POLITECNICO
MILANO 1863

SAFESTREETS

REQUIREMENT ANALYSIS AND SPECIFICATION DOCUMENT

SAFESTREETS

edoardo putti

10 November 2019

Table of Contents

1. Introduction 3
   1.1 Purpose 3
   1.2 Scope 3
      1.2.1 Goals 4
   1.3 Definitions, Acronyms, Abbreviations 4
   1.3.1 Definitions 4
   1.3.2 Acronyms 6
   1.3.3 Abbreviations 6
   1.4 Revision history 6
   1.5 Document Structure 6
2. Overall Description 7
   2.1 Product perspective 7
   2.2 Product Functions 7
      2.2.1 Scenario 1 7
      2.2.2 Scenario 2 8
      2.2.3 Scenario 3 8
      2.2.4 Scenario 4 8
      2.2.5 Scenario 5 9
   2.3 User characteristics 9
      2.3.1 Actors 9
   2.4 Assumptions, dependencies and constrains 10
      2.4.1 Domain Assumptions 10
3. Specific Requirements 11
   3.1 External Interface Requirements 11
      3.1.1 User Interfaces 11
      3.1.2 Hardware Interfaces 11
      3.1.3 Software Interfaces 12
      3.1.4 Communication Interfaces 12
   3.2 UML modeling 12

3.2.1 Use case diagrams 12
3.2.2 Domain Models 17
3.2.3 Activity diagrams 18
3.3 Functional Requirement 20
3.4 Performance Requirement 21
3.5 Design Constrains 22
3.5.1 Standard Compliance 22
3.5.2 Hardware limitations 22
3.5.3 Any other constraint 22
3.6 Software System Attributes 23
3.6.1 Reliability 23
3.6.2 Availability 23
3.6.3 Security 23
3.6.4 Maintainability 23
3.6.5 Portability 23
4. Formal Analysis Using Alloy 24
4.1 Alloy Code 24
4.2 Results of Alloy Analysis 34
4.3 Alloy Model 35
5. Effort Spent 35

Table of Figures

Figure 1 mockups of the interfaces in order Home Page, New Report, User map, Authority Map 11
Figure 2 Person Use Case diagram 12
Figure 3 User Use Case Diagram 13
Figure 4 Authority Use Case Diagram 14
Figure 5 Authority sign-up Activity Diagram 18
Figure 6 User new report activity diagram 19

1. Introduction

1.1 Purpose

This document is the Requirements Analysis and specification Document (RASD) for the Safestreets application offers, about requirements and goals that the system must present. This document offers also an analysis of the world and shared phenomena regarding Safestreets. RASD contains class diagrams to show domain models and other diagrams which illustrates, with more details, transaction of the functionalities of the application.

1.2 Scope

Safestreets is a crowd-sourced application that allows users to send reports about traffic violations. A person become a user of Safestreets by registering himself into the application. After this phase, user can start to use the basic functionalities of the app (e.g. sending reports and retrieving traffic information). Safestreets allows, also, to authorities to register, log-in and use advance functionalities of the app.

The app allows users to send reports to the authorities and querying the app for traffic and violations highlights. The goal of Safestreets is to send the reports, made by the users, to the authorities helping and facilitating their job.

When a user creates a new report, he can add a picture of the car plate so that the app can extrapolate the license plate automatically or insert it manually, user can also add the geographical position of the violation using the GPS or insert it manually, and of course the type of the traffic violation and its timing. Moreover users are allowed to visualize traffic highlights using the app.

The app allows authorities to querying all the reports and also to extrapolate useful information and statistic from the app.

The system interfaces with other firms (e.g. municipality services, territory maps companies) to offer a more comprehensive costumer experience.

1.2.1 Goals

[G1] a Person should be able to have a profile on Safestreets

[G2]: a User should be able to send reports regarding traffic violations.
[G2]#1 specify type of traffic violation
[G2]#2 attach a picture of the car plate
[G2]#3 include the geographical position

[G3]: Authorities should be able to access to all the reports

[G4]: Costumers should be able to access to different kind of information depending on their role
[G4]#1 users can access only to traffic highlights, and their reports
[G4]#2 authorities can access to traffic highlights, violation statistics and all the reports

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

Here is provided a list of definitions of words and expressions used in the document

- **Users:** the “normal” costumer of the application that exploits the application only to send traffic violations and to retrieve information from it.

- **Authorities:** the customer of the application that exploits it to monitor the reports and take adequate measures.

- **FIN:** is the Force Identification Number that is unique and is associated to an Authority

- **Customers:** people that uses Safestreets, can be an authority or a common user

- **Report:** a module reporting a traffic violations containing useful data

- **Traffic violations:** violation of the laws that regulate vehicle operation on streets and highways.

- **Affiliated company:** a company that has deals with the S2B

- **Welcome page:** the page where the user is redirected after completing the sign-up process and logging in for the first time.

- **Valid Credentials:** Name, Surname, personal email address and password.
  - **Name:** it should be non-empty and it should contain only alphabetical characters
  - **Surname:** it should be non-empty and it should contain only alphabetical characters
  - **Email:** it should be non-empty and a valid email address, with an alphanumerical string followed by a ‘@’, followed by an alphanumerical string, a dot, and the domain name
  - **Password:** it should be a string with at least 8 characters

1.3.2 Acronyms

- API = Application Programming Interface
- GPS = Global Positioning System
- UI = User Interface
- S2B = Software To Be

1.3.3 Abbreviations

- G*n* = *nth* goal
- D*n* = *nth* domain assumption
- R*n* = *nth* requirement

1.4 Revision history

RASD-1.0 contains all the basics field of a RASD

1.5 Document Structure

After purpose and scope, used to briefly introduce the topic, are delineated the goals that the S2B should achieve coupled with a list of useful definitions and acronyms. Subsequentially, the text proceeds with an analysis of the functions that the app should provide. The analysis starts with a general exposition of the scenarios and becomes gradually more detailed passing through the analysis of the actors that will interact with the S2B and the statements of the domain assumptions. After that, the specific requirements are exposed focusing firstly on the external interfaces and then providing the models used to highlights the relations between the actors and S2B and describe the internal structure of the latter. After that, Functional and non-Functional

requirements are sequentially discussed. Before ending with the effort spent the references is provided a formal analysis performed with alloy.

