Unimojo : An university database and analytics tool

Undergraduate Thesis

Michaël Gerber

HEIG-VD - School of Business and Engineering, Vaud, Switzerland

Professor : **Stephan Robert**

February 11, 2011

Contents

1 Introduction

This document is an overview of the work done on the Unimojo project. It was done at IBM Research Almaden (San José, CA) from July to December 2010 in order to accomplish my undergraduate thesis for the HEIG-VD. This work will complete my studies in telecommunications. Unimojo is a Web application that displays relevant information regarding universities and their surrounding in a way to simplify the search of information.
The idea of this project came from the fact that nowadays there is more and more interaction between universities and business. We need to know which university will be the next one but with the globalization it's hard to be aware of every new emerging universities in the world. Now every university has a chance to grow, new technologies can be developed everywhere and not only in large universities. Unimojo tries to answer to this problem by displaying the most useful facts.

1.1 Goals of the project

The Goal of this project is to complete a major work in the field of telecommunications. We have to use skills learned during studies to deal with the implementation of the project, the redaction of the report and the presentation.

1.2 State of the Art

Many data are available regarding universities but you need to consult several sources of information. If you want to know the ranking of a university you can found it on one of the websites of the Shanghai university [1] but if you want to obtain information on the university country you have to search on another website like for example the International Monetary Fund [5]. There are not many websites that gather all this information. WolframAlpha [27] may be one of the few good examples.

2 Project planning

2.1 Project management

The project was separated into five distinct parts.

1. Data gathering and creation of the database. The database is the base of the project, if there's a design error it will be passed on to the next steps and some difficulties will appear throughout the application development.

2. Implementation of the website functions, as written, is based on the database. This is the part that will display and format the relevant information to the user.

3. Creation of a nice GUI (Graphical User Interface) will determine the quality of interaction for the user. This step is important because it gives the appearance to the work done in the two first parts. With an unusable interface the application loses his appeal.

4. Preparation of the presentation shows in a short way the work that has been done during the development of the website.

5. Writing the report in order to show results of the project with more details than the presentation. Every details must appear in it.

2.2 Organisation

For the project we have used a tool called SVN (Apache Subversion [10]) that is a software versioning and a revision control system. Its advantages are the backup of the sources, the possibility of resuming an old version and the possibility to work with several people on the same project.

3 Technology

In this part of the report we will discuss about technologies used to develop the application. We will try to highlight the pros and cons.

3.1 Eclipse

Figure 1: Eclipse logo[1]

The choice of Eclipse [7] was done because it allows integration of the followings tools : Google Web Toolkit (See 3.2) and Google App Engine (See Section 3.3) which will be discussed later in this report. Eclipse is a free development environment, extensible by the plugin system which is the foundation of its architecture.

Version :

- Stable version 3.6.1 Helios, used for the project.
- Preview version 3.7M4.

3.2 Google Web Toolkit

Figure 2: GWT logo[2]

Google Web Toolkit [8] is an open source toolkit used to build complex Web applications. The goal of GWT is to offer a convenient environment to develop application without beeing an expert in the usual technology used to develop a website like JavaScript. The development is done in Java and some basic CSS (Cascading Style Sheets) knowledge. GWT takes care of generating the JavaScript, from the Java code, to be compatible with the most popular Web browsers

[1] http://goo.gl/lIV1q
[2] http://code.google.com/intl/fr/webtoolkit/images/gwt-logo.png

(Internet Explorer, Firefox, Chrome, Safari, Opera, ...). We can develop AJAX (Asynchronous JavaScript and XML) applications that allow to built interactive and dynamic website.

Version :

- Stable version 2.1.1, used for the project.

3.2.1 CSS

Figure 3: CSS3 logo[3]

Cascading Style Sheets [19] is a style language used to describe the presentation of a markup language, for example HTML. It has been used to design the website user interface in order to obtain a usable and enjoyable application. CSS is built upon the last version, adding new features. For this project we implemented CSS1, CSS2 and some properties of CSS3 that is under development for 5 years. We will have a look at the property "border-radius" that is a CSS3 property used to add rounded borders to an element. To add it to an element we have to add followings lines :

```
border-top-left-radius: 15px;
border-top-right-radius: 15px;
```

Those lines add rounded borders to the top of the element. The problem is that the property is compatible only with Opera. For Firefox we have to add thoses lines :

```
-moz-border-radius-topleft: 15px;
-moz-border-radius-topright: 15px;
```

It's compatible with the layout engine of Firefox, Gecko[4], but not with the CSS3 recommendations. The display is correct but the CSS3 validator gives errors.[5] For Safari and Chrome we add :

```
-webkit-border-radius-topleft: 15px;
-webkit-border-radius-topright: 15px;
```

and for Konqueror we add :

```
-khtml-border-radius-topleft: 15px;
-khtml-border-radius-topright: 15px;
```

We will discuss this problem more in detail in thoses parts : 5.3 and 5.4.

[3] http://en.wikipedia.org/wiki/File:Html5_css3_styling.svg
[4] https://developer.mozilla.org/en/Gecko
[5] goo.gl/wxAs3

3.3 Google App Engine

Figure 4: GAE logo[6]

Google App Engine [9] is a platform for developing and hosting web applications on servers of Google. We have used GAE for the database and to host the GWT application.

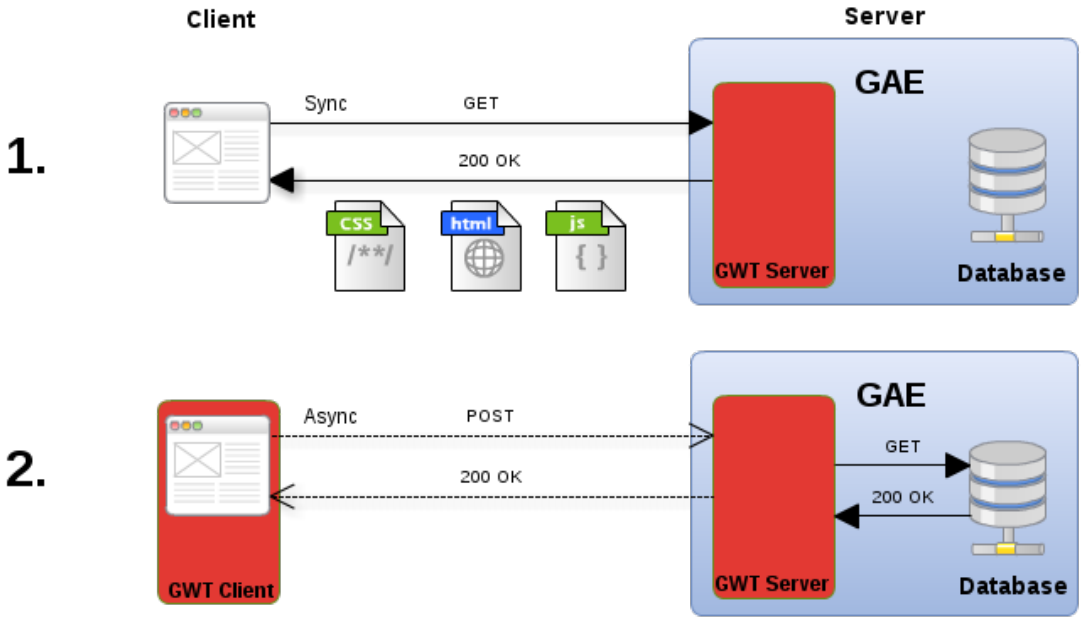


Figure 5: GWT + GAE flow chart

When you want to access the website (See Figure 5) you will type the URL http://unimojo.appspot.com that will send a "GET" request to the server. It will return necessary files to display and interact with the website. In this case we obtain a CSS, HTML and JS file. With these three files we can show the GWT client side in the browser and request data from the GAE database. The first request and those done to the database are synchronous means they block the process concerned because it's waiting on the response. GWT uses for the interaction between the client and server asynchronous requests so the JavaScript doesn't block and the website stays interactive. This means that the client side doesn't wait on the server response and can deal with user events.

[6] http://media.biologeek.com/images/google_appengine.png

Python version :

- Stable version 1.4.1.

Java version :

- Stable version 1.4.0, used for the project.

3.3.1 Java

Figure 6: Java logo[7]

With GAE we have the choice between Python and Java to develop applications. We have chosen to use the programming language Java [6] that is a multi-platform language. That means Java can be implemented on several computing platform like Microsoft Windows or Linux. The choice is justified by the fact that Java is a language known and used during studies at the HEIG-VD and because the GWT part is developed in Java. On anoter side the Java version of GWT is maintained after the Python version. The development is first done for Python and not everything is implemented for Java. Some functionalities have been added 6 months after the release of the Python version. This delay is seen especially in libraries around GWT and not directly in the latter.

