

SERVICE-BASED WEB PORTAL
FOR EMPLOYMENT OF STUDENTS AND GRADUATES



PROF.DR.WLADIMIR BODROW
HTW - BERLIN

PROF.DR.JOSE LUIS POZA LUJÁN
ETSINF/UPV - VALENCIA

VÍCTOR VICENTE, QUEROL GARCÍA
HTW – BERLÍN 2010-11

INDEX.

CHAPTER 1:INTRODUCTION -----	5
DESCRIPTION OF THE PROBLEM -----	5
OBJECTIVES -----	5
BENEFITS -----	6
CHAPTER 2: THEORETICAL FRAMEWORK -----	7
HTML -----	7
JSP -----	8
JAVASCRIPT -----	9
AJAX -----	9
GOOGLE MAPS -----	10
SQL -----	11
POSTGRESQL -----	11
APACHE TOMCAT -----	12
XML FILES -----	12
ECLIPSE -----	13
MULTITIER ARCHITECTURE -----	14
CHAPTER 3: SPECIFICATION OF REQUIREMENTS -----	15
DESCRIPTION -----	15
OBJECTIVES -----	15
OVERALL DESCRIPTION -----	15
PRODUCT PERSPECTIVE -----	16
SERVICE REQUIREMENTS -----	16
OBJECTS -----	17
MAINTAINABILITY REQUIREMENTS -----	18
CONCLUSION -----	18
CHAPTER 4: DEVELOPMENT OF IMPLEMENTATION -----	19
ACTORS -----	19
USE CASES DIAGRAM -----	20
CLASS DIAGRAM -----	22
ACTIVITY DIAGRAM -----	23
RELATIONAL MODEL DESIGN -----	24
APPLICATION ARCHITECTURE -----	26
INTERFACE & PROTOTYPE DESIGN -----	28
IMPLEMENTATION -----	30

CHAPTER 5: CONCLUSIONS & FUTURE WORK	36
CONCLUSIONS	36
FUTURE WORK	37
APPLICATION EXAMPLE: APPENDIX 1	38
CHECKING PLACES	40
REGISTER-APPLICATION	42
CREATE A NEW PLACE	44
REMOVING PLACES	46
CHANGING PASSWORD	47
INTERNET BIBLIOGRAPHY	49
BOOKS BIBLIOGRAPHY	50

INTRODUCTION.

Description of the problem.

This project is the solution to the need to search a centralized web portal of jobs or study to any speciality.

If we stop to think, there are almost no web sites (www.aftercollege.com- www.youtern.com) that gives us the chance to search by a particular speciality, companies where can realize an internship for new graduates. This project is enriched beside of the possibility of finding an internship in a company, adding the possibility of finding a specialty colleges, which offer studies in a particular speciality on a map shown all of google maps for cities.

We are aware of the difficulties that exist in certain countries, to find an internship or university bid us the possibility of study a bachelor or degree of our preferences or praktikum to initiate us into the world of work, due to ignorance and confusion. With this application, try to solve this problem existed before.

Today, if we look the job offers from companies which are looking for workers, we see that companies need workers with experience and to make an internship is a good way to get it and for the companies a way to gain productivity with lower costs. But also, working in an internship within a company, we will learn about what is the different forms of work and then decide whether we want to work in similar jobs.

OBJECTIVES.

The objectives of this application are very clear: to enable the new graduates getting a first job to improve their personal experience and to get a general idea about the way to work in a selected specific sector, besides of the pre-students in universities be able to find some universities where carry out their studies.

We can distinguish two kinds of user in this web page:

Pre-students & New-graduates:

The main idea of this web site is the simplicity. Simplicity to find some that could be complicate to get, as it could be a first job for a new-graduate or to find the proper university where to study a speciality. So, the information shown in the page will be public and anyone be able consult it.

Thus, the objective of the user will get into the page and choosing through their speciality, type of company or university, and location, will know where its possible get their most suitable university or companies which are offering a first job to him, all information will be displayed in a marker on a map of Google Maps in the correct geographical position.

Universities & Companies:

This characters will be responsible to begin the feedback in the web site, sending their

register application to appear in the options of the web site, thus one of the most interested in start this feedback have to be the companies and universities to attract new workers and students to their respective universities and companies. The user will get in touch with them and will arrange an job interview to know more information about the features of the job.

Admin:

The objective of the administrator will be ensure the proper behavior of the site, adding the new companies or universities that want to join to the application or removing them.

BENEFITS.

Developing this application, we are saving a lot of effort to the companies, universities, future students and new graduates, because we are computerizing several actions that in real life would be more complicate than by internet, centralizing everything in a web portal, where the students, new graduates, companies or universities just with few clicks be able to find new low cost young workers willing to learn or to get the, always complicate, first job, or to find the proper university.. etc

In addition, for the global world, this application will be a very good tool to the population acquire more knowledge having the experience of working on companies or studying in the university most proper. As well, it will be positive, because thanks of that, the unemployment rate will be reduced and overcoat in the fringe of the young people recently finished their studies.

THEORETICAL FRAMEWORK.

In this section, I will introduce the tools used for this project, as well as the technologies used giving a technical and theoretical explanation of:

HTML, Java Server Page, JavaScript, AJAX, Google Maps, Google Maps API, SQL, PostgreSQL, Apache Tomcat, XML files, Eclipse and Multi-tier Architecture.

HTML.



Which stands for HyperText Markup Language, is the predominant markup language for web pages. HTML is the basic building-blocks of webpages. A markup language is a set of markup tags, and HTML uses markup tags to describe web pages.

HTML is written in the form of HTML elements consisting of tags, enclosed in angle brackets (like <html>), within the web page content.

HTML tags normally come in pairs like <h1> and </h1>. The first tag in a pair is the start tag, the second tag is the end tag (they are also called opening tags and closing tags).

The purpose of a web browser is to read HTML documents and compose them into visual or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts in languages such as JavaScript which affect the behavior of HTML webpages.

Web browsers can also refer to Cascading Style Sheets (CSS) to define the appearance and layout of text and other material. The W3C, maintainer of both the HTML and the CSS standards, encourages the use of CSS over explicitly presentational HTML markup.

JSP.

JavaServer Pages (JSP) is a Java technology that helps software developers serve dynamically generated web pages based on HTML, XML, or other document types. Released in 1999 as Sun's answer to ASP and PHP; JSP was designed to address the perception that the Java programming environment didn't provide developers with enough support for the Web.

JSP allows Java code and certain pre-defined actions to be interleaved with static web markup content, with the resulting page being compiled and executed on the server to deliver an HTML or XML document. The compiled pages and any dependent Java libraries use Java bytecode rather than a native software format, and must therefore be executed within a Java virtual machine (JVM) that integrates with the host operating system to provide an abstract platform-neutral environment.

JSP syntax is a fluid mix of two basic content forms: scriptlet elements and markup. Markup is typically standard HTML or XML, while scriptlet elements are delimited blocks of Java code which may be intermixed with the markup. When the page is requested the Java code is executed and its output is added, in situ, with the surrounding markup to create the final page. Because Java is a compiled language, not a scripting language, JSP pages must be compiled to Java bytecode classes before they can be executed, but such compilation is needed only when a change to the source JSP file has occurred.

The JSP syntax adds additional XML-like tags, called JSP actions, to invoke built-in functionality. Additionally, the technology allows for the creation of JSP tag libraries that act as extensions to the standard HTML or XML tags. JVM operated tag libraries provide a platform independent way of extending the capabilities of a web server. Note that not all commercial Java servers are Java EE specification compliant.

JSPs are compiled into servlets by a JSP compiler. The compiler either generates a servlet in Java code that is then compiled by the Java compiler, or it may compile the servlet to byte code which is directly executable. JSPs can also be interpreted on-the-fly, reducing the time taken to reload changes.

JavaScript.

JavaScript is an implementation of the ECMAScript language standard and is typically used to enable programmatic access to computational objects within a host environment. It can be characterized as a prototype-based object-oriented scripting language that is dynamic, weakly typed and has first-class functions. It is also considered a functional programming language like Scheme and OCaml because it has closures and supports higher-order functions.

JavaScript is primarily used in the form of client-side JavaScript, implemented as part of a web browser in order to provide enhanced user interfaces and dynamic websites. However, its use in applications outside web pages—for example in PDF-documents, site-specific browsers and desktop widgets—is also significant. JavaScript uses syntax influenced by that of C. JavaScript copies many names and naming conventions from Java, but the two languages are otherwise unrelated and have very different semantics. The key design principles within JavaScript are taken from the Self and Scheme programming languages.

AJAX.