2. Overall Description

2.1 Product perspective

The system will be developed from scratch and it will use externals services including Google services and the services provided by affiliated companies. This is because the services provided by Google offers high quality and reliability and there is no point in trying to redevelop them. The services of affiliated companies are needed to interface with them information regarding traffic and vehicles.

2.2 Product Functions

Here are provided several scenarios to better delineate the purposes for which the app should be designed, the situations the S2B will deal with and more generally to have a better comprehension of the associated environment. The scenario are describe in an informal way in this section but they will be formalized in the next chapter.

2.2.1 Scenario 1

Carlo is a diligent citizen that riding his bike find a car parked on the cycle path. Worried about this parking violation take his phone and open the Safestreets app, in which he was recently registered, and start creating a new report. After having specified what type of violations occurred and the car model, the app asks him if he wants to insert the car plate or take a photo of it he choose to take a picture of

the car plate and attach it to the report, he also allows the app to collect his geographical position through the phone GPS and attach it to the report. Finally he confirm and send the report. Finished the report he received a mail containing the last report made by him to the mail address given during the registration. Now he can also see in his private area on the app the new report and his current status.

2.2.2 Scenario 2

Paolo is a citizen that wants to check the traffic condition in the various street that bring to his office, so he decide to rely on Safestreets to choose the best path. He opens the app and log-in, open the integrated map in the app, agreed to the acquisition of his position, and insert the office address, the app gives him the various rotes with the estimated time of arrival and also shows if some other users reported some traffic violation, that influence the estimated time, along the path. Paolo decide the fast and secure route to go to his office.

2.2.3 Scenario 3

Gabriele is a police officer that agreed to the collaboration with Safestreets and register himself in the application as an authority. During his watch he opens the application and check the reports. Today there are lots of reports so he open the integrated map and search for all the reports with a position far at most 5Km to his position. Using this service he optimize the watch`s route.

2.2.4 Scenario 4

Giuseppe is a police officer that wants to start collaborating with the Safestreets application. So he opens the app and start the registration . he insert his credential and his force identification number. After that a mail is sent to the municipality

that checks the FIN and after this accepts the request. Giuseppe is notified by mail that now can access to the application as an authority

2.2.5 Scenario 5

Piero is a chief police officer that wants to understand how to relocate efficaciously his men so decide to use the app Safestreets. As authority he can access to all statistical data and in particular, he is interested in the zone with an high density of reports so he opens the integrated map and the system highlights the zones with different colors based on the reports density in the zone.

2.3 User characteristics

2.3.1 Actors

- Person: a person that does not have a registered account. The only thing that he/she can do is to proceed with the Sign Up operation.

- User: a person passed through a successful registration process and now able to use Safestreets services. He/she can login to the system and, after that, use the platform's functionalities.

- Authority: a police officer that passed through a successful registration process and now able to use all Safestreets services. He/she can login to the system and, after that, use all the platform's functionalities.

- Google: the system with whom the S2B retrieves the maps and related information about routes, real-time traffic situations, estimated travel time.

2.4 Assumptions, dependencies and constrains

2.4.1 Domain Assumptions

[D1] the User's device should be allow the app to retrieve the language settings

[D2] when the registration process begin, the Person always insert his/her credential

[D3] when the S2B sends an email, it is always received by the receiver

[D4] every Person has an email address

[D5] the User and the Authority shall remember their password

[D6] the User and the Authority know only their password

[D7] the User's device has a working GPS and a camera installed, to which the app has access

[D8] the municipality is always a trusted source

[D9] the user always insert true information in a report

[D10] Google Maps services take traffic into consideration

[D11] every Person associated that works in the police have a unique FIN

[D12] the same car can't commit two different violations in the same day in less than 5 minutes from one and another

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

Here is provided some basic mockups to show how the interface should appear to the user:

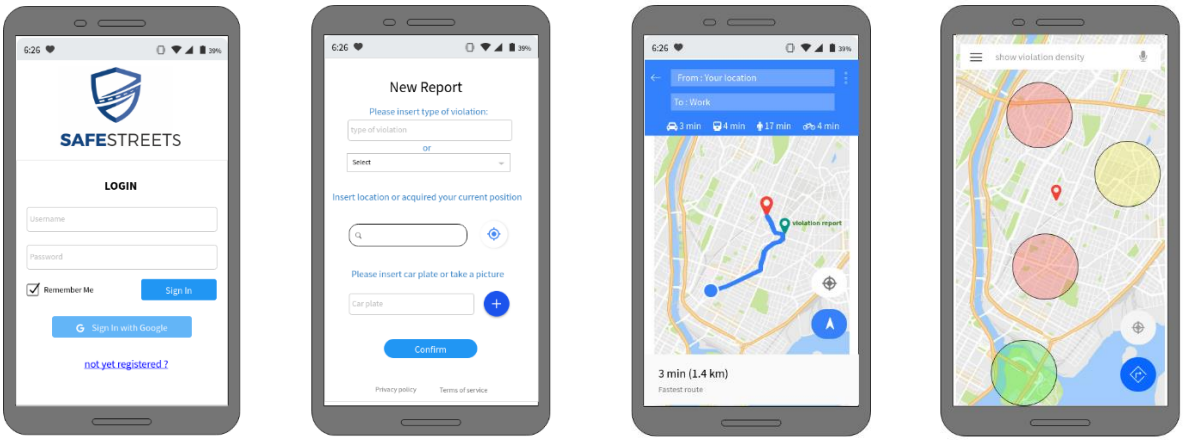


Figure 1 mockups of the interfaces in order Home Page, New Report, User map, Authority Map

3.1.2 Hardware Interfaces

The main hardware interface of the system consists in the access to the GPS data and the camera in the mobile application. The application also requires Internet connectivity.

3.1.3 Software Interfaces

The mobile application must support Android, iOS and the remaining main Oss (further details are discussed in paragraph 3.6.5 Portability).

3.1.4 Communication Interfaces

The communication between clients and server should be HTTP requests/responses based.

