



**POLITECNICO**  
**MILANO 1863**

Software Engineering Project  
Lorenzo Amici, Marina Ranghetti,  
Marta Rossi, Yinyao Zhang

# Testing Plan

---

**Deliverable:** TP

**Title:** Testing Plan

**Authors:** Lorenzo Amici, Marina Ranghetti, Marta Rossi, Yinyao Zhang

**Version:** 1.0

**Date:** 12-June-2019

**Download page:** <https://github.com/MarinaZen/SoftwareEng.git>

**Copyright:** Copyrightc 2019, Lorenzo Amici, Marina Ranghetti, Marta Rossi, Yinyao Zhang – All rights reserved

---

## Contents

<b>1. Customer who is not registered user .....</b>	<b>5</b>
1.1 Visualization app content	5
1.2 Reading user's posts	5
1.2.1 Blog section	5
1.2.2 reading posts (from the recent)	5
1.3 Visualization number of free bikes/stalls	5
1.3.1 Map page section	5
1.3.2 Update of initial position	5
1.3.3 Visualization of the nearest docking stations	5
1.3.4 Pop-up with information of the station (bikes, stalls)	6
1.3.5 Insert starting point and destination	6
1.3.6 Navigation to the destination	6
<b>2 Customer who is registered user .....</b>	<b>6</b>
2.1 Registration	6
2.1.1 Username	6
2.1.2 Mail	7
2.1.3 Password	7
2.1.6 Confirm password	7
2.1.7 Check PIN	7
2.2 Login	8
2.2.1 User mail	8
2.2.2 User password	8
2.2.3 Remember me	8
2.3 Retrieve password	9
2.3.1 Click forgot password	9
2.3.2 Insert email of registration	9
2.3.3 Check mailbox for assistance email	9
2.3.4 Click link in the email (user's mail)	10
2.3.5 Insert new password	10
2.3.6 Confirm new password	10
2.4 Updating profile	10
2.4.1 Login precondition	10
2.4.2 Account section	10
2.4.3 Change username	10
2.4.4 Change email	11

2.4.5 Update profile picture	11
2.5 Adding posts	11
2.5.1 Login precondition	11
2.5.2 New post section	11
2.5.3 Insert title	11
2.5.4 Insert content	11
2.6 Updating posts	12
2.6.1 Login precondition	12
2.6.2 New post precondition	12
2.6.3 Blog section	12
2.6.4 Click on your own post	12
2.6.5 Click update button	12
2.6.6.1 Insert new title	12
2.6.6.2 Insert new content	12
2.7 Deleting posts	13
2.7.1 Login precondition	13
2.7.2 New post precondition	13
2.7.3 Blog section	13
2.7.4 Click on your own post	13
2.7.5 Click delete button	13
2.7.6 Confirm deleting	13
2.7.7 Cancel	13
2.8 Displaying statistical information	13
2.8.1 Login precondition	13
2.8.2 Map section	13
2.8.3.1 Clicking a station	13
2.8.3.2 Pop-up with information of the station (bikes, stalls)	13
<b>3 Testing results .....</b>	<b>14</b>

# 1. Customer who is not registered user

## 1.1 Visualization app content

Table 1.1 Testing of Homepage section

Name/Action	Enter to see the homepage of website
Import/Input	<a href="http://127.0.0.1:5000">http://127.0.0.1:5000</a>
Initial state	Home page
Export/Output	Able to see home page
Final state	

## 1.2 Reading user's posts

### 1.2.1 Blog section

Table 1.2.1 Testing of Blog

Name/Action	Click Blog to see information
Import/Input	Click blog section
Initial state	Home page
Export/Output	History of recent posts
Final state	

### 1.2.2 reading posts (from the recent)

Table 1.2.2 Testing of Reading user's posts

Name/Action	Reading all posts
Import/Input	Click page 1 and 2 to see old posts
Initial state	History of recent posts
Export/Output	Able to see all posts
Final state	

## 1.3 Visualization number of free bikes/stalls


### 1.3.1 Map page section

Table 1.3.1 Testing of Visualization number of free bikes/stalls

Name/Action	Navigate to Map page to see all stations in map
Import/Input	Click map section
Initial state	Home page
Export outputs	All stations are shown in open street map
Final state	