Version :

- Stable version 1.6.0_20, used for the project.
- Latest stable version 1.6.0_23.
- Preview version 1.7b116, used in a part of the project (See Section C).
- Latest preview version 1.7b128.[8]

3.4 Libraries

We will discuss about libraries used in the project. They add functionalities to the application like charts or maps.

3.4.1 Google Chart Tools (aka Visualization) 1.1 Library

Google Chart Tools Library [15] enables to add interactive charts to Web page. This library is the adaptation of the Google Chart Tools [17], available in JavaScript, for GWT. We have image charts and interactive charts. We have used these so that the user can see the information he wants.

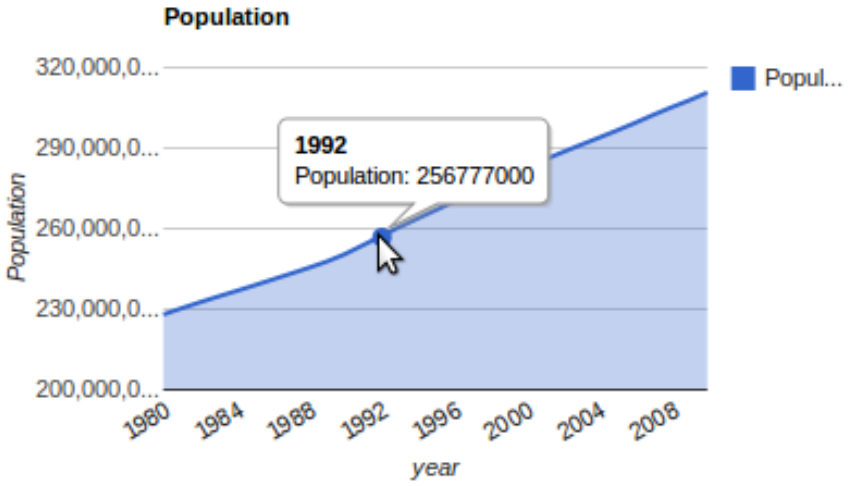


Figure 7: Chart of the population

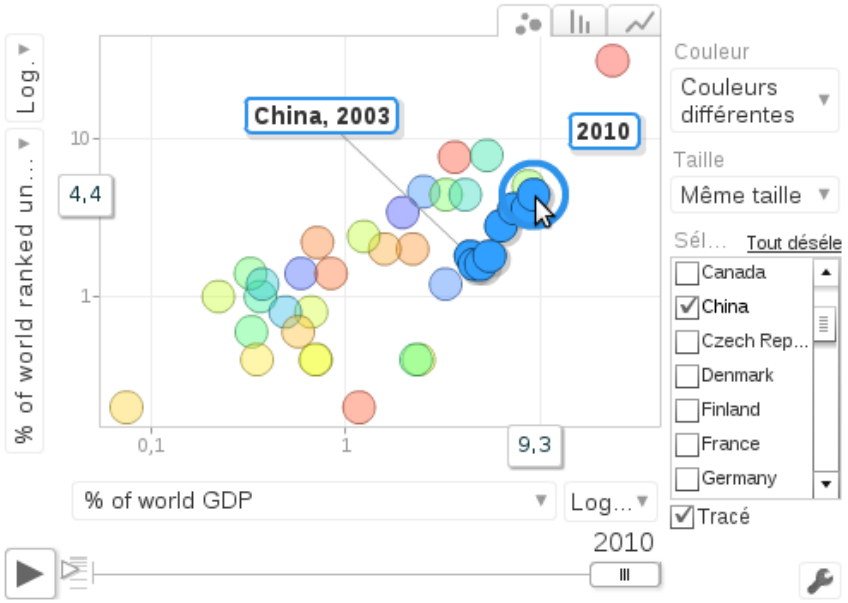


Figure 8: Motion chart

We have implemented Area chart (See Figure 7) and Motion chart (See Figure 8). The first one show us the evolution of the population over time. We have the possibility to know the exact value by using the mouse cursor. The second one allows more interaction with the chart. We can select countries, change the axis and the definition of the circle. The chart changes over time

[7] http://www.bazingaweb.fr/uploads/prestations/java_logo.png
[8] http://dlc.sun.com.edgesuite.net/jdk7/binaries/index.html

with the possibility of interrupting it.

Version :

- Stable version 1.1.0, used until January 31th.
- Latest stable version 1.1.1, use for the project.

The version 1.1.0 was used until January 31th because the new version (1.1.1) was released. The new one implemented the possibility to set the direction of the vertical axis of the chart. We needed this function because the understanding of the chart was flawed, for ranking charts the highest ranking appeared at the bottom and vice versa.

3.4.2 Google Maps API Library 1.1

This Library [15] allows to access to Google Maps API, implemented in JavaScript [18], from a GWT project.

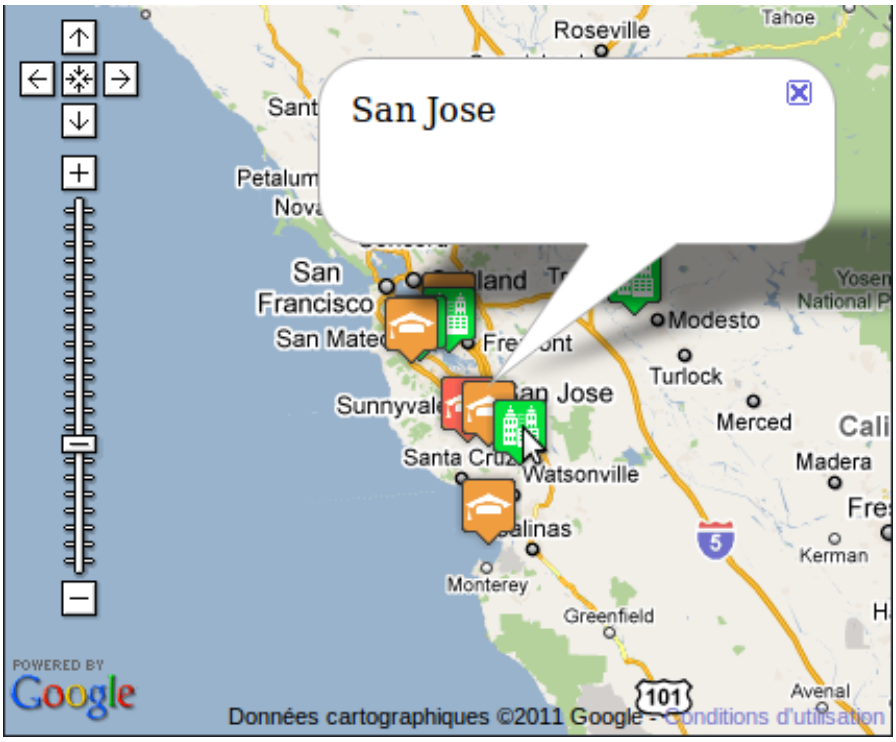


Figure 9: Map of a university and its surrounding

As we can see on the Figure 9 the map displays the current university (in red), the surrounded universities (in orange) and cities (in green). We can display the name by selecting an icon. The map allows the drag-and-drop to navigate. To use this service we have to apply for Google Maps API key[9] because there's a limitation on the geocode requests and the rate. We can send 2500

[9] http://code.google.com/intl/fr/apis/maps/signup.html

geocode requests in a day from a single IP address.[10]

Version :

- Stable version 1.1.0, used for the project.

3.4.3 appengine-utils

Appengine-utils [16] is a library that solves the problem of compatibility between GWT and GAE. Some data types of GAE (Key, Text, ShortBlob, Blob, Link, User, PostalAddress, PhoneNumber, Rating) aren't available on the GWT client-side. In this project we are using the Key type.

Version :

- Stable version 1.1, used for the project.

4 Database

The first step in the project was to find relevant information and assemble it in a database (SQL). To do that we had to find information sources and for each one we had to implement, in Java, a parser. We used the followings websites to gather data :

- Academic Ranking of World Universities [1] gives the ranking (by field, subject and for the world) of the best 500 universities. It starts in 2003 and continues to 2010. This part is explained in detail in the Section A.
- Top500 Supercomputing Sites [2] gives the ranking of nations with the most processing power throw supercomputers. The ranking starts in 1993 to 2010. This part is explained in details in the Section B.
- Wikipedia, the free encyclopedia [3] gives data on universities themselves. We have for example the number of students, undergraduates, postgraduates, the position (latitude, longitude), the budget, the endowment and the website. We have gather 604 universities. This part is explained in details in the Section C.
- GeoNames: Geographical database [4] gives information on cities. We have 22370 cities with the position and the population. This part, the following and the gather of all the data in the database has been done by Sébastien Keller who has done another part of the Unimojo project, the Android application.
- IMF: International Monetary Fund [5] gives information on countries. We have 248 countries with the country code, the population among others.

4.1 SQL to GAE

After having put the information in the database we have to set the data in the GAE database so that the Web application can interact with it. The SQL database is in the data folder.

