

# GARBAGE DISPOSAL INFORMATION SYSTEM

Team 3

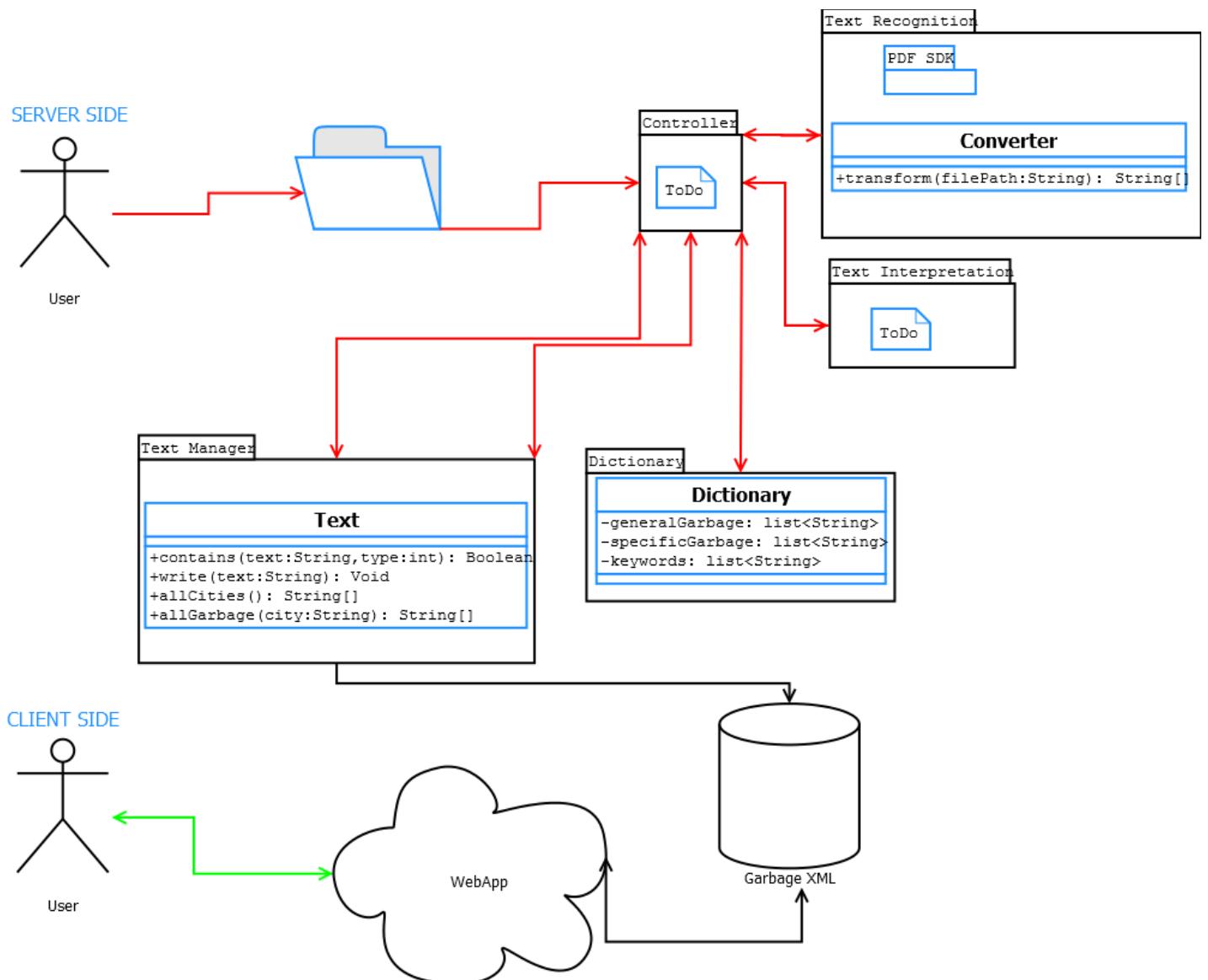
Christian Dallago  
Stanko Franz Ovkacic

- Bolzano and other cities enforce garbage separations through different guidelines that indicate how to dispose each kind of garbage.
- At times, it is not clear as to what is the container to use (for instance, in Bolzano even though an eggshell is organic, it should be disposed in the regular garbage).
- Furthermore, such regulations are not the same from city to city, so users may as well get confused about what bin to use.
- We need to create a software system that, giving a certain city and certain kind of product, determines what is the proper bin to dispose it.

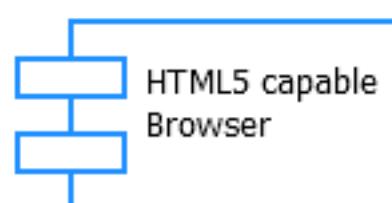
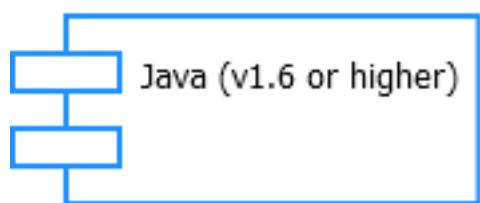
# Table of contents

page 3~8	<b>4+1 Views</b>	page 9	<b>Sprint 1 revision</b>
page 10	<b>Sprint 2 revision</b>	page 11	<b>Sprint 3 revision</b>
page 12	<b>Sprint 4 revision</b>	page 13	<b>Sprint 5 revision</b>
page 14	<b>Sprint 6 revision</b>	page 15	<b>Software Proposal reviewd</b>
page 16	<b>Class &amp; Package diagrams</b>	page 17	<b>Deployment &amp; Use Case diagrams</b>
page 18	<b>Sequence diagrams</b>	page 19~20	<b>Client Design</b>
page 21	<b>Administrator Design</b>	page 22	<b>User Stories</b>

