

2021



POLITECNICO
MILANO 1863

TEST PLAN: RODI – MILANO ROAD NETWORK MONITOR

SE4G – SOFTWARE ENGINEERING FOR GEOINFORMATICS
AMIEVA, AUSTONI, SONDOQAH

Content

Purpose of the document	2
Application overview.....	2
Testing Scope	2
Out of the scope.....	3
Use cases Testing	3
1. RegisterCitizenMember.....	3
2. RegisterSpecializedMember.....	7
3. LoginMember.....	11
4. ViewMap	15
5. VisualizeData	17
6. MyRequestList.....	18
7. RequestStatusModification.....	21
8. VisualizationUpdatedRequest	22
Conclusion and future work	23

Purpose of the document

This document has as a main objective to explain how will be performed the test plan of the web application “RODI – MILANO ROAD NETWORK MONITOR”. The document is also focused in getting some results of manual controls performed in the different sections of the application in order to understand the current level of development of each part and try to figure out possible improvements for each one of them.

Application overview

The users of the application are divided in three main groups:

- *Citizen users:* Common citizens which use the application in order to inform about distresses in the road network of Milano. They can also visualize a map with their requests of maintenance and see the status of them.
- *Specialized users:* Members of the Ministry of Transport and Infrastructure of Milano. They use the application with the aim to get data about the distresses that are reported in the road network of Milano and they use that information to plan the interventions and update the status of the different requests of maintenance.
- *Super User:* It is the admin of the webpage, the one with higher security permissions. He is also the receiver of the requests for being registered as specialized user. He is the only user that can access to ‘/admin-register’ section, and from that register specialized users as such.

Testing Scope

This testing plan performs tests in each of the core sections of the web app which are the following ones:

1. Registration of citizen users
2. Registration of specialized users
3. Log in
4. Visualization of Map
5. Dashboard visualization
6. Visualization of requests uploaded by citizen users

Out of the scope

Here were not considered in the scope of the current testing plan, tests related with performances.

Use cases Testing

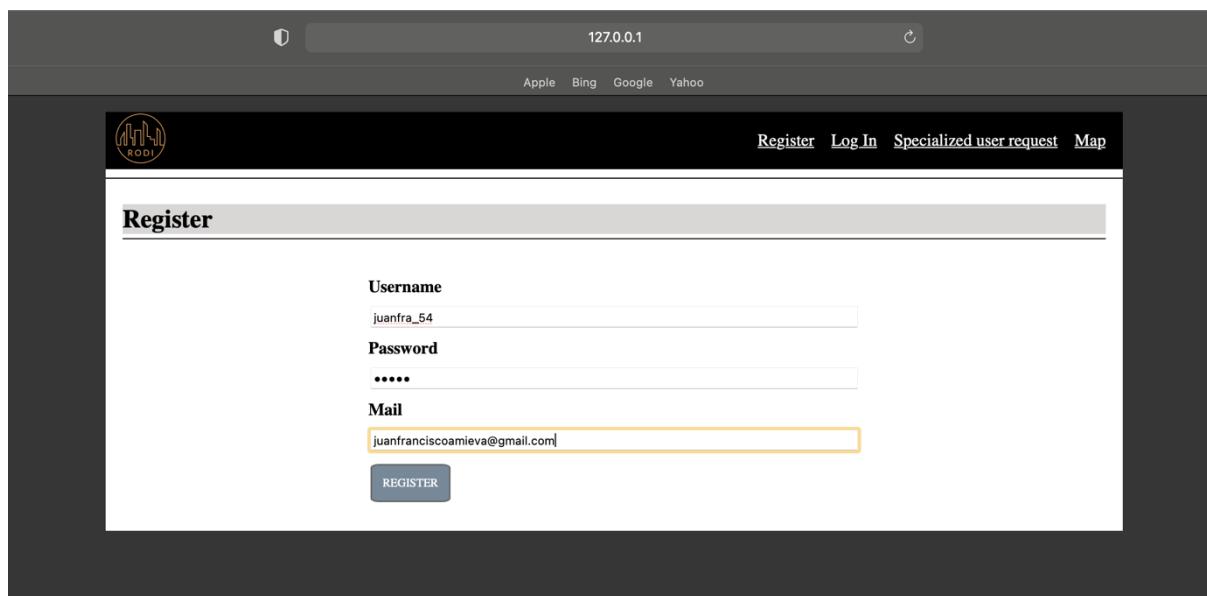
The present section explains how were performed the different manual tests on the different use cases proposed in the RASD document. It is important to highlight that here were used the same titles for each of the use cases because it is necessary to uniformize the notation in order to allow a good identification of them among the different documents.

1. RegisterCitizenMember

In order to control this use case was inspected how are performed the different steps in the registration process of a citizen user in the web app.

Considering this, it was necessary to click in the Register link in the web application, then complete the information required in the presented form and press the submit button.

Figure 1: Successful normal user registration



The screenshot shows a web browser window with the URL '127.0.0.1' in the address bar. The page title is 'Register'. The main content is a registration form with three fields: 'Username' containing 'juanfra_54', 'Password' containing '*****', and 'Mail' containing 'juanfranciscoamieva@gmail.com'. A yellow box highlights the 'Mail' field. Below the form is a 'REGISTER' button.

The registration was performed successfully.

Then it was checked the response of the system when a user forgot some of the inputs. In this case, it appears a popup on the empty field that indicates that this data is required.

Figure 2: Missing field in citizen user registration

The screenshot shows a web page titled "Register". At the top right are links for "Register", "Log In", "Specialized user request", and "Map". Below the title, there are three input fields: "Username", "Password", and "Mail". The "Username" field is highlighted with a yellow border and has a tooltip "Fill out this field" above it. The "Password" field contains four dots ("...."). The "Mail" field contains an email address. A "REGISTER" button is at the bottom.

Next a test on a too short password was performed. If the user chooses a password that has less than 5 characters, then an error message is displayed to the user to inform about password requirements. Like the following picture demonstrates, this test was performed successfully.

Figure 3: Password shorter than 5 characters

The screenshot shows a web page titled "Register". At the top right are links for "Register", "Log In", "Specialized user request", and "Map". Below the title, there are three input fields: "Username", "Password", and "Mail". An error message "Password should be at least 5 characters long" is displayed in a yellow box above the "Password" field. The "Username" and "Mail" fields are empty. A "REGISTER" button is at the bottom.

Finally it was executed a test on the email field. The picture below reports the error displayed correctly if the user inserts an email without using the “.” and “@” symbol. The system recognizes the missing characters and informs the user that the inserted email is invalid.

Figure 4: Missing “@” or “.” in email

The screenshot shows a web browser window with the URL 127.0.0.1. At the top, there are links for Apple, Bing, Google, and Yahoo. The main content area has a header "Register". Below it, a yellow rectangular box contains the text "Insert a valid e-mail address". There are three input fields labeled "Username", "Password", and "Mail". A blue "REGISTER" button is at the bottom.

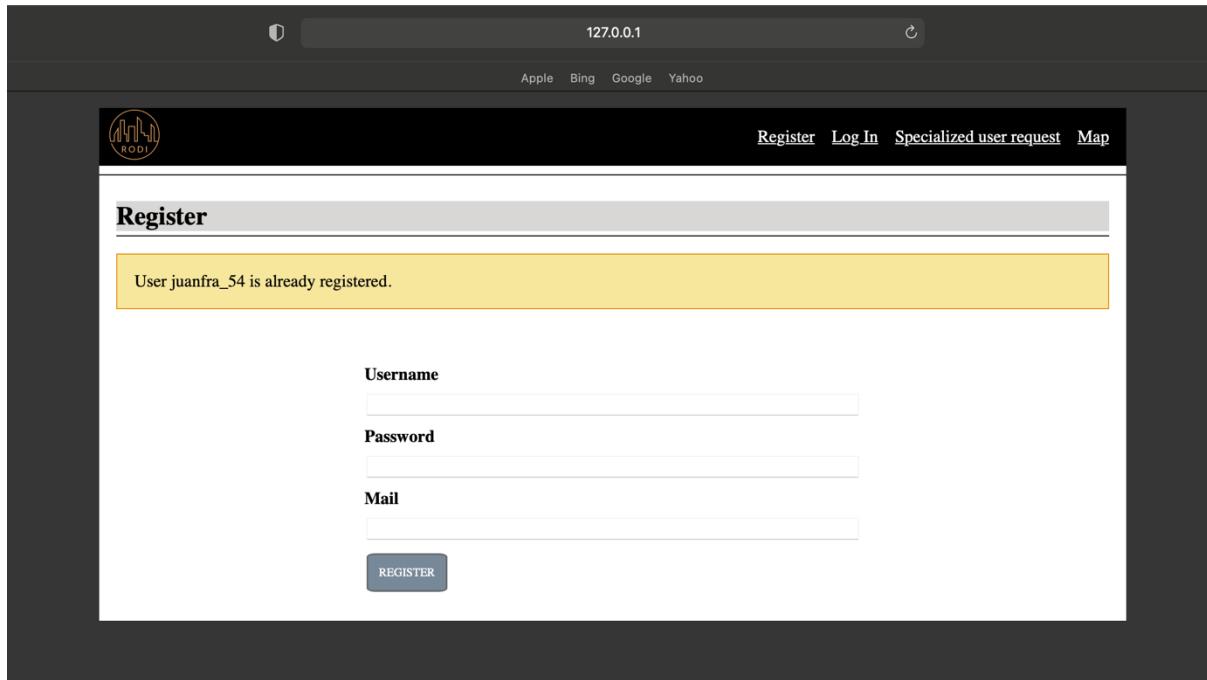
When instead the registration is performed successfully, the system redirects to the login section.

