



SAPIENZA
UNIVERSITÀ DI ROMA

FACOLTÀ DI
INGEGNERIA DELL'INFORMAZIONE, INFORMATICA E STATISTICA

DIPARTIMENTO
DI INGEGNERIA INFORMATICA, AUTOMATICA E GESTIONALE

Human Computer Interaction course
Final project

A.Y. 2020/21

R.O.M.A.

Professors:
Tiziana Catarci
Valeria Mirabella

Students:
Giovanni Pecorelli 1799865
Jacopo Rossi 1801667
Giacomo Venneri 1810169

Contents

1	Introduction	2
1.1	Work plan	2
1.2	Overview	3
2	Questionnaire	4
2.1	Conclusions	7
3	Personas and Scenarios	8
3.1	Personas	8
3.1.1	Persona: Caterina	8
3.1.2	Persona: Gabriele	9
3.1.3	Persona: Franco	9
4	Competitor analysis	11
4.1	R.O.M.A. functionalities	12
5	HTA and STN	13
5.1	Definitions	13
5.2	Start using R.O.M.A.	14
5.3	Add a new report	15
5.4	Check the progress of a report	16
6	First prototype	17
6.1	Expert-based heuristic evaluation	19
6.1.1	Heuristic evaluation	20
7	Second prototype	22
7.1	Prototype description	22
7.2	Correction of detected usability issues	22
8	Third prototype	27
8.1	User-based evaluation technique – think aloud	27
8.2	Correction of detected defects	28
8.2.1	Registration and login alternatives	28
8.2.2	Change username	29
8.2.3	Fullscreen images	29
8.2.4	Logout, delete account and settings page	30
8.2.5	Remove report	30
8.3	User-based evaluation technique – controlled experiment	31
8.3.1	The problem	31
8.3.2	ANOVA	32
9	Final version	34
10	Conclusions	39
10.1	Future works	39

Chapter 1

Introduction

This report describes the workflow about the development of a mobile application called R.O.M.A.

Currently in Italy the procedure for reporting urban problems is quite inconvenient: after finding out which is the responsible entity (municipality, province, region, private companies...) the most used mean of communication is a registered letter with return receipt, or PEC, or a simple email. Our idea is to simplify this process with the development of R.O.M.A., which aims to reduce waiting times and distances between citizens and competent authorities.

The idea behind this application is to give people the possibility to report episodes of urban problems when they encounter one. All the user has to do is to take a picture of the problem with his smartphone, and add a brief description to create a new report, ready to be taken care of by the authorities. These problems can be of any nature, from infrastructures malfunction to potholes in the streets, and many more. The app is potentially addressed to everyone and its goal is to help people no matter their age or speciality.

An important aspect is that we developed the idea of this application by considering the user centered design approach, focusing on developing an app that is suitable for everyone.

1.1 Work plan

To work as a team, we took advantage by several tools and communication means. All of the interaction between the three authors happened through video calls platforms, since we couldn't meet in presence.

To develop the app we used Android Studio, using Java and XML languages. To share the same software and code and to manage their versioning we used GitHub.

After the collection of the requirements, the analysis of the data and the analysis of the competitors, we developed a first prototype through the use of mockups, realized with Balsamiq. This first prototype was then evaluated through an expert-based technique to come up with its possible defects. These defects were corrected and the second prototype was developed as an interactive interface made thanks to Android Studio. The second prototype was analyzed through a user-based technique to come up with its defects. Another time, these defects were corrected and the third prototype was developed as a fully working Android app. This was the final version of R.O.M.A.

1.2 Overview

In **Chapter 2** are presented the results of the *questionnaire* given to a sample of people of different age, occupation, style of life, etc. The results are used to find out the characteristics of our main audience and to tune both the functionalities and the appearance of our product.

In **Chapter 3** we will introduce three *personas* (possible users) of R.O.M.A., and analyze three different sample of *scenarios* where they can use our app to solve the problems they encounter.

In **Chapter 4** we will then perform a *competitor analysis*, exploring the app stores to find other applications which offer similar features as our and comparing them to improve our offer.

In **Chapter 5** the design of the possible behaviors and interactions between the users and our app is discussed thanks to the use of some task modelling tools like *HTA* and *STN*. We will also explain more in detail three of the different functionalities offered by the app.

In **Chapter 6** the implementation of the first prototype from the initial design is shown. This first prototype has been implemented with the use of mockups. In the next chapter there is the application of an *expert-based evaluation* technique to the prototype, done to come up with the second prototype.

In **Chapter 7** is shown the implementation of the second prototype after the expert-based evaluation.

In **Chapter 8** there is the application of a *user-based evaluation* technique to the prototype. It also contains an *ANOVA* analysis.

In **Chapter 9** there is the presentation of the *final product* and its features.

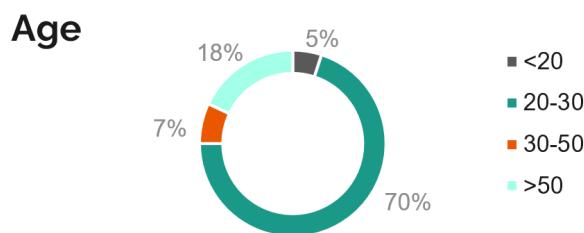
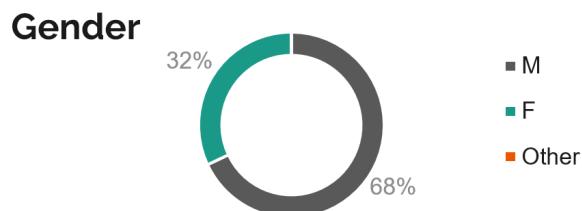
In **Chapter 10** the *conclusions* of the work and the possible future work are finally presented.

Chapter 2

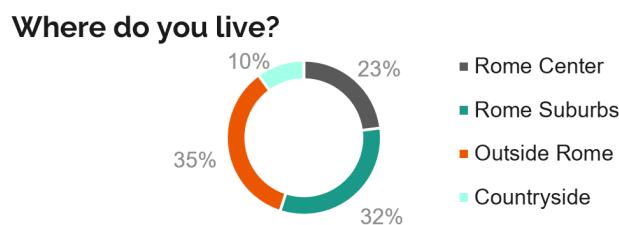
Questionnaire

In this chapter we will present and discuss the result of the questionnaire we gave to the possible users of the app, to discover which are the common traits and habits of the user base of R.O.M.A. This information will later be used to define the identity of our average user.

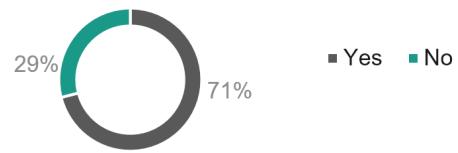
The survey was created in Google Modules. The questions and the answer were in Italian. The reason for this is that the language of the app will be Italian, since the app is meant to be used only in Rome. Another reason for this choice is to be more inclusive toward our user base less experienced with the English language. The survey was shared in most of our University group chats and with our familiars and people close to us. We collected 49 answers in total.



From this answer we confirm what we thought: our app has a userbase with a very heterogeneous age range, spanning from teenagers to older people.

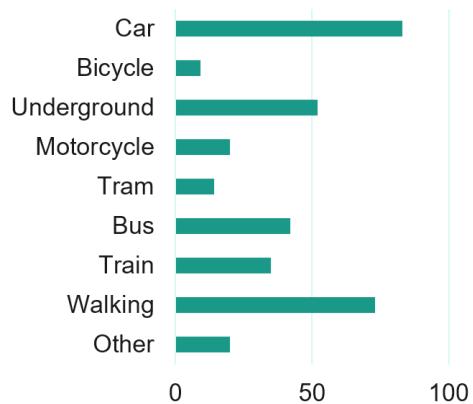


If you live outside Rome, would you still use the app?



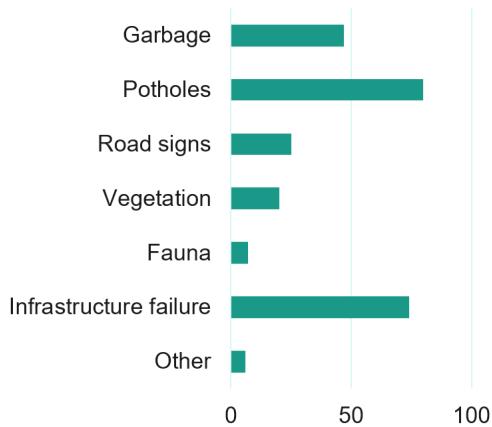
From these two questions, we find out that our userbase is either located in the city of Rome, or, if located outside Rome, is still interested in using the app to report problems in the capital. From this results, we decided to focus the scope of our app only in the territory of the city of Rome, allowing our user to create reports only in a restricted area.

Which of these means of transport do you use on a typical day?

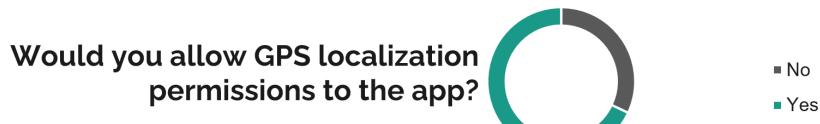


Since our userbase has a very heterogeneous utilization of public transports, we will keep this in consideration when deciding which categories of problems will be available in our app. If the survey had highlighted a means of transport more used than the others by a large margin, we would have given more attention to it, creating an entire category only for that particular transport.

Which of these problems regards you the most?



In this question we wanted to find out which problems were the most common among our surveyed. All the categories that we decided beforehand received at least one vote, and will be present in the app. The three answers that selected "Other" suggested some kind of problem related to the road. This induced us to consider the idea of adding another category ("Other road problems") in the final version of R.O.M.A.