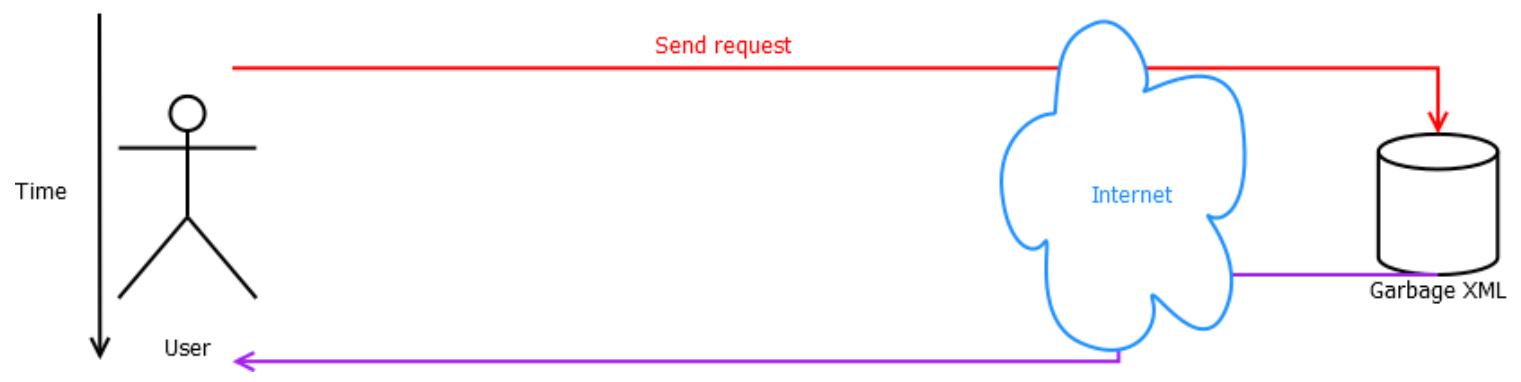
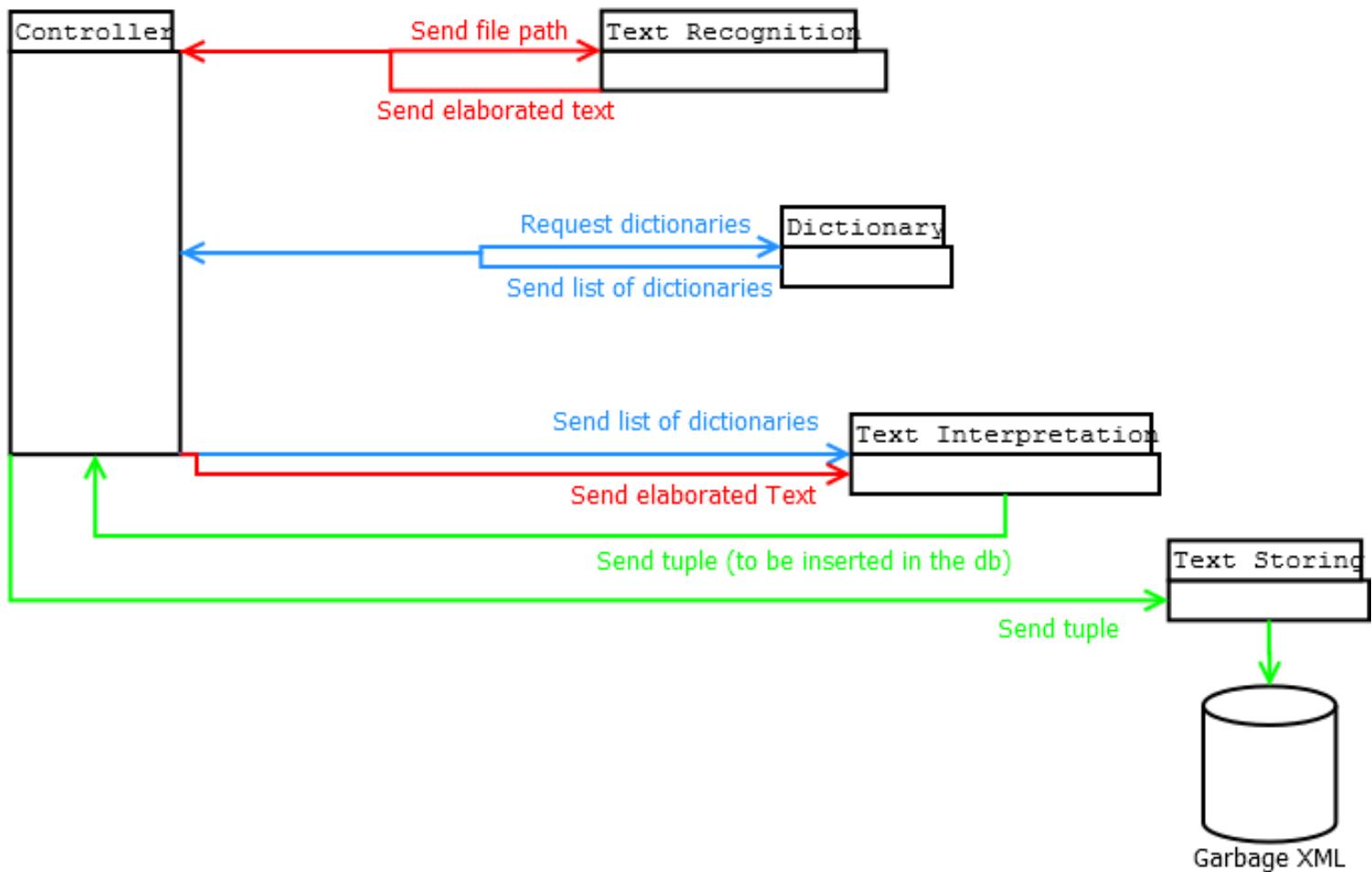
# Class & Package Diagram



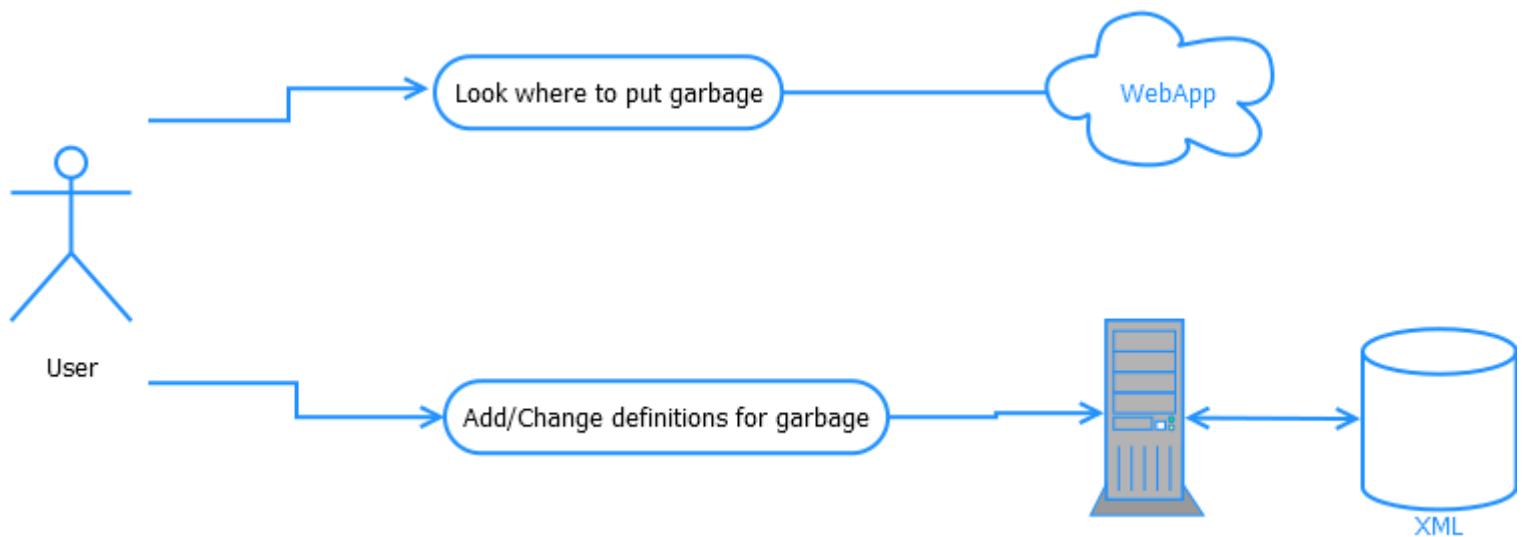
# Deployment diagram



# Sequence Diagram



# Use Case Diagram



# Client design Desktop/Tablet

Pare tu sia a : Bozen, 39100.  
Quale tipo di Rifiuto?  Cerca

Search for some item!