Figure 5: Redirection to login after normal user registration

The screenshot shows a web browser window with the URL 127.0.0.1. At the top, there are links for Apple, Bing, Google, and Yahoo. The main content area has a header "Log In". There are three input fields labeled "Username", "Password", and a blue "LOG IN" button.

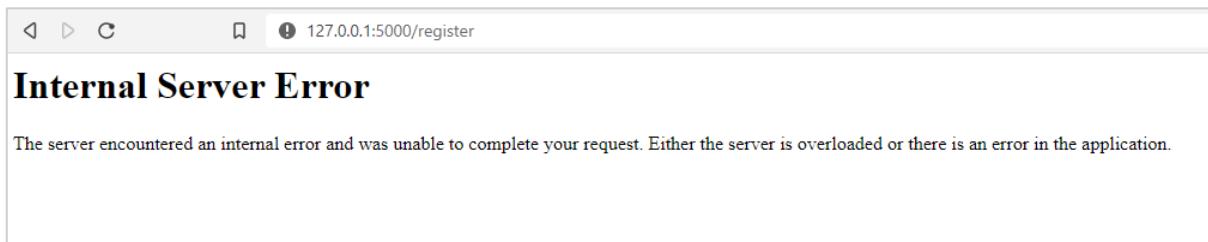
In case a user tries to register with the same username of another one already existing, the app will display the following popup.

Figure 6: Already registered username



In the testing process of this use case was detected a fail in case a user tried to register with the same email than another user, reporting an internal server error.

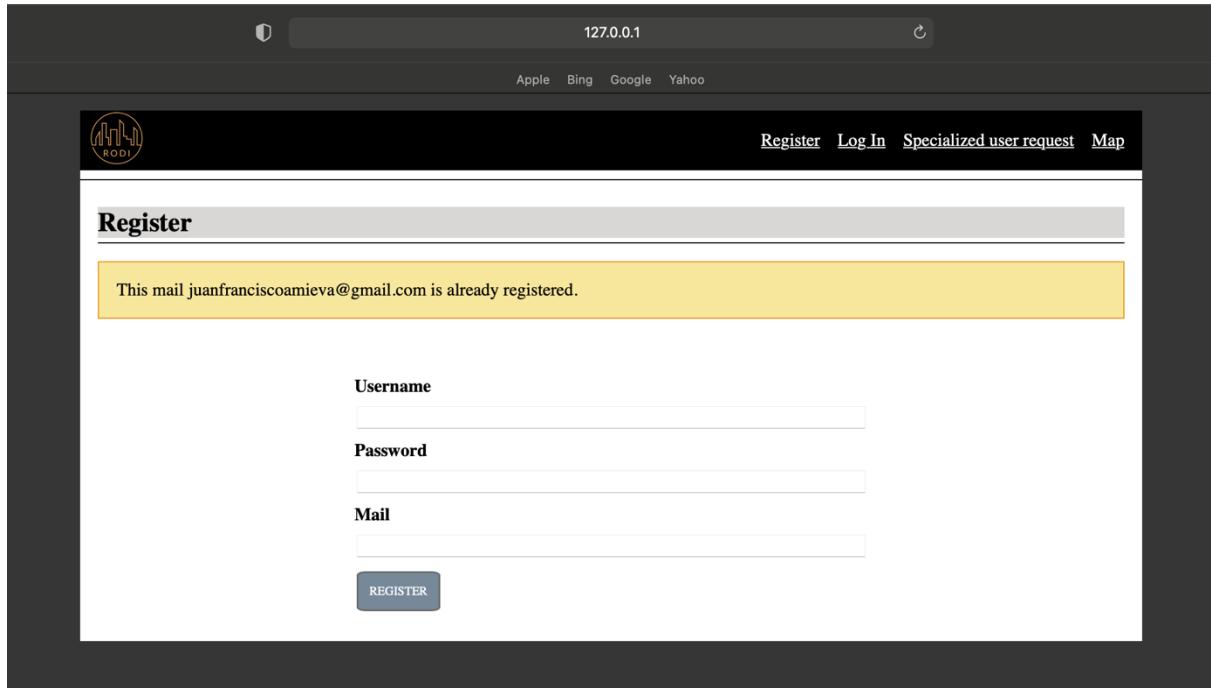
Figure 7: Detected error



Although is right to consider that it is not a possible scenario because the e-mail should be unique for each user, the system should render a page informing the user about the problem in the input received.

For this reason the code was edited in order to solve that. The final result after the control was a popup that indicated the related problem (following picture).

Figure 8: Already registered email



2. RegisterSpecializedMember

For this use case it was tested the registration form for “Specialized users”. In order to develop the test it is required to click on the “Specialized user request” and after that the web page renders the following section that has to be filled with the proper information.

Figure 9: Specialized user correct registration request

The screenshot shows a web browser window with the URL 127.0.0.1. At the top, there are links for Apple, Bing, Google, and Yahoo. The main content area has a header "Want to became a Specialized User?". Below it, there are five input fields: Name (Alessandro), Surname (Austoni), Role (Road Planning Manager), Work department (Maintenance Planning), and Phone number (+39 02 3511 1325). A "SUBMIT" button is at the bottom. The "Role" field is highlighted with a yellow border.

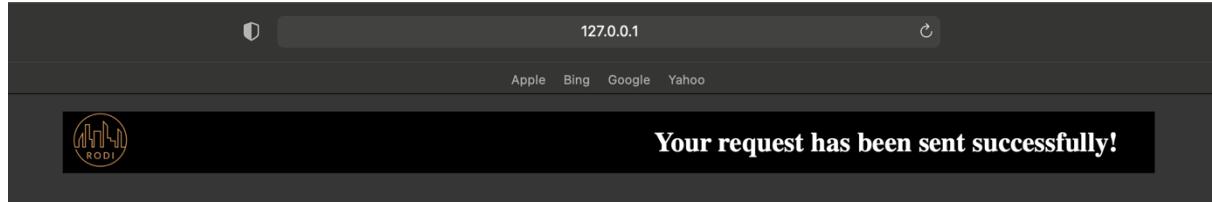
This is the result of a user that forgets to insert one of the following fields in the specialized user request section (here just one field missing is reported for simplicity, but the test was successfully performed on each of them).

Figure 10: Role field missing in request submission

The screenshot shows a web browser window with the URL 127.0.0.1. At the top, there are links for Apple, Bing, Google, and Yahoo. The main content area has a header "Want to became a Specialized User?". Below it, there are five input fields: Name (Alessandro), Surname (Austoni), Role (highlighted with a yellow border and a tooltip "Fill out this field"), Work department (Maintenance Planning), and Phone number (+39 02 3511 1325). A "SUBMIT" button is at the bottom.

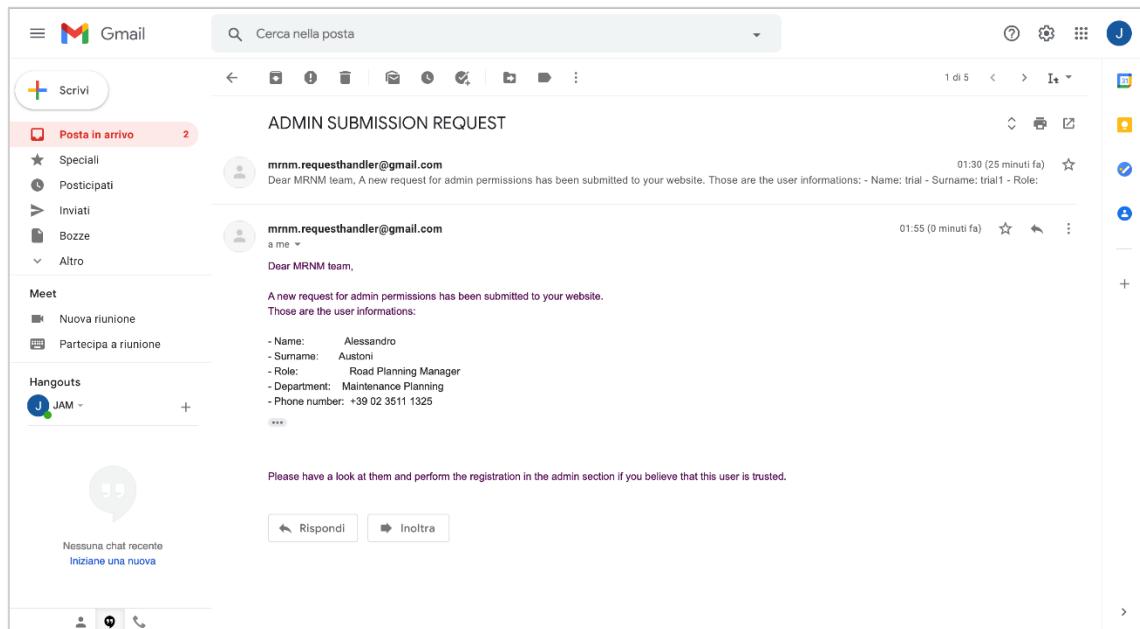
Once a user asks for becoming a specialized user in the correct way, the web page renders a message indicating that the request for the new user was sent successfully.