With these three questions we wanted to decide if these three features received enough approval from our surveyed and so if we had to implement them in R.O.M.A.

The first feature, GPS localization, was considered useful, and so will be present in the app.

Same thing with the feature regarding urgent reports.

The third feature, anonymous reports, did not receive enough votes for us to be considered, and so we decided to discard the idea of implementing it.

User profile

Age	18-100
Gender	Any
Job Titles	Any
Education	Any
Location	The app allows reports only in the territory of the City of Rome, but the reports can be added from anywhere
Technology	Some experience with using Android apps
Passions	A desire to improve the city of Rome

Table 2.1: User profile

2.1 Conclusions

From the analysis of the data coming from the questionnaire some important conclusions about our userbase can be taken.

We got confirmation of our hypothesis that the R.O.M.A. app has a very heterogeneous target, because our survey was answered by very different people. The one thing they all have in common is a connection to the city of Rome: they either live inside it, or they spend enough time in it to still want the app installed.

The differences within our surveyed sample make it difficult to identify one single average user of R.O.M.A. As we will see in the next chapter we have created three different *personas* with three different *scenarios* to represent the variety of our userbase.

Chapter 3

Personas and Scenarios

Let's introduce a selection of possible users of R.O.M.A. In the next sections we will examine more closely their personas and when they would use R.O.M.A. in a real-life scenario.

3.1 Personas

3.1.1 Persona: Caterina

User profile

Age: 23

Gender: Female

Job title: Medical student

Location: Rome, Piazza Bologna

Family: Single

Income: 0€/year



Persona

Caterina is a medical student in Sapienza. She is currently at the fourth year, and wants to specialize in dentistry in the future. She transferred from her hometown in Tuscany and rented a student home in San Lorenzo, near a metro station that takes her to the University campus. She doesn't have a job yet so she tries to keep her expenses low, taking public transportation or walking when she can. She usually uses a mobile 4G internet connection to browse the internet and use social apps on her phone.

Scenario

Caterina takes the metro to her University every morning. One day, she is late for the first lesson of the morning. Unfortunately she can't exit quickly from the metro station because there is a crowd on the stairs on the way out. The reason for this is that the escalators are

broken, once again.

Caterina, disappointed by this setback, decides to take action and searches for a way to report this problem. She then finds out about R.O.M.A.

3.1.2 Persona: Gabriele

User profile

Age: 45

Gender: Male

Job title: Civil engineer

Location: Rome, Parioli

Family: Married with children

Income: 40000€/year



Persona

Gabriele is a civil engineer. He is happily married, has two lovely children, but the true passion of his life are cars. He changes them frequently to experience new feelings on the road. He is always up to date with computer and smartphone technologies.

Scenario

After a hard day of work Gabriele, tired, is ready to go home. Today he didn't park his car in the company parking, but in another farther spot. When he gets to his car, he finds an unpleasant surprise on his windscreen: a fine. Apparently he parked in an unsignaled no parking area. The reason for this mistake was that the road sign forbidding the parking was almost entirely covered by the vegetation.

Irritated, he reports the problem to R.O.M.A.

3.1.3 Persona: Franco

User profile

Age: 71

Gender: Male

Job title: Retired

Location: Ostia

Family: Married

Income: 12000€/year



Persona

Franco is retired. He has lived in Ostia all his life. He loves the city and know all of its places. Even if he is not very interested in technology, he has a smartphone and he is able to use all its basic functionalities (downloading and using simple app). Anyway, he has a niece who helps him when he has problems with it.

Scenario

Every day during the last months, Franco is annoyed by the presence of a deep pothole right in front of his garage. Each time he wants to get in or out of it, he has to maneuver to avoid it. One day, his niece comes to visit him, and hearing his complaints about the pothole, she helps him reporting this problem with R.O.M.A.

Chapter 4

Competitor analysis

After a research on the Play Store, we found some possible competitors of R.O.M.A.:

- **WeDU! Decoro Urbano**
- **Bucapp**
- **ROMA Al tuo Fianco**
- **+Firenze**

Here we briefly analyze each of them, listing their features and their limits.

1. *WeDU! Decoro Urbano* is an app released back in 2016 with a lot of features present in R.O.M.A. It offers the possibility to login with Facebook, GPS localization is integrated, and overall seems very good.
However, it has few categories (only 4), lacks a feature of urgent reports and most importantly it isn't accessible since 2019 since the login is broken and the app isn't supported anymore by the developers.
2. *Bucapp*, as the name implies, is limited to reports of street potholes only. It isn't limited geographically, meaning that you can report wherever you want. It also lacks GPS localization.
3. *ROMA Al tuo Fianco* is the official app developed by the City of Rome for reporting urban problems. Unfortunately it's very basic, since the reports can't be customized in any way (no title, description, images...) and the categories present are only four. It also doesn't offer a profile page, and doesn't even require an account to be used.
4. *+Firenze* isn't really a direct competitor of R.O.M.A., since it's a local app for the city of Florence. However its functionalities are very similar to ours. It offers a very large number of different categories, maybe even too many. The main difference it's the lack of interaction with other users: you can't visit other profiles or see reports made by other users.

4.1 R.O.M.A. functionalities

R.O.M.A. is an app that requires users to login to use the app, so of course it offers the functionalities of registration and login, with email and password or with Google and Facebook.

After the login, the user can see all of the reports made in the area of Rome by the other users. They are represented as colored markers on a map. Different problem categories (for example potholes, garbage, road problems, infrastructures failures...) have different colors to differentiate them. The color scheme is used consistently in all the app pages.

In the same page, the app offers the possibility to quickly access your actual position with the press of the GPS button, if the authorization was given at the launch of the app.

Furthermore, the map in the homepage is also the place where new reports are created. Simply long-press anywhere on the map, as long as it's contained in the City of Rome territory, and you will be zoomed in on the map location of the problem to be sure that you have chosen the exact spot.

The reports can be explored visually on the map in the homepage or browsed in the Explore page, where they are shown with full details.

If the user is interested in seeing all of the reports made by a single user, he has the possibility to view their profile, where he can find information about the user and a list of all the reports he made. Of course, a user can also access and see his own profile.

Chapter 5

HTA and STN

5.1 Definitions

HTA

Hierarchical task analysis is a task description method and a variant of the task analysis. This is used to produce an exhaustive description of tasks in a hierarchical structure of goals, sub-goals, operations and plans. In this kind of analysis, tasks are broken down into pieces progressively smaller.

STN

A state transition network is very intuitive diagram. It assumes that a dialog essentially refers to a progression from one state (or node) of the system to the next in the system state space.

In the next sections we will explain more in detail three of the different functionalities offered by the app, representing them through HTA and STN visualizations.

5.2 Start using R.O.M.A.

When a user first launches the app, the first page he sees is the login page. From here, he must complete the login procedure to enter the app, so he has two options. If he isn't a registered user already, he has to go to the registration page by clicking the "Register" button, and fill in the form with the required information. To select an image he can scroll through the images in his gallery in the window opened by the button "Load image". From this time on, he only has to perform the login procedure with his chosen credentials (username and password).

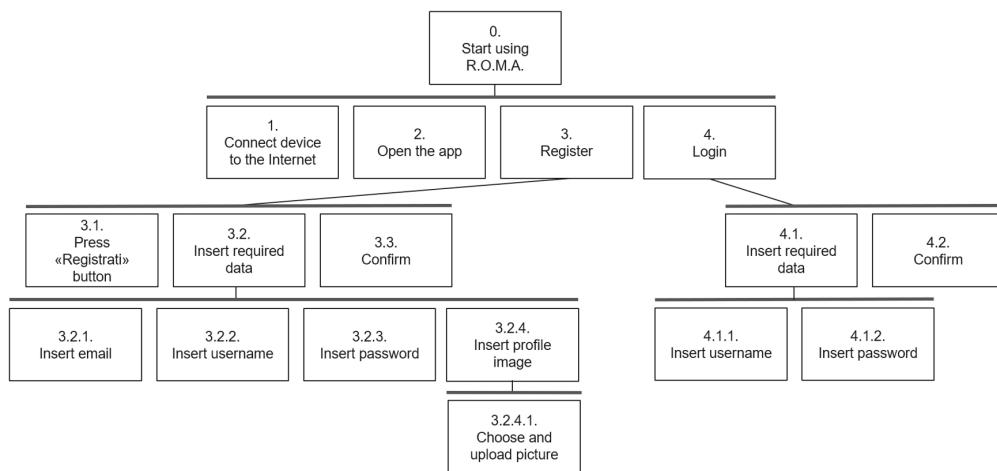


Figure 5.1: HTA: Start using R.O.M.A.

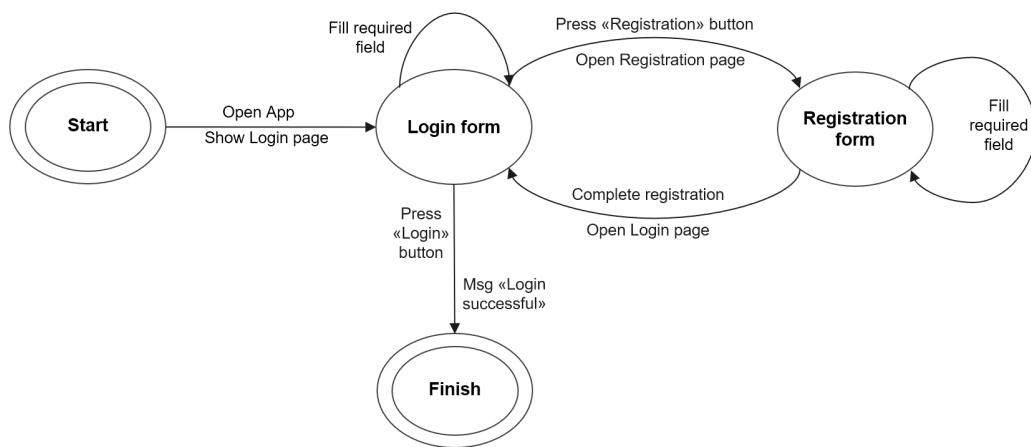


Figure 5.2: STN: Start using R.O.M.A.