3.2 UML modeling

In this section, is formalized the S2B in terms of UML models.
The Use Case Diagrams are divided into parts to slightly improve the readability of the Diagrams. After the Diagrams, descriptions of the main Use Cases are provided.
After that is provided a Class Diagram of the whole system and then some Activity Diagrams, to better explain the structure of the S2B and its behavior.

3.2.1 Use case diagrams

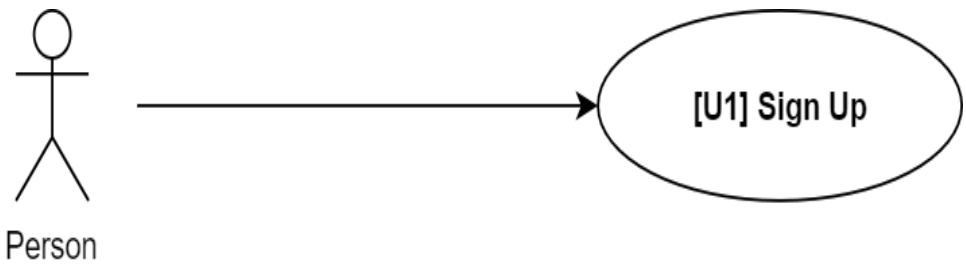


Figure 2 Person Use Case diagram

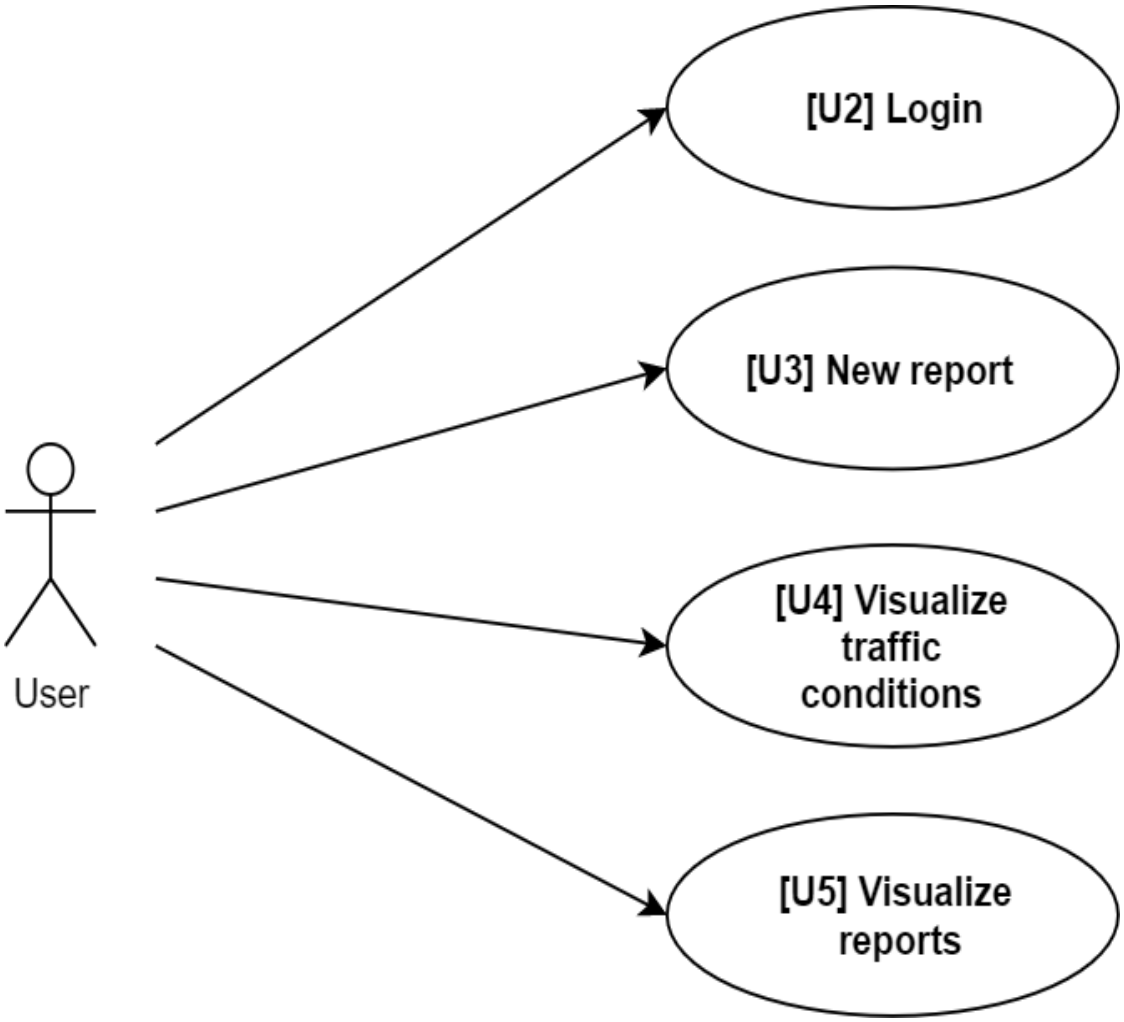


Figure 3 User Use Case Diagram

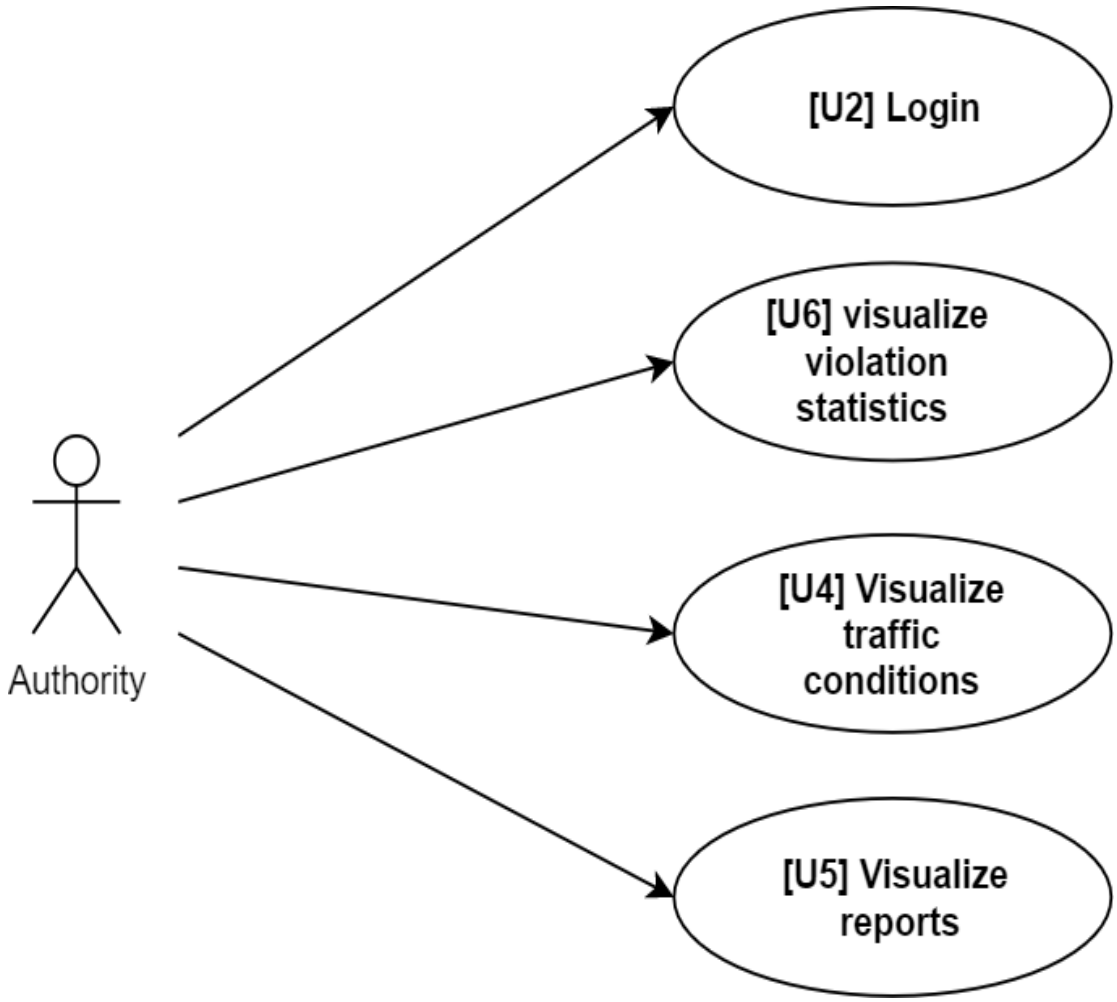


Figure 4 Authority Use Case Diagram

Sign Up for regular user

Use Case:	Sign Up
Actors:	Person
Entry Condition:	None.
Flow of events:	the Person insert *valid credentials*; the app sends an email with the confirmation link; the Person give the confirmation through the link on the email; the app shows the *welcome page* to the new User.
Secondary flows:	None.
Exceptions:	The Person inserts non *valid credentials;* The sign-up cannot proceed.
Post conditions:	The person is successfully signed up and become an actual User.

Sign Up for an authority

Use Case:	Sign Up
Actors:	Authority
Entry Condition:	None.
Flow of events:	the Person insert *valid credentials*; the Person insert his *FIN;* the app sends an email with the confirmation link to the municipality; the Municipality check the *FIN* and give the confirmation through the link on the email; the app send a welcome email ; the app shows the *welcome page* to the new Authority.
Secondary flows:	None.
Exceptions:	The Person inserts non *valid credentials;* The Person doesn't insert a *FIN;* The Municipality doesn't approve the request; The sign-up cannot proceed.
Post conditions:	The person is successfully signed up and become an actual Authority.