Figure 11: Successfull request



The application uses the role of a Super User (Admin) to manage the permissions for the specialized user requests. After the user compiles this form, the system sends an email to the super user, containing the information filled in the form by the user.

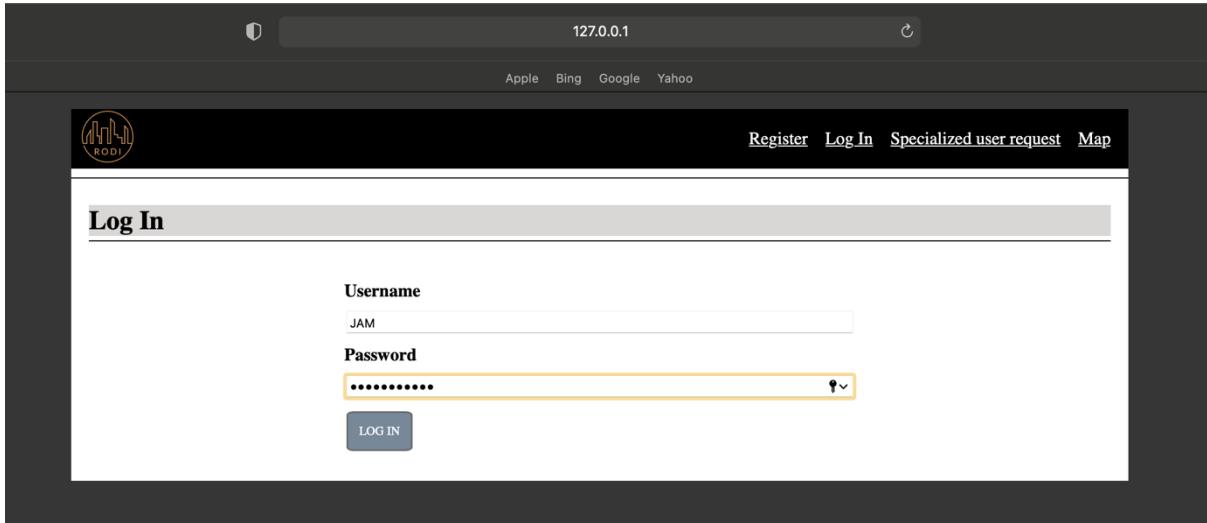
Figure 12: Submission request email



Finally, if the admin considers that the user is trusted, the super user ("JAM" in this case) can register the specialized user in the web app by accessing the root /admin-register.

Here we can see a test where the Super User, first, logs in

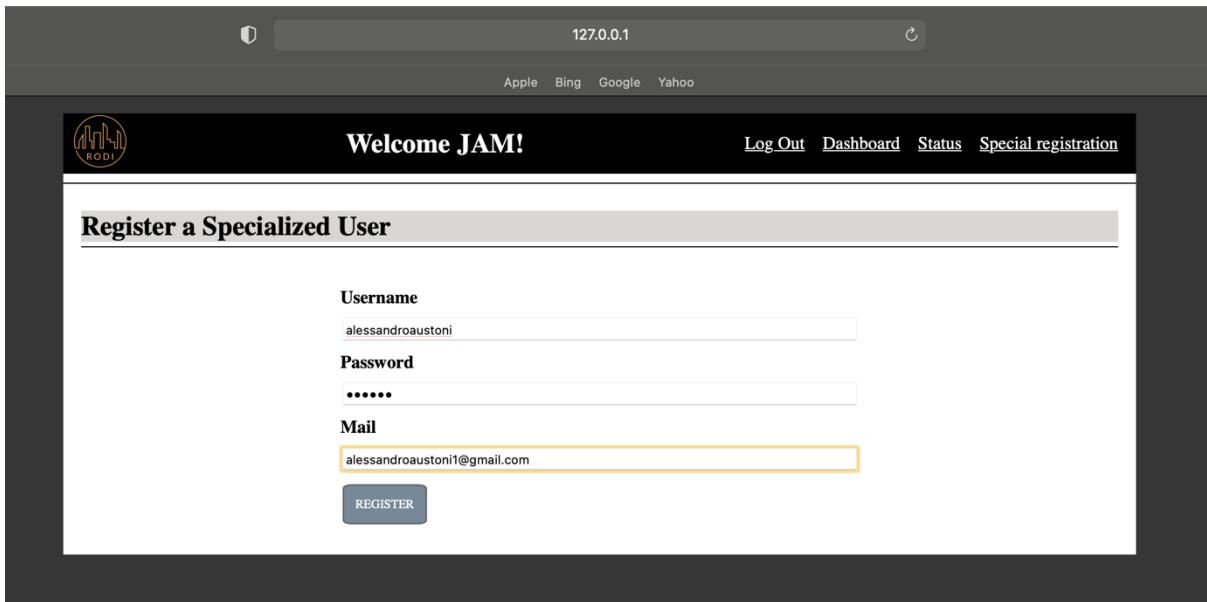
Figure 13: Super User Log-in



A screenshot of a web browser showing a log-in form. The address bar shows '127.0.0.1'. The page title is 'RODI'. The navigation bar includes links for 'Register', 'Log In', 'Specialized user request', and 'Map'. The main content area has a header 'Log In' and contains fields for 'Username' (containing 'JAM') and 'Password' (containing '*****'). A 'LOG IN' button is at the bottom.

And then access the specialized user registration web page thanks to the special registration link, and performs a specialized registration (here the specialized user's username is: 'alessandroaustoni').

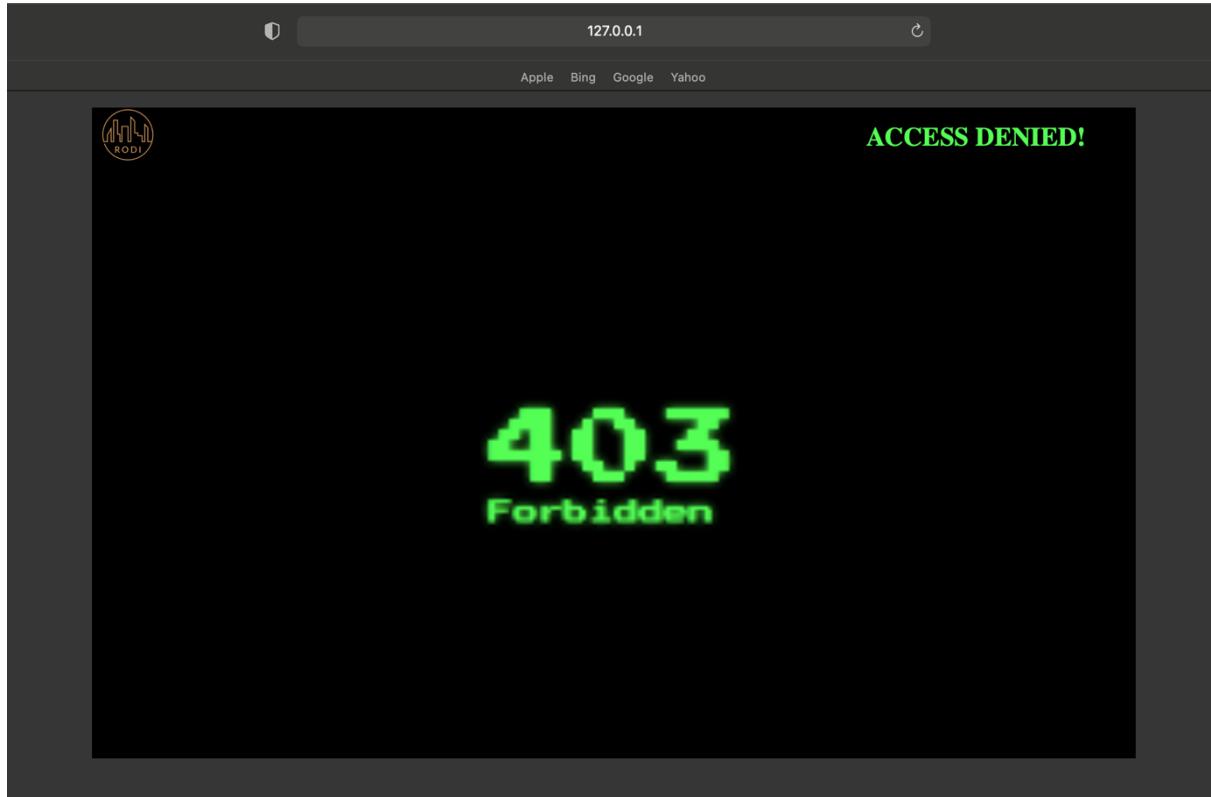
Figure 14: Super User registers Specialized User



A screenshot of a web browser showing a registration form. The address bar shows '127.0.0.1'. The page title is 'RODI'. The navigation bar includes links for 'Log Out', 'Dashboard', 'Status', and 'Special registration'. The main content area has a header 'Welcome JAM!' and a sub-header 'Register a Specialized User'. It contains fields for 'Username' (containing 'alessandroaustoni'), 'Password' (containing '*****'), and 'Mail' (containing 'alessandroaustoni1@gmail.com'). A 'REGISTER' button is at the bottom.