5.3 Add a new report

To add a new report the user must be in the Home page, where the map is. The procedure for adding a new marker is the same as Google Maps, that is long-pressing on the desired location. To find this location it can be useful to use the GPS localization button which focuses the view on the current user position. The long-press opens up a popup containing a form that has to be filled with all the required information that is mandatory. After this, he can press the "Confirm" button to see his new report on the map.

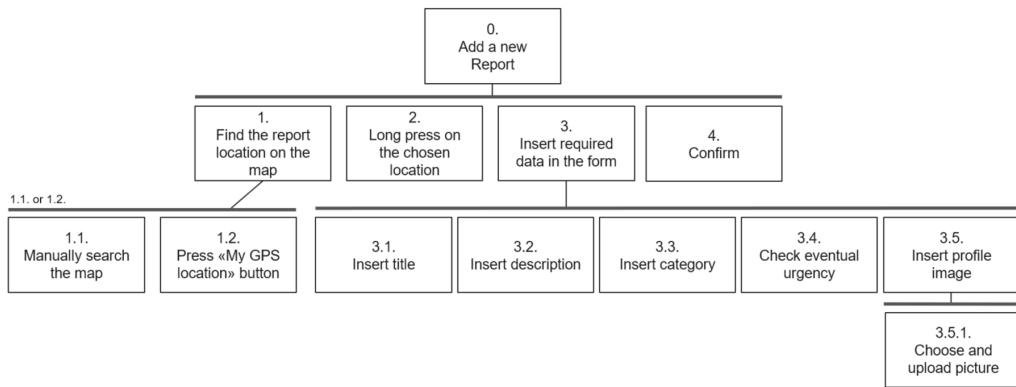


Figure 5.3: HTA: Add a new report

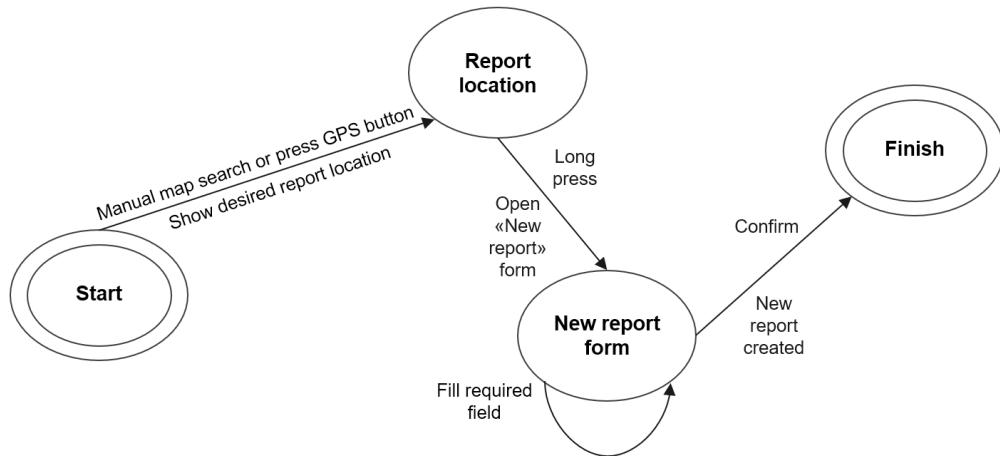


Figure 5.4: STN: Add a new report

5.4 Check the progress of a report

Users have many options to see the details of a specific report. These details are shown in cards and popups in all of the app's pages.

For example when a user is scrolling the map, he can click on a marker to see a preview of the details (category, description, image) of the report related to that marker in a popup above the map. He then can click on the popup to read even more information, such as the time of publishing, the status of the report, and the user who made it. All of this is shown in the profile page of the person who posted that report.

Alternatively the user can decide to browse the Explore page to see all the reports made in the app by the other users, in chronological order, with all the details.

Finally, to see the reports he made himself, the user can go to his Profile page and check the status of all the reports made by him.

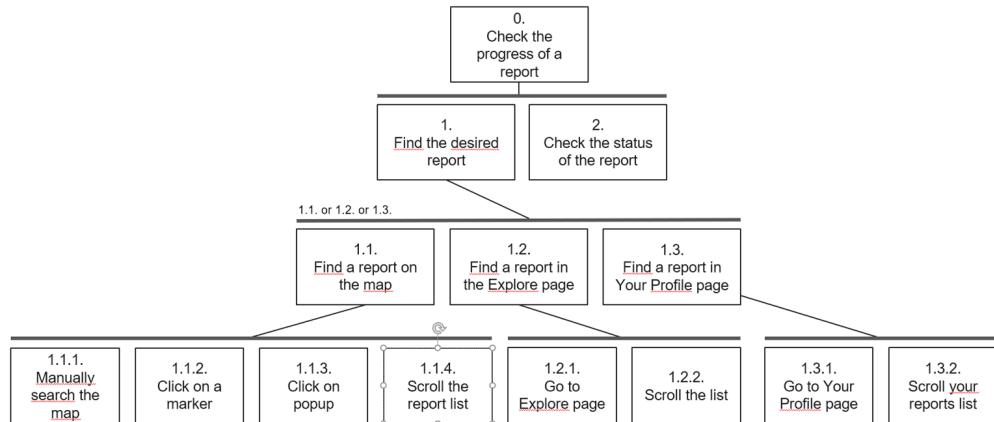


Figure 5.5: HTA: Check the progress of a report

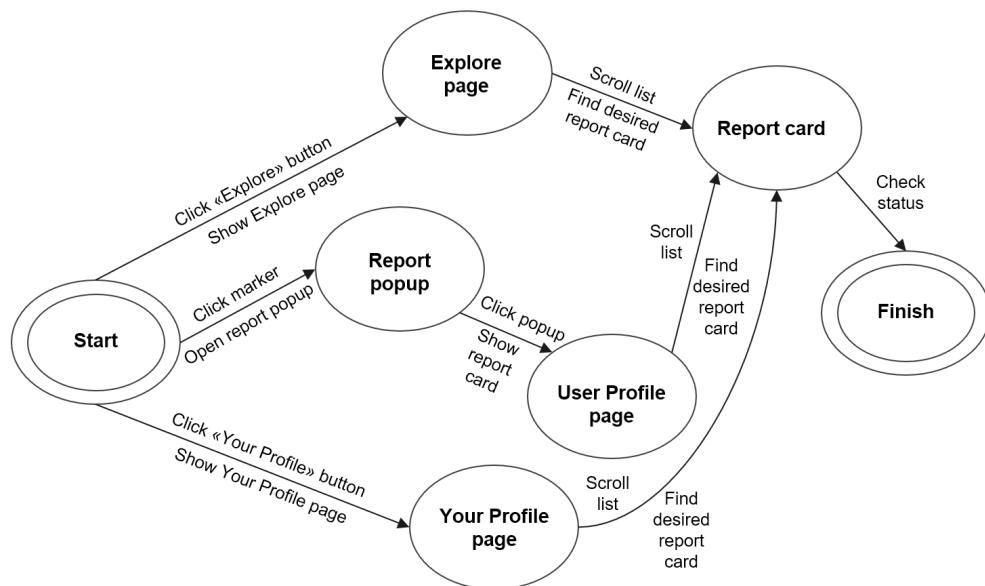


Figure 5.6: STN: Check the progress of a report

Chapter 6

First prototype

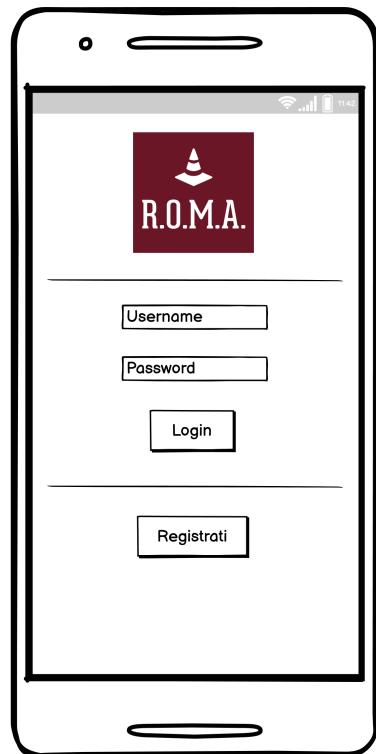
Our first prototype was realized through mock-up made in Balsamiq, a free program that allows to quickly build low-fidelity user interfaces.

It had an essential graphic user interface, to avoid a too heavy design that, even if fancy, wouldn't be easy to use.

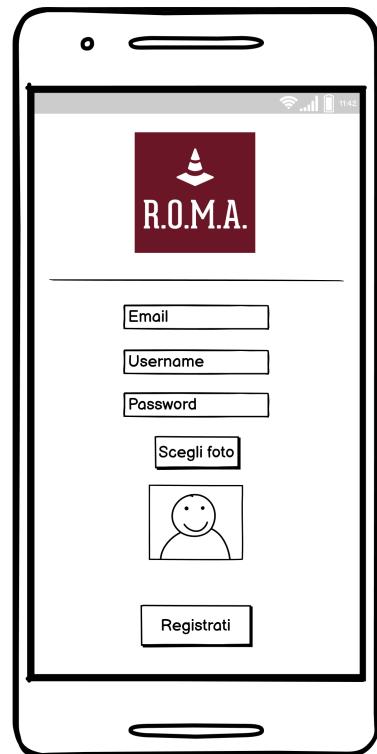
We chose to use familiar icons, in such a way that the user can take advantage of his knowledge of other applications that use similar icons and that the user is facilitated in his utilization of the app by recognizing familiar icons instead of recall the meaning of icons that he never saw before.