[10] http://code.google.com/intl/fr/apis/maps/faq.html#geocoder_limit

4.1.1 Implementation

We have implemented an API in the package com.unimojo.api that read the database and send the information to Servlets on the GWT server-side. Then the Servlet set the data in the GAE database. To interact with the server we use URL with parameters :

```
http://unimojo.appspot.com/api/addCountry?code=US&iso3=USA&...
```

The client opens a connection to this URL to run the Servlet :

```
private static void request(URL url) {
  URLConnection connection;
  try {
    connection = url.openConnection();
    connection.getContent();
  } catch (IOException e) {
    System.out.println("X : " + url.toString());
  }
}
```

If there's a problem with the connection the method print an error in the console. On the server side, it receives the request and deal with it to create an Object, set the parameters and set it in the GAE database :

```
  public void doGet(HttpServletRequest req, HttpServletResponse ↩
     res)
         throws IOException {

      PersistenceManager pm = PMF.get().getPersistenceManager();

      // Create a new country.
      Country country = new Country(req.getParameter("name").trim↩
         ());

      // Set attributes of the country.
      country.setCode(req.getParameter("code"));
      country.setContinent(req.getParameter("continent"));
      country.setIso3(req.getParameter("iso3"));
      country.setLastUpdate(Integer.valueOf(req.getParameter("↩
         last_update")));
      country.setName(req.getParameter("name"));

      //Add the country to the GAE database.
      pm.makePersistent(country);
      pm.close();
  }
```

In this example we create a new Country, we set the parameters and set it in the database using :

```
pm.makePersistent(country);
```

and finally we close the connection to the database. This operation is done for every universities, countries, cities, ... The problem is that GAE is free but limited for the resources. To add data we need CPU time. We have only 6.5 CPU hours/24h and to add the SQL database we need more because sometimes we have errors so we have to restart some parts. More CPU times can be buy but this's very cheap. As we can see in the Figure 10 the CPU time has exceed the free quota.

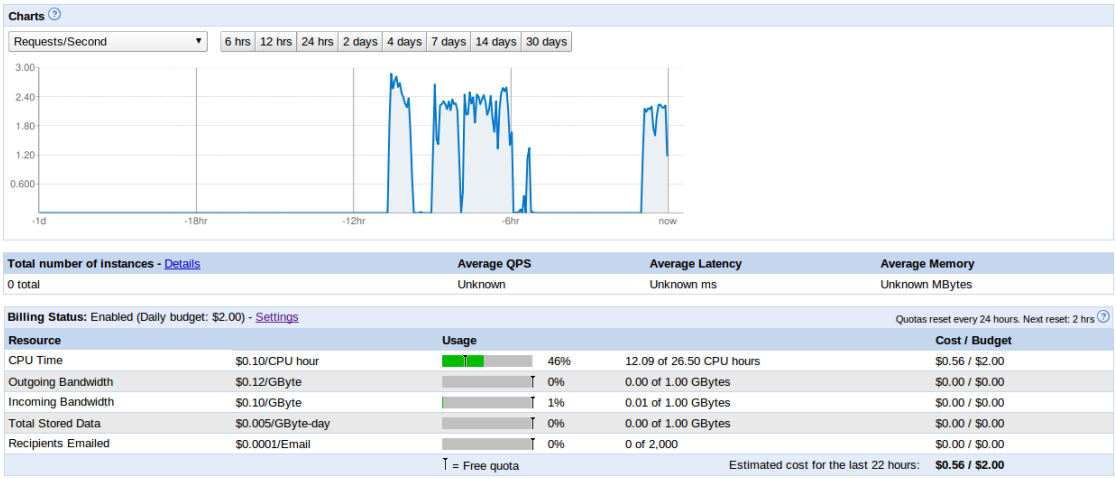


Figure 10: Screenshot of the GAE admin console

We can have access to other kind of information with the GAE admin console like statistics on the database (See Figure 11).

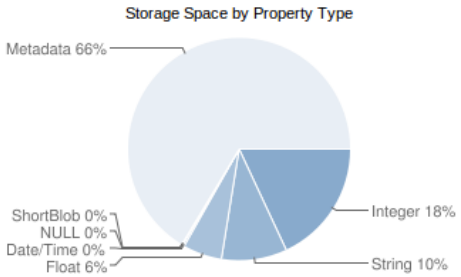
On the right we can see a chart of the storage space by entity kind. The majority is rankings and cities. On the left we can see the storage space by property type. The 66% of the database is metadata! This is due to the indexes. We have to index every kind of queries because GAE is design to be efficient with small database and large one. With indexes it takes the same time to perform a query on 1 million entities than on 10. To perform a query on the Country and to sort it by name we have to add an index as following to the datastore-indexes.xml that is in the /war/WEB-INF/ folder.

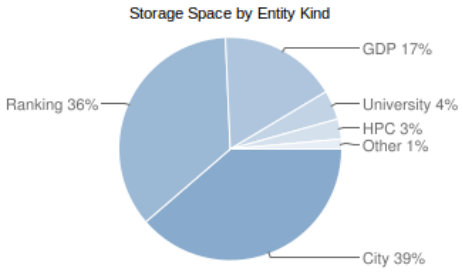
```
<datastore-index kind="Country" ancestor="true">
  <property name="name" direction="asc" />
</datastore-index>
```

After deploying the Web application on GAE the indexes appear in the admin console (See Figure 12).

In this example we have the Country and GDP indexes. It's notice that those indexes are serving. This means that the index is ready, until that you can't perform a query on the database.

Last updated	Total number of entities	Size of all entities
0:52:57 ago	22,766	9 MBytes





Breakdown by Property Type

Property Type	Size
Integer	2 MBytes
String	916 KBytes
Float	546 KBytes
NULL	30 KBytes
Date/Time	8 KBytes
ShortBlob	4 KBytes
Metadata	6 MBytes

Figure 11: Statistics on the GAE database

Country	
countries_INTEGER_IDX ▲ Includes ancestors	Serving
countrys_INTEGER_IDX ▲ Includes ancestors	Serving
name ▲ Includes ancestors	Serving
GDP	
gdps_INTEGER_IDX ▲ Includes ancestors	Serving
year ▲ , value ▲	Serving
year ▲ , value ▼	Serving

Figure 12: Indexes serving on the GAE database

4.1.2 Tricks

Sometimes the deployment of the application on GAE ends with an error seeing we have to rollback the version. Here's what you will have to type in your console :
For Windows :

```
C:\Program Files\eclipse\plugins\com.google.appengine.eclipse.sdkbundle.
X.X.X_X.X.X.vXXXXXXXXXXXX\appengine-java-sdk-X.X.X\bin>appcfg.sh --email
=YOUR_EMAIL --passin rollback C:\Users\USERNAME\workspace\PROJECT_NAME\war
```

For Linux :

```
USERNAME@USERNAME-laptop:~/.eclipse/plugins/com.google.appengine.eclipse.
sdkbundle.X.X.X_X.X.X.vXXXXXXXXXXXX/appengine-java-sdk-X.X.X/bin$ ./appcfg.sh
--email=YOUR_EMAIL --passin rollback /home/USERNAME/workspace/PROJECT_NAME/war/
```

If you want to access to the GAE console when you are on localhost use the following URL :

```
http://localhost:8888/_ah/admin
```

5 Unimojo

The Web application Unimojo is the main part of the project. The database has taken a lot of time but finally what will be usable by the user is the website. Unimojo is the tool that allows to display what contains the database. This is the front-end of the information.

5.1 Features

The application has several parts :

- The search engine helps us finding the university with suggestions. You can also type what you want, for example the name of a city, the name of an area in italian, or your address and it will return the nearest university.
- The ranking charts are the representation in time of rankings of three types : subject, field, world.
- The map displays the current university, the surrounding universities and cities with an icon. The map is dynamic, you can drag-and-drop it with the mouse to search for other universities. The universities display on the map will appear in a suggest list.
- Suggest list regroups the visible universities on the map that evoluates dynamically while the user moves the map. There's a second list with universities that are better than the current university in every concerned fields and subjects.
- Cities are the nearest ones. The distance between cities and current university is display in kilometers and miles.
- The Country tab adds information regarding the country itself with chart on the evolution of the GDP, unemployment rate, the population and HPC information. We can also do comparison between countries with a motion chart.

