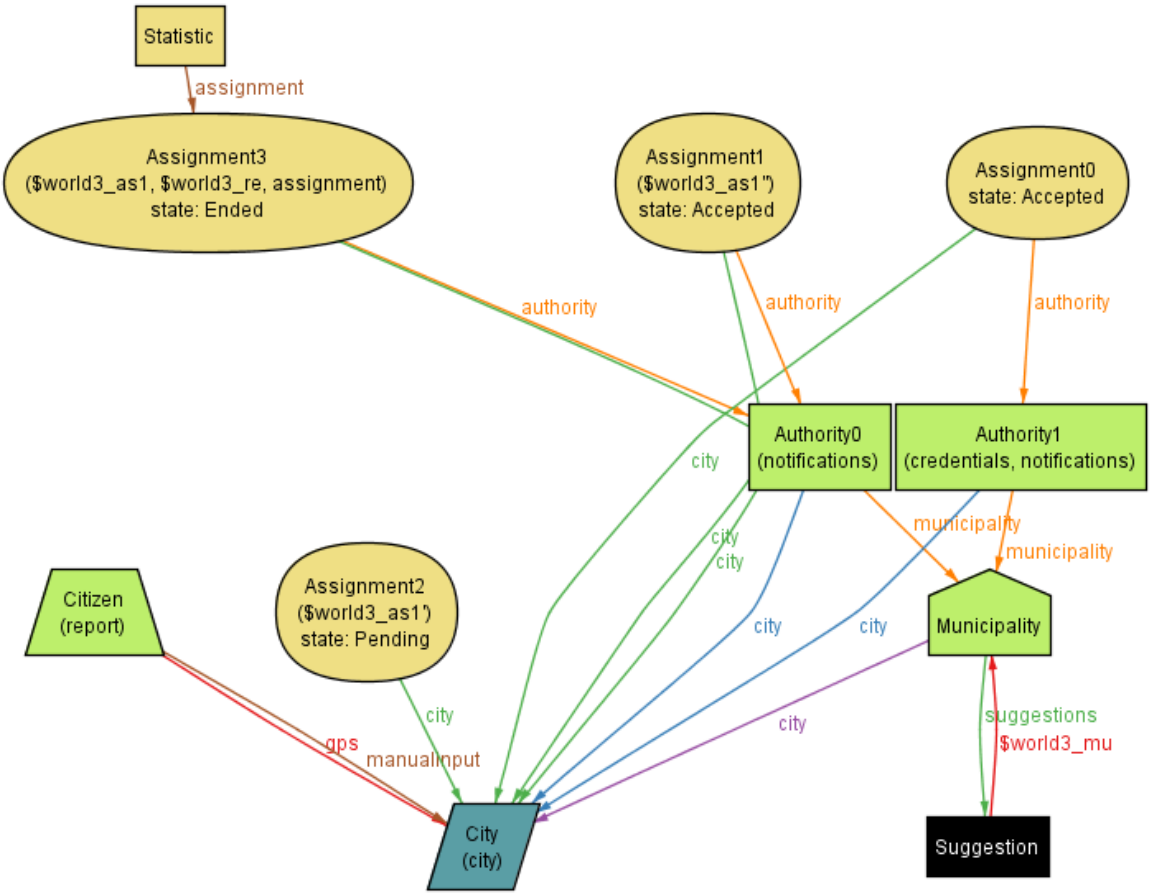

Figure 32: Third world
This world represents a more generic world where assignments pending,accepted,finished are all generated. Also here credentials are projected away for the sake of clarity.

5 Effort Spent

Maldini Pietro	
Section	Effort
Discussion about project, organization and project text analysis	5
Introduction	3
Overall Description	7
Specific Requirement's	9
Scenarios	3
UML Modelling	10
Alloy Modelling	9
Total	46 h

Paone Angelo	
Section	Effort
Discussion about project, organization and project text analysis	5
Introduction	2
Overall Description	6
Specific Requirement's	7
Scenarios	5
UML Modelling	14
Alloy Modelling	6
Total	45 h

6 References

- GPS Performances
- article about traffic violations in milan
- Vliet, Hans : van copertina Software engineering : principles and practice 2008
- Pressman, Roger S. Principi di ingegneria del software 2004