If a user that is different from the Super User tries to access that page to register a specialized user, he/she would get redirected to an access denied web page, that prevents the access to the section. This is the result of that trial for a not registered user, a citizen user (Username: juanfra_54) and specialized user (Username: alessandroaustoni)

Figure 15: Access Denied



3. LoginMember

All the users log in by implementing the same steps which are the next ones:

- Access to the link “Log in”
- Then inserting their username and password in the system
- Finally pressing the “Log in” button

To control this procedure different tests were implemented for each kind of user. The usernames considered in the tests were the ones exposed in the previous use cases with the objective to maintain the same convention.

Figure 16: Normal user correct Log-in

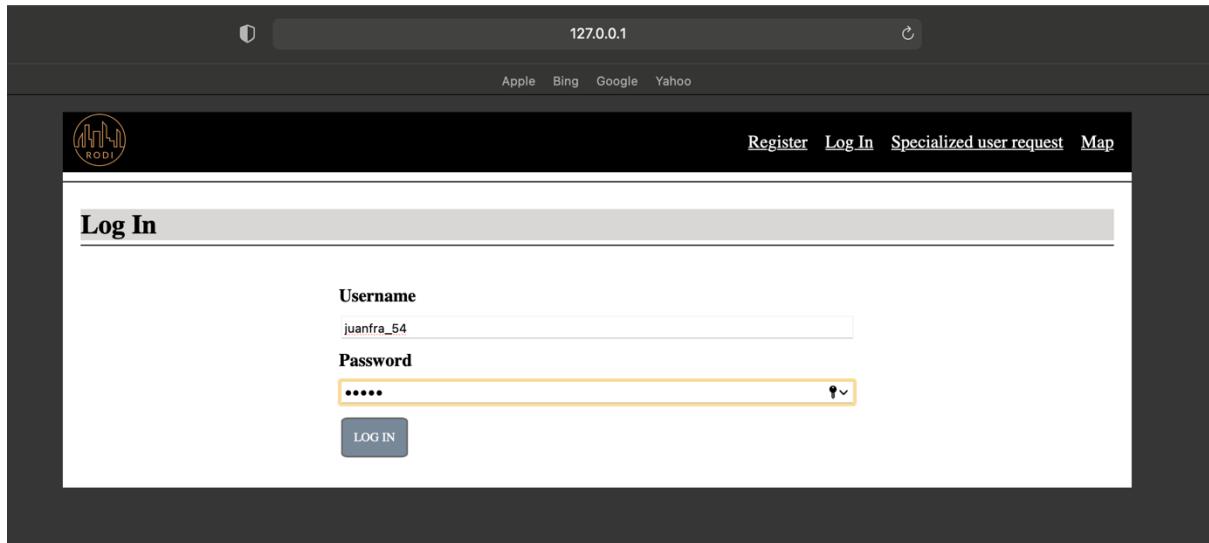


Figure 17: Normal User correctly logged in

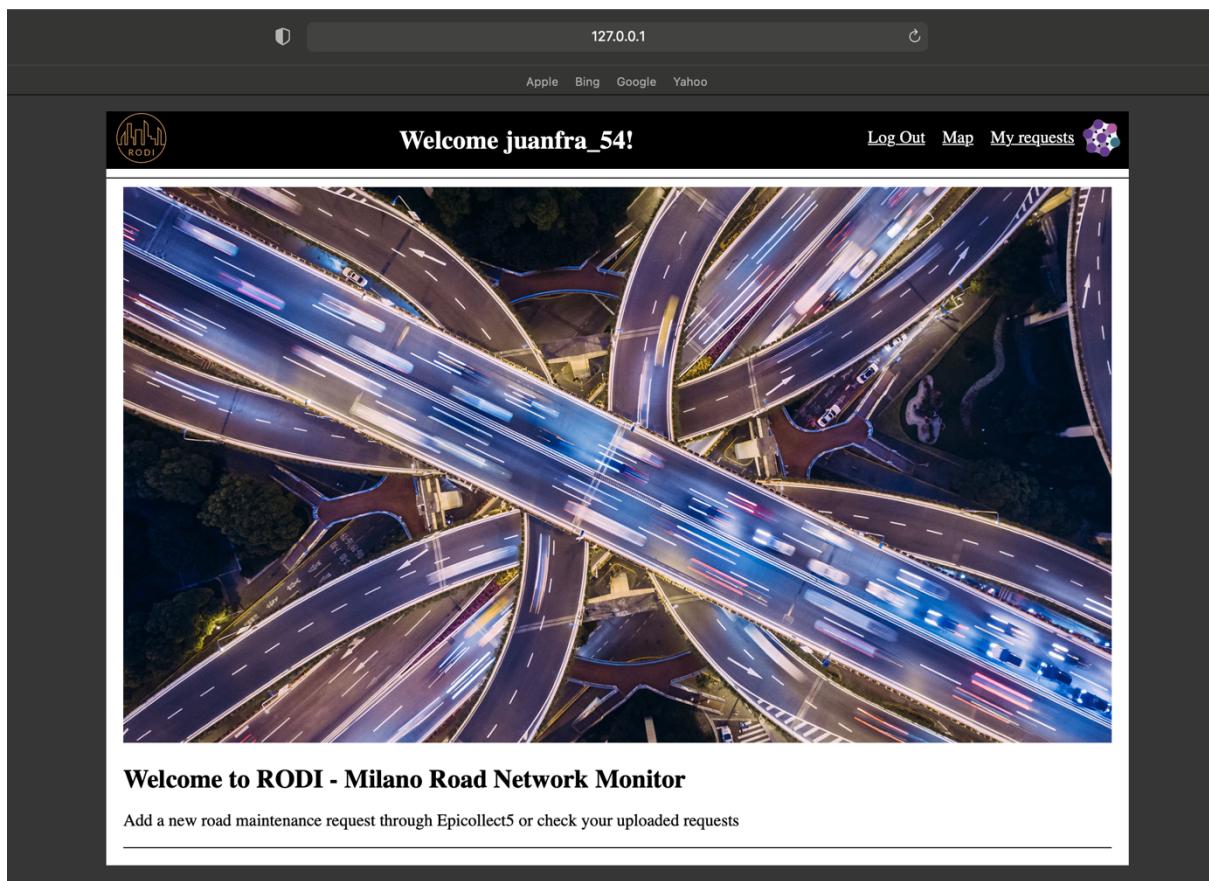
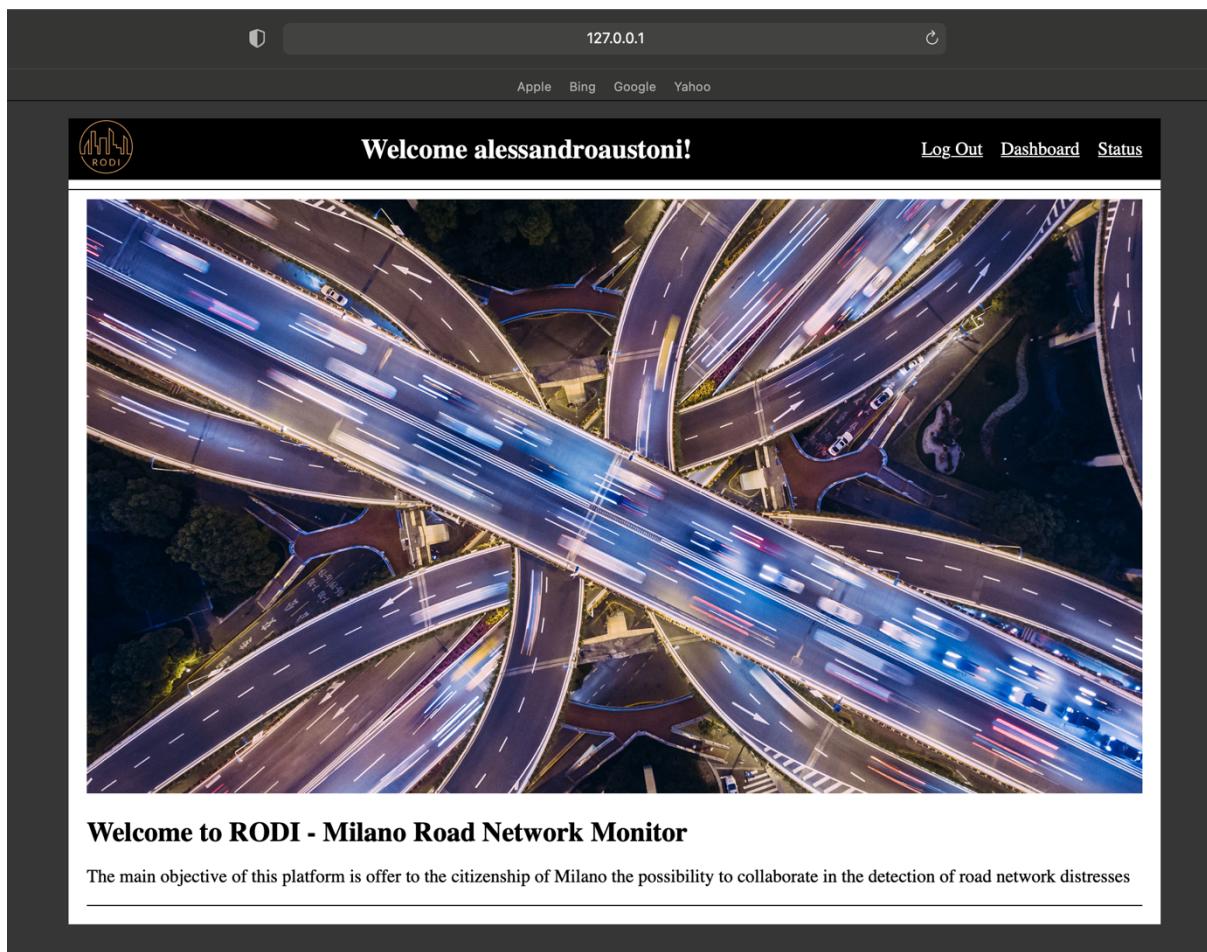