### 1.3.2 Update of initial position

Table 1.3.2 Testing of update of initial position

Name/Action	Click  to automatically have the real-time position
Import/Input	Click "my position" icon
Initial state	Map page with all stations
Export/Output	Pop-up window for allowing getting the position, if accepted a blue point will be render on the map with your real-time position and accuracy
Final state	

### 1.3.3 Visualization of the nearest docking stations

Table 1.3.3 Testing of visualization

Name/Action	Display the stations
Import/Input	Stay on map page
Initial state	Map page
Export/Output	Visualization of the stations around the user
Final state	

### 1.3.4 Pop-up with information of the station (bikes, stalls)

Table 1.3.4 Testing of every station

Name/Action	Click to get information of every station
Import/Input	Choose different stations
Initial state	Map page
Export/Output	Display numbers of stalls, location about every station and other information
Final state	

### 1.3.5 Insert starting point and destination

Table 1.3.5 Testing of initialization of routing machine

Name/Action	Click on the map to insert coordinates of the desire and actual position
Import/Input	Actual location and destination
Initial state	Map page
Export/Output	The chosen position are shown in a designed window
Final state	

### 1.3.6 Navigation to the destination

Table 1.3.6 Testing of navigation

Name/Action	Show the path from origin to destination
Import/Input	Actual location and destination
Initial state	Map page
Export/Output	The route to get the destination is shown with a red line
Final state	

## 2 Customer who is registered user

### 2.1 Registration

#### 2.1.1 Username

Table 2.1.1 Testing of insert username

Name/Action	Click field to input username
Import/Input	Username of the user for registration
Initial state	Registration page
Export/Output	If the form is completely filled, once clicked on the button sign up, there will be a pop up for making the user understand if the form was well-filled. In case the username was already taken an alert will be display to the user

Final state	Username, mail and password of the user will be save in the database, only if the for was well filled.
-------------	--

### 2.1.2 Mail

Table 2.1.1 Testing of insert mail

Name/Action	Click field to input email
Import/Input	user's mail
Initial state	Registration page
Export/Output	If the form is completely filled, once clicked on the button sign up, there will be a pop up for making the user understand if the form was well-filled. In case the email was already taken or was wrongly written an alert will be display to the user
Final state	Username, mail and password of the user will be save in the database, only if the for was well filled.

### 2.1.3 Password

Table 2.1.3 Testing of insert password

Name/Action	Click field to input password
Import/Input	Password of th user
Initial state	Registration page
Export/Output	If the form is completely filled, once clicked on the button sign up, there will be a pop up for making the user understand if the form was well-filled. In case the password doesn't fulfill the requirement an alert will be display to the user
Final state	Username, mail and password of the user will be save in the database, only if the for was well filled.

### 2.1.6 Confirm password

Table 2.1.6 Testing of confirm password

Name/Action	Click field to input password again
Import/Input	Password of th user
Initial state	Registration page
Export/Output	If the form is completely filled, once clicked on the button sign up, there will be a pop up for making the user understand if the form was well-filled. In case the password doesn't fulfill the requirement and it is not equal to the previous field (see 2.1.3) an alert will be display to the user
Final state	Username, mail and password of the user will be save in the database, only if the for was well filled.

### 2.1.7 Check PIN

Table 2.1.7 Testing of insert PIN

Name/Action	Click field to input PIN
Import/Input	PIN provided by Comune di Milano
Initial state	Registration page
Export/Output	If the form is completely filled, once clicked on the button sign up, there will be a pop up for making the user understand if the form was well-filled. In case the PIN is not inside our database an alert will be displayed to the user
Final state	The PIN will be deleted from our database. Username, mail and password of the user will be saved in the database.

## 2.2 Login

### 2.2.1 User mail

Table 2.2.1 Testing of login - user's email

Name/Action	Open Login page
Import/Input	User email in the form
Initial state	empty form
Export/output	Only an email existing in the database will be accepted. If it doesn't follow the standards it won't be recognised as valid email and a pop up will alert the user that it is invalid.
Final state	Email accepted only if compliant with standards and already existing in the database.