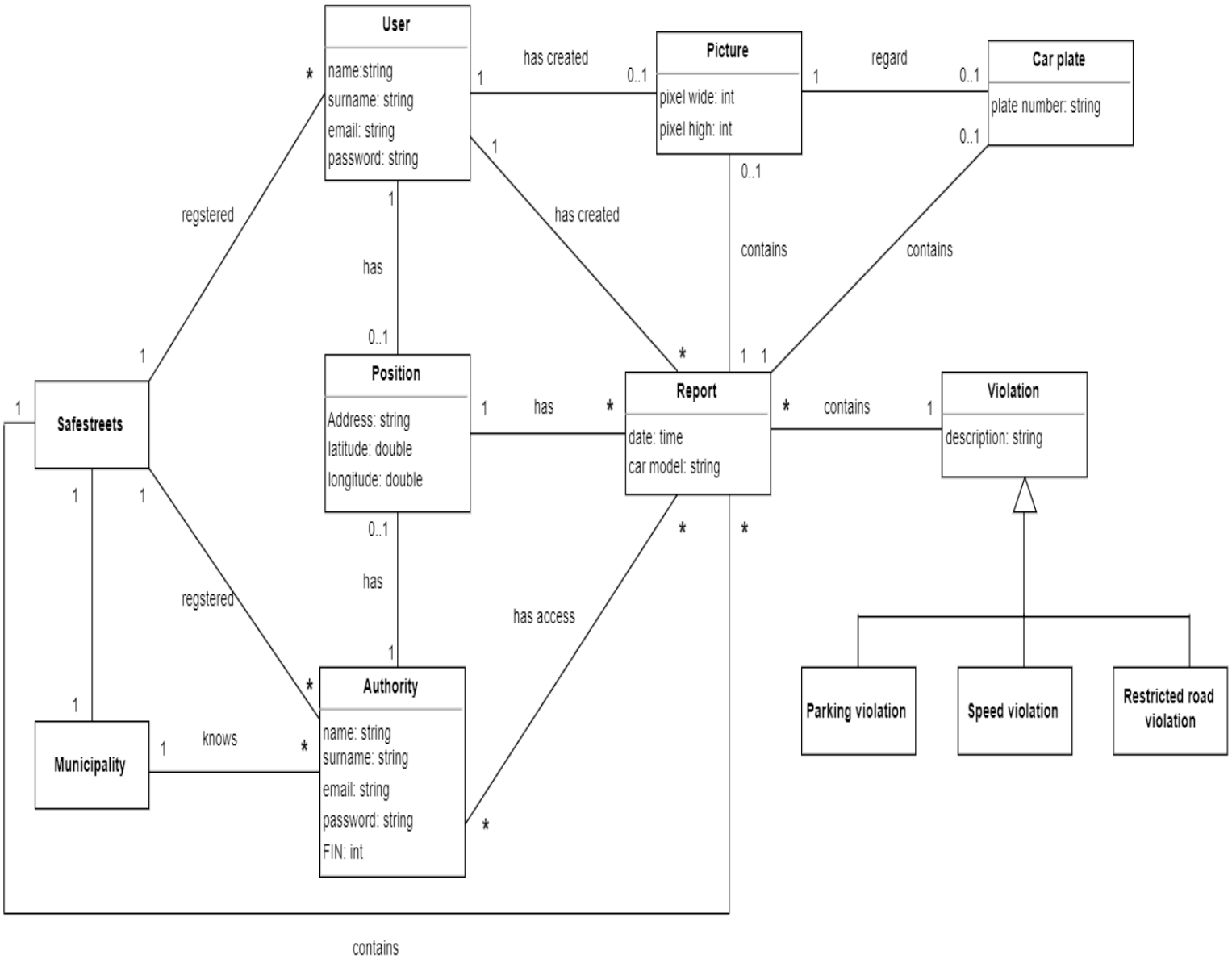
User creates a new report

Use Case:	New Report
Actors:	User
Entry Condition:	The user completed the sign-in process.
Flow of events:	The User creates a new report; User specifies the type of violation is occurred; User specifies car model: User specifies the violation's position with his phone GPS; User takes a pictures of the car plate: User confirms the current report; The app redirect the User to the main page.
Secondary flows:	User insert manually the violation's position; User insert manually the car plate. User confirms the current report; The app redirect the User to the main page.
Exceptions:	None.
Post conditions:	The report is saved.

Authority choose a report

Use Case:	Choose a report
Actors:	Authority
Entry Condition:	The Authority completed the sign-in process.
Flow of events:	The authority insert his location; The app shows all the reports located nearby; The authority picks one; The app shows the path to the location; The authority sets the status of the report to terminated.
Secondary flows:	Authority insert manually a position;
Exceptions:	None.
Post conditions:	The report's status is set to terminate.