GreeniFy [Home](#) [About](#) [Lista di città](#)

# Mobile

Pare tu sia a : Bolzano - Bozen, 39100.

Carta  
39100  
Cerca

## Carta A Bolzano

L'oggetto Carta di tipo Carta, a Bolzano si butta nel cassetto di colore Arancio

**Home**  
**Lista di città**  
**About**

# Use Cases

As a user I would like to know where to throw plastic So that I can throw a plastic bottle in the right place.

As a user I would like to know how to classify egg shells So that I can differentiate them correctly.

As a user I would like to have access to local recycling policies So that I can differentiate my waste or to know policies of a particular location.

As a user I would like to know what kind of waste a certain bin-color corresponds to So that I can identify bins better.

# First sprint revision

After dicussing the first project draft, we were told to implement the system in such a way that a user could interact with it through a mobile device.

The following steps have been redesigning the diagrams and thinking about the implementation, which at first, was though as two different projects: The mobile application and the server-side application.

The mobile application would allow the user to view information about garbage disposal *in it's (a) current - automatically retrieved - location*, offering then (b) a searching mechanism to know where to dispose a specific item, (c) a consulting mechanism which offers the user the possibility to have a list of items and their correct disposal. The application would also allow the user to (d) *change to a specific - user requested - location*, in order to gather information about a different disposal system of a different city.

The server side, instead, would allow the specific entity of the city, responsible for garbage differentiation, to edit and create new definitions for the city and it's disposal system.

The two separated projects had to interact between themeselves trough a common entity which was identified by a database or similar.

After this new guidelines were identified, focus was put on implementing first the -what appears to be more feasible- client-side, to then have enough time to implement the text retrieval of the server-side.

The idea of a specific Android app seemed the most appealing at first, but due to concerns of time, scalability, and app-market policies, we decided to take a wider approach, implementing the client-side in a more efficient and scalable way.

The new idea has therefore grown towards implementing the client-side as a web-application, allowing to bypass any limitation imposed by mobile-software manufacturers.

Using this approach *we are confident that within sprint 2*, we will conclude the client-side of the application, leaving a considerable amount of time to test and implement the server-side and the text recognition, which, since project start, have been our biggest concerns.

We agree that by whisely using a combination of HTML5, CSS3 and JavaScript, we can meet all the previously stated points for the andorid-app development (see points a-d above). HTML5 introduces, in fact, geolocation retrieval, which is a key element to this development. It will allow to get the location *even from desktop users, or from any-kind of mobile user* (since the HTML standard 5 has been implemented in every smartphone mobile browser). CSS3 will allow us to easily model different usage scenarios (use on phone, use on tablet or on desktop), and JavaScript will give us the flexibility we need to write this web-app (and is also supported by every modern mobile/desktop browser)

We also have deeply analized the different "interface entities" which could be used to "speak" between the server-side and the client-side, and we agree that the idea of an XML is the best, for scalability issues and simplicity of use/understanding. An XML would allow us a seamless interfacing with the web-app, while still delivering the content of interest and all the information needed.

We have therefore identified the objectives for sprint two, which are:

- 1) Creation of the XML rules and creation of a XML test file
- 2) Creation of the web-app itself (CSS3 for desktop and mobile, HTML5 structure and JavaScripts)
- 3) Deployment of the web-app and the service on a custom server @ <http://uni.christiandallago.com>

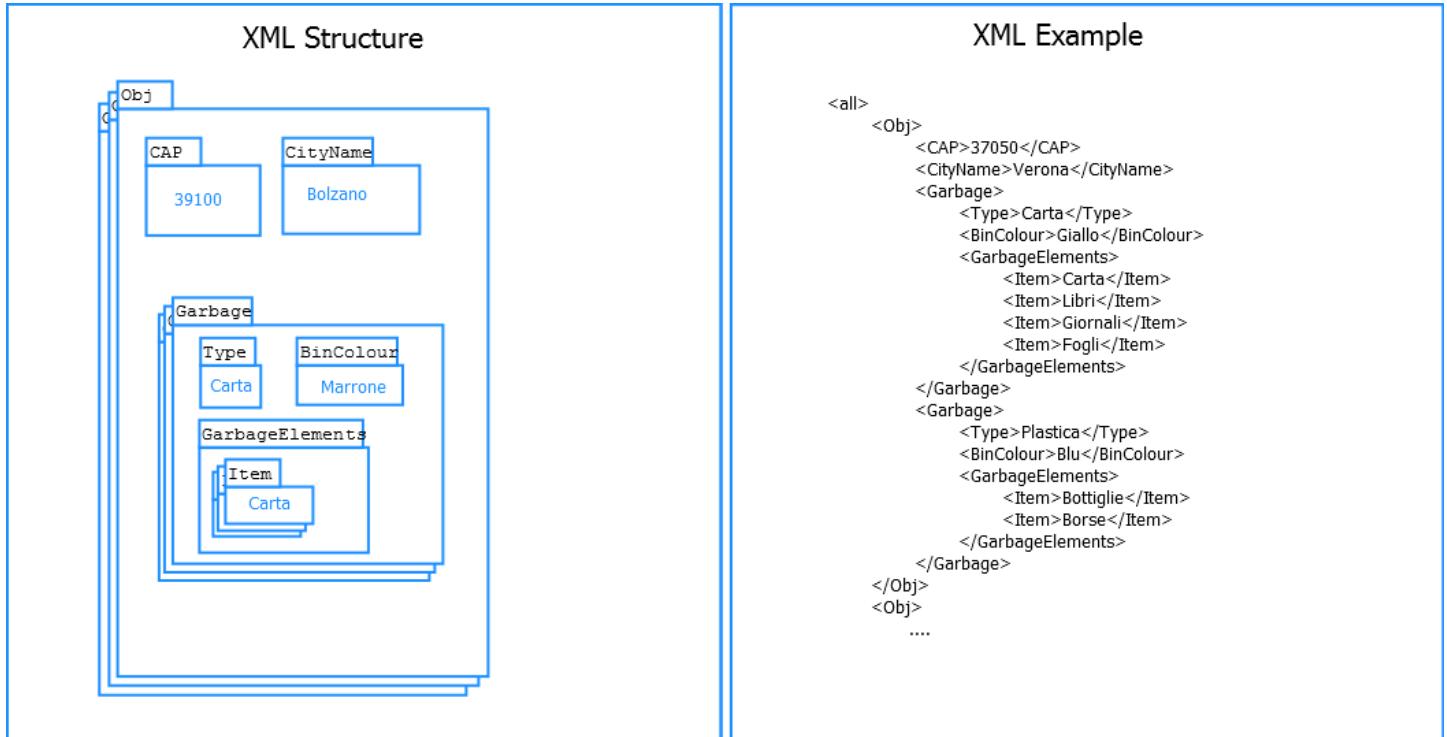
We are confident that the web-app will be available by the end of sprint 2.