5.2 Scenarios

The possibility you have at the beginning is to search for a university name or a place. Then the website tries to find universities that corresponds to the input and if there no suggestions, it tries to find the nearest university of the input. It follows the next scenario (See Figure 13) :

After displaying the university data, the map and the suggestions the user have several possibilities. He can search for another university or place but he can also move the map, click on a suggestion or go to the city and country tab. It follows the next scenario (See Figure 14) :

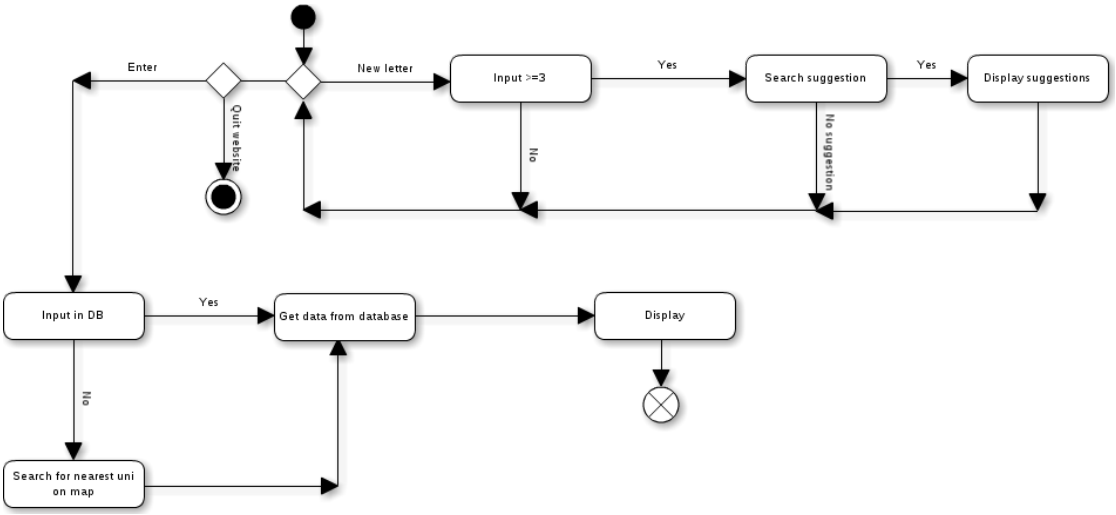


Figure 13: Activity diagram of the search engine

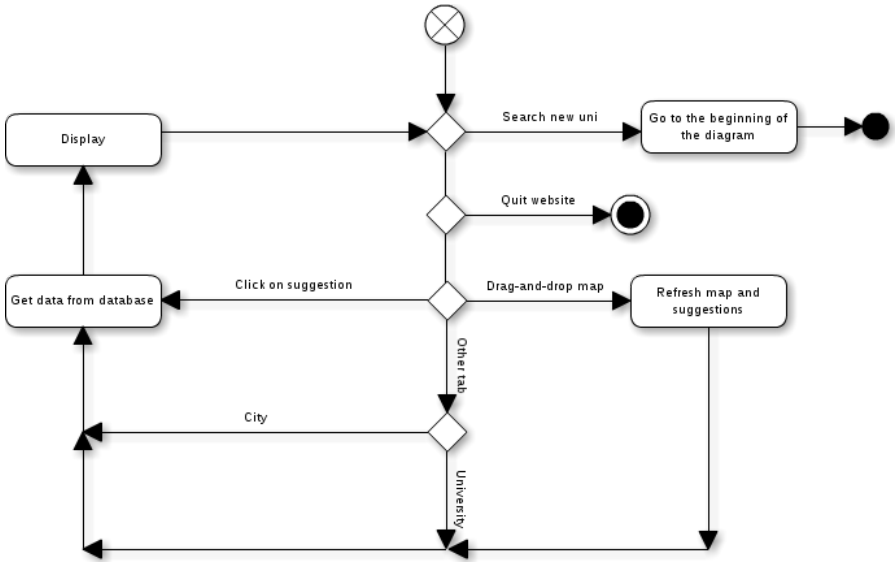


Figure 14: Activity diagram of the website

5.3 Interface

In this part we will discuss about the interface and explain each parts and their functionalities. As we can see on the Figure 15 we have several panels described in the next lines.

- Search engine allows to search for an university name or a place.
- Data and charts displays information on the university, cities or country.
- Navigation allows to navigate throw tabs.

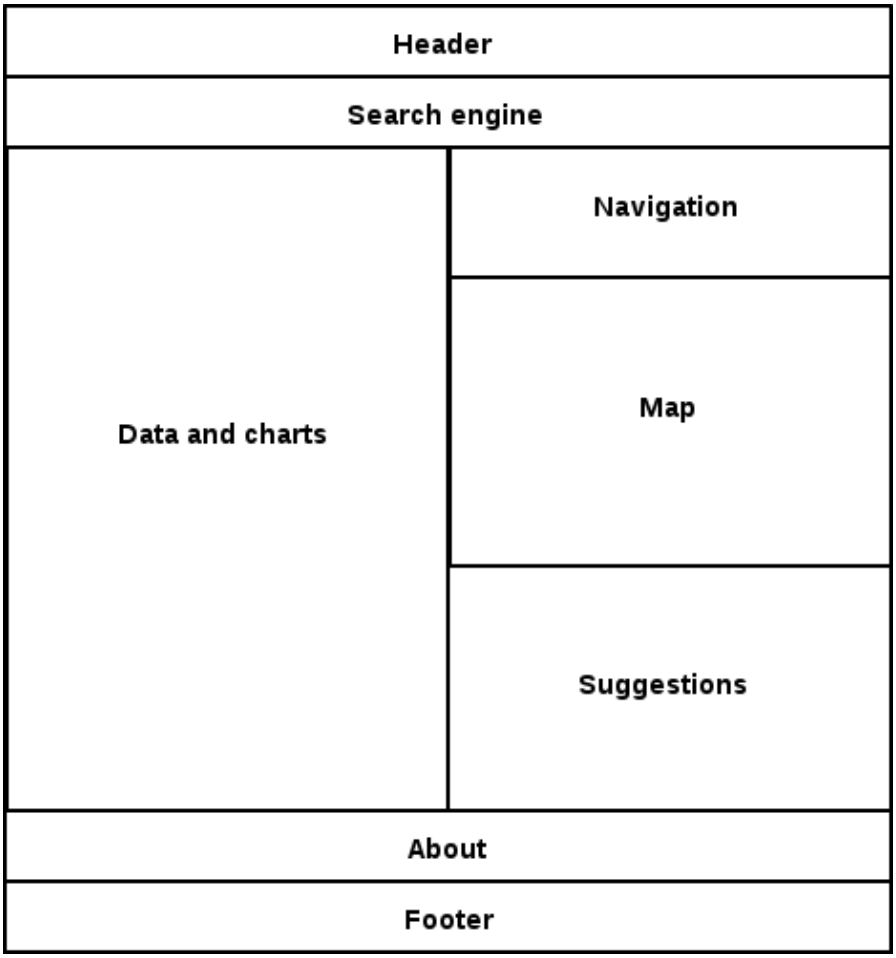


Figure 15: GUI representation

- Map displays the current university, cities and the suggest universities.
- Suggestions give to the user some help to navigate between universities.
- About gives help to the user.

5.3.1 Search engine

When we arrive on the website (See Figure 16) we have can only use the search engine. We have to type an university name or a location to display data regarding the latter.

Then we can start typing in the box and some suggestions will appear (See Figure 17, if available.

It works well with an input having mistakes. For example typing santfrod univrsety instead of Stanford University. You will then click on Enter and the website will display data regarding the input.

Figure 16: The Unimojo website

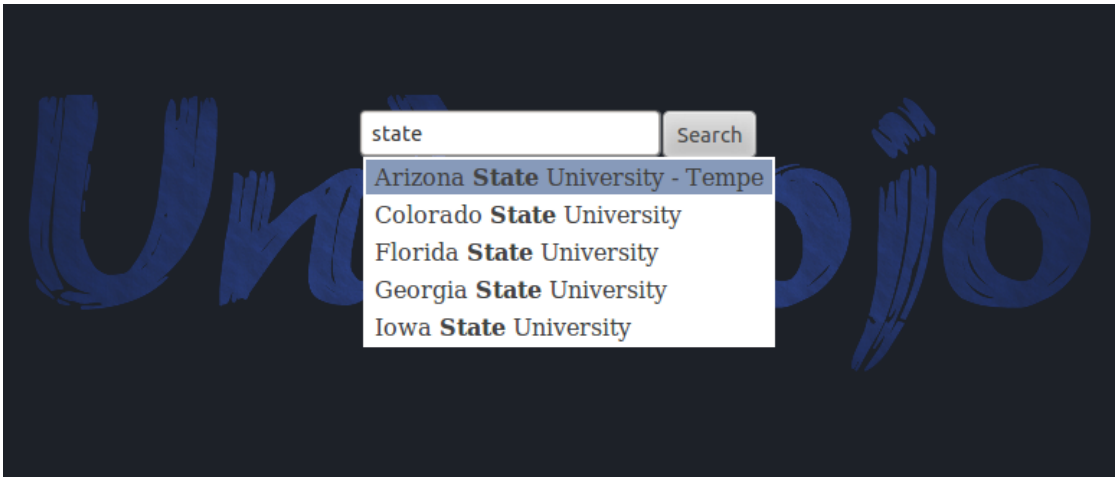


Figure 17: The suggest engine

5.3.2 Footer

The footer adds a few information on the website (See Figure 18).

