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

Requirement Analysis and Specification Document

Deliverable: RASD
Title: Requirement Analysis and Verification Document
Authors: Lorenzo Amici, Marina Ranghetti, Marta Rossi, Yinyao Zhang
Version: 1.0
Date: 29-April-2019
Download page: https://github.com/MarinaZen/SoftwareEng.git
Copyright: Copyrightc 2017, Lorenzo Amici, Marina Ranghetti, Marta Rossi, Yinyao Zhang – All rights reserved

Contents

1.Introduction

The goal of this project is to develop a web-app for managing bike-sharing's service information.

Bike-sharing is a short-term bicycle rental service available in different urban locations characterized by bicycle transit.

The service is accessible through applied technology (smart cards and/or mobile phone). In particular, it offers 24h/7 service.

The development of this web application has been commissioned from Comune di Milano that will eventually release privileged permissions to obtain statistical information and other utilities. In the following a more precise description is provided.

1.1World Phenomena

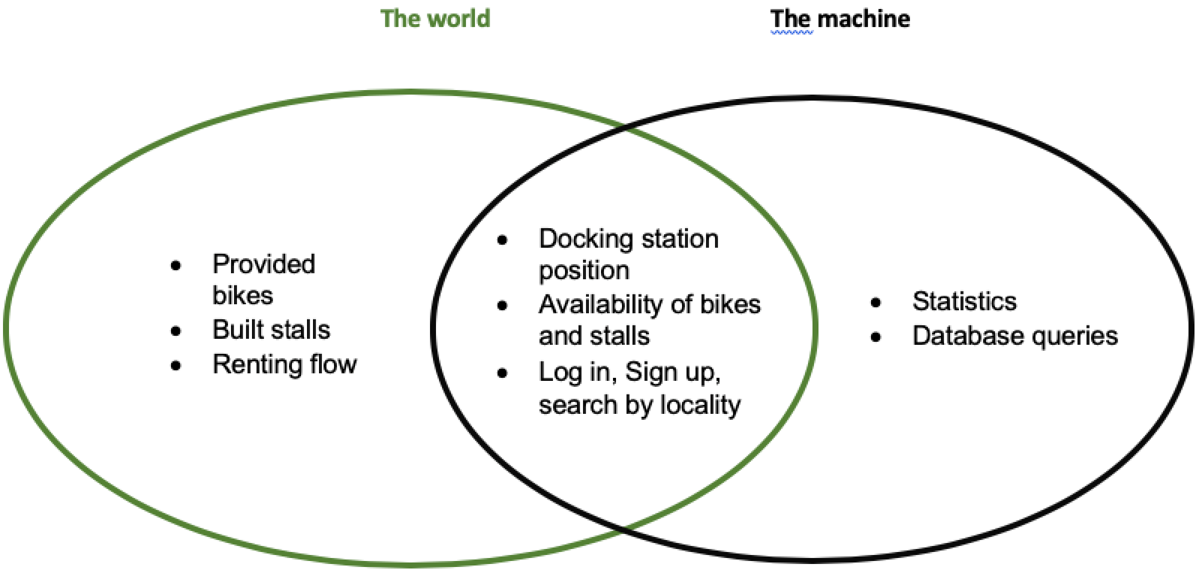


Fig.1: Wold and Machine Phenomena Scheme

World Phenomena:

- **Provided Bikes**

  Comune di Milano has provided a sufficient number of bikes to satisfy the necessity of the citizen.

- **Built Stalls**

  Comune di Milano has provided a sufficient number of stalls in strategic area of the city to satisfy according to the number of bikes .

- **Renting Flow**

  Movement of bikes in each station during the day.

Shared Phenomena:

- **Docking Station Position**

  Exact position in a specific reference system of the docking station.

- **Availability of bikes and stalls**

  The movement of bikes in each station during the day produces a different availability of bikes and stalls in every docking stations.

- **Log In, Sign Up, Search by attribute**

  Procedures the users can apply, particularly register to our app filling each field with his data and then log in.
  Once open the app he can search each docking station by name of the locality.

Phenomena located entirely in the machine:

- **Statistics**

  Having the flow of bikes in each stations during the day, it is possible to provide a full vision of this movement.

- **Database Queries**

  Retrieving data from the database (post, user data, bikes data etc...).

1.2 Scenario

Bob is common user. He wants to hire a bike for going from position A to position B. Particularly, he wants to discover if there are some docking stations near him with available bikes. Lastly when he will reach the destination he would be sure that there are docking stations with free stalls near his position.