We also used standard Google material and Android components, such as the bottom navigation bar and the expandable cards to contain the reports information.

In the following pictures are presented some of the pages of this first mock-up prototype:



(a) Login



(b) Registration



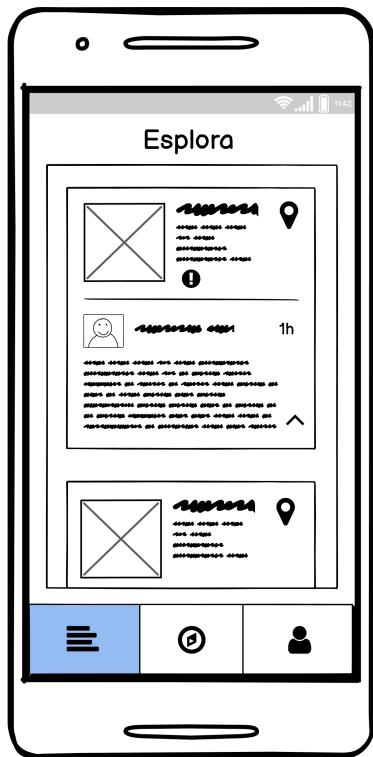
(a) Homepage



(b) Homepage with report popup



(a) Homepage with new report form



(b) Explore page



(a) Profile page

(b) My Profile page

6.1 Expert-based heuristic evaluation

A heuristic evaluation is a usability inspection methodology for computer software that helps to identify usability problems in the user interface design. It specifically involves evaluators examining the interface and judging its compliance with recognized usability principles (the "heuristics"). These evaluation methods are now widely taught and practiced in the new media sector, where UIs are often designed in a short space of time on a budget that may restrict the amount of money available to provide for other types of interface testing.

The main goal of heuristic evaluations is to identify any problem associated with the design of user interfaces. Usability consultant Jakob Nielsen developed this method on the basis of several years of experience in teaching and consulting about usability engineering.

The heuristics as published in Nielsen's book Usability Engineering are as follows:

Heuristics

1. Visibility of system status:

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

2. Match between system and the real world:

The system should speak the user's language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

3. User control and freedom:

Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

4. **Consistency and standards:** Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
5. **Error prevention:** Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.
6. **Recognition rather than recall:** Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.
7. **Flexibility and efficiency of use:** Accelerators — unseen by the novice user — may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.
8. **Aesthetic and minimalist design:** Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.
9. **Help users recognize, diagnose, and recover from errors:** Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
10. **Help and documentation:** Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

6.1.1 Heuristic evaluation

So, the evaluation of the first prototype was a heuristic evaluation made by experts. For this step the evaluation was made by our professor Valeria Mirabella. Here below there is her heuristic evaluation of our project:

Frame	Heuristic violated	Severity	Description / Comment
<i>Login page</i>	Help users recognize, diagnose, and recover from errors	4	Include a Forgot password? link
<i>Registrati</i>	Error prevention	4	In case of accidental mistyping consider asking the new password twice entering it in the "Confirm password" box and/or provide function to show password in clear text
<i>Home page</i>	Recognition rather than recall	2	Consider the possibility to provide a brief description of the app at least on the first login
<i>All (esplora)</i>	User Control and Freedom	3	Allow users to go back a step

Table 6.1: Expert report: Heuristic Evaluation

Severity scale

0 = I don't agree that this is a usability problem at all

1 = Cosmetic problem only

2 = Minor usability problem

3 = Major usability problem

4 = Usability catastrophe

Chapter 7

Second prototype

7.1 Prototype description

Our second prototype is realized through Android Studio. It is a demo of the app, meaning that it has no database of users, but it still shows all the pages preset in the mock-up and the interactions inside the app are all implemented.

Its main differences with respect to the previous prototype were essentially the ones linked to the corrections described at the end of the past chapter. In the next section we will discuss the solutions adopted to correct the heuristics problems, comparing the aspect and the differences between the mock-up and the Android app.

7.2 Correction of detected usability issues

After receiving the expert-based evaluation with the heuristics violated in Prototype 1 we made some changes to solve these issues:

1. Frame: "Login page", Heuristic violated: "Help users recognize, diagnose, and recover from errors", Severity: 4

The problem with this page is that if a user forgets his password, he will not be able to access the app again, because the app doesn't provide a way to recover it.

Our solution adopted the expert suggestion of adding a "Forgotten password?" link, which then brings up a popup where the user can write his email and initiate the procedure for obtaining a new password.

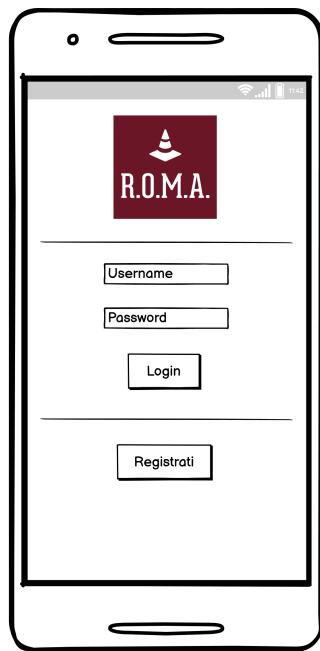


Figure 7.1: Login page in the first prototype with Balsamiq, with violated heuristic

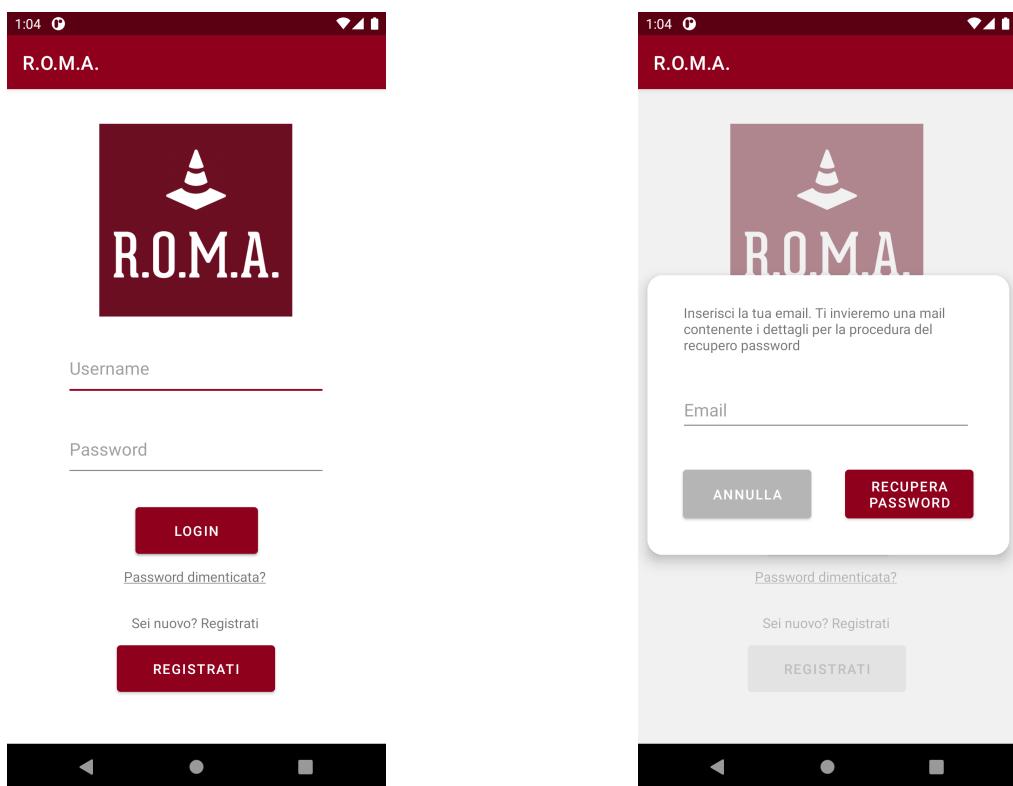


Figure 7.2: Login page in Android corrected with the "Password dimenticata" link and relative popup

2. Frame: "Registrati", Heuristic violated: "Error prevention", Severity: 4

The problem with this page is that a user could accidentally mistype the password without noticing the error until it's too late.

To prevent this issue, we adopted the first proposed solution, which is to ask the user to input his password twice. In this way he can't see it, since it's shown as a string of asterisks, but the two passwords have to match, so that the user has a way of knowing that he entered the password he intended to.