### 2.2.2 User password

Table 2.2.2 Testing of login - user's password

Name/Action	Fill the password form
Import/Input	Personal user's password
Initial state	empty form
Export/output	If the user writes his correct password some extra functionalities will be available both in the Blog and Map page. If the password doesn't correspond to the one stored in the database a pop up alert will inform the user that the password corresponding to email is wrong.
Final state	Login accepted/refused

### 2.2.3 Remember me

Table 2.2.3 Testing of remember me checkbox

Name/Action	Click on the checkbox at the bottom of Login page
Import/Input	Close web app page and then open it again



Initial state	Box checked, logged in
Export/output	The user will be logged in even if the page has been closed.
Final state	The user can continue to use the web app, avoiding to login each time the web page is closed.

## 2.3 Retrieve password

### 2.3.1 Click forgot password

Table 2.3.1 Testing of change password - 1 step

Name/Action	Click on the link related to forgotten password in the Login page
Import/Input	User's email
Initial state	The user forgot the password and is unable to log in: clicking the link a page related to password reset is opened
Export/Output	The user's email will be recognised only if it corresponds to an existing email in the database
Final state	The user's email is accepted if already registered. If so the possibility to create a new password is guaranteed.

### 2.3.2 Insert email of registration

Table 2.3.2 Testing of change password - 2 step

Name/Action	Reset password
Import/Input	Click "Request Password Reset"
Initial state	The user's email is written in the form and if already existing in the database is recognised
Export/Output	The Login page is again displayed and the user is informed with an alert message that an email has been sent with related instructions to reset the password.
Final state	

### 2.3.3 Check mailbox for assistance email

Table 2.3.3 Testing of received email - 3 step

Name/Action	The user checks that the email to reset the password has been successfully received
Import/Input	Opening the email the user checks if the user's name displayed in the email is the one he registered with in the web app
Initial state	A reset password request was sent from the web app, an email with user's name is received

Export/Output	The user can click on a link that allows to reset password.
Final state	

#### 2.3.4 Click link in the email (user's mail)

Table 2.3.4 Testing of reset link - 4 step

Name/Action	Check the availability of link
Import/Input	The user can click the link to reset password
Initial state	The link should redirect the user to reset password page
Export/Output	The reset password page opens and the user can create the new password.
Final state	

#### 2.3.5 Insert new password

Table 2.3.5 Testing of insert new password

Name/Action	Write the new password in the form
Import/Input	New password is chosen and put in the form
Initial state	Previous password still available in the database but forgotten by the user
Export/Output	New password accepted in the form and no alerts are sent to the user
Final state	

#### 2.3.6 Confirm new password

Table 2.3.6 Testing of reset password

Name/Action	Write the new password in the form
Import/Input	Same new password written in the second form
Initial state	Previous password still available in the database but forgotten by the user
Export/Output	If the new password is the same in the two forms it is then stored in the database and associated to user's email
Final state	The user can now login using his email and the new password.

### 2.4 Updating profile

#### 2.4.1 Login precondition

#### 2.4.2 Account section

Table 2.4.2 Testing of account section

Name/Action	Account functions
Import/Input	Click on Account
Initial state	Home page
Export/Output	Show information about user's account
Final state	

#### 2.4.3 Change username

Table 2.4.3 Testing of change of information

Name/Action	Click on username field
Import/Input	New username
Initial state	Show exist username in the table
Export/Output	Table borders become blue when insert characters, and show new username in the table
Final state	

#### 2.4.4 Change email

Table 2.4.4 Testing of change email

Name/Action	Change information
Import/Input	New mail address
Initial state	Show existing email in the table
Export/Output	Table borders become blue when insert characters, and show new mail in the table
Final state	

#### 2.4.5 Update profile picture

Table 2.4.5 Testing of upload of profile

Name/Action	Click "choose file" icon and click "Update"
Import/Input	Choose picture from local computer
Initial state	Preinstalled image
Export/Output	Update the profile picture and see the message: "Your account has been updated!" in green behind home page bar.
Final state	

### 2.5 Adding posts

#### 2.5.1 Login precondition

#### 2.5.2 New post section

Table 2.5.2 Testing of post section

Name/Action	Click "New Post" in the home page bar
Import/Input	Null
Initial state	Home page
Export/Output	Jump to "New Post" section
Final state	