Alice is a registered user. She has recently used the bike-sharing app and would like to leave a comment about the service. Moreover, she wants to check the bike flows of her usual station.

John heard about this application and would like to join as a new registered user.

1.3 Use Cases

Visualization app content

Actors:

- Common User
- Customer

Entry condition:

- Information about the app and bike sharing service

Flow of events:

- About/Home page section

- Reading description and aim of the app

Exit condition:

- Closing the web app

Exceptions:

- App crash

Reading users' posts

Actors:

- Common User
- Customer

Entry condition:

- Reading posts

Flow of events:

- Blog section
- Reading posts (from the recent)

Exit condition:

- Closing the web app

Exceptions:

- App crash

Visualization number of free bikes/stalls

Actors:

- Common User
- Customer
- GPS
- Map service

Entry condition:

- Pick up/drop off a bike

Flow of events:

- Map page section
- (optional) Update of initial position
- Visualization of the nearest docking stations
- Pop-up with information of the station (bikes,stalls)
- (optional) Insert input and final destination.
- (optional) Navigation to the final destination

Exit condition:

- Closing the web app

Exceptions:

- GPS not working
- App crash

THE FOLLOWING USE CASES ARE RELATED ONLY TO A REGISTERED USER

Registration

Actors:

- Customer

Entry condition:

- Filling registration form

Flow of events:

- Username (unique)
- Mail (unique)
- Confirm mail
- Password (length condition)
- Confirm password
- Check PIN (provide by Comune di Milano)

Exit condition:

- Closing app
- Submit the form

Exceptions:

- Warning alert if user has already registered
- Warning alert for each field wrongly compiled
- App crash

Login

Actors:

- Customer

Entry condition:

- Filling login form

Flow of events:

- User mail
- User password
- (optional) Remember me

Exit condition:

- Closing app
- Log In

Exceptions:

- Warning alert if user doesn’t have registered

- Warning alert for each field wrongly compiled
- App crash

Retrieve password

Actors:

- Customer

Entry condition:

- Forgot password during login

Flow of events:

- Click forgot password
- Insert email of registration
- Check mailbox for assistance email
- Click link on the email
- Insert new password
- Confirm new password

Exit condition:

- Closing app
- Submit new password

Exceptions:

- Warning alert if user doesn't have registered
- Warning alert for each field wrongly compiled
- App crash

Updating Profile

Actors:

- Customer

Entry condition:

- Change profile settings

Flow of events:

- Login precondition
- Account section
- (optional) change username
- (optional) change email
- (optional) update profile picture

Exit condition:

- Closing app
- Submit
- Change page

Exceptions:

- Warning alert if username is not unique
- Warning alert if email is not unique
- App crash

Adding posts

Actors:

- Customer

Entry condition:

- Creating a new post

Flow of events:

- Login precondition
- New post section
- Insert title
- Insert content

Exit condition:

- Closing app
- Submit
- Change page

Exceptions:

- Warning alert if no title inserted
- Warning alert if no content inserted
- App crash

Updating posts

Actors:

- Customer

Entry condition:

- Change content or title of a post

Flow of events:

- Login precondition
- New post precondition
- Blog section
- Click on your own post
- Click update button
- (optional) Insert new title
- (optional) Insert new title

Exit condition:

- Closing app
- Post changement

- Change page

Exceptions:

- Warning alert if no title inserted (empty field)
- Warning alert if no content inserted (empty field)
- Warning alert if not customer's post
- App crash

Deleting posts

Actors:

- Customer

Entry condition:

- Deleting a post

Flow of events:

- Login precondition
- New post precondition
- Blog section
- Click on your own post
- Click delete button
- (optional) Confirm deleting
- (optional) Cancel

Exit condition:

- Closing app
- Submit form
- Change page

Exceptions:

- Warning alert if not customer's post
- App crash

Displaying statistical information

Actors:

- Customer
- GPS
- Map service

Entry condition:

- Visualize statistics of the docking stations

Flow of events:

- Login precondition
- Map section
- Clicking a station
- Pop-up with information of the station (bikes, stalls)

- Click statistic button
- Statistical information (to define) in a new page

Exit condition:

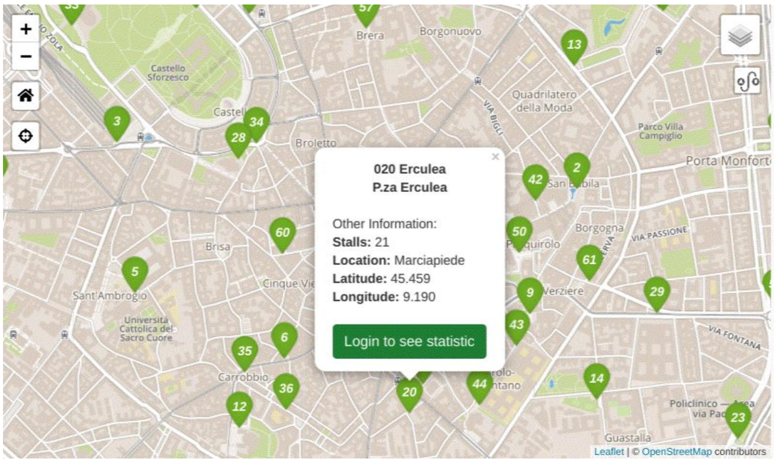
- Closing app

Exceptions:

- Login button instead of statistic button if customer is not logged in
- App crash

2.Overall description

Here is the description of the functionalities of our web-app.
First it is necessary to distinguish between users that are/are not registered.
Both can visualize a map with all the docking stations: by selecting one of them the user can come to know the station's exact position, the number of available bikes and stalls and other information.



020 Erculea
P.za Erculea

Other Information:
Stalls: 21
Location: Marciapiede
Latitude: 45.459
Longitude: 9.190

Fig.2: Overview of the map embedded in the web application.

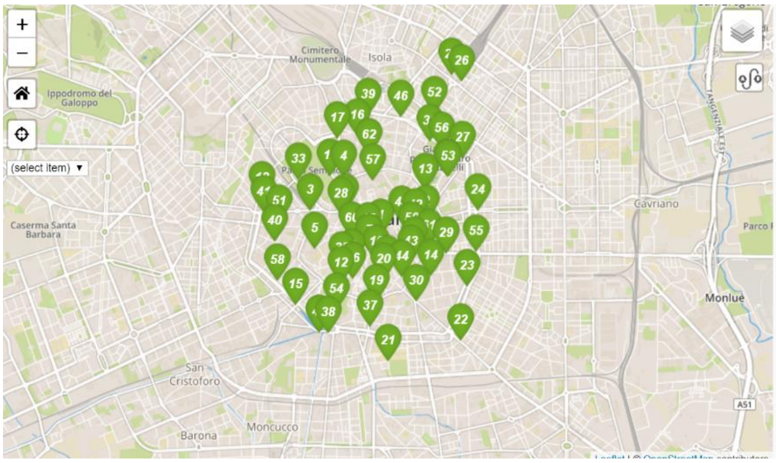


Fig.3: Example of a popup displaying information of a station.

The registered user will be able to and participate to the blog and visualize statistical information. In particular, an interactive real-time graph representing the flow of bikes' utilization. Those graphs can be customized in a period of time that the user can specify. The user will also be able to add preferences to specific stations in order to facilitate future researches.

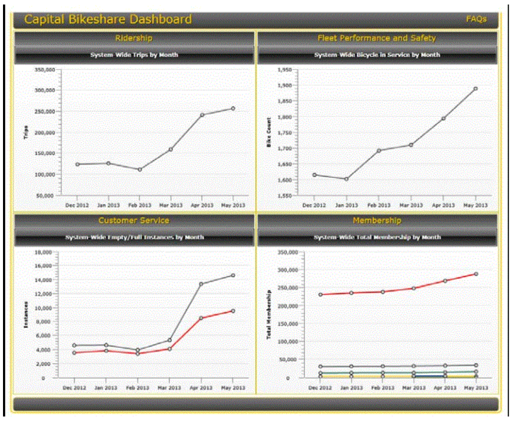


Fig.3: Example statistic of a station.

3. Specific requirements

3.1 Technical requirements

The system should be implemented in:

- Python
- HTML, CSS
- Easy interface
- English and Italian languages
- Software for managing databases

3.2 Non-functional requirements

- The system should be available 24h/7
- The system should be updated in real time
- Supportability on mobile phones

3.3 Functional requirements

- The system shall allow multiple users to access a same service at the same time
- The system shall allow users to access the basic services (availability of bike and stalls) without any registration needed
- The system shall allow registered users to enter the privileged area
- The system shall allow registered user to see statistics about bike movements

4. Effort Spent

Marina Ranghetti: RASD and web application (building website and map)
Marta Rossi: RASD and web application (building website)
Lorenzo Amici: First draft of the RASD
Yinyao Zhang: Support