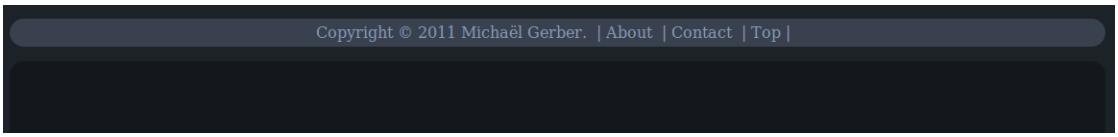


Figure 18: The footer of the website

"About" gives information on the reason of this website. "Contact" allows you to send an

email if you want to obtain more details and “Top” leads you to the top of the website.

5.3.3 University tab

The university tab displays information about the rankings (See Figure 19).

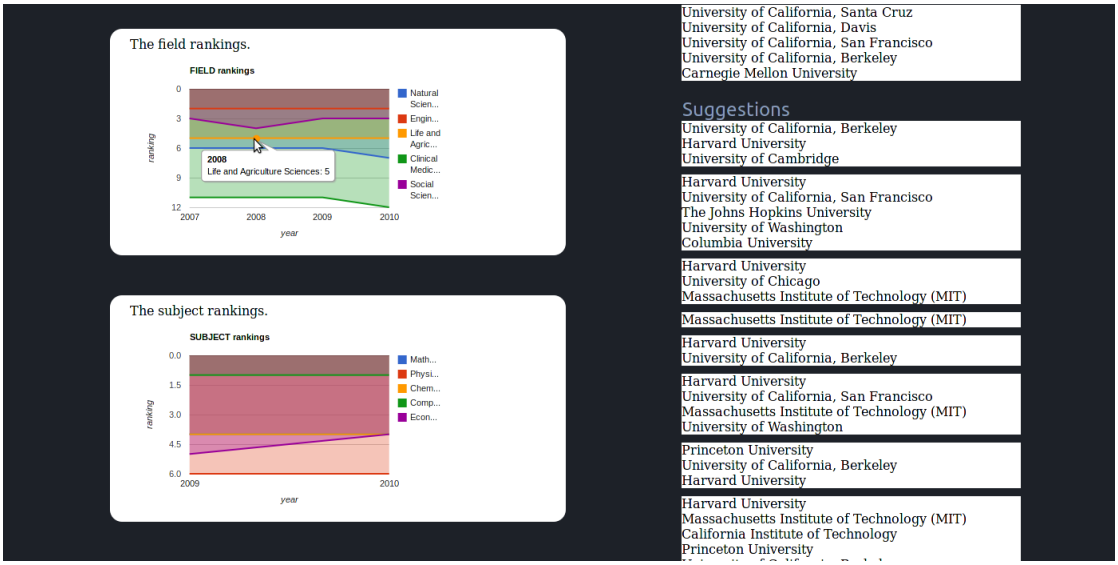


Figure 19: The University tab

Suggestions When the university data are displayed some suggestions are added on the right. You have the nearby universities and the ones that are better in the rankings (See Figure 20).

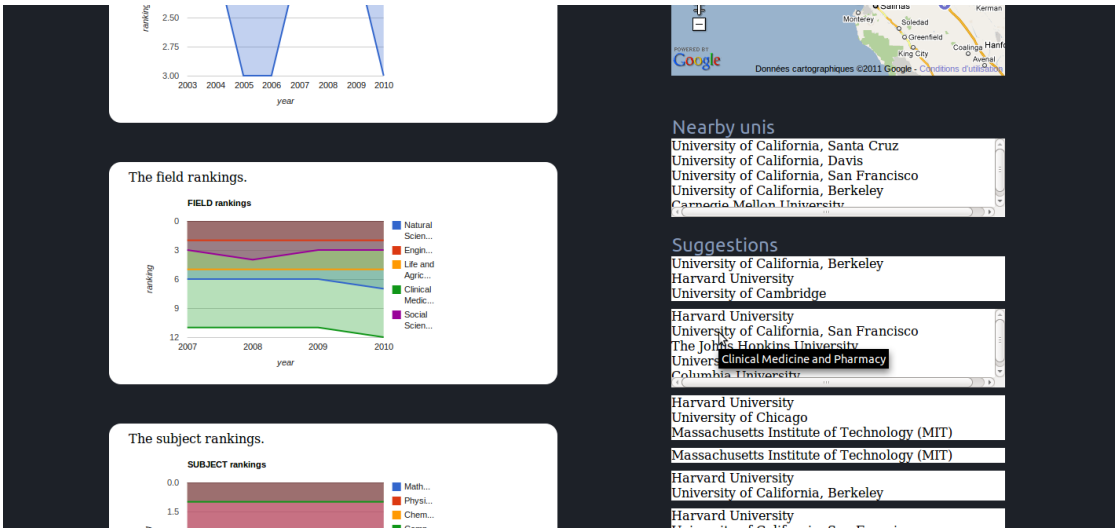


Figure 20: The suggestions of the website

5.3.4 Cities tab

After clicking on the City button, you access to information on the nearest cities around the current universities. As we can see on the Figure 21 cities are added to the map in green and they are displayed with the population and the distance to the university (in kilometers and miles).

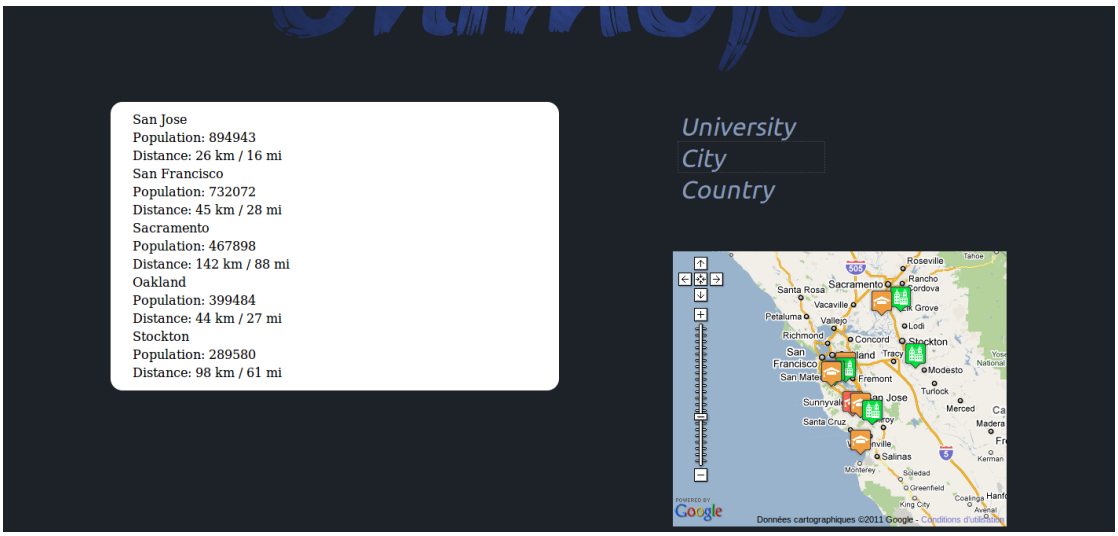


Figure 21: The City tab

5.3.5 Country tab

The country tab gives us information on the evolution of the country like the unemployment rate, the GDP or the number of supercomputers. We have also the possibility to use the motion chart that is a dynamic chart. The country tab can be shown on the Figure 22 and 23.

5.3.6 Icons

As you can see there's some icons used for the website. For the ones display on the map we used a collection of icons dedicate to Google Maps.[26]. For icons on the Country tab in the first panel we have used the website Iconfinder that is an icon search engine.[25]

5.4 Tests

The test that we can do with a website is to display the latter in every popular Web browser. Before speaking about results we can guess that there will be some problems because not every browser respects the recommendations of the World Wide Web Consortium (W3C)[11] who is "the main international standards organization for the World Wide Web"[12]. The Acid3 test[13] shows us the performance and the respect of the standards by a Web browser. Actually less Web browsers obtain 100% to the test. We will see the results, the problems and maybe the solutions

[11] http://www.w3.org/
[12] http://en.wikipedia.org/wiki/World_Wide_Web_Consortium
[13] http://acid3.acidtests.org