# Second sprint revision

During the last week, we have, as planned, developed the fornt-end web app, which is multiplatform compatible with desktop, tablet and mobile devices.

The design of the web page has been accurately studied, and we decided to take a 'gamification' approach, in order to stimulate children in using the service as well as giving an adult public a pleasant experience.

We also defined the XML structure as well as all functions to access, query it, also.



The webservice is already available at <http://uni.christiandallago.com>

The next steps consist in writing the XML handler class (and the text-storing responsible package), part of the controller package (responsible for interaction between GUI and text-storing) and the server-part GUI.

# Third sprint revision

This week we have worked on defining the XML structure in terms of Java Classes and Objects.

We have created the City class, the ftp uploader (which we will use to update the XML), the xml parser, which parses the XML to then create a (to-do) tree, which will contain an ordered list of cities, and a utility class, which will standardize the user inputted/automatically generated new definitions.

We have experienced some problems with the working environments (one had java 1.6, the other one was using 1.7) and therefore with the compatibility of the SDK's. Finally we were able to solve our problems, working both with Java 1.7.

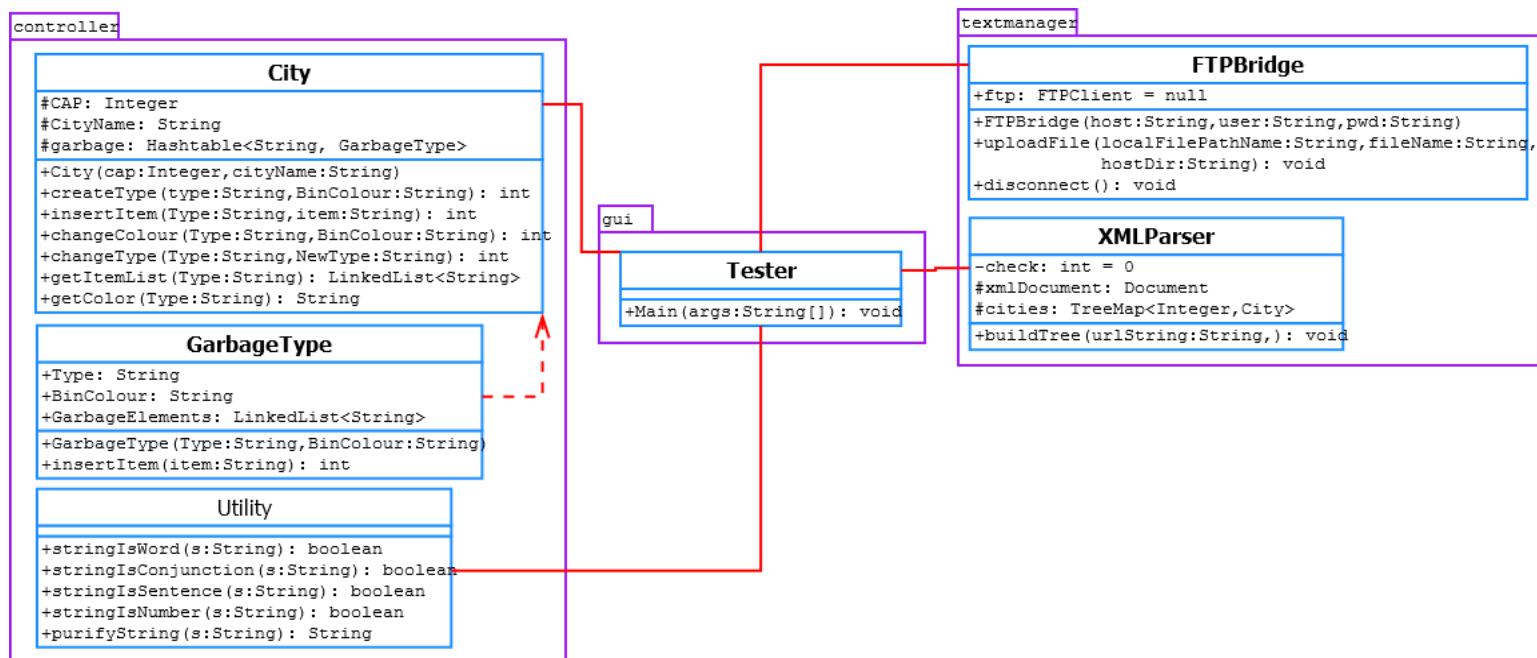
We were not able to set up prominator to take measurements (although we have both the framework and the eclipse-plugin installed, and although we have followed the instructions on the user guide), but we used the webpage to keep track of our work.

We have not been working on the GUI so far, nor on the dictionary or the definition retrieval from documents.

We identified the next steps as being:

- 1) Implementing the construction of the tree of cities from the XML "document" (as called from the SDK)
- 2) Implementing the mechanism to store the tree of cities into an XML file and then updating the online XML file via FTP
- 3) Construct the GUI anche the possibility to create (a) new definitins for a city and (b) update definitions for a city

At the current development stage, we have the following classes, atrtributes and methods up and running.



# Fourth sprint revision

This week we have coped with major problems with the dom4j libraries.

We have spent most of the week trying to solve problems related to the representation of XML elements through the library and the way these elements can be handled.

Finally we were able to retrieve information and write new definitions in a sorted fashion to the XML.

This has unfortunately led to a delay in the creation of the GUI, which at this stage we consider as being the next focus point.

The coming sprint we will focus on the GUI, on the correctness of the code (refactoring), on the documentation (JavaDocs, diagrams) and on testing what has been implemented so far but still not been fully tested.

If all of these steps are to be completed ahead of time, we will begin with the definition of a dictionary and a natural language interpreter.

# Fifth sprint revision

As planned, we have created the GUI, the test cases (through JUnit), refined our code and documented it. We have tested the City, GarbageType, Definition and FTPUpload classes.

We have also updated the webpage scripts and created a page to list all cities defined in the XML.

On the creation of the GUI we have coped with some problems in the design, which we wanted to unify as much as possible. The outcome is a dynamic design which we will present during the sprint revision at the lab.

Goals for next week are (in order of importance):

- investigate on information retrieval from pdf's/xml's and try to implement such a mechanism.
- refactor the webpage and this document, updating diagrams and illustrating the design and functionalities of the GUI
- document the GUI and refactor the code of the GUI.

# Sixth and last sprint revision

This last sprint we have slightly changed our objectives with respect to what written in the past sprint revision.

We have agreed that at this stage of the development, it would be better to focus on refining the last details of the working parts of the project, rather than to waste energies on implementing new parts.

We have therefore rewritten most of the webpages, the scripts and styles, in order to offer a simple and coherent experience between desktop and mobile users. We have significantly reduced the amount of code and therefore made the pages load faster and better maintainable.