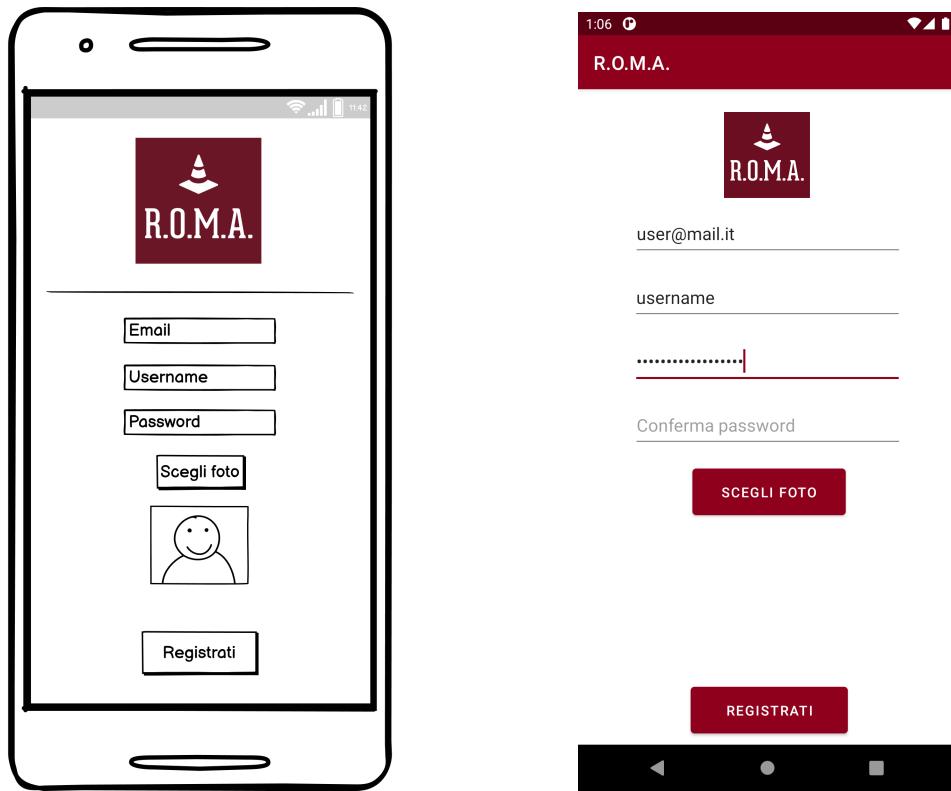


Figure 7.3: Registration page: before and after

**3. Frame: "Home page", Heuristic violated: "Recognition rather than recall",
Severity: 2**

Following the expert advice we decided to add an Info page into the app. This page is shown after the login, the first time the user opens the app. It can also be reached from the profile page, clicking a "help" button. The page offers an introduction about the app's purpose and features.



Figure 7.4: Info page

4. Frame: "All (esplora)", Heuristic violated: "User Control and Freedom", Severity: 3

In this case we believe that the user in every moment already has the possibility to arbitrarily switch between every page of the app with a single click.

For example, if he is in the Profile or Explore page and he wants to go back to the map, the user can simply use the navigation bar on the bottom of the page to open the map again on the position he was previously at. If he is visiting the profile page of another user, to go back he can use the "back" arrow button on the top left of the page. To go back to the login page, the user has to use the "logout" button on his profile page.

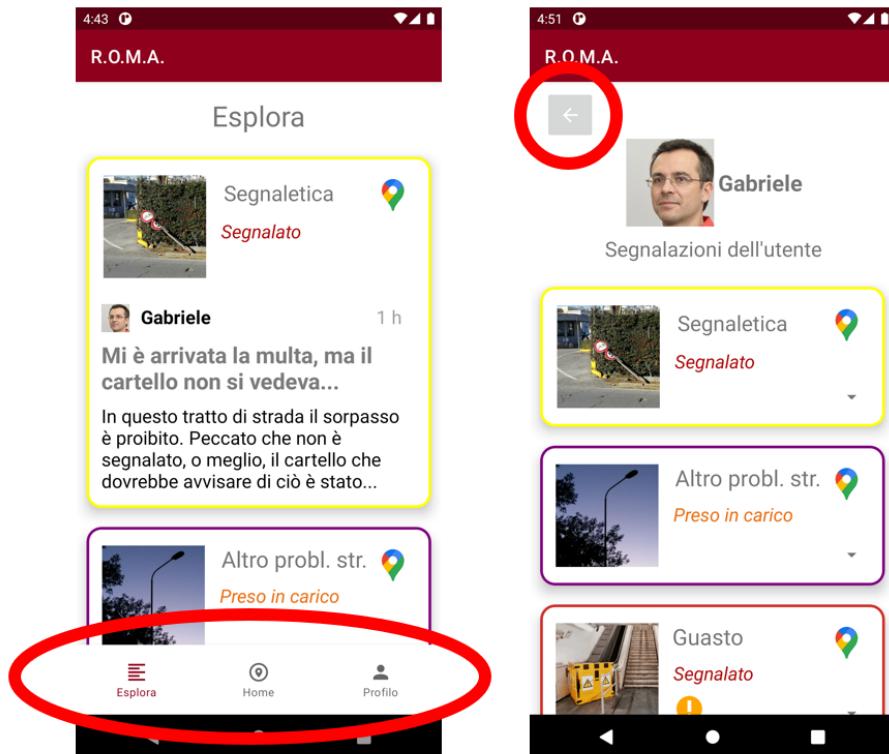


Figure 7.5: UI elements that allow to go back a step

Chapter 8

Third prototype

8.1 User-based evaluation technique – think aloud

The think aloud is a kind of evaluation based on some simple rules. We chose a group of 12 people of different ages. In particular, 6 of these people were our parents, while the other 6 were people of our age, like siblings and friends. This was chosen to represent the heterogeneous demographic of the userbase: we want to know if the design choices we made were suitable for everyone.

We performed the experiment using these criteria:

- We first explained to the users the purpose of the app.
- We explain that we are testing our application, and not testing them.
- Each member has to accomplish the same two tasks individually: adding a new report on their current position, and checking the profile of the user who made the report in a certain area of Rome.
- The experiment took place in a room without distractions, and each person had a smartphone with R.O.M.A. installed.
- While executing the task, each user has to say aloud what he's doing, what he thinks it's happening, any doubt etc.
- During the experiment, we took note of their behaviours and their comments.

The survey found out that every user was eventually able to find their way through the app, and in the end complete the tasks they were assigned. Generally speaking, none of them found any major problem to accomplish the task. However, we noticed some slowdowns in certain activities so we decided to update parts of the app to improve the overall usability of the app. We also added new important features that we felt were missing from the previous version.

8.2 Correction of detected defects

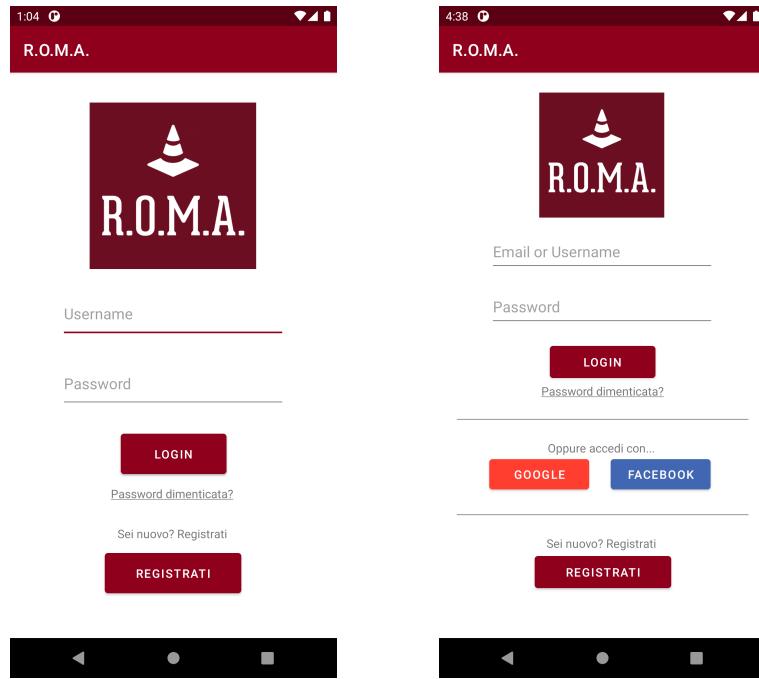
8.2.1 Registration and login alternatives

During the evaluations we noticed that many users were particularly slow in the first mandatory step, which is to enter the app with a registration and then a login. One of them even complained, saying out loud that he wished for a login with third-parties.

It's a fair complaint: lately it's more and more common for sites and applications to offer the possibility to register and login using Google or social media accounts. Logging in with email and password feels like a slower way of doing the same thing. In addition, the user has to remember yet another password, and it can be annoying.

So, we decided to implement the possibility of logging in using the most common external social accounts: Google and Facebook. Pressing one of these two button immediately brings the user inside the app, ready to use it.

Furthermore, now if a user registers with email and username, now he can login using either one of them, together with the password.

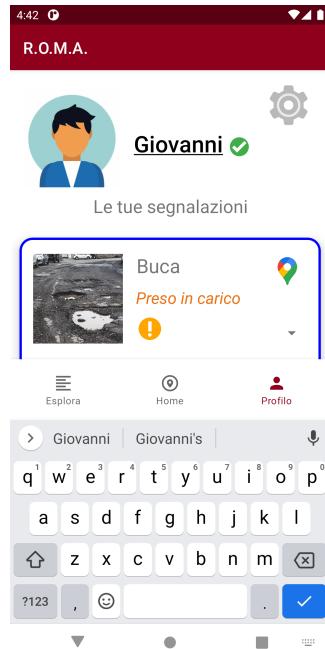


Login page: before and after

8.2.2 Change username

A feature that is missing from the second prototype of the app is the possibility of changing your username: once it is chosen, it cannot be changed anymore. This issue emerged when one of the surveyed accidentally mistyped his username in the registration page.

The lack of this feature is corrected in the third prototype by adding a button that offers the possibility of changing your username in your profile page any time you want.



8.2.3 Fullscreen images

During the interview one of the feature that was requested by a user was the possibility to see fullscreen images of the reports. We added this feature, so now clicking on any image in the app opens a window with that image in fullscreen.