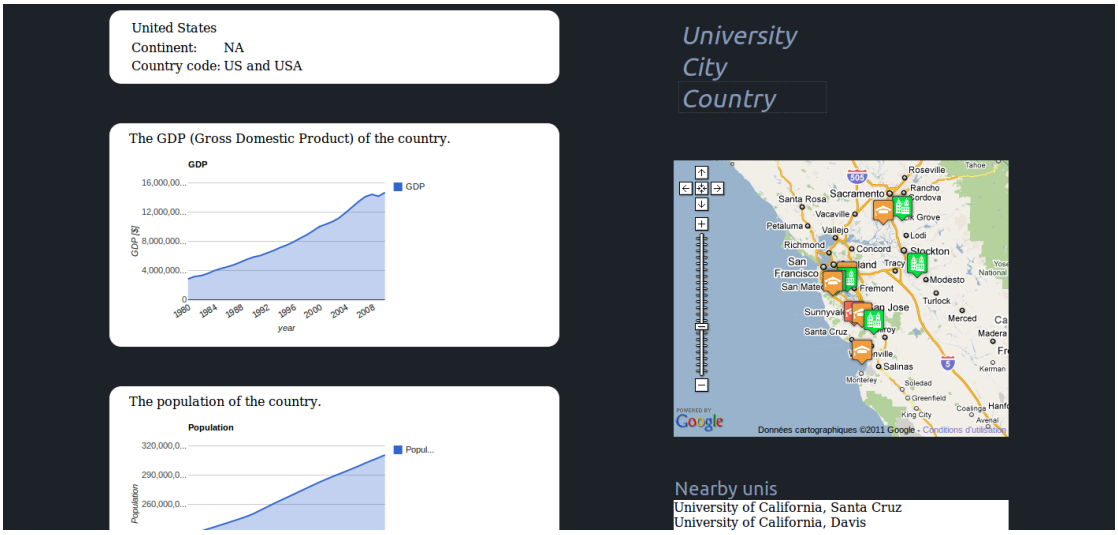


Figure 22: The Country tab

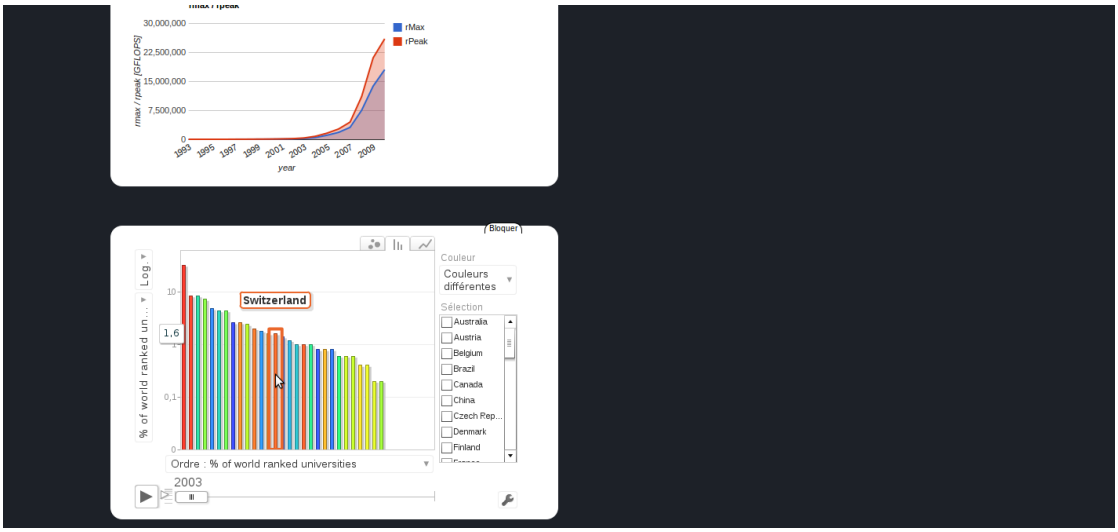


Figure 23: The Country tab

to the web browser performances.
Here's the results of tests done on several Web browser (The web browser will be sort by decreasing popularity) :

5.4.1 Firefox

Mozilla Firefox is a free and open source web browser developed by the Mozilla Foundation.

Versions tested :

- Stable version 3.6.13 (Ubuntu 10.10)

- Preview version 4.0 Beta 11 (Ubuntu 10.10)
- Global Popularity 42.8%

As said in Section 3.2.1 we have to use a trick to display rounded borders with Firefox (Chrome, Safari and Konqueror too). The only browser that displays correctly the website without trick is Opera. Firefox 3.5+ is based on Gecko 1.9.x (the layout engine) that implements the -moz-border-radius property. But when the W3C recommendations on CSS3 were released the name for this property was border-radius. So the version 2 of Gecko, that is on development, implements the border-radius property. This version is used by Firefox 4. For the release it will be 100% compatible with CSS3.

Firefox 3.6.13

- Popularity 36.1%
- Acid3 test 94%

Problems

1. When you type on the key Enter nothing appends. This problem appears only with Firefox.

Solutions We don't have found solutions for the moment.

Firefox 4.0 Beta 11

- Popularity 1.5%
- Acid3 test 97%

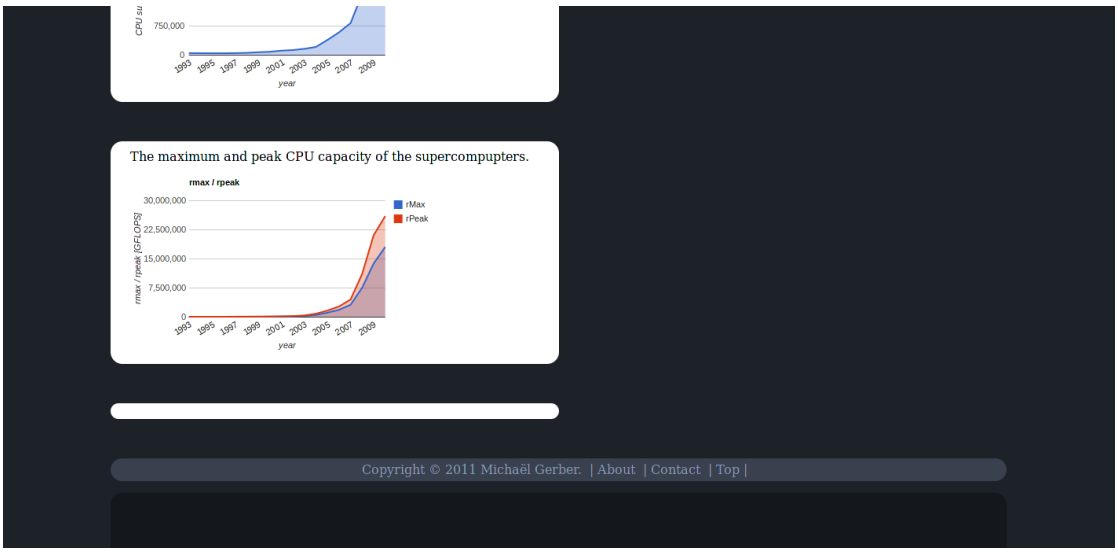


Figure 24: Problem with Firefox 4.0 Beta 11 to display the motion chart

Problems

1. The Motion chart doesn't display. This is maybe due to a JavaScript problem (See Figure 24).
2. When you type on the key Enter nothing appends. This problem appears only with Firefox.

Solutions We don't have found solutions for the moment but we can see that they are trying to fix bugs because the Acid3 result has increased a little bit. Now this is close to the perfection.

5.4.2 Internet Explorer

Internet Explorer is a Web browser developed by Microsoft.

Versions tested :

- Stable version 8.0.7600.16385 (Windows 7)
- Global Popularity 26.6%

Internet Explorer 8.0.7600.16385

- Popularity 16.6%
- Acid3 test 20%

Problems Internet Explorer doesn't support the rounded borders. So has we can see on the Figure 25 the corners aren't rounded. The rest of the application works well.

Figure 25: Problem with Internet Explorer 8.

Solutions They exist some solutions that use JavaScript to display the rounded borders or you can add pictures to every corners.

5.4.3 Chrome

Chrome is a web browser developed by Google.
Version tested :

- Stable version 9.0.597.94 (Ubuntu 10.10)
- Global Popularity 23.8%

Chrome 9.0.597.94

- Popularity 1.3%
- Acid3 test 100%

Problems No known issues.

5.4.4 Safari

Safari is a graphical web browser developed by Apple and included as part of the Mac OS X operating system.

Version tested :

- Stable version 5.0.3 (Windows 7)
- Global Popularity 4.0%

Safari 5.0.3

- Popularity 3.5%
- Acid3 test 100%

Problems No known issues.

5.4.5 Opera

Opera is a web browser and Internet suite developed by Opera Software.

Version tested :

- Stable version 11.01 (Ubuntu 10.10)
- Global Popularity 2.5%

Opera 11.01

- Popularity 1.7%
- Acid3 test 100%

Problems No known issues.

5.4.6 Konqueror

Konqueror is a web browser and file manager developed by KDE.