#### 2.5.3 Insert title

Table 2.5.3 Testing of edit post

Name/Action	Edit Title field of a post
Import/Input	New Title for the post
Initial state	Null
Export/Output	Table borders become blue when the user starts to insert characters, and show input characters in the table
Final state	

#### 2.5.4 Insert content

Table 2.5.4 Testing of edit post

Name/Action	Edit Content field of a post
Import/Input	New Content for the post
Initial state	Null
Export/Output	Table borders become blue when the user starts to insert characters and show input characters in the table
Final state	

## 2.6 Updating posts

### 2.6.1 Login precondition

### 2.6.2 New post precondition

Table 2.6.2 Testing of post section

Name/Action	New post
Import/Input	Click "Post" in the Update Post section
Initial state	Update Post page
Export/Output	User is redirected to the blog section and the message "Your post has been created!" appears
Final state	The post is created

### 2.6.3 Blog section

### 2.6.4 Click on your own post

Table 2.6.4 Testing of update post

Name/Action	Update post
Import/Input	Click user's post
Initial state	Blog page
Export/Output	Jump to user's own page with "Update" in grey and "Delete" in red.
Final state	

### 2.6.5 Click update button

Table 2.6.5 Testing of edit post

Name/Action	Click update
Import/Input	Null
Initial state	User's own post
Export/Output	Jump to "Update Post" page
Final state	

#### 2.6.6.1 Insert new title

#### 2.6.6.2 Insert new content

Table 2.6.6 Testing of edit post

Name/Action	Edit post
Import/Input	New title and Content and click "post"
Initial state	Former user's post
Export/Output	Table borders become blue when the user starts to insert characters and shows input characters in the table. After clicking "Post", the page shows the new post, and the message "Your post has been updated!" appears in green behind the home page bar.

Final state	
-------------	--

## 2.7 Deleting posts

### 2.7.1 Login precondition

### 2.7.2 New post precondition

### 2.7.3 Blog section

### 2.7.4 Click on your own post

### 2.7.5 Click delete button

Table 2.7.5 Testing of delete post

Name/Action	Delete post
Import/Input	Click "delete"
Initial state	User's own post
Export/Output	Pop-on window shows "Delete post?" in red with "Close" and "Delete" options
Final state	

### 2.7.6 Confirm deleting

Table 2.7.6 Testing of deleting post

Name/Action	Confirm deleting
Import/Input	Click "Delete"
Initial state	Delete Post window
Export/Output	User is redirected to the blog page and the message "Your post has been deleted" appears in green behind the home page bar.
Final state	Post is deleted

### 2.7.7 Cancel

Table 2.7.7 Testing of cancel deleting

Name/Action	Cancel deleting
Import/Input	Click "Close"
Initial state	Delete Post window
Export/Output	User is redirected to his post
Final state	

## 2.8 Displaying statistical information

### 2.8.1 Login precondition

### 2.8.2 Map section

Table 2.8.2 Testing of map section

Name/Action	Map section
Import/Input	Click "Map"
Initial state	Home page
Export/Output	The interactive map appears, with every station location on it
Final state	

### 2.8.3.1 Clicking a station

### 2.8.3.2 Pop-up with information of the station (bikes, stalls)

Table 2.8.3 Testing of station information

Name/Action	Show stations' information
-------------	----------------------------

Import/Input	Click the selected station on the map or select it from the dropdown menu
Initial state	Map page
Export/Output	Pop-on window containing the number the station, the stalls number, the address, latitude and longitude coordinates, the available bikes at the moment and the "Statistic" button.
Final state	

### 3 Testing results and conclusions

The web app seems to work properly for every case that was tested.

The filling in of all the fields works perfectly both in the right case (action is completed) and in the wrong one (warning message shows up).

All the functionalities that were tested (maps, statistics, blog, account and posts management) responded correctly, and all the requirements previously established were fulfilled by the app.

The limitation of access for non registered users works as it's meant to, denying the access to the pages of statistics and account/posts management.

Overall, the app didn't show any kind of unexpected behavior, and all the requirements that the application was supposed to achieve passed the testing phase.

### 4 Effort spent

Marina Ranghetti: RASD, DD and web application (website and map)

Marta Rossi: RASD, DD and web application (website and code comments)

Lorenzo Amici: RASD, DD, Testing document, statistics and their implementation

Yinyao Zhang: DD, testing of use cases and code comments