Figure 18: Specialized user correct log-in

The screenshot shows a web browser window with the address bar set to 127.0.0.1. The page title is "Log In". At the top right, there are links for "Register", "Log In", "Specialized user request", and "Map". On the left, there is a circular logo with a stylized building icon and the word "RODI". The main form area has two input fields: "Username" containing "alessandroaustoni" and "Password" containing "*****". Below the password field is a dropdown menu. A large blue "LOG IN" button is at the bottom of the form.

Figure 19: Specialized user correctly logged in



There were also tested cases with incorrect usernames and passwords and the application displayed a popup indicating the error in the input data received by the user who tried to log in.

Figure 20: Incorrect username

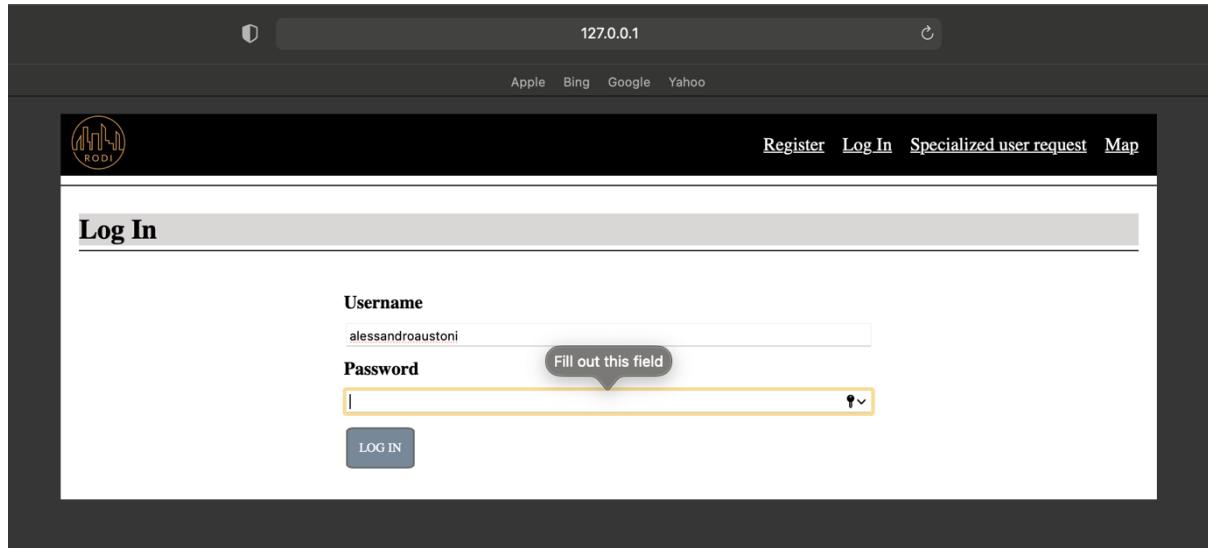
A screenshot of a web browser window. The address bar shows '127.0.0.1'. The page title is 'Log In'. At the top right, there are links for 'Register', 'Log In', 'Specialized user request', and 'Map'. Below the title, a yellow horizontal bar contains the text 'Incorrect username.'. Below this bar are two input fields labeled 'Username' and 'Password', and a blue 'LOG IN' button.

Figure 21: Incorrect password

A screenshot of a web browser window. The address bar shows '127.0.0.1'. The page title is 'Log In'. At the top right, there are links for 'Register', 'Log In', 'Specialized user request', and 'Map'. Below the title, a yellow horizontal bar contains the text 'Incorrect password.'. Below this bar are two input fields labeled 'Username' and 'Password', and a blue 'LOG IN' button.

Considering the use case in analysis, it was also checked the situation in which the user does not includes some of the fields requested in the log in form. In this case the screen shows a label indicating that the field is required to advance with the log in.

Figure 22: Field empty – Log in



4. ViewMap

Focusing on the map it was analyzed the possibility to analyze the information rendered in the map displayed in the “Map” section of the web application. This map is available for any kind of user, even for those users who are not registered in the application.

In order to display the map, the user should press the link “Map” in the index page.

Figure 23: Index page – Map link



Then the map is rendered and there are different possible interactions to perform in order to analyze the presented information.

Figure 24: Map functionality testing (clusters and zoom)