We have structured the scripts in such a way that every script performs an action and each action can be performed before page load or after page load (for example, loading the css is done before, but checking the position of the user, which might take some time, is done after, so the user can already utilize the page).

We have also implemented autocomplete mechanisms in the search fields, which will match every CAP number defined and every substring of every string of all items in every city and every garbage super type.

We have then worked on the server side by:

- 1) Documenting the code.
- 2) Refactoring the GUI
- 3) Adding checks at the GUI level for inserted strings and warnings if the insertion of something is done incorrectly.
- 4) Modified the GUI such that it would fit on screens of different platforms.

We have also produced our poster and rewritten the documentation, and following this page we will insert software proposal, diagrams and screenshots of the finished work.

# Software Proposal reviewed

## Requirements

- 1) Bolzano and other cities enforce garbage separations through different guidelines that indicate how to dispose each kind of garbage.
- 2) At times, it is not clear as to what is the container to use (for instance, in Bolzano even though an eggshell is organic, it should be disposed in the regular garbage).
- 3) Furthermore, such regulations are not the same from city to city, so users may as well get confused about what bin to use.
- 4) We need to create a software system that, giving a certain city and certain kind of product, determines what is the proper bin to dispose it.

## Does our software meet the requirements?

Our software is composed by two parts:

- The user part, which is concerned about the end user of the application, therefore the homeowner or tourist in city A, that wants to know where to correctly dispose a certain kind of item.
- The administrative part, which is concerned about creating the correct definitions for the users.

These two parts also come with two different implementations:

- The first trough a web application written in HTML5, CSS3, JavaScript and JQuery, therefore available for every device, be it computer, smartphone, tablet or smart-TV.
- The second one as Java Application (.jar), therefore cross platform and it gives a more concrete sense of security and reliability to the entities which are responsible to create definitions for garbage disposal.

These two parts communicate trough an XML file which contains the definitions.

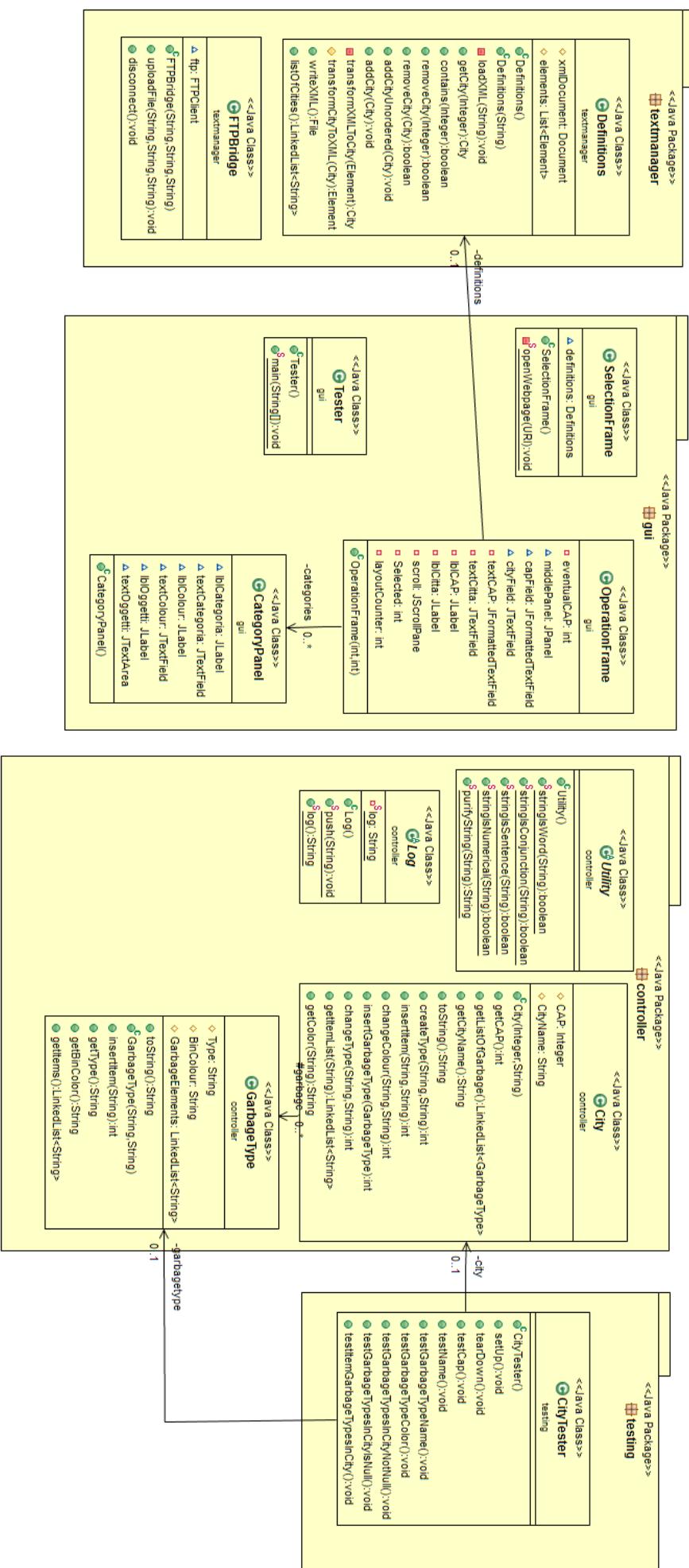
The main goal of the project has therefore been met, giving a user the possibility to look up, given its position and the garbage he/she wants to dispose, where this item gets disposed to.

The webservice offers also automatic location retrieval, which can facilitate the user in finding out the CAP of the city he/she finds him/herself in.

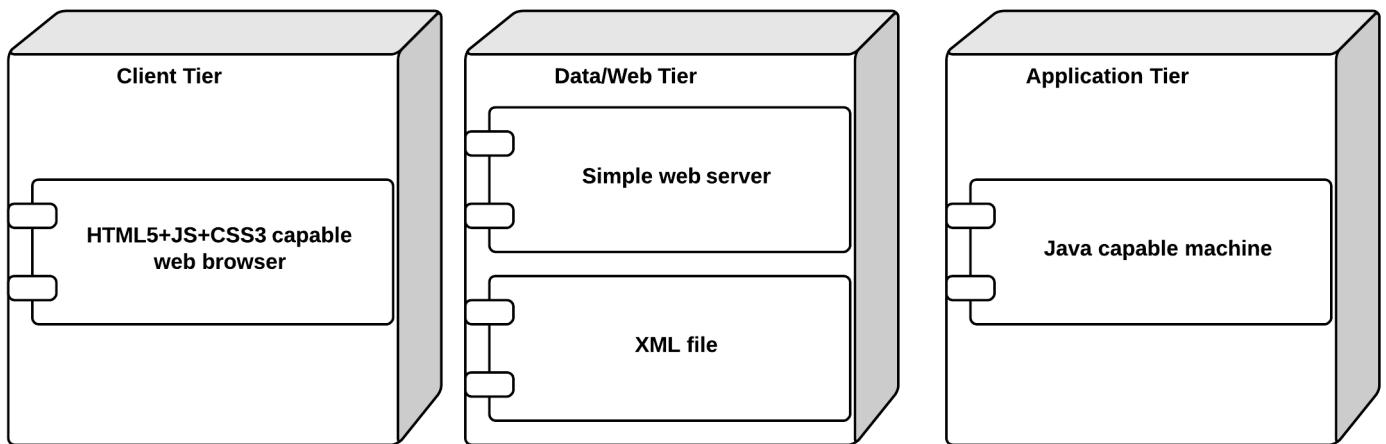
There is also a webpage listing all tuples of city-CAP, giving therefore a match for both name and CAP to the users.