Version tested :

- Stable version 4.5.1 (Ubuntu 10.10)
- Global Popularity unknown

Konqueror 4.5.1

- Popularity unknown
- Acid3 test 89%

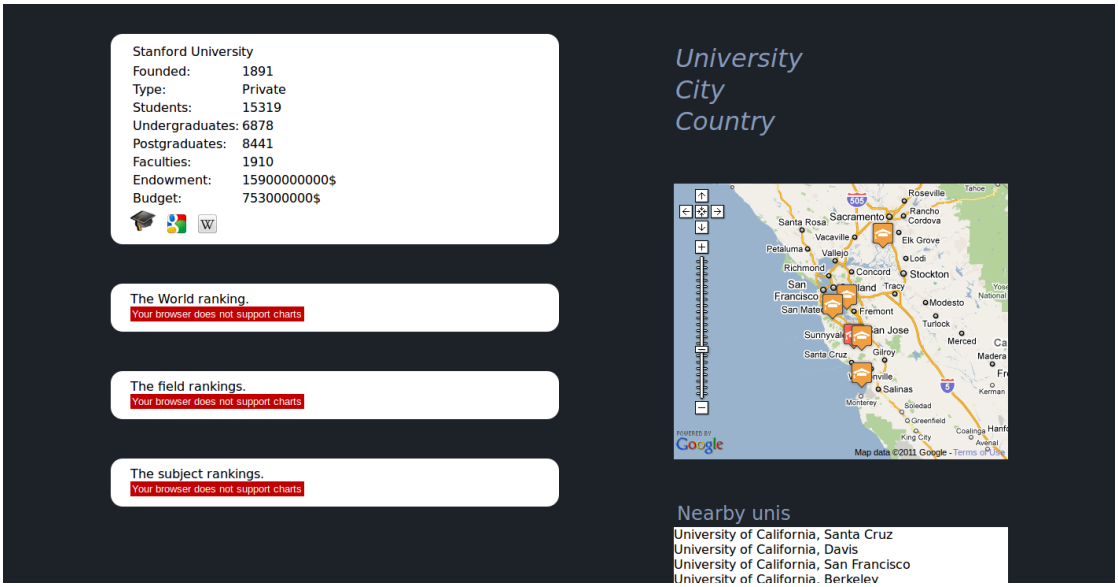


Figure 26: Problem with Konqueror 4.5.1 to display charts

Problems

1. No charts are displayed (See Figure 26). There's a warning saying that"Your browser does not support charts".
2. As for Firefox the Enter key doesn't work.

Solutions We don't have found solutions for the moment

5.4.7 Midori

Midori is a web browser that aims to be lightweight and fast.

Version tested :

- Stable version 0.2.4 (Ubuntu 10.10)
- Global Popularity unknown

Midori 0.2.4

- Popularity unknown
- Acid3 test 100%

Problems No known issues.

5.4.8 Arora

Arora is a free and open source lightweight cross-platform web browser.

Version tested :

- Stable version 0.10.2 (Ubuntu)
- Global Popularity unknown

Arora 0.10.2

- Popularity unknown
- Acid3 test 100%

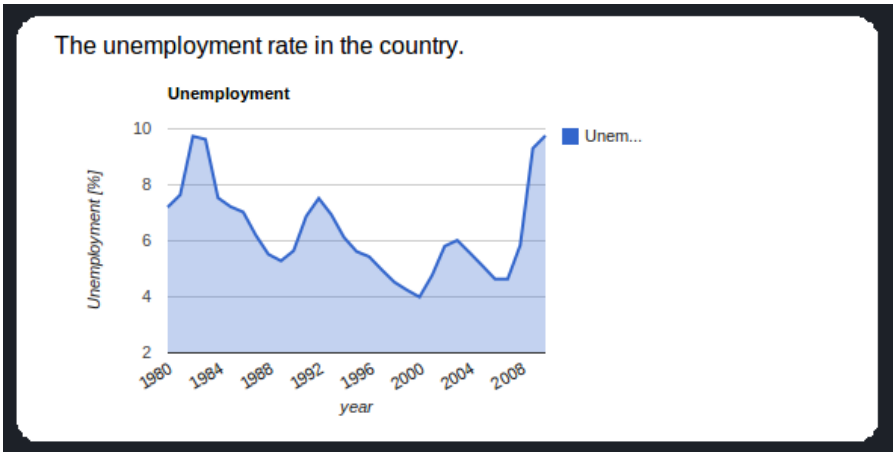


Figure 27: Problem with Arora 0.10.2 with rounded borders

Problems

1. The rounded border are displayed but there's a few aliasing (See Figure 27).

Solutions We don't have found solutions for the moment but what is surprising is that the layout engine, Webkit, is used by Chrome and Safari.

5.4.9 Conclusion

As we can see, web browsers, that don't respect or implement the recommendations of the W3C, don't support some features of a website using the last technologies. We are using HTML5, CSS3 and JavaScript and we have to find tricks to make the website correctly viewable on most browsers.

The popularity statistics come from W3C.[14] This is the statistics of January 2011.

5.5 User guide

If you simply want to try the application go to : http://www.unimojo.appspot.com
If you want to run the application locally :

1. Download and install Eclipse.[7].
2. Install the Google plugin and SDKs (GWT and GAE). Follow the Quick Start tutorial [11].
3. Import the Unimojo project in Eclipse.
4. Run it as Web Application.
5. Then go to http://127.0.0.1:8888/Unimojo.html?gwt.codesvr=127.0.0.1:9997 (the URL will be display in the Development Mode console) with Firefox, Internet Explorer, Chrome, or Safari. You will have to install a plugin for your Web browser. A popup will appear.

6 Modeling

In this part we will see the relations in the GAE database.

6.1 UML chart

This chart (see Figure 28) shows us the relations between the tables of the database.

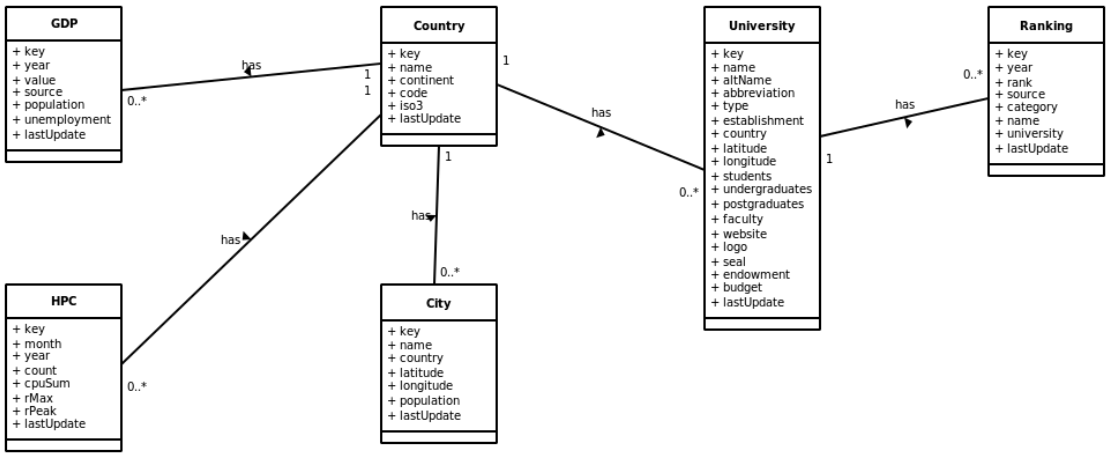


Figure 28: UML chart

[14] http://www.w3schools.com/browsers/browsers_stats.asp

6.2 Description of tables

Here is the description of tables in the GAE database.

City

key : Unique identifier of the City.
name : The name of the City.
country : The Country of the City.
latitude : The latitude of the City.
longitude : The longitude of the City.
population : The population of the City.
lastUpdate : The last time the City has been updated in the database.

Country

key : Unique identifier for the Country.
name : The name of the Country.
continent : The continent of the Country.
code : The two-letter Country code (ISO 3166-1 alpha-2).
iso3 : The three-letter Country code (ISO 3166-1 alpha-3).
lastUpdate : The last time the City has been updated in the database.

GDP

key : Unique identifier of the GDP, Gross Domestic Product.
year : The year of the GDP.
value : The value of the GDP.
source : The source of the GDP.
population : The population of the Country.
unemployment : The unemployment rate of the Country.
lastUpdate : The last time the GDP has been updated in the database.

HPC

key : Unique identifier of the HPC, High-Performance Computers.
month : The month of the HPC.
year : The year of the HPC.
count : The number of High-Performance Computers.

cpuSum : The sum of processors.
rMax : The maximum of FLOPS (FLoating point OPerations per Second).
rPeak : The peak of FLOPS (FLoating point OPerations per Second).
lastUpdate : The last time the HPC has been updated in the database.

Ranking

key : Unique identifier of the Ranking.
year : The year of the Ranking.
rank : The value of the Ranking.
source : The source of the Ranking.
category : The category of the Ranking.
name : The name of the Ranking.
university : The university ranked.
lastUpdate : The last time the Ranking has been updated in the database.

University

key : Unique identifier of the University.
name : The name of the University.
altName : The alternative name of the University.
abbreviation :The abbreviation of the University.
type : The type of University.
establishment : The date of establishment.
country : The Country of the University.
latitude : The latitude of the University.
longitude : The longitude of the University.
students : The number of students in the University.
undergraduates : The number of undergraduate students in the University.
postgraduates : The number of postgraduate students in the University.
faculty : The number of staff in the University.
website : The website URL of the University.
logo : The logo of the University.
seal : The seal of the University.
endowment : The endowment of the University.
budget : The budget of the University.
lastUpdate : The last time the University has been updated in the database.