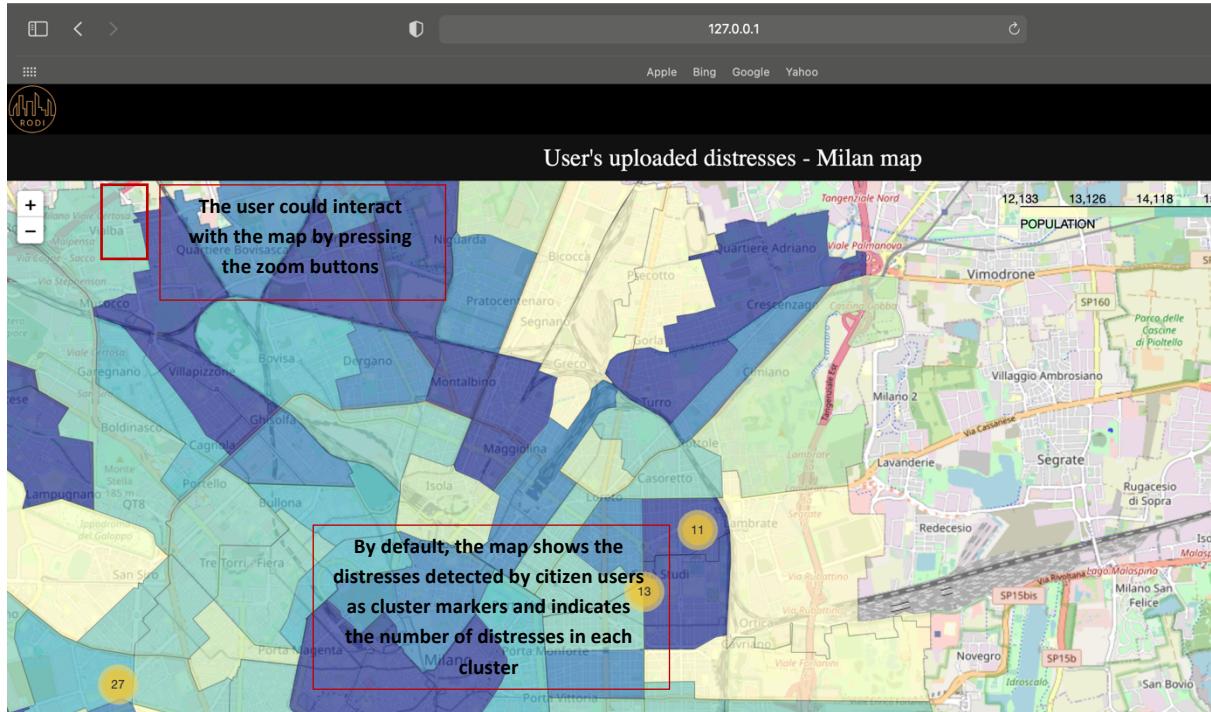
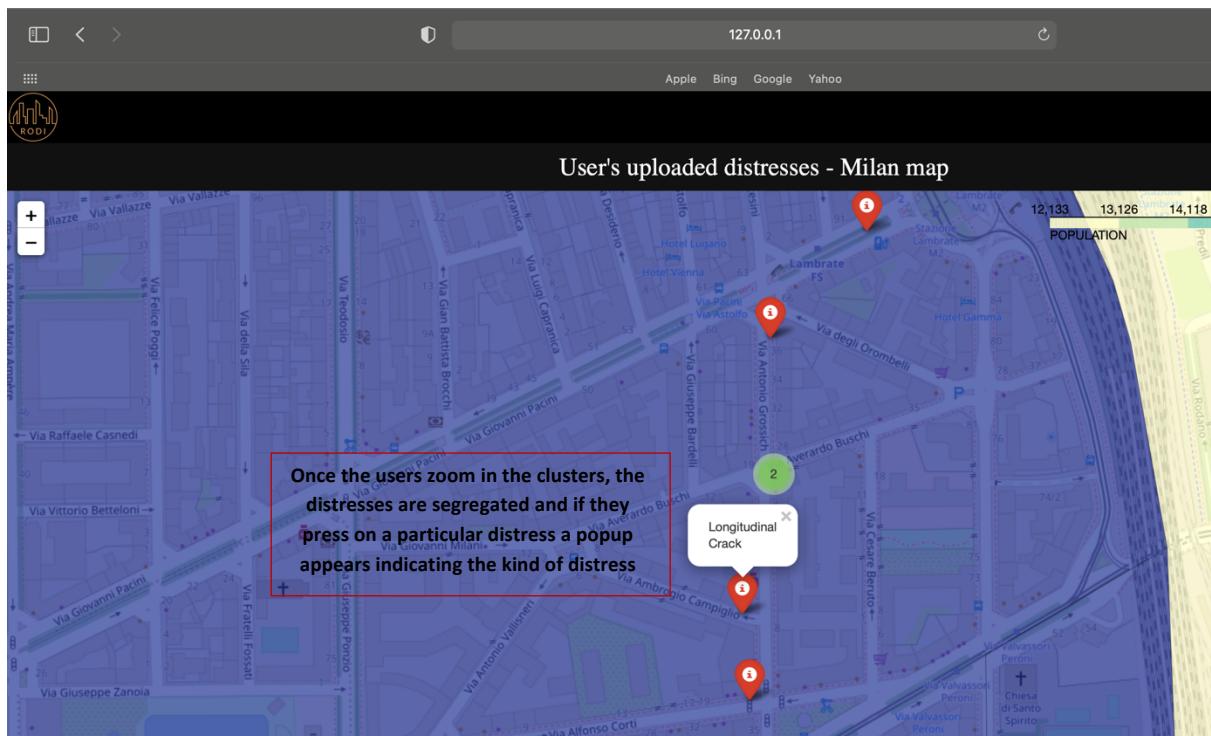


Figure 25: Map marker testing



5. VisualizeData

The main objective here, is to check the interaction with the different graphs that expose the data in the dashboard and to control the access of different users to this section.

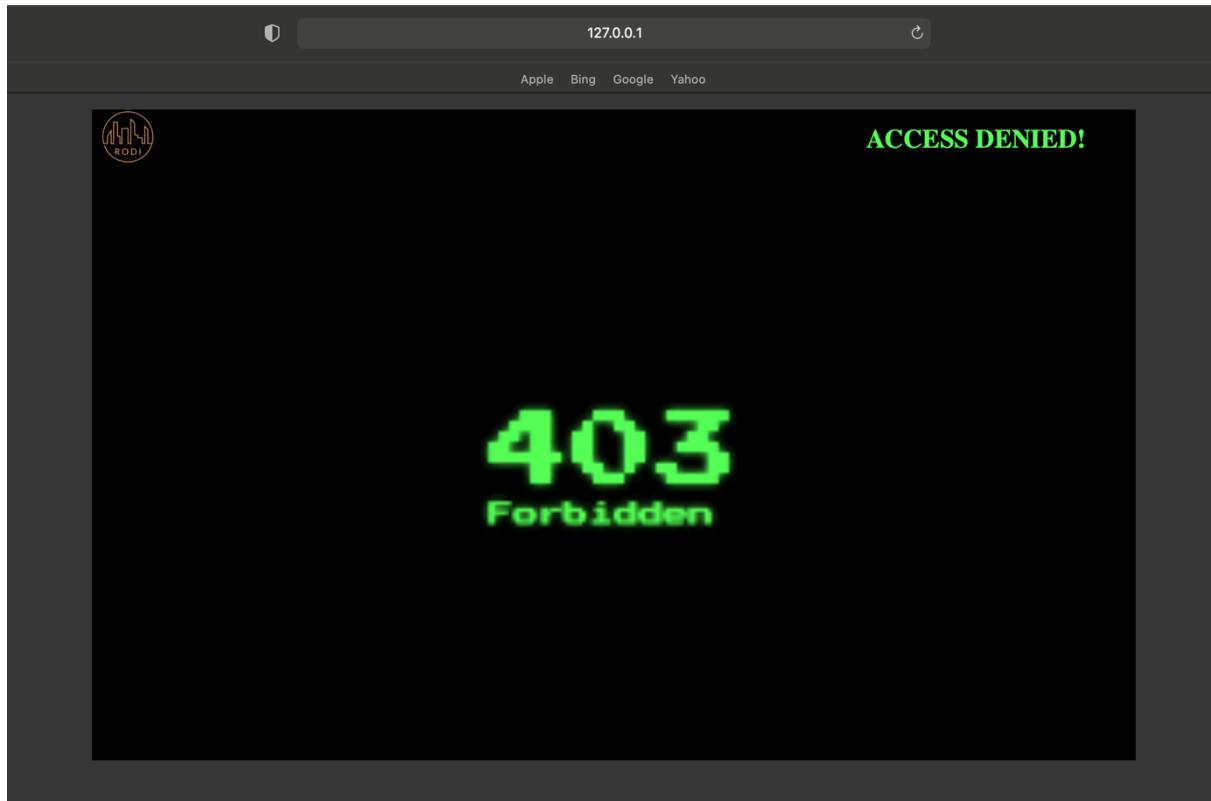
Only the super user “JAM” and the specialized users has the permission to access to the dashboard section.

Figure 26: Dashboard section access



In case a citizen user or a not-registered user tries to access to this section the app will render a page informing that the access is denied.

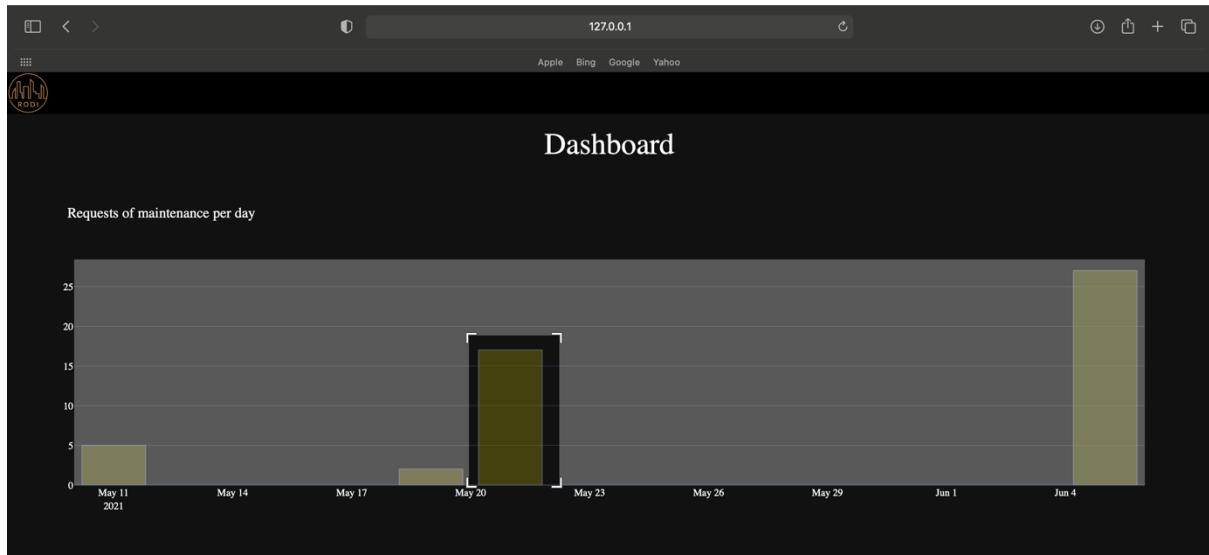
Figure 27: Access denied – dashboard section



Once a specialized user or the super user (“JAM”) accesses to this section a few interactions are then possible.