Ajax (shorthand for Asynchronous JavaScript and XML) is a group of interrelated web development methods used on the client-side to create interactive web applications. With Ajax, web applications can retrieve data from the server asynchronously in the background without interfering with the display and behavior of the existing page. Data is usually retrieved using the XMLHttpRequest object. Despite the name, the use of XML is not needed, and the requests need not be asynchronous.

Like DHTML and LAMP, Ajax is not one technology, but a group of technologies. Ajax uses a combination of HTML and CSS to mark up and style information. The DOM is accessed with JavaScript to dynamically display, and to allow the user to interact with the information presented. JavaScript and the XMLHttpRequest object provide a method for exchanging data asynchronously between browser and server to avoid full page reloads.

The term Ajax has come to represent a broad group of web technologies that can be used to implement a web application that communicates with a server in the background, without interfering with the current state of the page. The following technologies are incorporated in AJAX:

- HTML or XHTML and CSS for presentation
- the Document Object Model (DOM) for dynamic display of and interaction with data
- XML for the interchange of data, and XSLT for its manipulation
- the XMLHttpRequest object for asynchronous communication
- JavaScript to bring these technologies together

Google Maps.



Google Maps is a web mapping service application and technology provided by Google, free (for non-commercial use), that powers many map-based services, including the Google Maps website, Google Ride Finder, Google Transit, and maps embedded on third-party websites via the Google Maps API. It offers street maps, a route planner for traveling by foot, car, or public transport and an urban business locator for numerous countries around the world. Google Maps satellite images are not in real time; they are several months or years old.

Like many other Google web applications, Google Maps uses JavaScript extensively. As the user drags the map, the grid squares are downloaded from the server and inserted into the page. When a user searches for a business, the results are downloaded in the background for insertion into the side panel and map; the page is not reloaded. Locations are drawn dynamically by positioning a red pin (composed of several partially-transparent PNGs) on top of the map images.

A hidden IFrame with form submission is used because it preserves browser history. The site also uses JSON for data transfer rather than XML, for performance reasons. These techniques both fall under the broad Ajax umbrella.

Google Maps API.

Google launched the Google Maps API in June 2005, to allow developers to integrate Google Maps into their websites. It is a free service, and currently does not contain ads, but Google states in their terms of use that they reserve the right to display ads in the future.

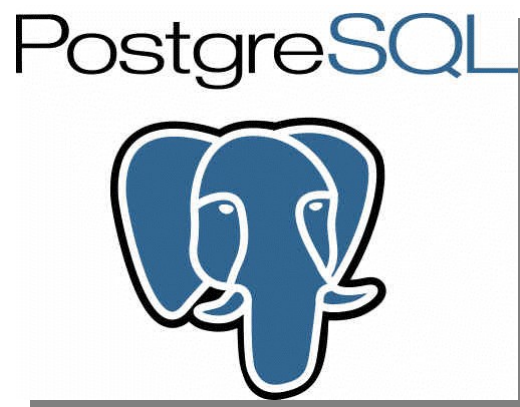
By using the Google Maps API, it is possible to embed Google Maps site into an external website, on to which site specific data can be overlaid. Although initially only a JavaScript API, the Maps API has since expanded to include an API for Adobe Flash applications, a service for retrieving static map images, and web services for performing geocoding, generating driving directions, and obtaining elevation profiles. Over 350,000 web sites use the Google Maps API, making it the most heavily used web application development API.

SQL.



Often referred to as Structured Query Language, is a database computer language designed for managing data in relational database management systems (RDBMS), and originally based upon relational algebra and calculus. Its scope includes data insert, query, update and delete, schema creation and modification, and data access control. SQL was one of the first commercial languages for Edgar F. Codd's relational model, as described in his influential 1970 paper, "A Relational Model of Data for Large Shared Data Banks". Despite not adhering to the relational model as described by Codd, it became the most widely used database language.

PostgreSQL.



PostgreSQL, often simply Postgres, is an object-relational database management system (ORDBMS). It is released under an MIT-style license and is thus free and open source software. As with many other open-source programs, PostgreSQL is not controlled by any single company — a global community of developers and companies develops the system.

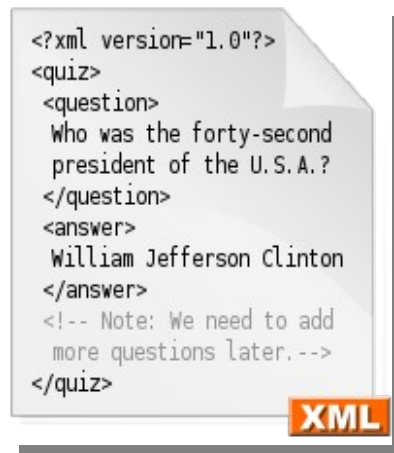
APACHE TOMCAT.



Apache Tomcat is an open source servlet container developed by the Apache Software Foundation (ASF). Tomcat implements the Java Servlet and the JavaServer Pages (JSP) specifications from Sun Microsystems, and provides a "pure Java" HTTP web server environment for Java code to run.

Tomcat should not be confused with the Apache web server, which is a C implementation of an HTTP web server; these two web servers are not bundled together. Apache Tomcat includes tools for configuration and management, but can also be configured by editing XML configuration files.

XML.



Extensible Markup Language (XML) is a set of rules for encoding documents in machine-readable form. It is defined in the XML 1.0 Specification produced by the W3C, and several other related specifications, all gratis open standards.

XML's design goals emphasize simplicity, generality, and usability over the Internet. It is a textual data format with strong support via Unicode for the languages of the world. Although the design of XML focuses on documents, it is widely used for the representation of arbitrary data structures, for example in web services.

Many application programming interfaces (APIs) have been developed that software developers use to process XML data, and several schema systems exist to aid in the definition of XML-based languages.

ECLIPSE.

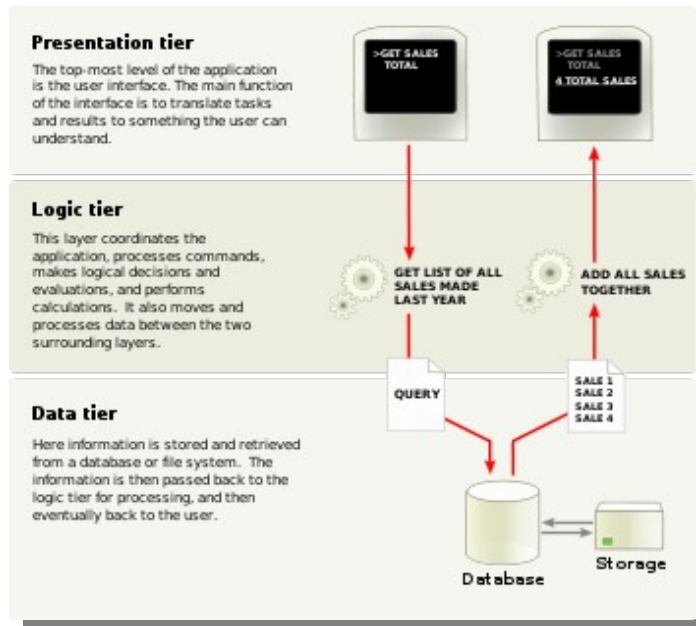


Eclipse is a multi-language software development environment comprising an integrated development environment (IDE) and an extensible plug-in system. It is written mostly in Java and can be used to develop applications in Java and, by means of various plug-ins, other programming languages including Ada, C, C++, COBOL, Perl, PHP, Python, Ruby (including Ruby on Rails framework), Scala, and Scheme. The IDE is often called Eclipse ADT for Ada, Eclipse CDT for C/C++, Eclipse JDT for Java, and Eclipse PDT for PHP.

The initial codebase originated from VisualAge. In its default form it is meant for Java developers, consisting of the Java Development Tools (JDT). Users can extend its abilities by installing plug-ins written for the Eclipse software framework, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules.

Released under the terms of the Eclipse Public License, Eclipse is free and open source software.

MULTITIER ARCHITECTURE.



Three-tier is a client–server architecture in which the user interface, functional process logic, computer data storage and data access are developed and maintained as independent modules, most often on separate platforms.

The three-tier model is a software architecture and a software design pattern. Apart from the usual advantages of modular software with well-defined interfaces, the three-tier architecture is intended to allow any of the three tiers to be upgraded or replaced independently as requirements or technology change. For example, a change of operating system in the presentation tier would only affect the user interface code. Typically, the user interface runs on a desktop PC or workstation and uses a standard graphical user interface, functional process logic may consist of one or more separate modules running on a workstation or application server, and an RDBMS on a database server or mainframe contains the computer data storage logic. The middle tier may be multi-tiered itself (in which case the overall architecture is called an "n-tier architecture").