The GUI of the end-user webpage is thought as a gamification of the problem, giving a lighter approach to the issue and bringing the webpage closer to younger citizens.

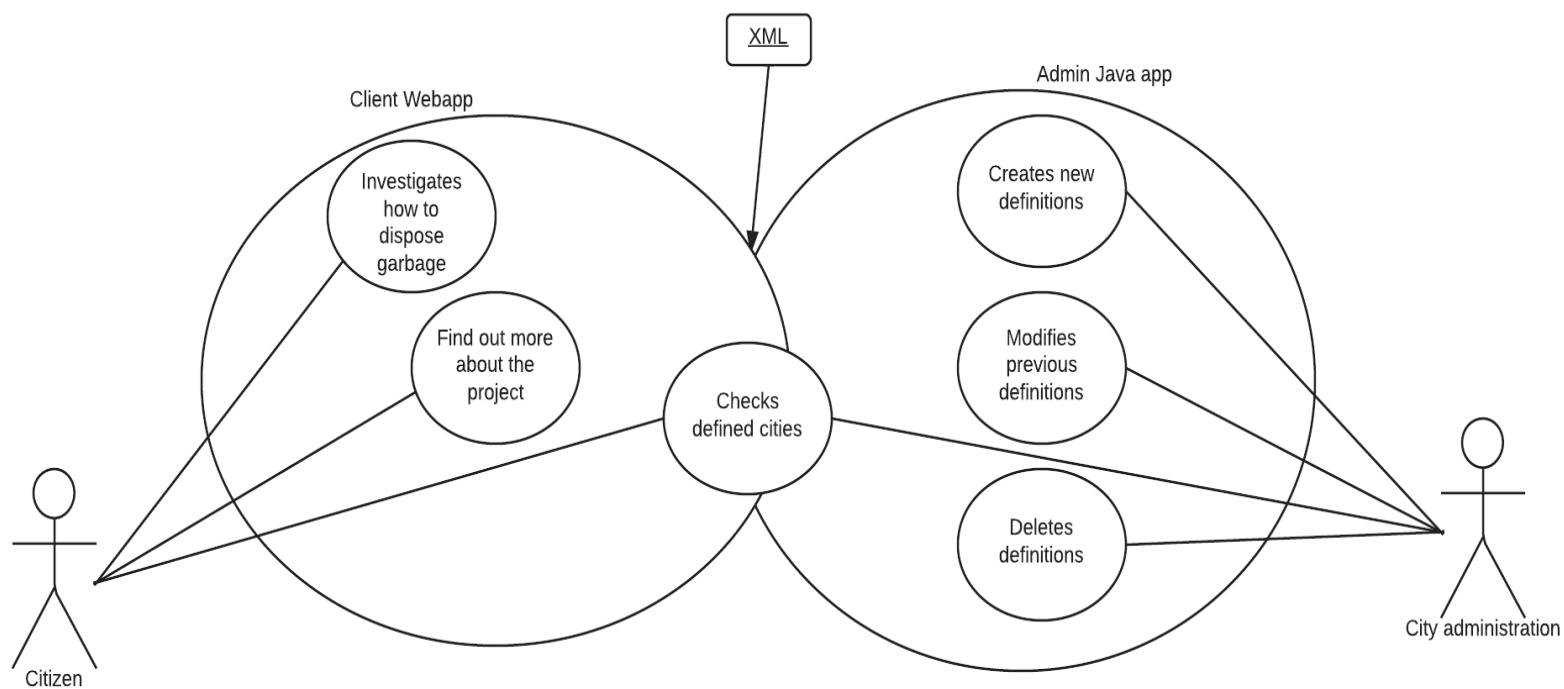