3.2.2 Domain Models



3.2.3 Activity diagrams

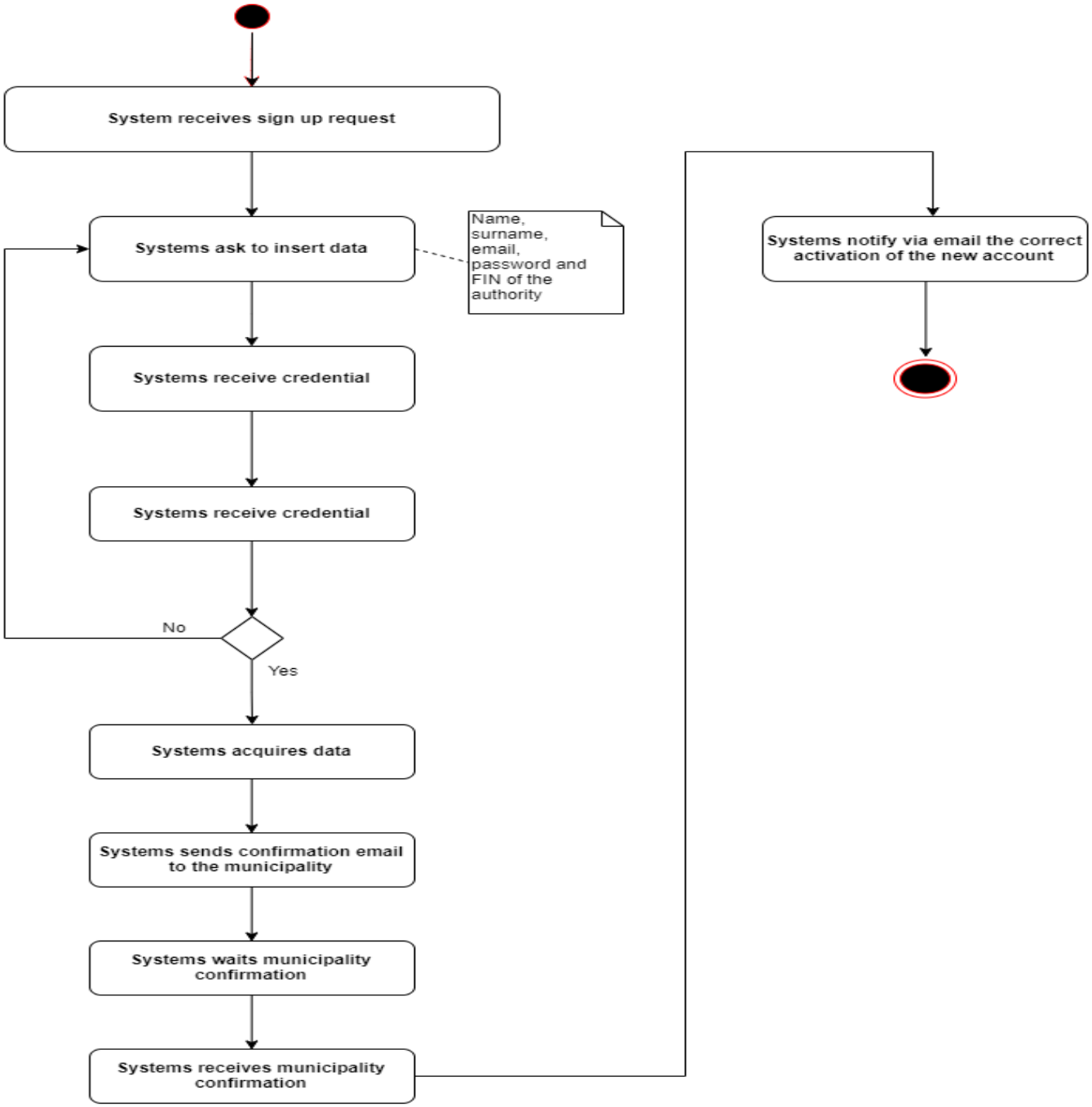


Figure 5 Authority sign-up Activity Diagram

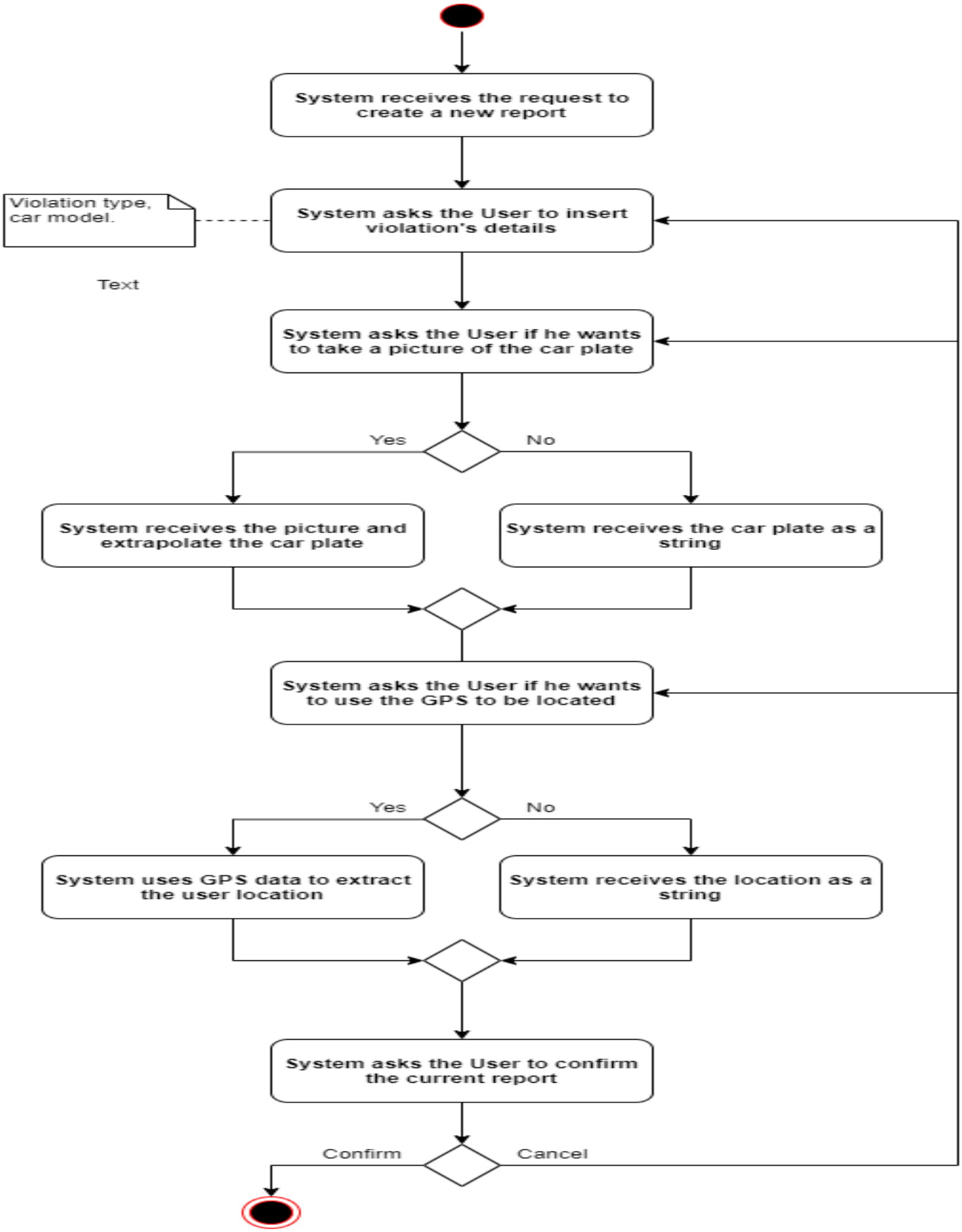


Figure 6 User new report activity diagram

3.3 Functional Requirement

[G1] a Person should be able to have a profile on Safestreets

[R1] the S2B must provide to every person a way to begin the registration process
[R2] after the insertion of the credential and their validation, the S2B has to send, to the provided email address, an email with the activation link
[R3] the S2B must refuse the registration if the inserted email is already associated to an account
[R4] when the Person confirms through the activation link, he/she becomes a User
[R5] in the case of non-valid credentials, the system must reject them and restart the registration process
[R6] the S2B must grant access to the User if and only if the User insert an existing email and the associated password

[D2] when the registration process begin, the Person always insert his/her credential

[D3] when the S2B sends an email, it is always received by the receiver

[D4] every Person has an email address

[D5] the User and the Authority shall remember their password

[D6] the User and the Authority know only their password

[G2] a User should be able to send reports regarding traffic violations.

[R7] the system must provide a way to start the creation of a new report
[R8] during the process the User shall fill all the report's fields

[D7] the User's device has a working GPS installed, to which the app has access

[D9] the user always insert true information in a report

[D12] the same car can't commit two different violations in the same day in less than 5 minutes from one and another

[G3] Authorities should be able to access to all the reports