Three-tier architecture has the following three tiers:

PRESENTATION TIER

This is the topmost level of the application. The presentation tier displays information related to such services as browsing merchandise, purchasing, and shopping cart contents. It communicates with other tiers by outputting results to the browser/client tier and all other tiers in the network.

BUSSINES LOGIC

The logic tier is pulled out from the presentation tier and, as its own layer, it controls an application's functionality by performing detailed processing.

DATA TIER

This tier consists of database servers. Here information is stored and retrieved. This tier keeps data neutral and independent from application servers or business logic. Giving data its own tier also improves scalability and performance.

SPECIFICATION OF REQUIREMENTS.

DESCRIPTION.

A Software Requirements Specification (SRS) is a complete description of the behavior of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. Use cases are also known as functional requirements. In addition to use cases, the SRS also contains non-functional requirements. Non-functional requirements are requirements which impose constraints on the design or implementation.

OBJECTIVE.

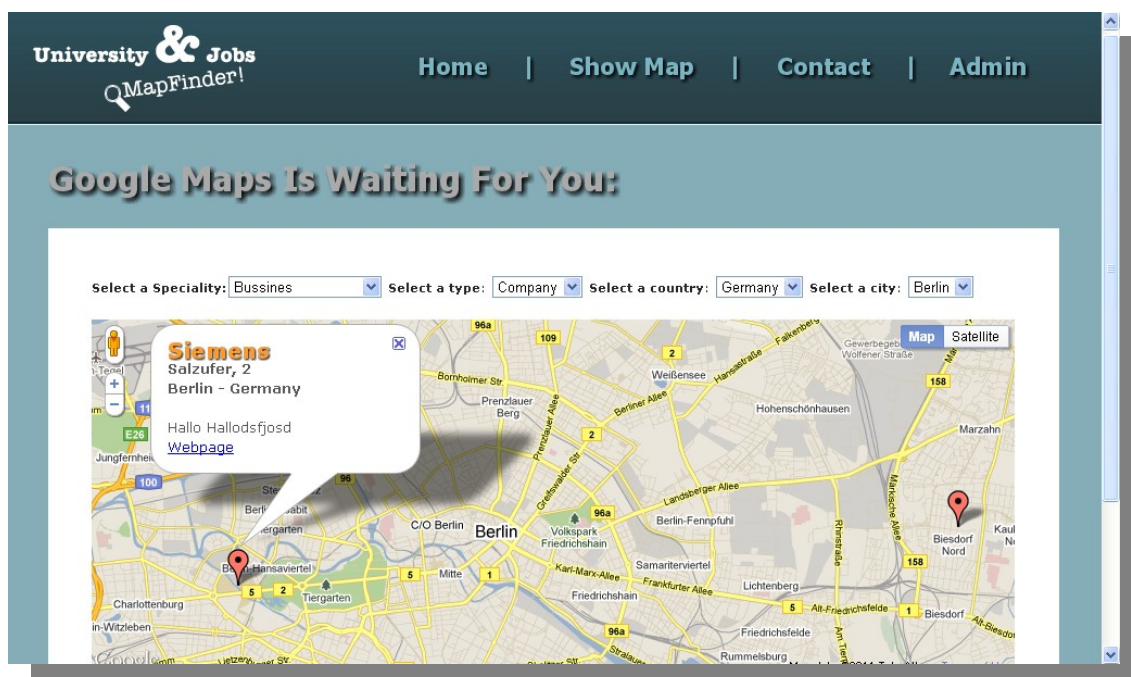
In this section, the objective is to detail the specific requirements that this application needs, as well the information related to the analysis of the application.

This application analysis will oscillate according to the scenarios and characters which has this web site, then, it will be a way to justify decisions taken to develop the project.

OVERALL DESCRIPTION.

Universities & Job - Map Finder! is a web-application or specialized portal that offers the chance for the new-graduates to get a first job or internship, besides of to find a university where the future-students carry out their studies, this, in any speciality or any city of the world.

When the user enter in the website, it appears a menu, with "Home", "Show Map", "Contact" "Admin", but we are going to focus in the main function "Show Map", which, through 4 combo boxes, among all the available specialities in the data base we can choose one of our liking, the type available for that speciality, it can be university, company or both, and the country and city where there are some university or company available for our request. Then, it will appear a map of Google Maps, with the markers in the correct position where the companies/universities are located exactly.



We can see the bubble, the name, address, comment about the company and web page to get in touch with the university/company.

As well, to complete the feedback, it exists an special area "Admin" where the administrator, introducing the password will access to the intranet, where he be able to register the applications from the universities/companies that the web page has received, likewise to delete any company/university that requires.

PRODUCT PERSPECTIVE.

Universities & Job - Map Finder! Is a project developed in JSP and JavaScript language, which will need a Web Server as Apache Tomcat 6.0 and a domain, besides of a data base to store all the information, which will be inserted by an administrator.

User Features.

In this project will have five types of users:

Pre-Student & New-Graduate: Both be able check the available universities to study an specific specialization or check for to realize some internship in a available company. They have two distincts profiles and each one can check everything in the web site because the content is public, thus, they are two but, in a some way, we could classify both as a 'generic user'.

University & Company: This type of user will be the responsible for to advertise in the website. Due to the intern implementation, the University or Company can't register themselves directly in the system due to the 'principle of user distrust'. Then, the universities/company will have to send a register application to join to the web site. Therefore, it will be necessary that an administrator manage the website and register/remove the new/old universities or companies.

Administrator: As above is detailed, and due to the 'principle of user distrust', it will be necessary that exists an administrator who manage the site, adding or removing the places. It's not compulsory that this administrator have some knowledge about programming, data base or computer, because there will be an a web interface to add or remove the places.

SERVICE REQUIREMENTS.

This section presents the functional requirements that must be satisfied by the system. All requirements shown here are essential and have been specified, considering all the software and hardware and services that this project have and need.

User Interfaces.

Both the Universities/Companies, pre-students/new-graduates and Admin, will need a browser and internet connection to check, manage or send an register application. Overcoat the admin, existing an interface in the intranet to add/delete the places. Thus, the admin doesn't need to face to the data base, having the web site a section to manage that kind of things.

Hardware Interfaces.

This application have been developed using JavaScript, Jsp, Ajax, Google Maps, PostgreSQL, XML files and a Tomcat Web Server.

Then, it will need internet, due to Api Google Maps is not integrated in the project, if not, this is linked on the code files by a link, that's why we depend of internet and Google Maps. As well, we need installed the Java Sun plugin in the browser and a Web Server (Tomcat) besides of a data base PostgreSQL.

OBJECTS.

Create new place:

Once the university or the company have sent the register application. If the admin pass the application, it will be registered in the system by the admin. And automatically it will appear in the options.

Attributes of the place:

DESCRIPTION	TYPE	EXAMPLE
Code	Serial – String (text)	'1'
Name	String (text)	'Siemens'
Address	String (text)	'Unter den Linden, 1'
City	String (text)	'Berlin'
Country	String (text)	'Germany'
Web	String (text)	'www.siemens.com'
Mail	String (text)	'info@siemens.com'
Type	String (text)	'Company'
Comment	String (text)	'Siemens is a company of...'
Speciality	String (text)	'Electronics'

Operations:

As you can see above of this, I don't care if the place is a company or a university, because in the table will be both together, and in the logic level I will differentiate among both, according the preferences of the user.

The main operation here will be the register of the place in the data base for then to be available in the preference's search if the user requests it.

MAINTAINABILITY REQUIREMENTS.

It exists two kinds of this kind of requirements:

Admin: Due to the 'principle of user distrust', the webmaster can't trust in the user, that's why it's necessary to create the character of the administrator, to insert in the data base the information about universities or companies that will appear in the options where to study or to get an internship for the user.

Google Maps: This website depends strictly on Google Maps and his API, being this one, the provider of the map where we can show the results of the search. Theoretically, this could be a negative point against us, but, everybody knows the reliability and seriousness of Google, and for sure that this service should to work properly and to add more usefully and simplicity to this web site.

CONCLUSION.

Thanks to the ERS (Software Requirements Specification), the developer realize that which things really need in his project. Thus, I think that it's a very good way or step to think in the future of the application and to distinguish among the different needs that each part of the website needs. Therefore, I could conclude that it has been a great help to know the most of the things that I will need to develop this project.

DEVELOP OF IMPLEMENTATION.

In this section, I'm going to detail the development process that I have carried out, describing the necessary previous steps to realize a project of Software Engineering.

ACTORS.

This application has two destiny users, one of them would be the students who want to study a specific studies and want to know where they can study it, and the other user would be the new graduates who are looking for an internship or to get the first job once their university period is over.

There are as well, 3 users else, University, Company and Manager. The University and the Company will send applications to appear in the search of the webpage, and the Manager will be the person in charge of to manage the website for the proper behavior of the site.