# Class & Package Diagram



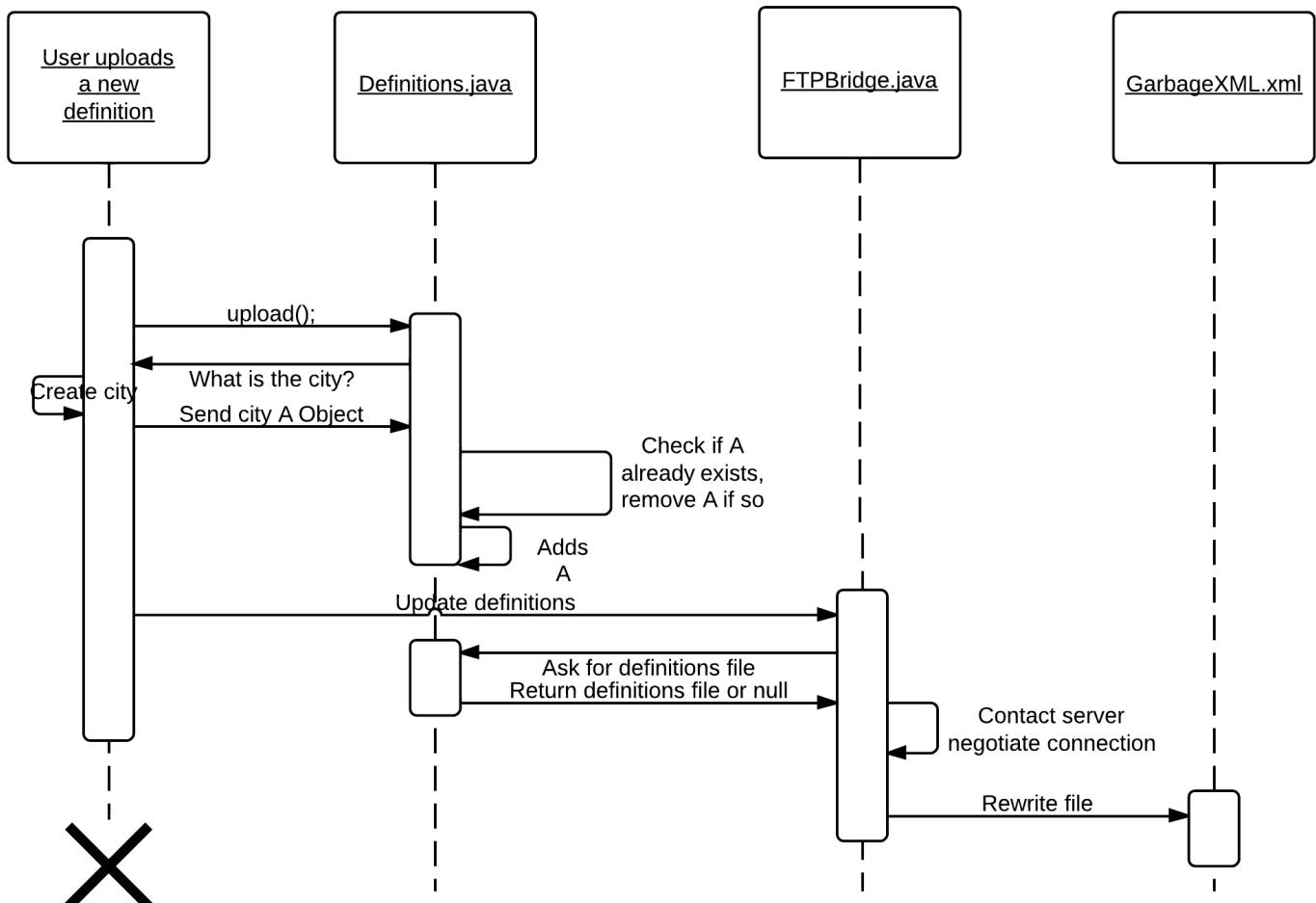
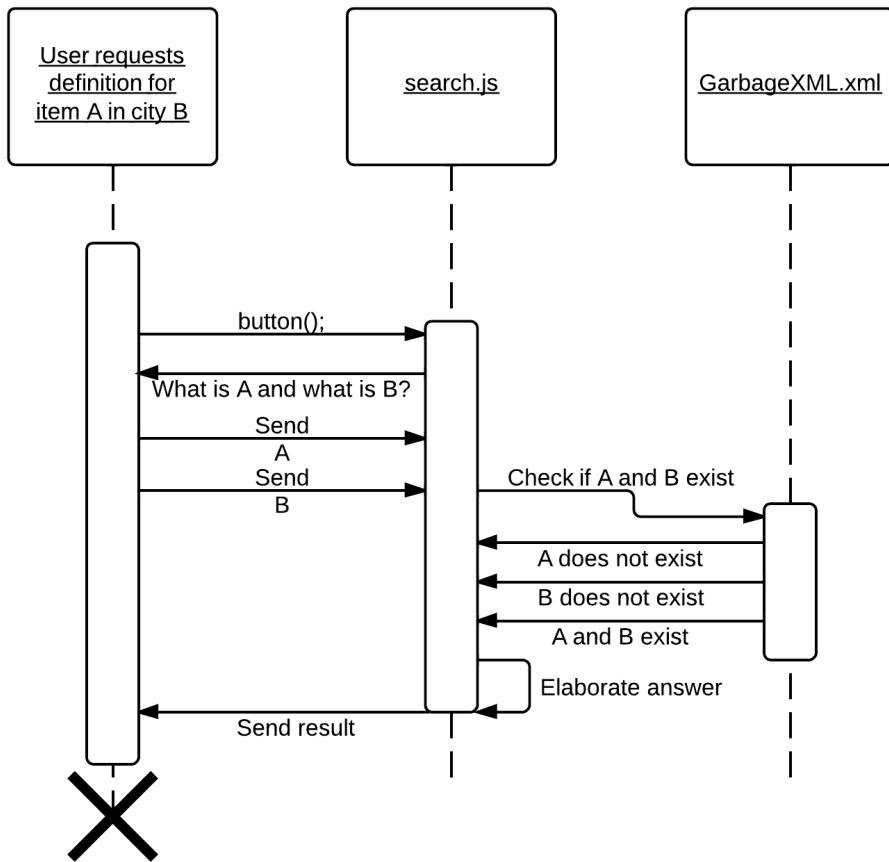
# Deployment diagram



# Use Case diagram

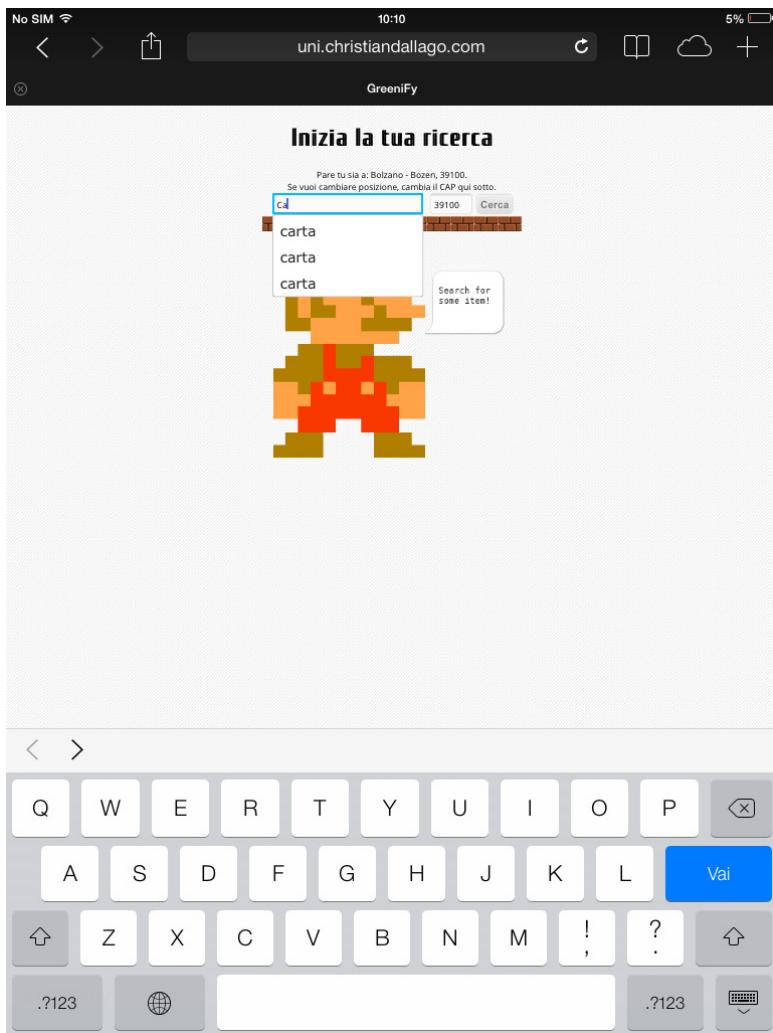


# Sequence diagram



# Design on mobile devices

Tablet (ipad, safari)