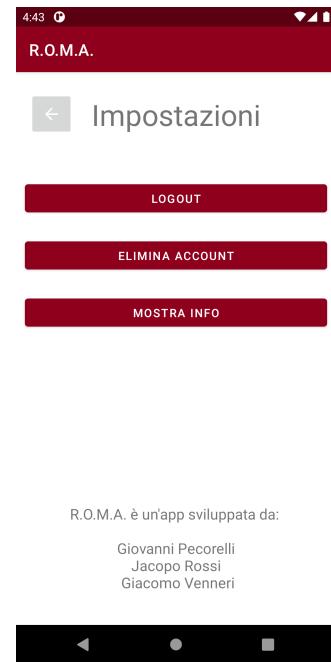


8.2.4 Logout, delete account and settings page

During the interview we found out that we needed to implement the features of logout, to go back to the login page and enter with different accounts, and also the feature of deleting an account. These are two important features that must be provided by any app.

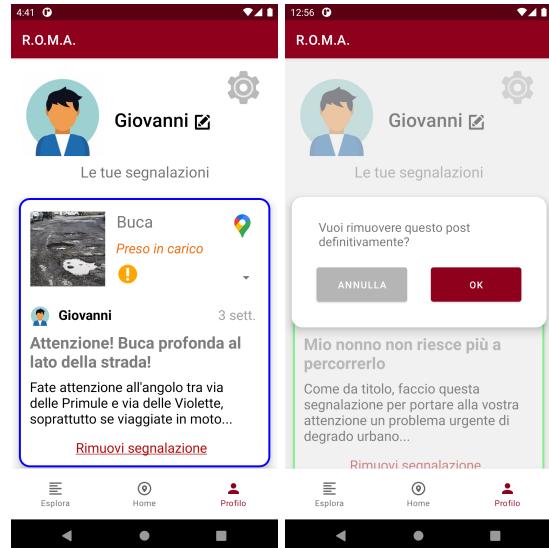
We decided to put these two button, together with the "Show info" button in a separate "Settings" page, reachable from the Profile page.

Both of the buttons open a popup that asks the user for confirmation about what he is about to do, to allow him to cancel his choice or go forward with it.



8.2.5 Remove report

The last feature we decided that had to be implemented in the final version of the app is the possibility to remove a report you made, making it disappear the relative card from your profile page and from the other users' explore page, and the relative marker from the map.



8.3 User-based evaluation technique – controlled experiment

A controlled experiment is when an observer tests a hypothesis by looking for changes brought on by alterations to a variable. In a controlled experiment, an independent variable is the only factor that is allowed to be manipulated, with the dependent variable as the factor that the independent variable will affect. The big advantage of a controlled experiment is you can eliminate much of the uncertainty about your results.

8.3.1 The problem

We decided to make two different interfaces to accomplish the same task: adding a new report. The image on the left is a screenshot from the first interface that we implemented and to complete the task you simply have to long press on the desired location to open the "new report" form.

Instead, in the interface on the right there are two new elements: a search bar and a button "+". To add a new report, the user has to search for an address and then, once he finds it, he has to press the button to open the "new report" popup.

We decided to make two different interfaces because we didn't know which one was going to be easier to use, and more intuitive for the users. We thought that the first interface might be better, but to be sure we decided to do a controlled experiment, to see if our assumption was right.

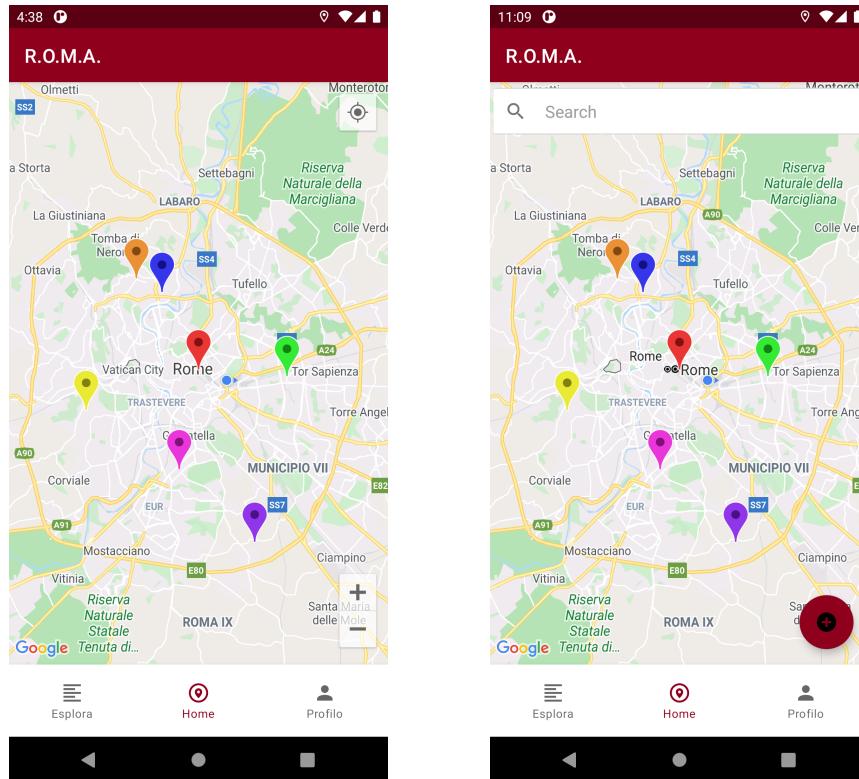


Figure 8.2: Adding a new report: Interfaces 1 and 2

8.3.2 ANOVA

The controlled experiment has been performed with ANOVA One-Way analysis. We asked to all of our 12 different testers to perform the same task. Since usually to run a Between Groups controlled experiment it is necessary to have available a great number of users, we decided to run a Within Groups controlled experiment. This means that each user tests both of the interfaces. To mitigate the consequences of the learnability effect issue in our experiment we divided our testers in two groups and we asked to the members of the groups to test both the interfaces in different order.

We measured with a chronometer how much time each user takes to perform the same task requested with the two different interfaces. All the values are collected in order to compute the analysis. The results are shown in the label below.

- **Problem:** “Having implemented two distinct interface styles which differ in the way users perform the same task, which of them is easier to use?”
- **Who?** 12 people (in a range of age between 18-60 years old according to user profiles)
- **Variables:**
 - *independent*: the two interfaces
 - *dependent*: the time in seconds to execute the task
- **Hypothesis:**
 - *null*: there are no differences between the two interfaces
 - *our*: users will complete the task in less time using Interface 1 rather than Interface 2
- **Experiment:**
 - *task*: "Add a new report"
 - *assumptions*: User is already logged in, on the homepage, and has to create a report given its attributes (title, description, image...)

Interface 1	Interface 2
71	91
65	117
49	81
98	111
101	130
55	72
39	65
98	107
59	75
52	85
62	88
46	78

Figure 8.3: User task completion time (in seconds)

ANOVA

RIEPILOGO						
Gruppi	Conteggio	Somma	Media	Varianza		
Interface 1	12	794,5	66,20833333	464,1572		
Interface 2	12	1099,5	91,625	403,23295		

ANALISI VARIANZA						
Origine della variazione	SQ	gdl	MQ	F	Valore di significatività	F crit
Tra gruppi	3876,041667	1	3876,041667	8,9372508	0,006756612	4,300949502
In gruppi	9541,291667	22	433,6950758			
Totale	13417,33333	23				

Figure 8.4: ANOVA analysis results

As we can see, $F > F \text{ crit}$ ($8.93 > 4.30$).

Therefore, we can reject the null hypothesis. This means that our collected data have statistical relevance and that the means of the two populations examined are not equal. Since the significance value is of 0.00675, so lower than the significance level threshold (usually chosen at 0.05), we can confirm that this is a significant result and that it was caused by the independent variable (as we wanted).

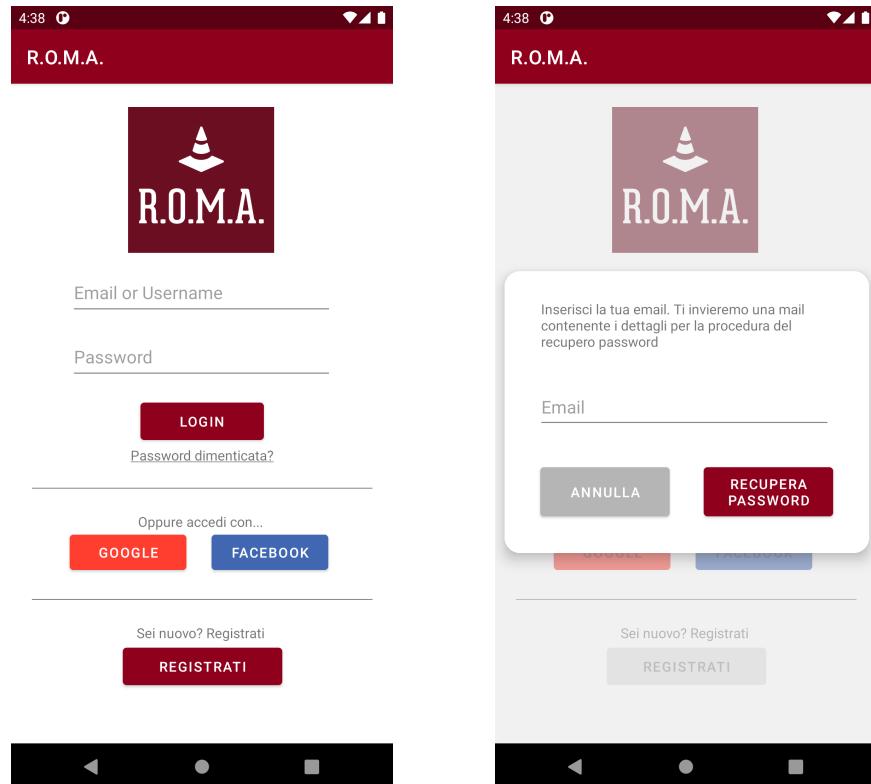
Given that the null hypothesis is rejected and these results are proved to be statistically significant, we can compare the means of the two populations and we can say that since the mean obtained with Interface 1 is less than the mean obtained with Interface 2 ($66.208 < 91.625$) the Interface 1 is better than the Interface 2.