Pre-Student



New-Graduate



**Hochschule für Technik
und Wirtschaft Berlin**

University of Applied Sciences

University

SIEMENS

Company



Manager

USE CASES DIAGRAM.

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

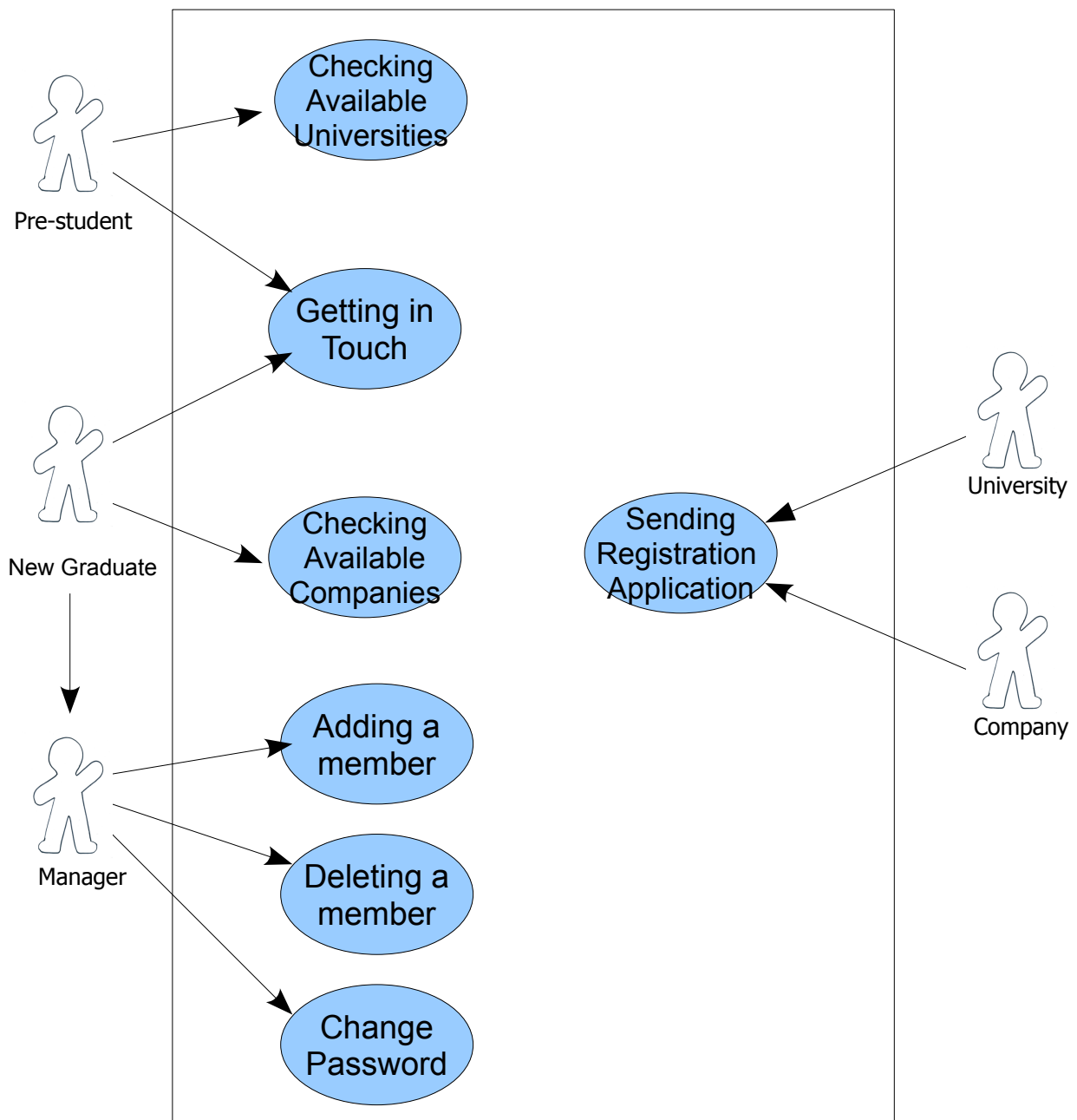
Use Case diagrams are formally included in two modeling languages defined by the OMG: the Unified Modeling Language (UML) and the Systems Modeling Language (SysML).

In my case, I have reflected all the uses that every users can reach and to realize from the website:

- Checking Available Universities {pre-student}
- Checking Available Companies {new graduate}
- Getting in touch with them {pre-student+new graduate}
- Sending Registration Application {universities+companies}
- Adding a Member {manager}
- Deleting a Member {manager}
- Change the Password {manager}

Next are detailed the above information down the context of a Use Case Diagram:

Use Case Diagram:



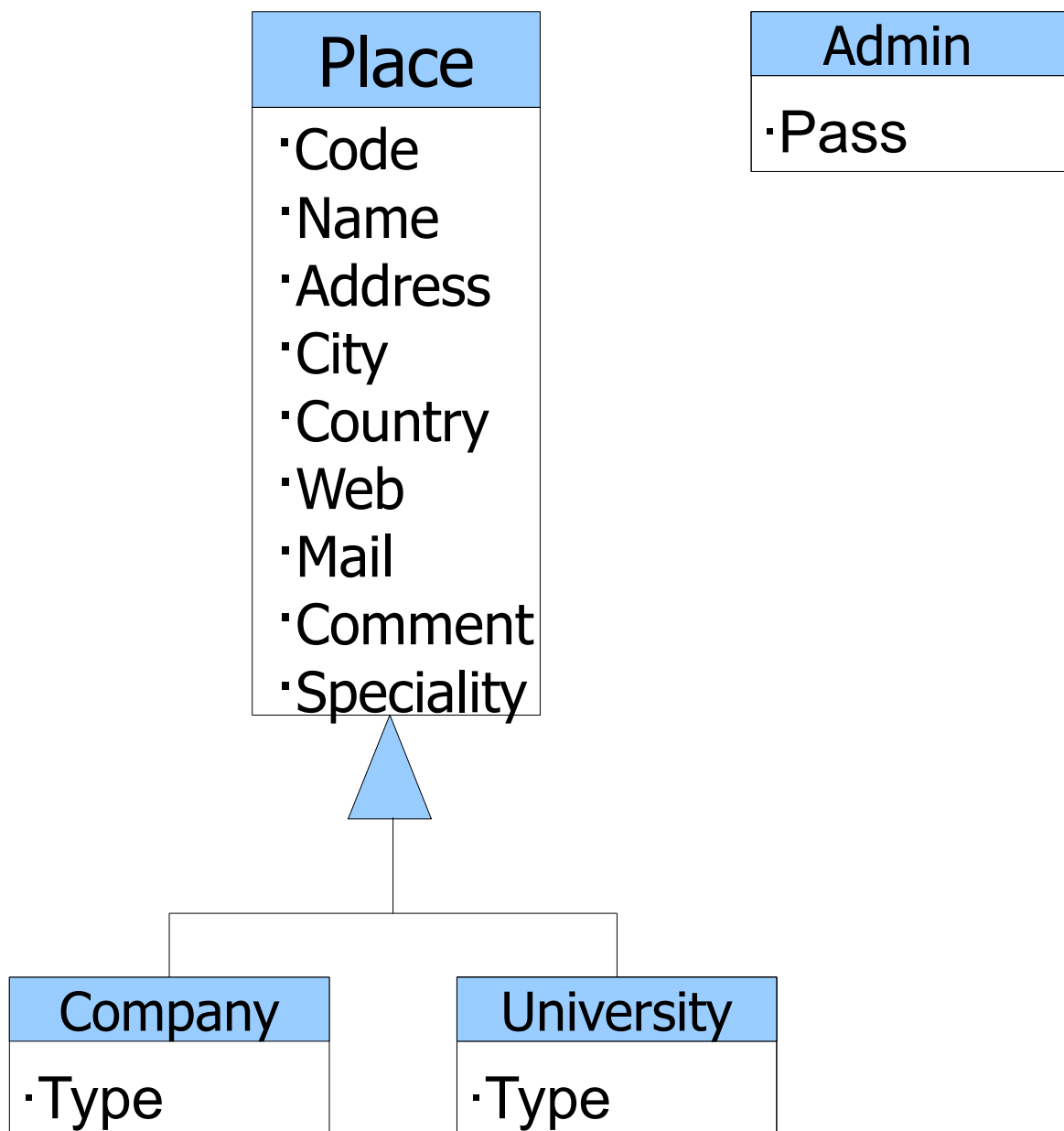
Cases Use Diagram.

CLASS DIAGRAM.

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes.

The class diagram is the main building block in object oriented modeling. It is used both for general conceptual modeling of the systematics of the application, and for detailed modeling translating the models into programming code. The classes in a class diagram represent both the main objects and or interactions in the application and the objects to be programmed.