[R9] existing reports can be viewed as a list or in the map based on the current location
[R10] from the list or the map the system provides a way to visualize a single report and its details
[R11] after managed the report the Authority can set is status to terminated

[D7] the User's device has a working GPS installed, to which the app has access

[G4] Costumers should be able to access to different kind of information depending on their role

[R12] the system must grant access to different type of users
[R13] the system must allow User to see their reports and their status
[R14] the system must allow an Authority to see all the reports stored in Safestreets and give him the possibility to consult both stats and aggregate data

[D4] every Person has an email address

[D5] the User and the Authority shall remember their password

[D6] the User and the Authority know only their password

[D11] every Person associated that works in the police have a unique FIN

3.4 Performance Requirement

The system has to be able to respond to a possibly great number of simultaneous requests, and more generally to a great number of request throughout the day. The S2B, at least for the start, will only be available for the Lombardy region. Based on demographic analysis (number of inhabitants, number of people under the age of 60,

number of smartphones sold over the past 2 years), it was decided to design the S2B to support 100,000 users simultaneously, but scalability needs to be guaranteed.

3.5 Design Constrains

3.5.1 Standard Compliance

To ensure interoperability the S2B will follow the W3C web standard and will be as adherent as possible to code practices in relation to the use of HTML/XHTML, CSS and Java programming language. Moreover, the use of non-opensource libraries will be avoided.