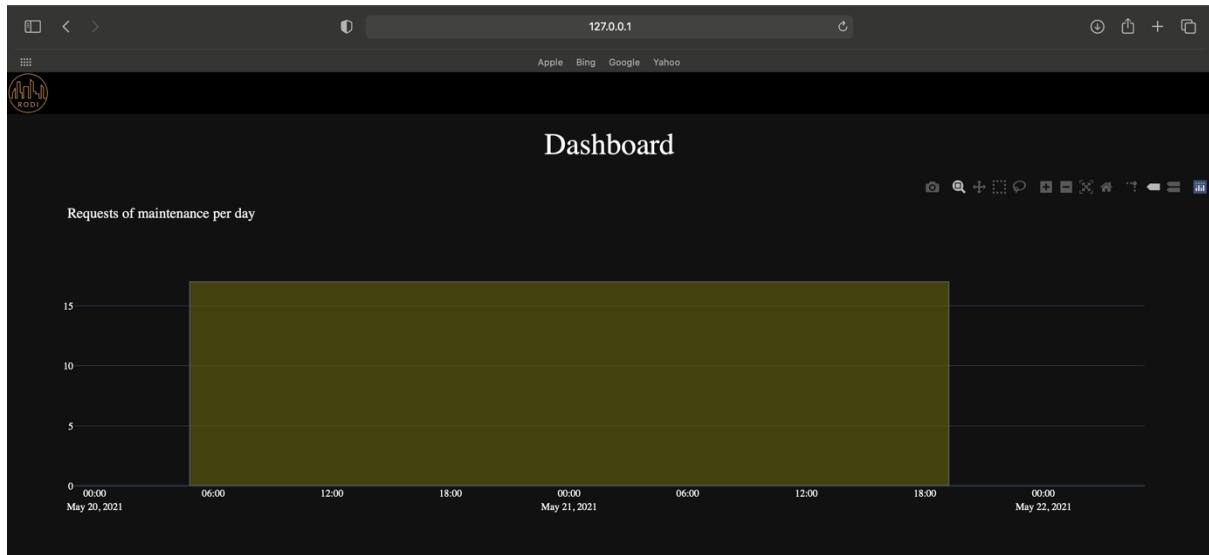
Here were tested the possible interactions with the graphs exposed below that allows the users to filter different categories just by selecting the desired section with the mouse.

Figure 28: Dashboard graphs testing functionalities



This will result in a correct zoom of the desired section, which will expose more details about the desired information (in this case a denser time interval).

Figure 29: Dashboard section zoom



6. MyRequestList

In this case it was important to detect if the users could retrieve their own requests uploaded previously in Ep5 and it also was checked that different citizen users cannot visualize the requests uploaded by others in the “my requests” section.

Figure 30: Citizen user log in index

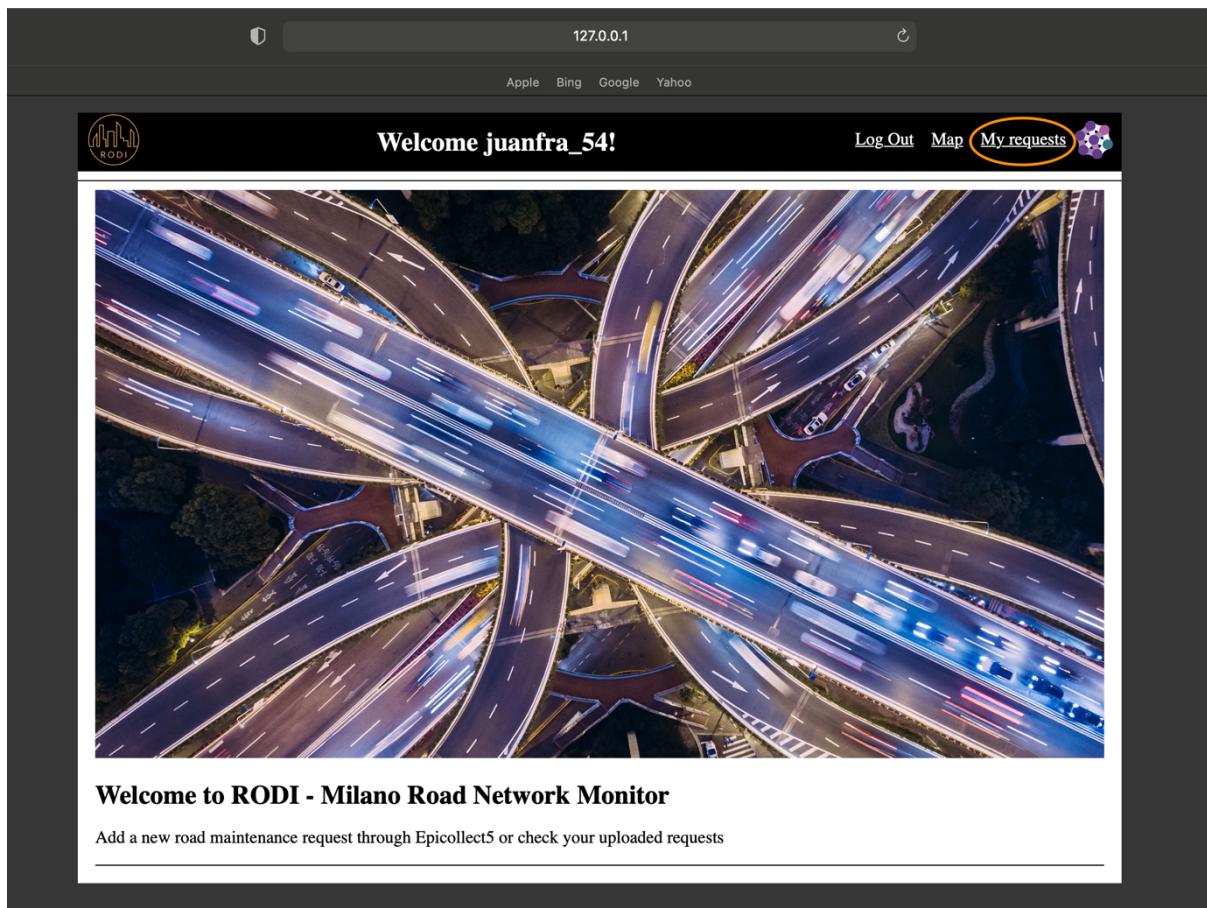


Figure 31: User maintenance requests

The screenshot shows a web browser window with the URL 127.0.0.1. The header includes links for Apple, Bing, Google, and Yahoo. On the left is a logo for RODI. The main content area is titled "User Requests". It lists three entries from the user "juanfranciscoamieva@gmail.com" under the category "On Going". Each entry includes a timestamp (e.g., 13:20:10 | 21/05/2021), a location (Flexible pavement (Asphalt)), a defect type (Block Crack! Higher than 2 m | Longitudinal Crack! Higher than 2 m | Pothole 0 - 0.5 m), and a small image of the road surface.

Then, was also fundamental, to inspect if a user without being registered to the web app could have access to this section. The following test demonstrates that this access is not allowed if the user is not correctly logged in.

Figure 32: User requests trial without been logged in

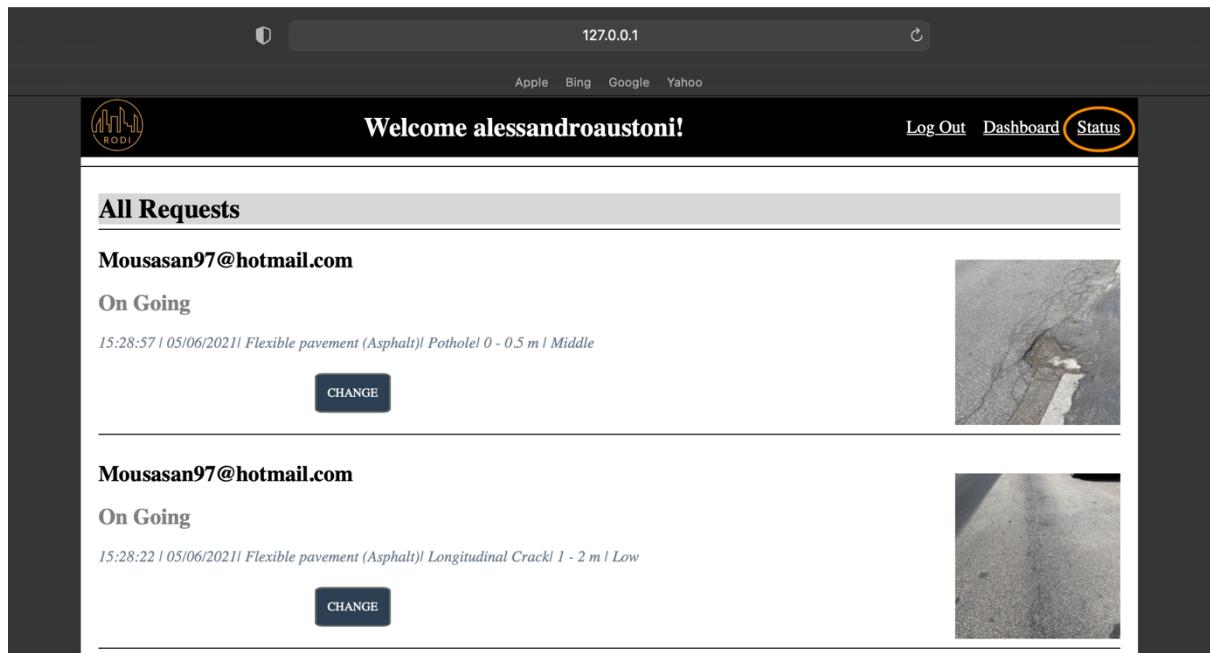
The screenshot shows a "Log In" page. At the top, there is a message in a yellow box: "you should login for seeing user requests". Below this, there is a form with two input fields labeled "Username" and "Password", and a "LOG IN" button.