In my case, the development has resulted very simple, because there isn't important relations in the internal logical of the project. Only there is input data about Universities or Companies which are shown according to the academic speciality of the user.



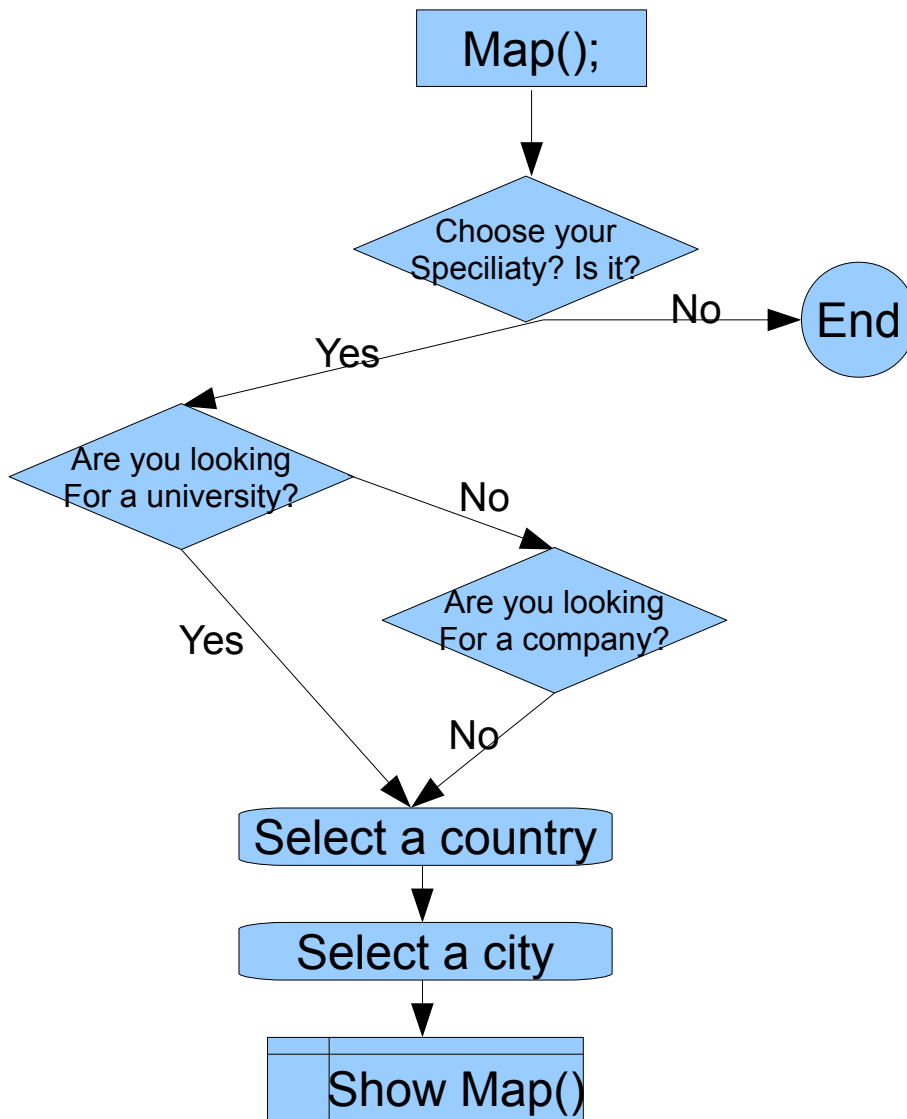
Class Diagram.

ACTIVITY DIAGRAM.

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency.

In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

In this case, I have represented the activity of the main function of the web site, it is the steps that the user have to reach to know which places can be interesting to get an internship or an University:



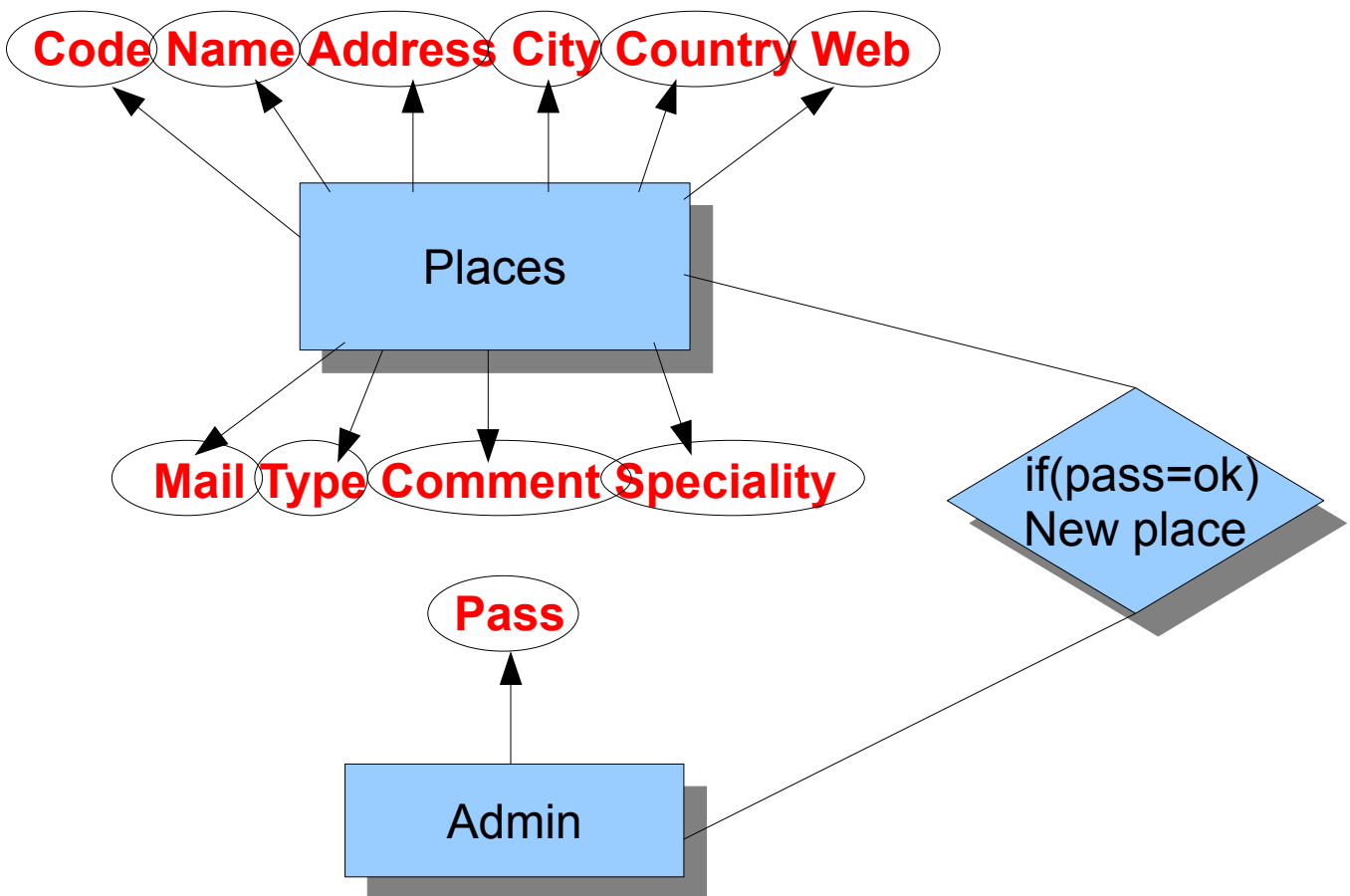
Activity Diagram.

RELATIONAL MODEL DESIGN.

The central idea was to describe a database as a collection of predicates over a finite set of predicate variables, describing constraints on the possible values and combinations of values. The content of the database at any given time is a finite (logical) model of the database, i.e. a set of relations, one per predicate variable, such that all predicates are satisfied. A request for information from the database (a database query) is also a predicate.

The purpose of the relational model is to provide a declarative method for specifying data and queries: we directly state what information the database contains and what information we want from it, and let the database management system software take care of describing data structures for storing the data and retrieval procedures for getting queries answered.

In my case, the requirements of the data base are very simple and easy, because we can have everything stored in just one table where will be each University and Companies and then, in the logical tier, I divide the information to show to the user according to the preferences of the user, Company or University. As well, It can see in the model that to add/delete a new place the admin have to insert properly the password of the system.



Data Base Model.