3.5.2 Hardware limitations

- Mobile App:
  - 3G or more powerful connection
  - GPS
  - Phone Camera
  - Space for app package
- Web App:
  - Modern browser able to retrieve positions
  - PC Camera

3.5.3 Any other constraint

The system will have to ask for user's permission in order to retrieve and use their position and the phone camera. Email addresses will not be use for commercial uses.

3.6 Software System Attributes

3.6.1 Reliability

The system must guarantee a 24/7 service. Very small deviations from this requirements will be obviously acceptable.

3.6.2 Availability

The S2B must guarantees a 3-nines availability (**99.9** percent) with a downtime not greater than 8 hours per year.

3.6.3 Security

User credentials will be stored. Data confidentiality is a primary concern. The S2B must be able to adopt access management protocols and communication protocols able to prevent not granted access and/or sniffing/Spoofing activities performed by thirds.

3.6.4 Maintainability

The S2B must be designed in a way to easily correct defects or their cause, repair or replace faulty or worn out components without having to replace still working parts, prevent unexpected working conditions, maximize its useful life, maximize efficiency, and safety, meet new requirements, make future maintenance and cope with a changed environment.

3.6.5 Portability

The S2B must be able to run in all main mobile OS (Android, iOS, Windows-Phone OS, MIUI), and to be supported by all the main Web Browser (Google Chrome, Safari, Firefox, Microsoft Edge)

4. Formal Analysis Using Alloy

4.1 Alloy Code

```
//=============================================================

//====================== PRIMITIVE SIGNATURES ========================


sig Name {}


sig Surname {}


sig Email {}


sig Address {}


sig FIN {}


sig Double {}


sig CarModel {}
```

```
sig CarPlate {}

enum Status {
                    Open,
                    Closed
                    }

//=============================================================
//========================== SIGNATURES ============================

sig Day { Value: Int } {Value>0}

sig Time { hours: Int,
                    minutes: Int
                    } {hours >0 and minutes>0}

sig Person  { name: Name,
                    surname: Surname,
                    email: Email,
                    isUser: lone User,
                    isAuthority: lone Authority
                    }

some sig  User { name: Name,
```

```
                surname: Surname,
                email: Email,
                createsReport: set Report,
                currentlyAtPos: Position
                }

some sig  Authority { name:one Name,
                      surname:one Surname,
                      email:one Email,
                      fin:one FIN,
                      currentlyAtPos: Position,
                      allReports: set Report
                      }

sig Position { address: one Address,
               latitude: one Double,
               longitude: one Double
               }

some sig Report { day:  Day,
                  time:  Time,
                  creator: one User,
                  carModel: lone CarModel,
                  position:  Position,
                  carPlate:  CarPlate,
```

violation: TrafficViolation,

status: one Status

}

```
abstract sig TrafficViolation{}

sig ParkingViolation extends TrafficViolation {}

sig SpeedViolation extends TrafficViolation {}

//=============================================================

//========================== FACT ============================

fact AllPersonAreOrAnAuthorityOrAUser

{

all p : Person | (p.isUser = User and p.isAuthority = none) or (p.isAuthority = Authority and p.isUser = none)

}

fact EveryUserHas2Person

{

all u : User | some disj p1,p2 : Person | p1.isUser = u => p1.isAuthority = none and samePerson[p1,p2]
```

```
}

fact EveryAuthorityHas2Person

{

all a : Authority | some disj p1,p2 : Person |  p1.isAuthority = a => p1.isUser = none and samePerson[p1,p2]

}

fact SameEmailImpliesSamePerson

{

 all p1,p2 : Person | p1.email = p2.email => (samePerson[p1,p2])

}

fact AllNameMustBelongToSomeone

{

all n : Name | some u : User, a : Authority | (a.name = n) or (u.name = n)

}

fact AllSurameMustBelongToSomeone

{

all s : Surname | some u : User, a : Authority | (a.surname = s) or (u.surname = s)

}

fact AllEmailMustBelongToSomeone

{
```

```
all e : Email | some  u : User, a : Authority | (u.email = e) or (a.email = e)
}

fact noMailShared
{
 all u : User, a : Authority | (u.email != a.email)
}

fact AllAddressesMustBelongToPositions
{
all a : Address | some pos : Position | pos.address = a
}

fact eachUserIsUnique
{
 all disj  u1 , u2 : User |  (u1.name != u2.name ) and (u1.email != u2.email) and (u1.surname !=u2.surname)
}

fact eachAuthorityIsUnique
{
        all disj a1 , a2: Authority | (a1.fin != a2.fin) and (a1.email != a2.email)
}

fact authorityHaveAccessToAllTheReports
```

```
{
 all r : Report , a : Authority | (r in a.allReports)
}


 fact justOneReportOfTheSameViolationPerDayReferringToTheSameCar
{
        all disj r1,r2 : Report |  ( r1.day != r2.day) and (r1.violation != r2.violation) and
(r1.carPlate != r2.carPlate) and (r1.position != r2.position)
}

fact SameCarCommitsMultipleViolationAfterAtLeast5Minutes
{
 all disj r1, r2 : Report | ((r1.day = r2.day)and(r1.carPlate = r2.carPlate)and(r1.violation !=
r2.violation)) =>( ((r2.time.minutes - r1.time.minutes) > 4)or((r1.time.minutes -
r1.time.minutes) > 4))
}

fact justOneCreatorForReport
{
 all r : Report, u : User | ( u.createsReport = r) <=> (r.creator = u)
}

fact compleatePosition
{
```

```
 all disj p1,p2 : Position |  (p1.address != p2.address) and (p1.latitude != p2.latitude and
p1.longitude != p2.longitude)

}




//=============================================================

//==========================ASSERTION============================


assert user_consistency

{

no disj u1,u2 : User  |   (u1.name = u2.name ) and(u1.email = u2.email) and(u1.surname =
u2.surname)

}


//check user_consistency


assert authority_consistency

{

        no disj a1 , a2: Authority | (a1.fin = a2.fin) and (a1.email = a2.email)

}


check authority_consistency
```

```
assert justOneReport
{
        no disj r1,r2 : Report |  ( r1.day = r2.day) and (r1.violation = r2.violation) and (r1.carPlate = r2.carPlate) and (r1.position = r2.position)
}

check justOneReport



//=============================================================
//============================PREDICATES============================
pred samePerson[p1,p2 : Person]
{
p1.email = p2.email and p1.name = p2.name and p1. surname = p2.surname
}

pred showAllReportsClosed
{
some r : Report | r.status= Closed
}

run showAllReportsClosed

pred showAllReportsOpen
```

```
{
some r : Report | r.status= Open
}

run showAllReportsOpen

pred showAllUserThatSendReports
{
some r : Report | r.creator = User
}

run showAllUserThatSendReports

pred show { }

run show
```

4.2 Results of Alloy Analysis

Executing "Check authority_consistency"
Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
8304 vars. 540 primary vars. 17912 clauses. 43ms.
No counterexample found. Assertion may be valid. 12ms.