We can conclude that our hypothesis is confirmed. For these reason, we decide to keep the Interface 1 and to discard the alternative Interface 2.

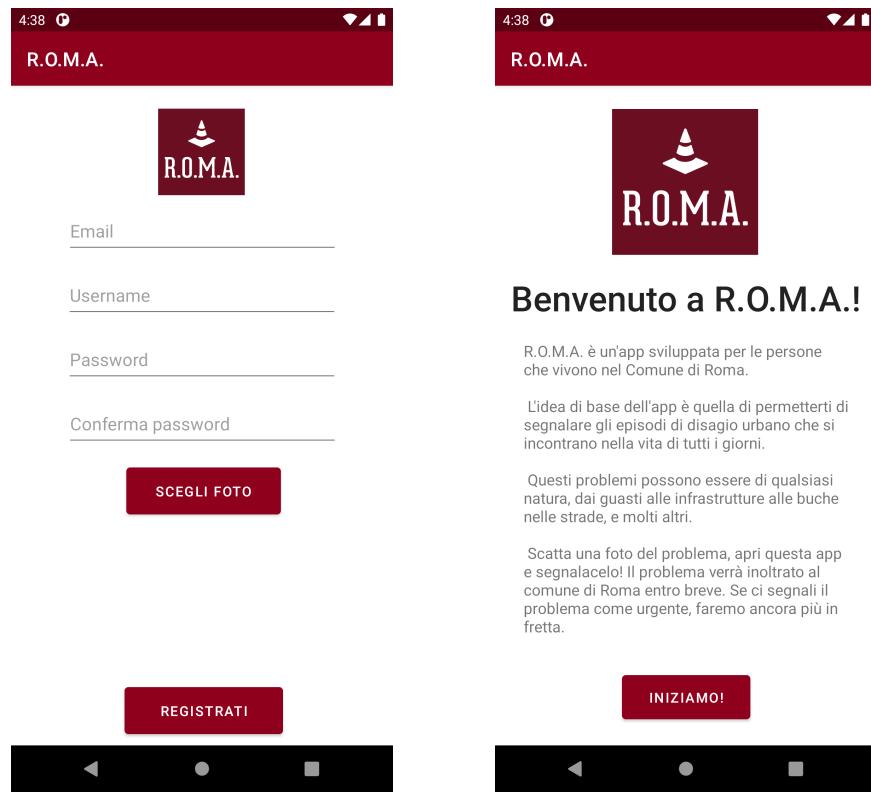
Chapter 9

Final version

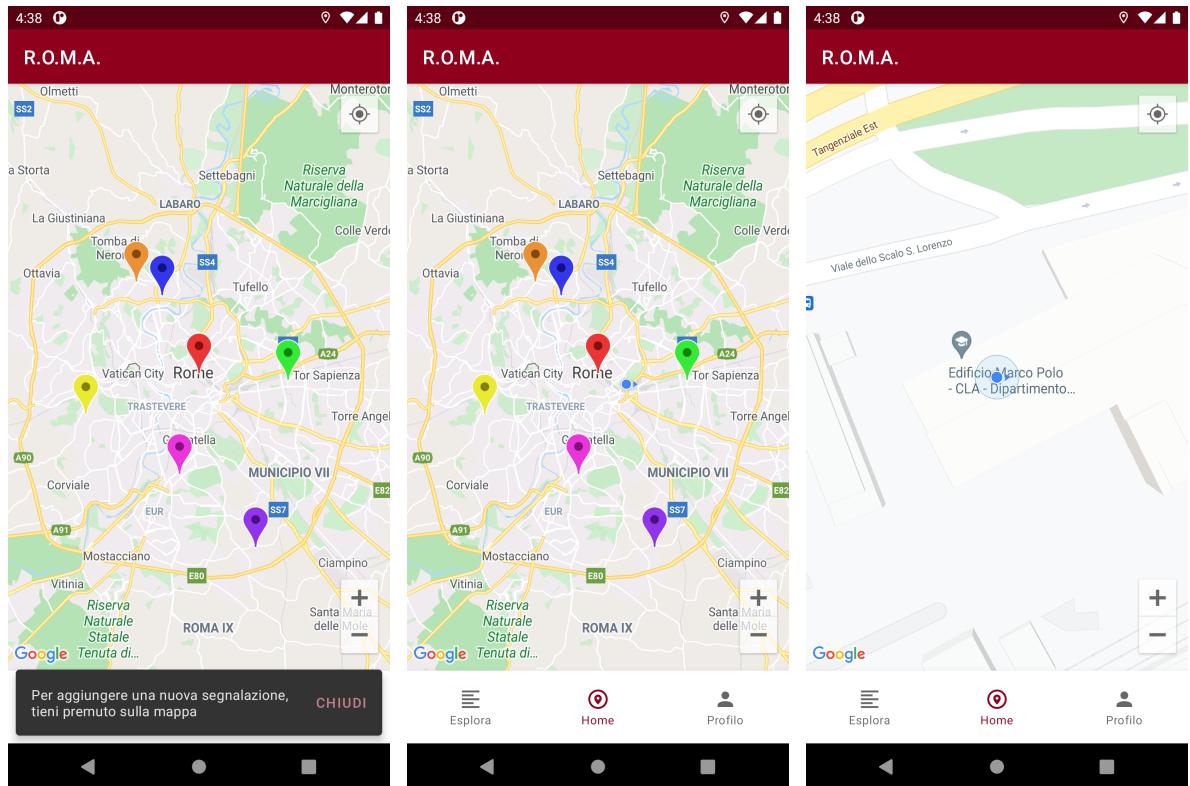
In this chapter we show the final version of the app, with all its different pages and popups.