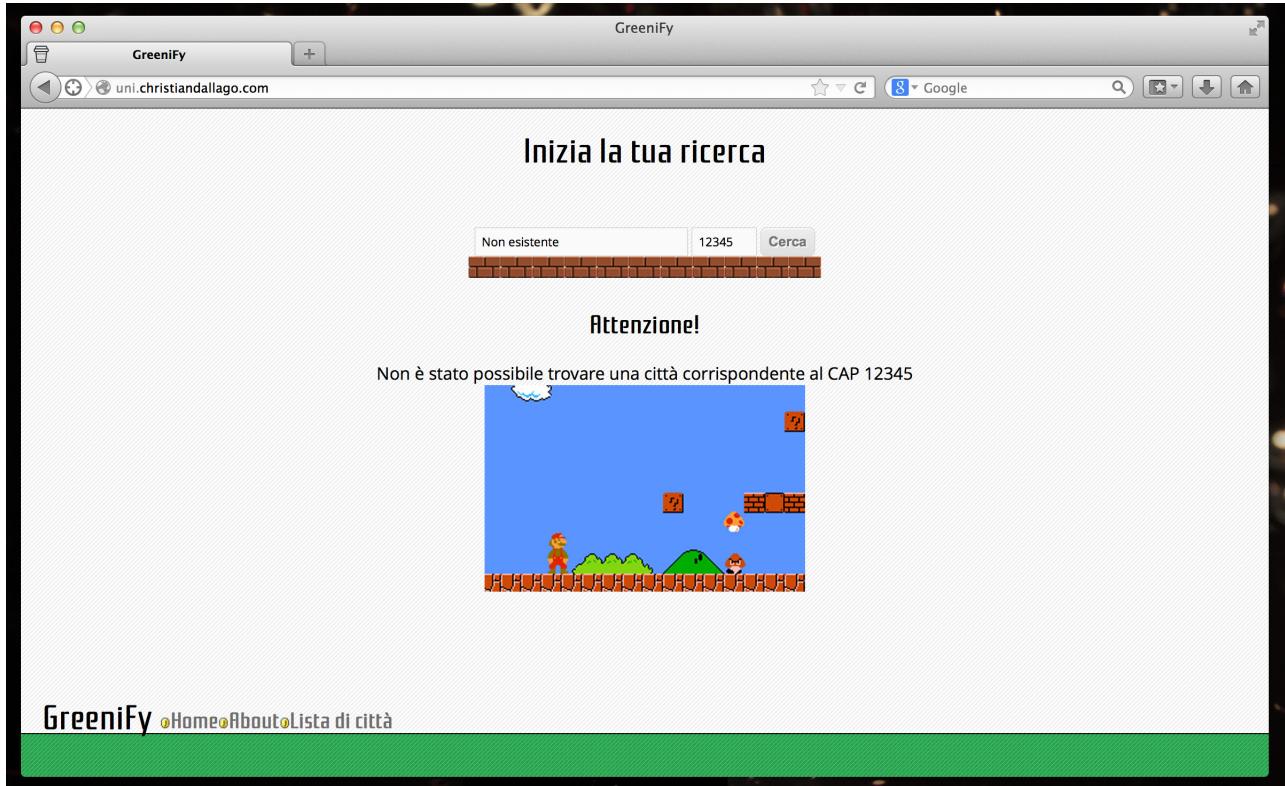


Smartphone (nexus 5, chrome)

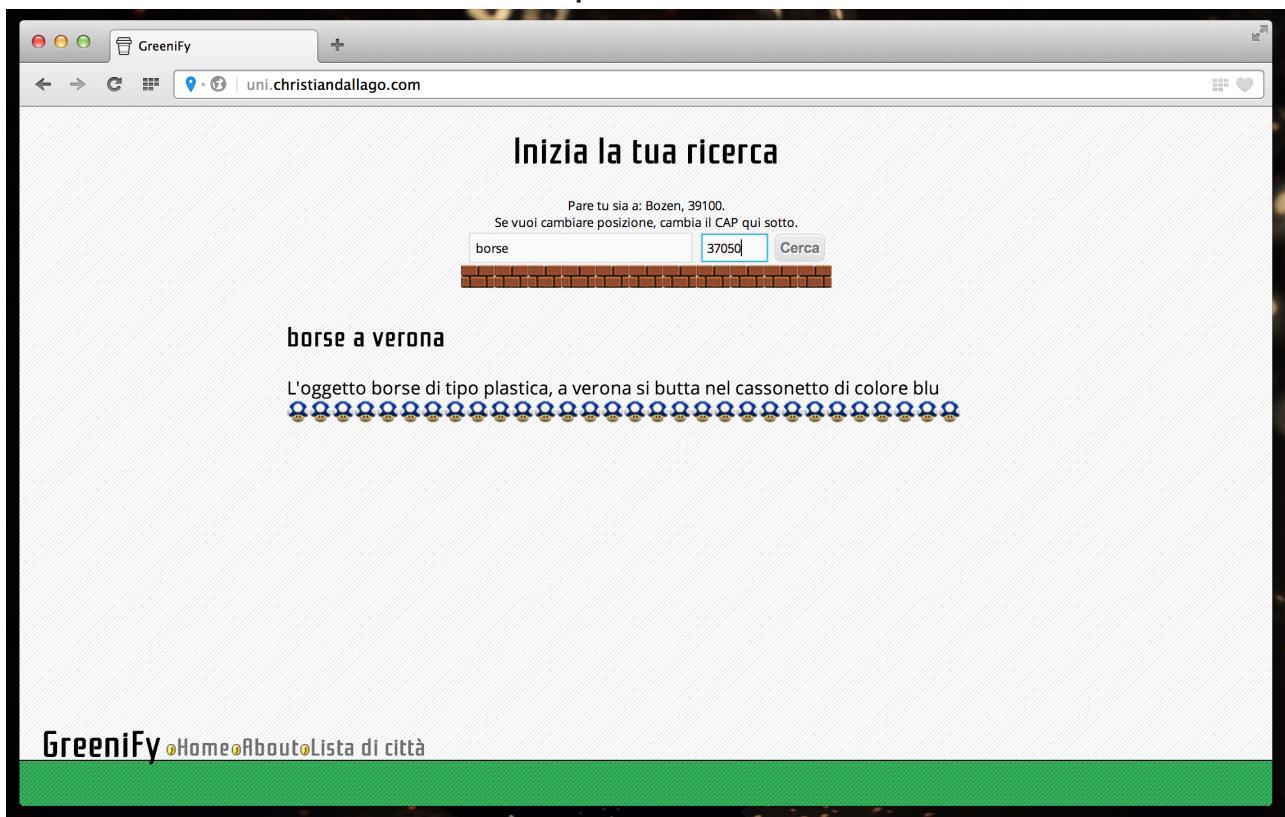


# Design on computer

Firefox



Opera next



# Server side design on computer



A screenshot of a software window titled "Inserimento di una nuova città". The window is labeled "Options" at the top. It contains several input fields and buttons. At the top left are fields for "CAP" and "Città", each with an associated input box. Below these are fields for "Categoria" and "Colore", also with input boxes. A larger input box labeled "Oggetti" contains the text "i.e. carne;frutta;verdura". At the bottom of the window are two buttons: "Nuova Categoria" on the left and "Aggiungi questa Citta" on the right.

# User Stories

As an administrator user I would like to add a new city garbage definition to the database.

As an administrator user I would like to modify the already existing garbage definitions of Bolzano.

As an administrator user I would like to completely eliminate a wrong city definition.

As an administrator user I would like to know who is responsible for the Java deskapp.

As a user/administrator I would like to see the cities available in the system.

As a user I would like to know where to throw an item in the town I am in.

As a user I would like to know where to throw an item in a certain town of my choice.