6.3 Description of packages

We will describe every package of the project to understand there utility and function.

6.3.1 com.unimojo

This is the main package of the project. It contains sub-packages and a configuration file. We can set, for example, inherited modules and shared package between the client and server.

com.unimojo.api This package allows to set the GAE database with the SQL database.

com.unimojo.client The functionalities on the client-side.

com.unimojo.client.rpc The remote methods that the client can access.

com.unimojo.data The data that the client and server shared.

com.unimojo.pmf The manager to interact with the GAE database.

com.unimojo.server The functionalities on the server-side.

com.unimojo.shared Everything that is shared between the client and server. But this package isn't really necessary because if you want to share a package you simply have to add the following line in the file Unimojo.gwt.xml (package com.unimojo) :

```
<source path='data' />
```

The package com.unimojo.data is shared. We have to do so because the client and server exchange Object from the data package and both sides must know the definition of the object.

7 Conclusion

7.1 Problems

The main problems encountered during this project have been described throughout this report. To summarize, what has caused most problems was to get started with the different SDKs (GWT and GAE) because both are very powerful and needed time to be master. Even during development when new releases appeared they were pros, like the implementation of new features, and cons, like the abandonment of features used in the project. So it must be often modified during the development and adjust to match with the API.

7.2 Future work

The future work can be done on the following list :

- Optimization can be done on some parts that consume too many resources. With a large traffic on the website it couldn't be possible with actual resources.
- IE compatibility can be done with some tricks that aren't very elegant.

- More data can be display. Now every tools are working so to add more charts it's very easy. But the large work is to gather data from various sources.

7.3 Conclusion of the project

This project was the largest project done during my study. It needed a lot of work to obtain what I expected at the beginning. It was a project with a lot of challenge like mastering new SDKs and leading a project from scratch. It was a huge opportunity to discover a new country and the life in such a large company.

7.4 Acknowledgments

The list of people I would like to thank :

- Professor Stephan Robert, who offered me the opportunity to do my last project in the USA.
- Jim Spohrer, who welcomed me at IBM and supervised my work.
- Jeffrey Brody, who supervised the progress of work and gave me advices.
- Sébastien Keller, who shared this adventure.
- My family and friends, for their help and support during the 6 months of my project and trip.

A ARWU parser

This application parses the Academic Ranking of World Universities website[1]. It takes the ranking of the 500 best universities for every available year.

A.1 Features

1. Get HTML webpages and save them in .html files so that we don't have to download them twice. Files are in the file folder in the project.
2. Parsing of webpages to get important information.
3. Filling the CSV file to gather the data. The CSV file is the file folder.

A.2 User guide

1. Import project in Eclipse.
2. Run it as an application.

A.3 Libraries

We will describe libraries used for the ARWU parser.

A.3.1 GeoGoogle

GeoGoogle is an address standardization API.[12] It uses the google's geocoding service. We need it to find the coordinates (latitude, longitude) of universities using their name. This is done just before filling the CSV file.

Version :

- Stable version 1.5.0, used for the project.

Dependencies The list of libraries that GeoGoole needs to work :

- activation-1.1.jar
- commons-codec-1.2.jar
- commons-collections-3.2.jar
- commons-httpclient-3.1-beta1.jar
- commons-io-1.3.1.jar
- commons-lang-2.3.jar
- commons-logging-1.0.4.jar
- jaxb-api-2.1.jar
- jaxb-impl-2.0.3.jar
- jsr173_api-1.0.jar
- stax-api-1.0-2.jar

A.3.2 jsoup

Jsoup is a Java HTML parser[13]. It allows to extract and manipulate data.

```
Document doc = Jsoup.parse(clientAnswer);
```

This line create a document by parsing the input. clientAnswer is a String representation of the HTML page.

```
String universityNames = doc.select("div[align=left]");
```

Then to extract information we have to define rule in the select part. In this example we have :

```
div[align=left]
```

It means we want the div tag that is align on the left. This library is very powerful because with a few lines you obtain exactly what you need.

Version :

- Stable version 1.2.3, used for the project.
- Latest stable version 1.4.1.

B HPC parser

This application parses the top 500 website[2] to obtain the ranking of countries throw supercomputers.

B.1 Features

See Section A.1.

B.2 User guide

See Section A.2.

B.3 Libraries

We will describe libraries used for the HPC parser.

B.3.1 HttpComponents Client

HttpComponents Client is an HTTP agent implementation[14]. We can create a Client that gets webpages throw the HTTP protocol.

Version :

- Stable version 3.1, used for the project.

B.3.2 jsoup

See Section A.3.2.

Version :

- Stable version 1.3.3, used for the project.
- Latest stable version 1.4.1.

C Wikipedia parser

This application parses the Wikipedia website to obtain information on universities.

C.1 Features

1. Get HTML webpages and save them in .html files so that we don't have to download them twice. Files are in the file folder in the project.
2. Parsing of webpages to get important information. If the application doesn't understand an input, that doesn't match with rules (example: a date of establishment can't have 5 digits.), we have to correct it by typing the solution in the console.
3. Getting logo of universities and saving them in PNG picture. Logos are in /file/img. We don't use them in the website because many pictures have a copyright. To correctly save the logos as PNG without losing quality we had to use Java 1.7b116.[15] This is a beta version of the next Java version. It deals with transparency needed to save pictures as PNG. The actual stable version of Java doesn't implement transparency.
4. Filling the CSV file to gather the data. The CSV file is the file folder.

C.2 User guide

See Section A.2. As mentioned above you have to type inputs that the application doesn't match.

C.3 Libraries

We will describe libraries used by the Wikipedia parser.

C.3.1 HttpComponents Client

See Section B.3.1.

C.3.2 jsoup

See Section A.3.2.

Version :

- Stable version 1.3.3, used for the project.
- Latest stable version 1.4.1.

[15] http://dlc.sun.com.edgesuite.net/jdk7/binaries/index.html

References

Data sources

[1] Academic Ranking of World Universities.
http://www.arwu.org/

[2] Top500 Supercomputing Sites.
http://top500.org/

[3] Wikipedia, the free encyclopedia.
http://en.wikipedia.org/wiki/Main_Page

[4] GeoNames: Geographical database.
http://www.geonames.org/

[5] IMF: International Monetary Fund.
http://www.imf.org/external/index.htm

Development tools

[6] Java: A programming language.
http://www.oracle.com/technetwork/java/javase/downloads/index.html

[7] Eclipse: Multi-language software development environment.
http://eclipse.org/

[8] Google Web Toolkit: Development toolkit for building and optimizing complex browser-based applications.
http://code.google.com/intl/en/webtoolkit/

[9] Google App Engine: Enables you to build and host web apps on the same systems that power Google applications.
http://code.google.com/intl/en/appengine/

[10] Apache SVN: A software versioning and a revision control system.
http://subversion.apache.org/

[11] Google Plugin for Eclipse.
http://code.google.com/intl/fr/eclipse/docs/getting_started.html

Libraries

[12] GeoGoogle - Google Geocoder Java API.
http://geo-google.sourceforge.net/

[13] jsoup: Java HTML Parser.
http://jsoup.org/

[14] HTTP/1.1 compliant HTTP agent implementation.
http://hc.apache.org/

[15] The Official Google API Libraries for Google Web Toolkit.
http://code.google.com/p/gwt-google-apis/

[16] Google Web Toolkit/Google App Engine integration libraries.
http://www.resmarksystems.com/code/

[17] Google Chart Tools: Enable adding live charts to any web page.
http://code.google.com/intl/fr/apis/charttools/

[18] Google Maps API: Let you embed Google Maps into a website.
http://code.google.com/intl/fr/apis/maps/

CSS

[19] CSS official website.
http://www.w3.org/TR/#tr_CSS

[20] CSS Design (french): Inspiration source for web designers.
http://www.css-design.fr/

[21] bbxdesign: CSS WordPress Web Design.
http://bbxdesign.com/

[22] Alsacréations (french): Tutorials XHTML, CSS, news and articles on the Web standards.
http://www.alsacreations.com/

[23] CSS3.info: Everything you need to know about CSS3.
http://www.css3.info/

[24] Popup creation (french) : Creation of a 100% CSS popup.
http://www.wks.fr/Creer-une-popup-100-CSS.html

Pictures

[25] Iconfinder: Icon Search Engine.
http://www.iconfinder.com/

[26] google-maps-icon: More than 1000 free and descriptive map POI markers, icons, for your maps.
http://code.google.com/p/google-maps-icons/

Other

[27] WolframAlpha: Computational Knowledge Engine.
http://www.wolframalpha.com/

This document has been written with LaTeX.