Table PLACES:

Cod: Code of the place - {PK} - **automatic serial**

Name: Name of the company/university - **text**

Address: Address of the company/university - **text**

City: City of the company/university - **text**

Country: Country of the company/university - **text**

Web: Web of the company/university - **text**

Mail: E-mail of the company/university - **text**

Type:Type of the place, can be university or company - **text**

Comment: Comment which will appear in the bubble of the marker in the map - **text**

Speciality: Speciality of the university or company - **text**

Table ADMIN:

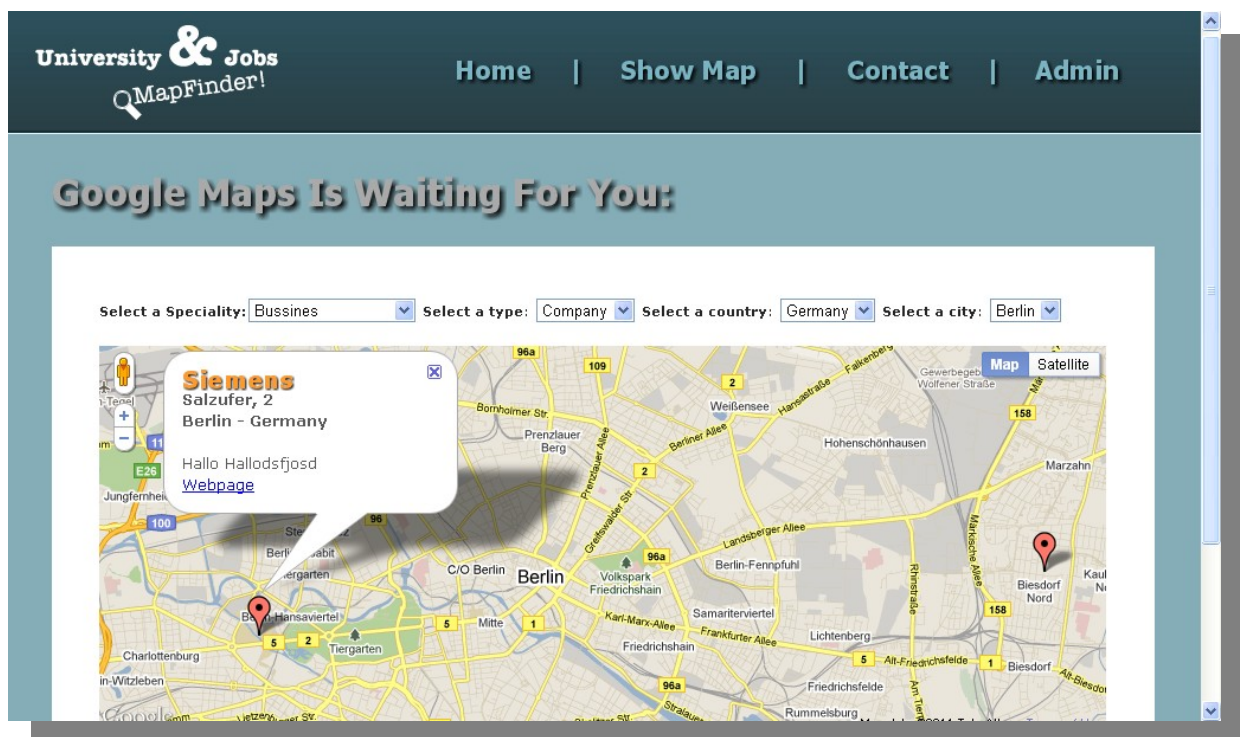
Pass: Password of the Admin Area - {PK} - **text**

APPLICATION ARCHITECTURE.

In software engineering, multi-tier architecture (often referred to as n-tier architecture) is a client-server architecture in which the presentation, the application processing, and the data management are logically separate processes. For example, an application that uses middleware to service data requests between a user and a database employs multi-tier architecture. The most widespread use of multi-tier architecture is the three-tier architecture.

Next are detailed how I he used this architecture with real examples of my application and their conections each other:

PRESENTATION TIER:



Logic Tier.

```

214     function createMarker (posi, nam, comm, add, ci, cou, w) {
215         var wi = 400;
216         var marker = new google.maps.Marker({
217             map: map,
218             animation: google.maps.Animation.DROP,
219             position: posi
220         });
221
222         var content='<span class="txt-map1">'+nam+'</br></span>'
223             +'<span class="txt-map2"><b>'+add+'</br>'
224             +'ci+ ' +'+cou+'</b></br></br>'
225             +'comm+'</br>'+
226             '<A href="http://'+w+' ">Webpage</A></br></span>';
227
228
229
230         var infoWindow = new google.maps.InfoWindow(
231             { content: content,
232               maxWidth: wi
233             });
234         google.maps.event.addListener(marker, 'click', function() {
235             infoWindow.open(map, marker);
236         });
237         markers.push(marker);
238     }

```

Persistence Tier.

	name text	address text	city text	country text	web text	mail text	type text	comment text	speci text
1	UPB	Calle Colon, 60	Barcelona	Spain	www.upb.com	upb@upb.com	University	HolaHola Como l	Compu
2	UPV	Calle Colon, 6	Valencia	Spain	www.upv.com	upv@upv.com	University	HolaHola Como l	Compu
3	Okode	Calle Alicante, 6	Valencia	Spain	www.okode.com	okode@okode.c	Company	Mola Okode Vale	Compu
4	Hospital Univer	Calle Valencia, 6	Alicante	Spain	www.hua.com	hua@hue.com	Company	hospital ffd alic	Health
5	Facultad de Enfi	Gran Via, 6	Madrid	Spain	www.hm.com	hm@hm.com	University	facultad madrid	Health
6	Facultad de Bell	Gran Via, 77	Madrid	Spain	www.hm.com	hm@hm.com	University	bellas artes	Arts
7	Pintame SL	Calle Colon, 77	Madrid	Spain	www.hm.com	hm@hm.com	Company	bellas artes2	Arts
8	Construcciones	Calle Valencia, 7	Alicante	Spain	www.hm.com	hm@hm.com	Company	mola construir	Archite
9	Diseños Perez	MunzStrasse,2	Berlin	Germany	www.o33.com	ff@tr.es	Company	halloo	Arts
10	Universidad de /	Calle Colon, 16	Albacete	Spain	"	"	University	Com mola	Archite
11	Universidad de f	Calle Colon, 1	Lleida	Spain	"	"	University	Com mola	Health
12	Universidad de f	Calle Colon, 1	Lleida	Spain	"	"	University	Com mola	Health
13	Fachochsule Ber	Treskowalle, 1	Berlin	Germany	"	"	University	Com mola	Health
14	Fachochsule Hai	Treskowalle, 1	Hanburg	Germany	"	"	University	Com mola	Health
15	Fachochsule Köl	Treskowalle, 1	Köln	Germany	"	"	University	Com mola	Health
16	Universite Paris	Champs Elisees, Paris	France	"	"	"	University	Com mola	Health
17	Hospital Paris	Champs Elisees, Paris	France	"	"	"	Company	Com mola	Health
18	Hospital Nantes	Nantes , 1	Nantes	France	"	"	Company	Com mola	Health
19	Universite Nanti	Nantes , 1	Nantes	France	"	"	University	Com mola	Health
20	University AAm	Amsterdam , 1	Amsterdam	Holland	"	"	University	Com mola	Health
21	Hospital Amster	Amsterdam , 1	Amsterdam	Holland	"	"	Company	Com mola	Health
22	Hospital Munich	AmsterdamStr,	Munich	Germany	aa	aaa	Company	Hallooo	Health
23	Hospital Lübeck	AmsterdamStr,	Lübeck	Germany	aa	aaa	Company	Hallooo	Health
24	Universidad de f	Calle Bilbao,1	Pamplona	Spain	www.up.com	up	University	Holaaaa	Compu
25	Autoescuela Per	Calle San Vicent	Valencia	Spain	www.sss.es	ss	University	asasa	Driver

INTERFACE & PROTOTYPE DESIGN.

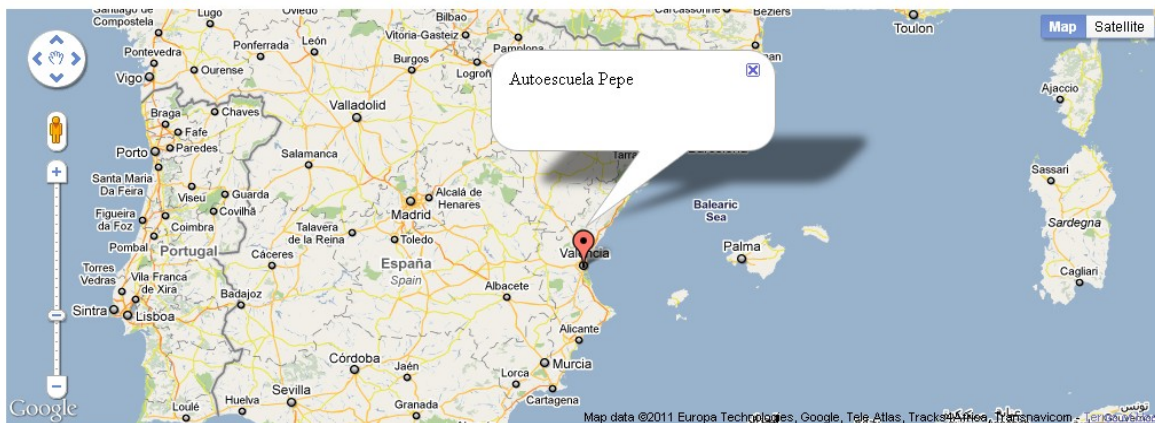
Before to nothing, once I understood all the requirements that I needed, I drew on paper a prototype of the web page, design, windows, way to connect every window, workflow, and data base design.