7. RequestStatusModification

The Super User (JAM) and the Specialized users should have the possibility to watch on screen the different requests uploaded by the citizens and at the same time they should have the option to modify their status in order to let the normal users know if the specialized users are already working (or not) on the uploaded request of the respective (normal) user.

When the Specialized user access to the “status” section, all the requests in the database are correctly displayed on screen together with a button for each entry, to allow the change of status of each specific request (see picture below).

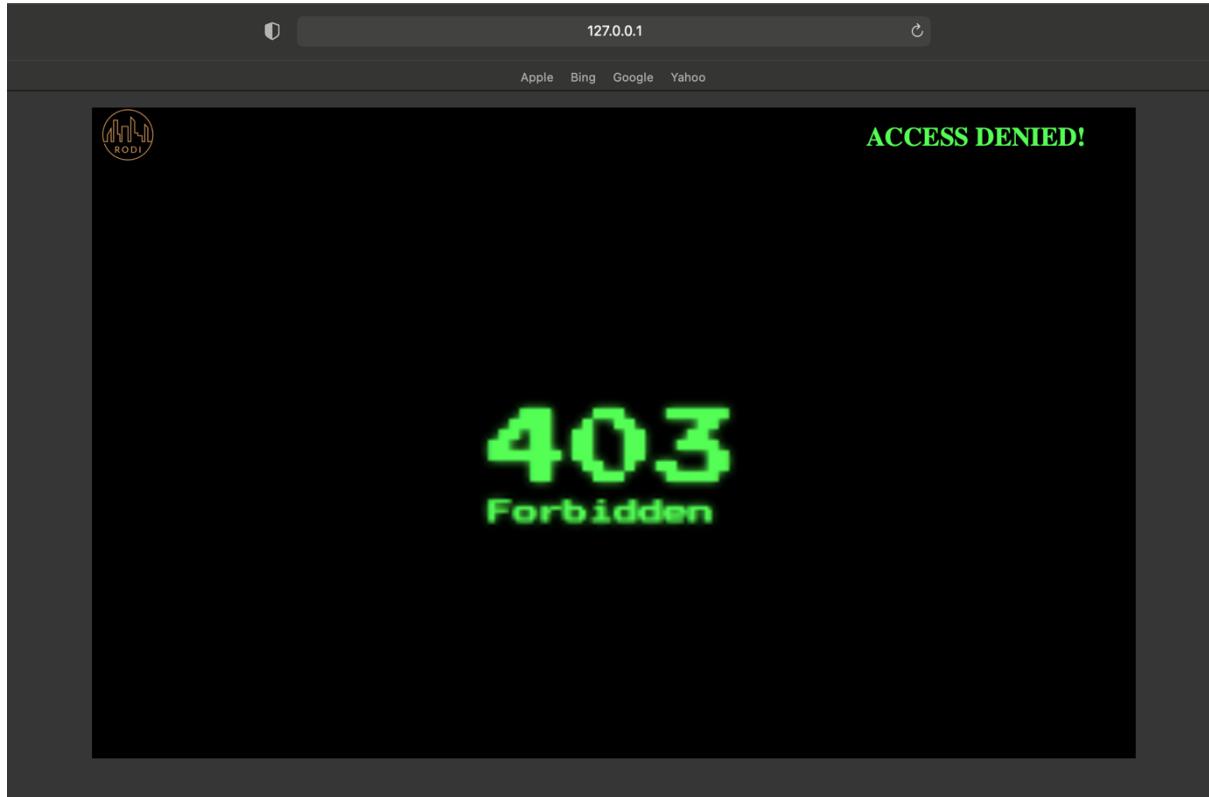
Figure 33: status section



A simple test was performed with the previous normal user () and not registered user, to make sure that only special users can have access to that page (and so modify the status of a request).

The result was successful, access denied for both kinds of users.

Figure 34: Access denied – status section



To assert the correct functioning of the changing request button, it was made a trial that consists in pushing a “change” button next to a specific request. Like the picture demonstrates the test was passed and the status was correctly update in the database and changed from “On going” (still to be taken in charge) to “complete” (processed).

Figure 35: status changed – specialized user interface

Mousasan97@hotmail.com

Complete

15:28:57 | 05/06/2021 | Flexible pavement (Asphalt) | Pothole 0 - 0.5 m | Middle

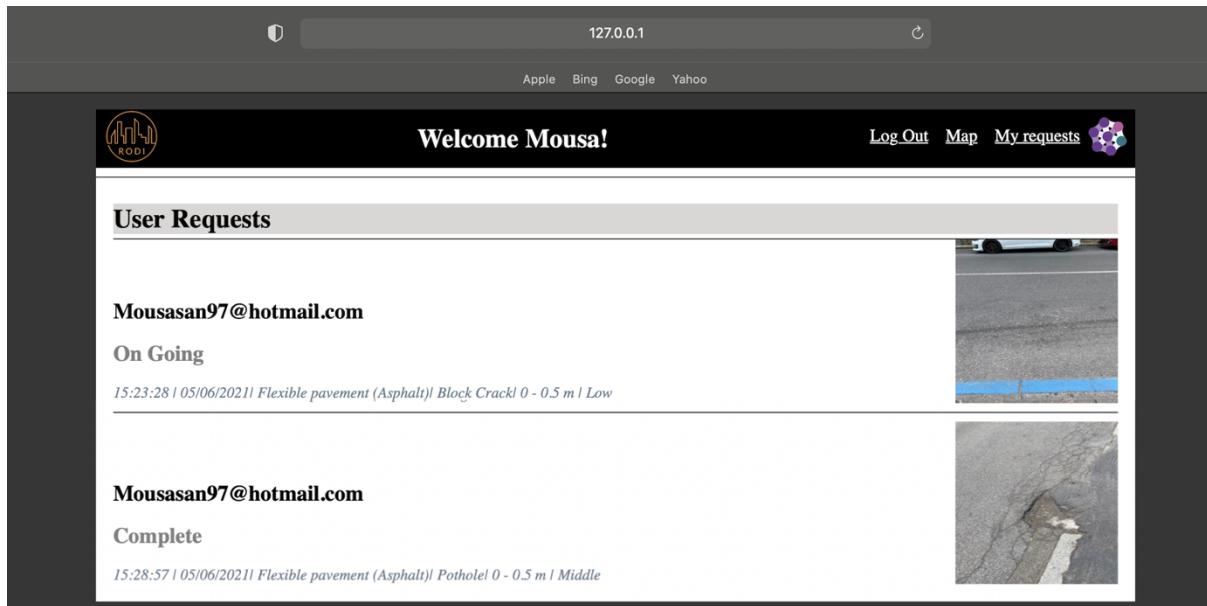
CHANGE



8. VisualizationUpdatedRequest

This success is also validated by the update seen also in the “my requests” section of the normal user (picture below)

Figure 36: status changed – Citizen user interface



Finally, the last test was performed again on the “change” button. This time unfortunately a fail occurred. Unfortunately, if the specialized user presses again the button after the status has been changed to “Complete” the system is not able to change it back to “ON _GOING”. This is for sure an un-wanted feature, because if a specialized user presses a button by mistake (human possible error), then the specialized user is no more able to change back the status of that request.

Conclusion and future work

As an important aspect in these tests which were performed manually, was detected a mistake in the case a user tried to register its account (citizen user account) with the same email than a previously registered user. Thanks to that detection the problem was solved and it is not more a gap to solve in terms of the related requirements for that use case.

A second aspect to highlight is the level of robustness observed in the log in and registration use cases after the previously mentioned correction. As were observed in their related sections, were performed different tests and all of them shows solid results with different advertences for the users in case they complete incorrectly the information required or if they omit some data.

Another relevant point is that the access to the dashboard section was restricted to specialized users and super user and the tests shows that if a non-registered user or a citizen user tries to access to them the access will be denied.

Finally, it was detected a small bug in the status change web section, that may lead to an irreversible error by the user, but it is planned to be fixed as soon as possible.