Executing "Check justOneReport"
Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
8380 vars. 540 primary vars. 18114 clauses. 39ms.
No counterexample found. Assertion may be valid. 27ms.

Executing "Run showAllReportsClosed"
Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
8229 vars. 537 primary vars. 17716 clauses. 36ms.
Instance found. Predicate is consistent. 39ms.

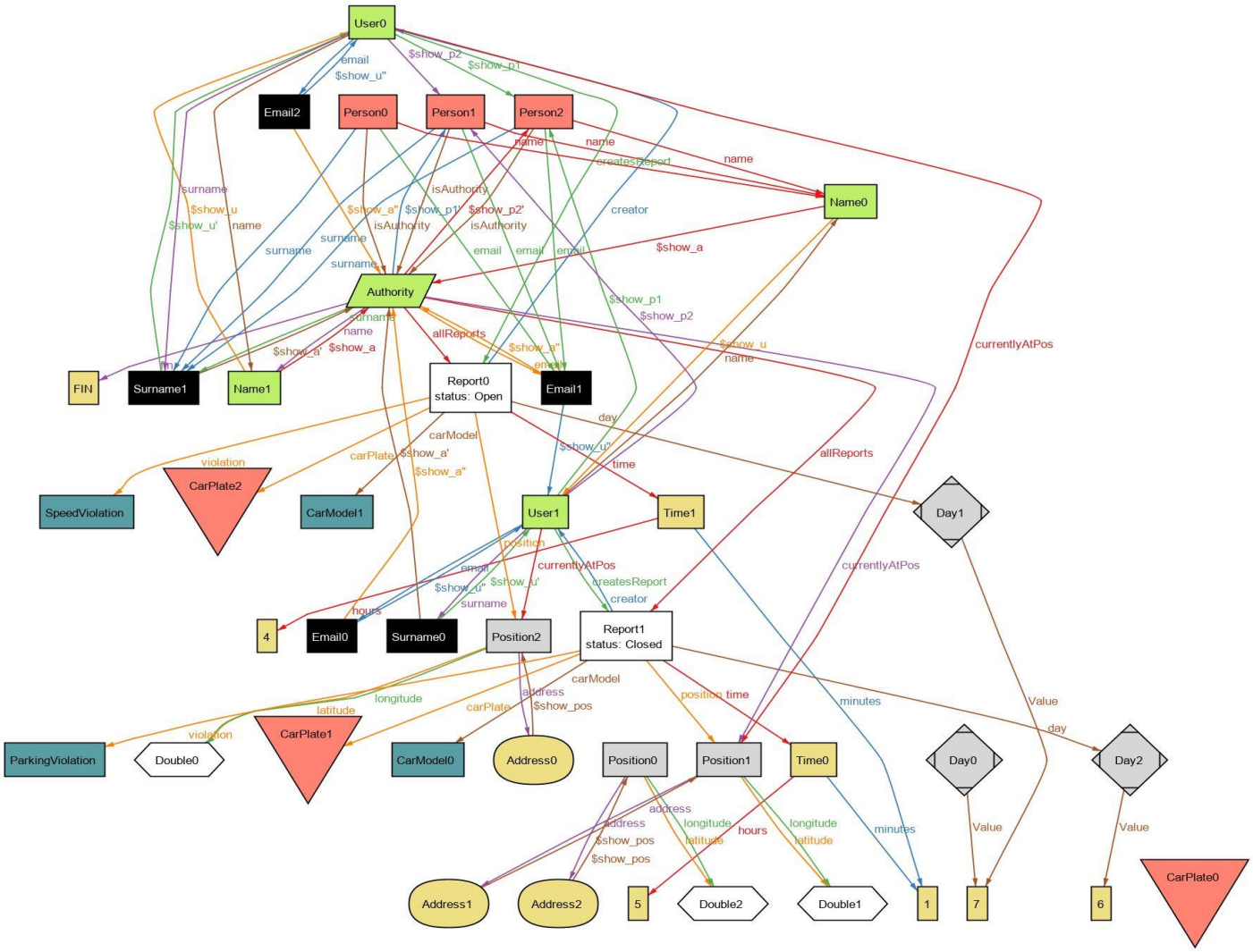
Executing "Run showAllReportsOpen"
Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
8229 vars. 537 primary vars. 17716 clauses. 40ms.
Instance found. Predicate is consistent. 38ms.

Executing "Run showAllUserThatSendReports"
Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
8241 vars. 537 primary vars. 17754 clauses. 21ms.
Instance found. Predicate is consistent. 29ms.

Executing "Run show"
Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
8213 vars. 534 primary vars. 17693 clauses. 25ms.
Instance found. Predicate is consistent. 36ms.

6 commands were executed. The results are:
#1: No counterexample found. authority_consistency may be valid.
#2: No counterexample found. justOneReport may be valid.
#3: **Instance found.** showAllReportsClosed is consistent.
#4: **Instance found.** showAllReportsOpen is consistent.
#5: **Instance found.** showAllUserThatSendReports is consistent.
#6: **Instance found.** show is consistent.

4.3 Alloy Model



5. Effort Spent

I spent more than 5 days to complete the project. This is due to the fact that I was alone and also because I never used Alloy.

I spent an entire day for the goal identification and World and Shared phenomena, Another day was dedicated to the domain model and the activity diagrams,

Two days were spent for the Alloy modeling and the last day was spent for the other paragraphs .