After that, I began to develop whole the main function about the map, to interact between data base and the map, and can show all the information stored in the data base, once got it, I developed the style of the web page using CSS's and tables to build a simple and intuitive web page able to transmit to the user the simplicity that really it doesn't exist internally.

Below of this It can see a screen capture about the aspect before to apply CSS's in the develop period:

Google Maps is waiting for your input

Select a Speciality: Select a type: Select a country: Select a city:

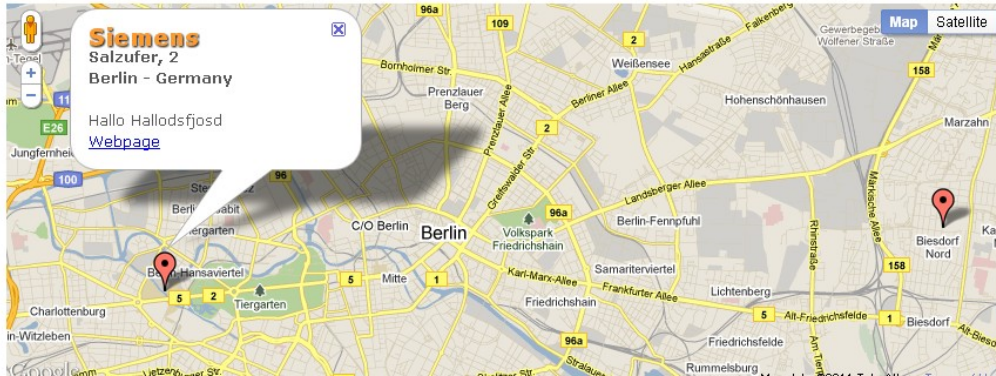


Note that any CSS was created and each bubble to lack of a internal design and I used the minimum information to test that the web page was working properly, and every data was circulating in a good way arriving to the function/service/data base suitable.

Then, once all the services and main functions was working correctly I carried out to add the style to the project:

Google Maps Is Waiting For You:

Select a Speciality: Select a type: Select a country: Select a city:



Below of this, we can see the internal menu for the administer with the CSS's activated, once introduced the password correctly, we would see this, clicking in **Create Map Marker**:

Create Map Marker

Speciality: [New Speciality?](#)

Type: University Company

Name:

Address:

City:

Country:

Web:

E-mail:

Comment:

IMPLEMENTATION.

The real implementation of the web page could have been in PHP, ASP or several dynamic languages, but I decided develop it in Java Server Page, due to better acknowledge and trust in the behavior of Java language.

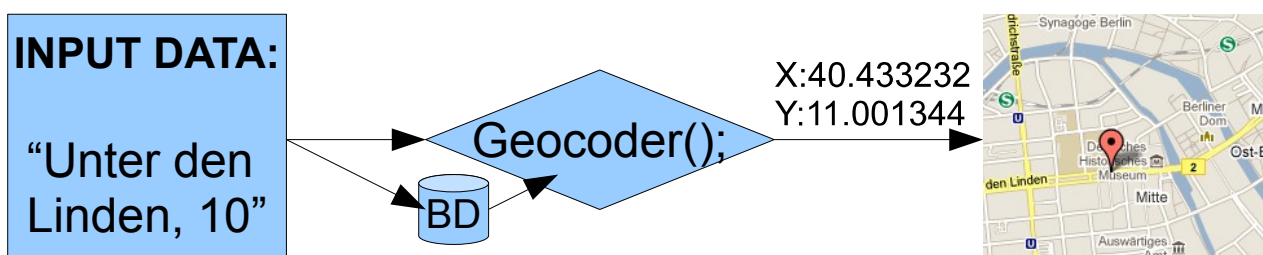
I have used Eclipse, which it is a development environment oriented to Java and it offers facilities to debugging and to integrate and manage the Apache Tomcat server. This tool is developed by OTI and his purpose is to design and programming with a organized way, centralizing all the resources and services.

The most of the functions have been realized in Javascript which is a language that runs on the client-side and very reachable and useful to programming web sites, as I have realized, in addition, If we mix this programming language with the API of Google Maps, possibilities multiply can develop, thanks of the services that Google Maps offer us. With a very good documentation in the site:

<http://code.google.com/intl/es/apis/maps/documentation/javascript/basics.html>

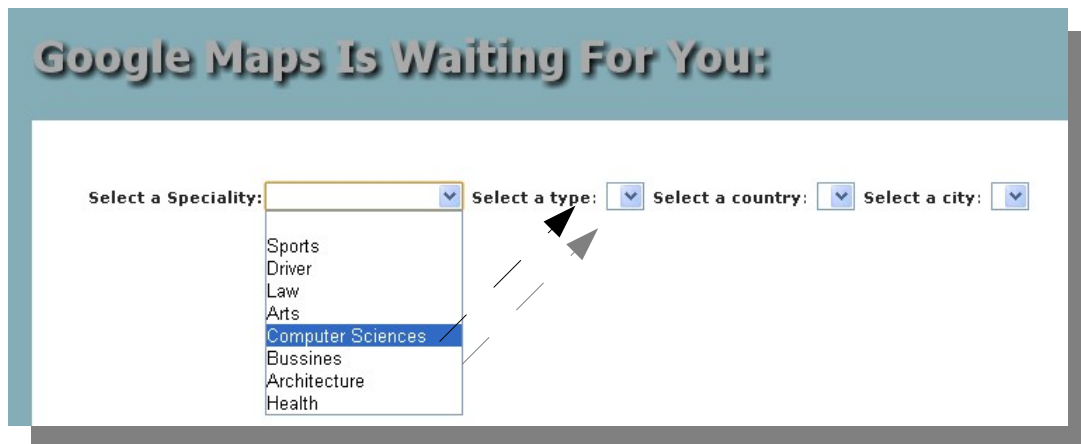
We can use a lot of services and methods related with Maps, Markers, Polygons, Polylines, InfoWindows, Street View, Layers, and one of the most famous and important for this project, the Geocoder service. One of the troubles that I thought in the previous analysis part was to find the way to keep the marker's addresses with dynamical way in the data base, as the user/admin doesn't know about coordinates, just streets, avenues.. Geocoder is a method that provide Google Maps to translate physical addresses, like, "Karl-Marx Strasse, 10" to coordinates, of this way: "X:-50.434327 , Y: 4.399343 "

Then, logically, the application only will store the addresses, and then, in the source code, it will be translated to coordinates, such as Google Maps works with the maps.



It is worth pointing out that the Geocoder service is not 100% right sometimes, because maybe one street can be registered in Google Maps data base with a different way that really it is. Real Example: "Calle San Vicente Martir, 10" would be wrong; right way would be: "Calle **de** San Vicente Martir, 10" or "Calle Colón, 1" would be wrong again, and the right way "Calle **Cristobal** Colón, 1". So that's why, it is necessary an admin or manager to make sure that the information are in a right way.