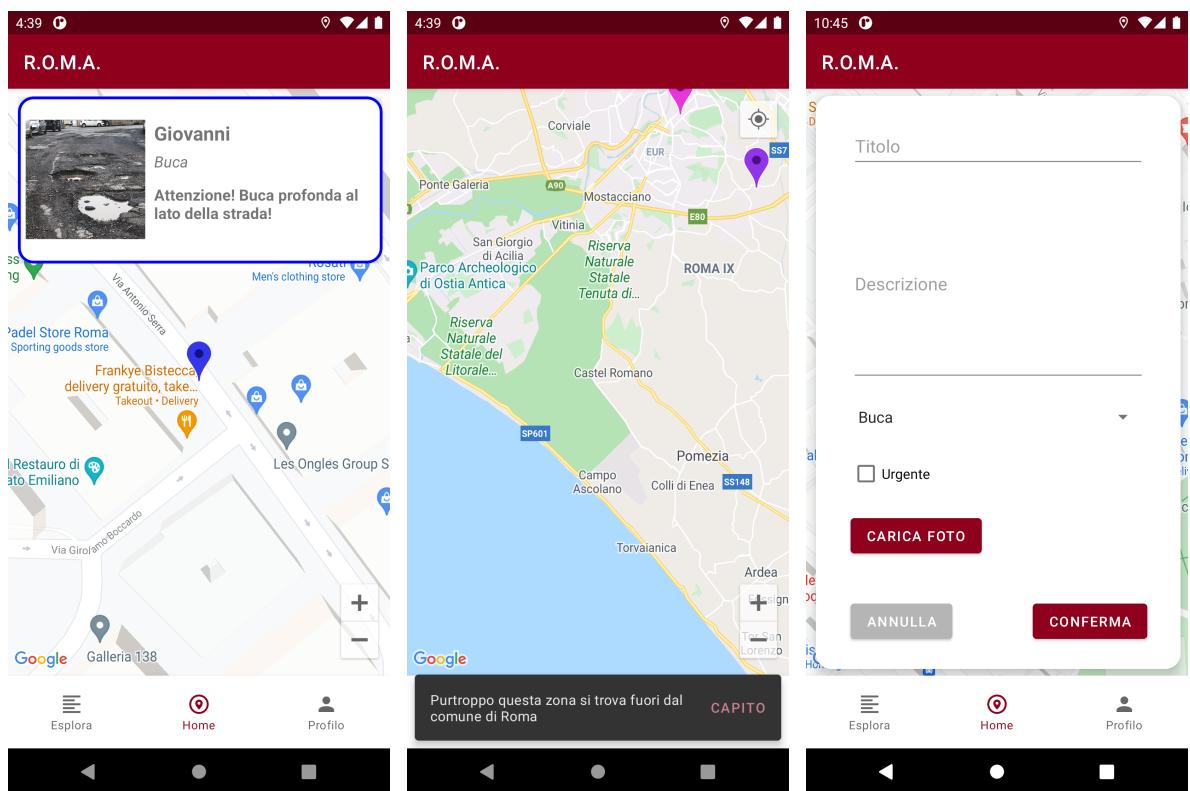
Login page



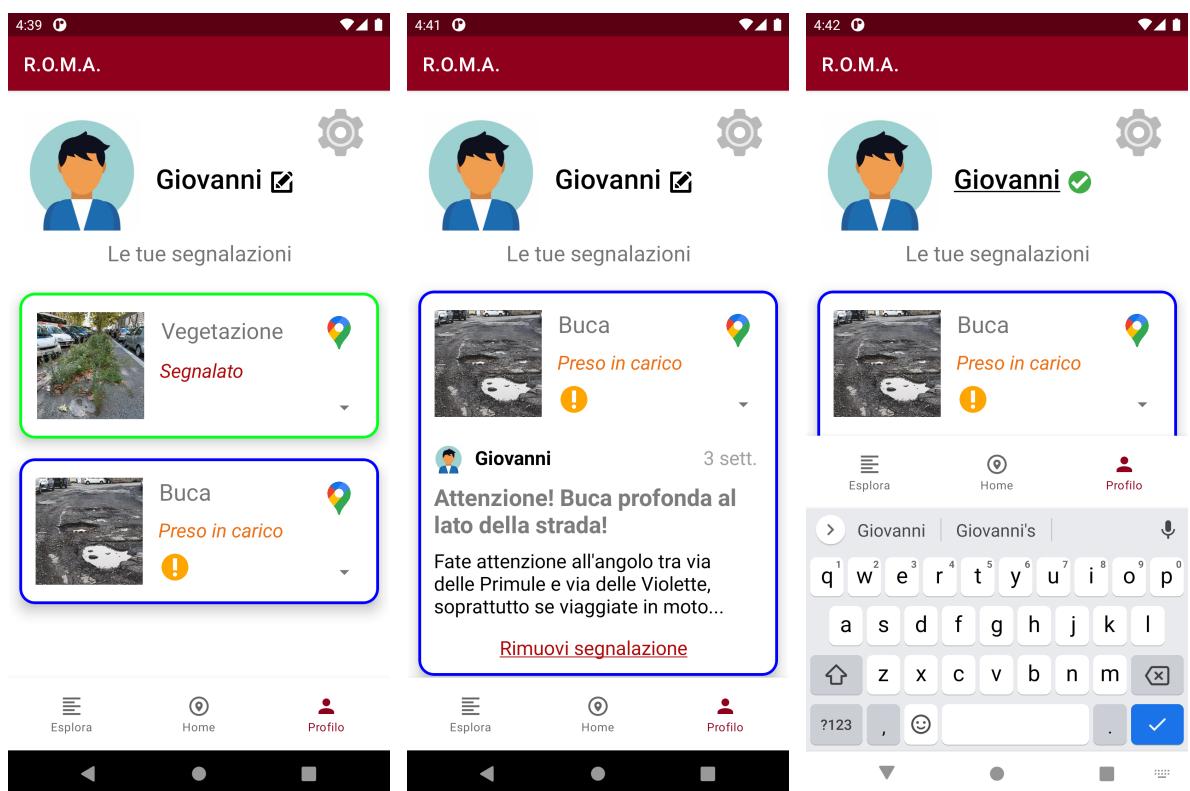
Registration page



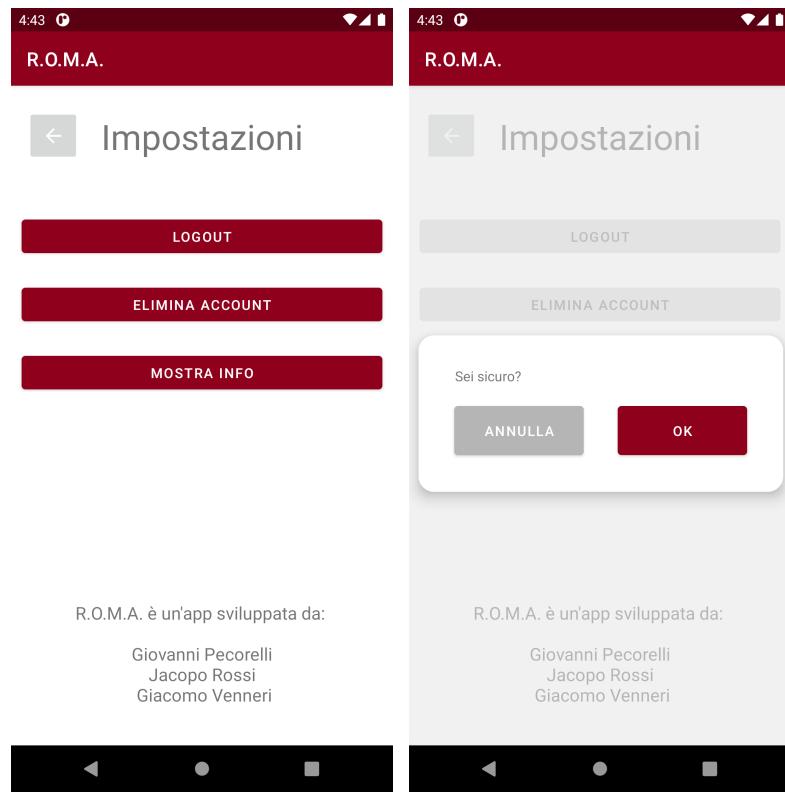
Homepage



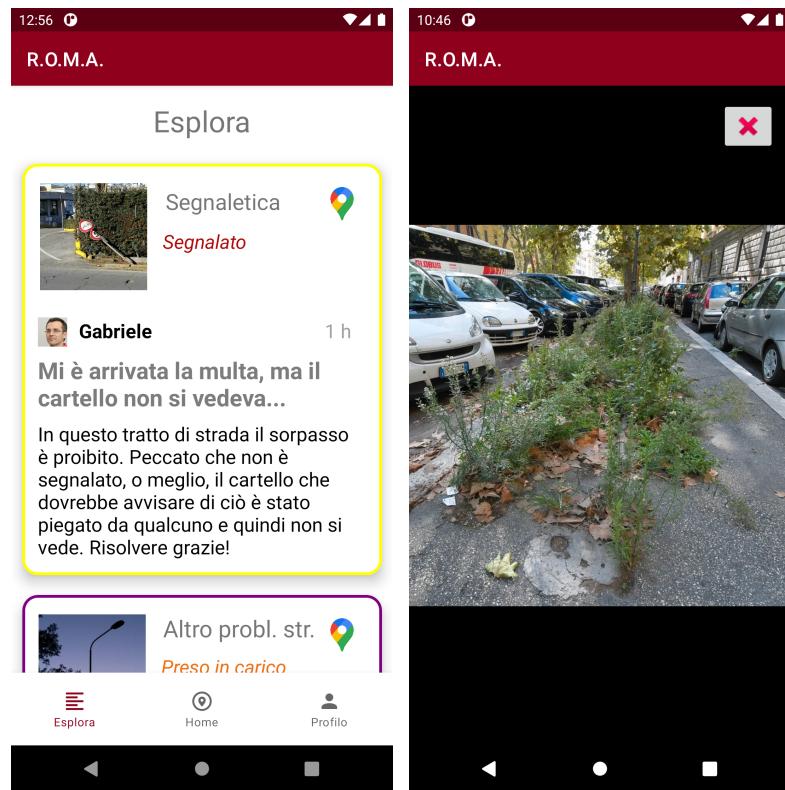
More Homepage



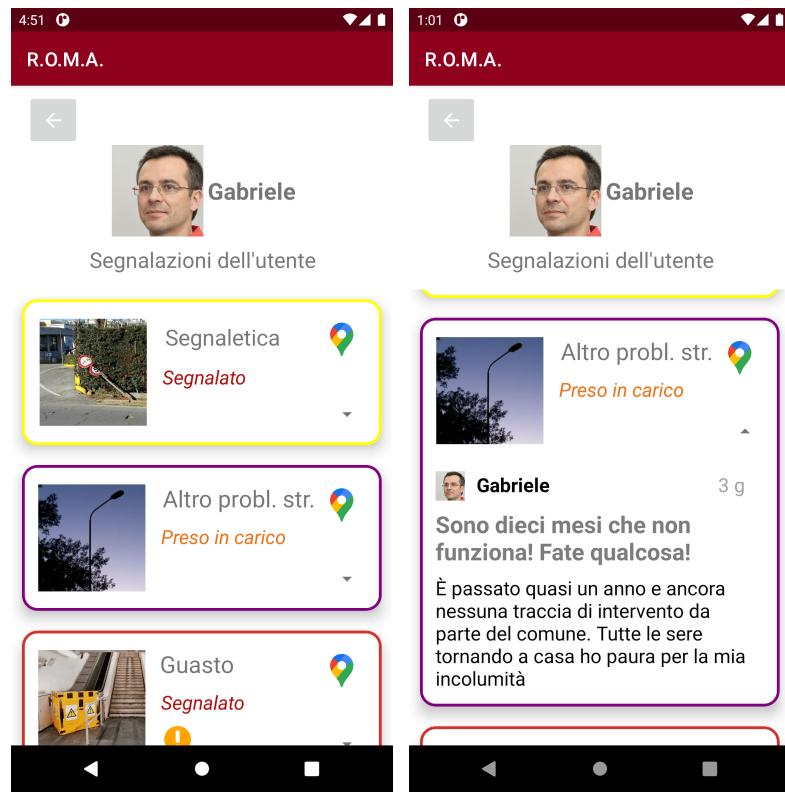
Profile page



Settings page



Explore page and fullscreen image



External profile page

Chapter 10

Conclusions

After working on this project we improved some of our technical skills, in particular those concerning developing mobile applications and interfaces. This was the first time we had a direct contact with the users in order to collect requirements, suggestions and feedback.

10.1 Future works

Some features were not implemented in this project, and have been left to future development. Here's a list of some interesting features that could be added:

- Implement notifications about changing statuses of your report, to let you know that a problem you reported was solved.
- Building social feature, for example likes and comments under reports.
- Bringing R.O.M.A. on iPhones with an iOS version.

Bibliography

- [1] Course material,
- [2] <https://www.excel-easy.com/examples/anova.html> to build the ANOVA OneWay Analysis,
- [3] <https://www.thispersondoesnotexist.com> for the personas images,
- [4] <https://www.balsamiq.com> for the mockups,
- [5] <https://www.developer.android.com> for building the app,
- [6] <https://www.overleaf.com> for writing this report.