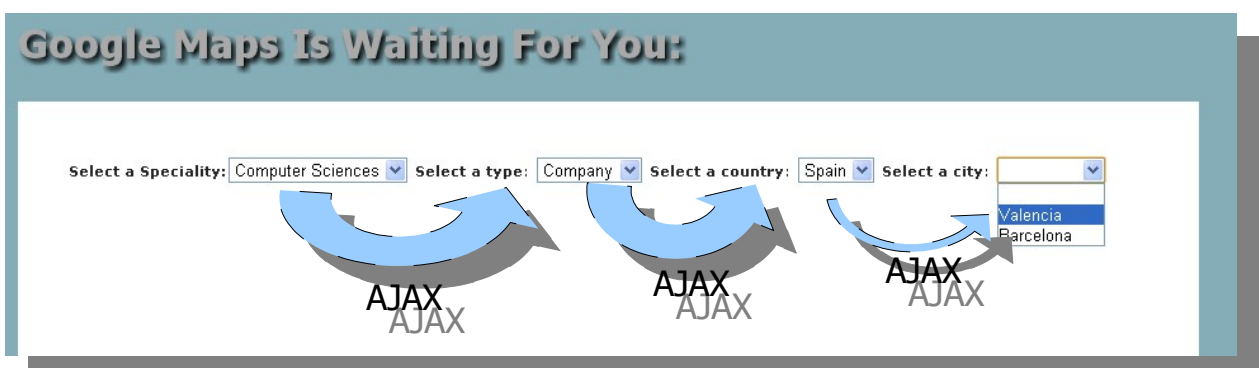
Once solved this problem, I was thinking about the logic design of the main function, "Show Map". This one, had to have several combo box to choosing the preferences of the search, but I decided to do it, with the latest features that are currently using, and I thought about to nest 4 combo boxes, such as show the next image:



Just in the moment at the user choose '**Computer Sciences**' the application by Javascript are sending the speciality chosen in a query to the data base, to know if there is some university or company about Computer Sciences. Well, maybe you are wondering right now, how the application, or in a better way, how JavaScript can send a query to the data base without refresh the web page and to show the available types, when this language is in the client side. Very simple, AJAX is the solution.

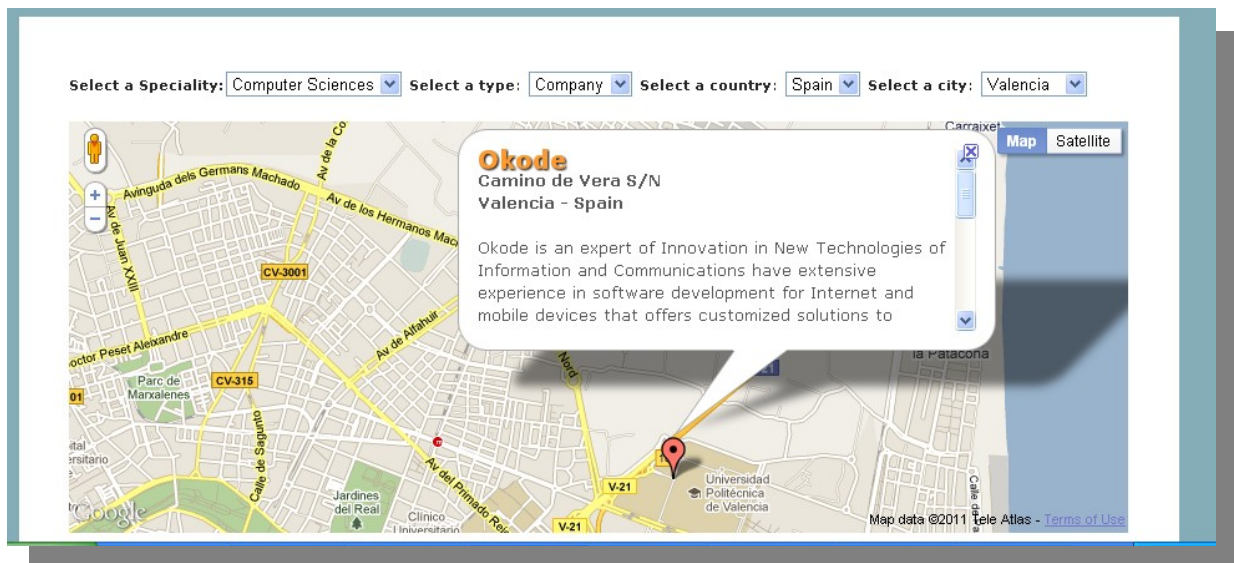
AJAX is a group of interrelated web development methods used on the client-side to create interactive web applications. With AJAX, web applications can retrieve data from the server or data base asynchronously in the background without interfering with the display and behavior of the existing page. Data is usually retrieved using the XMLHttpRequest object. Despite the name, the use of XML is not needed, and the requests need not be asynchronous.

In my case, I created with JavaScript a function called "DownloadURL()" using the AJAX technologies which take a dynamic file in JSP (getType(), getCities(), getCountries()), previously created by myself that it (JSP file) realizes the query with the data selected in the previous combo box and it creates a XML archive with the information which will be appear in the next combo box, once the XML arrives to the client part, there is a JavaScript function (parsexml();) which carry out the translation from XML to fill the next combo box, and like that, the following combo boxes:



Once the user click in the city of last combo box, the system calls to "ShowMap(city);"

where 'city' will be the last value that we need to do the last query to the data base to show the map with the markers pointing the available places:

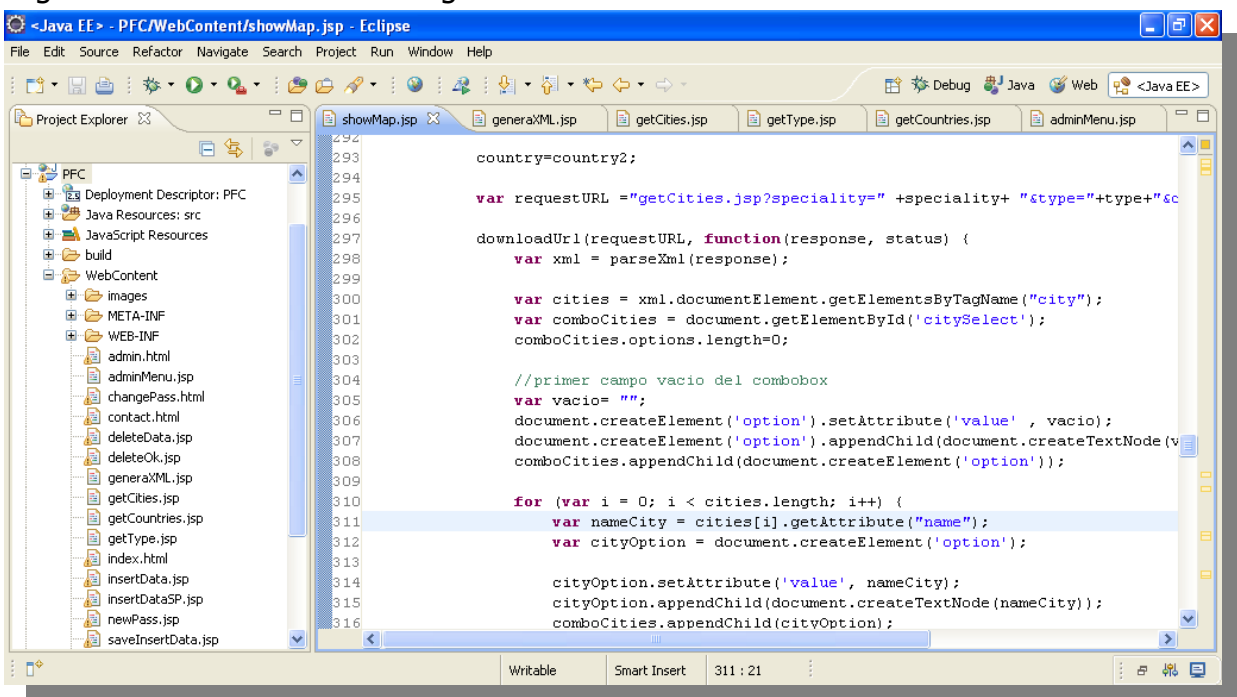


The browser used in the development have been Google Chrome and Mozilla Firefox because they are the browsers most used currently and their facilities to debug with their developers tools, but once finished the application I tested it with Internet Explorer 8.0, Google Chrome 9 and Mozilla Firefox 3.6.

The web server used is Apache Tomcat 6.0, the most commonly used and the best in web projects due to his simplicity, efficiency and proper behavior.

About the data base I have used PostgreSQL and pgAdmin III, in my task to develop about data bases have been simple and I used it because I have good acknowledges about this tool.

The development environment that I have used have been 'Eclipse Galileo' and I have programmed 18 files and it is organized like that:



index.html: This is the main page, there we can find the public menu with all the services that the website provides the user, in this case: "Home", "Show Map", "Contact", "Admin".

contact.html: Here are detailed instructions for a company or university to register in the system.

showMap.jsp: This is the largest file and where is developed the main function of this project. The user have chosen the values according to his academic preferences and showMap receive an XML file with whole the information about markers that has to show to him in the map and to parse it to know where the user can realize an internship or the Universities where can study.

getType.jsp: getType receive a parameter with the speciality chosen by the user and make a query to the data base to import the available types for that speciality: University or/and Company, in XML format.

getCountries.jsp: getCountries receive two parameters with the speciality and type chosen by the user and make a query to the data base to import the available countries for that specification, in XML format.

getCities.jsp: getCites receive three parameters with the speciality, type and country chosen by the user and make a query to the data base to import the available cities for that specification, in XML format.

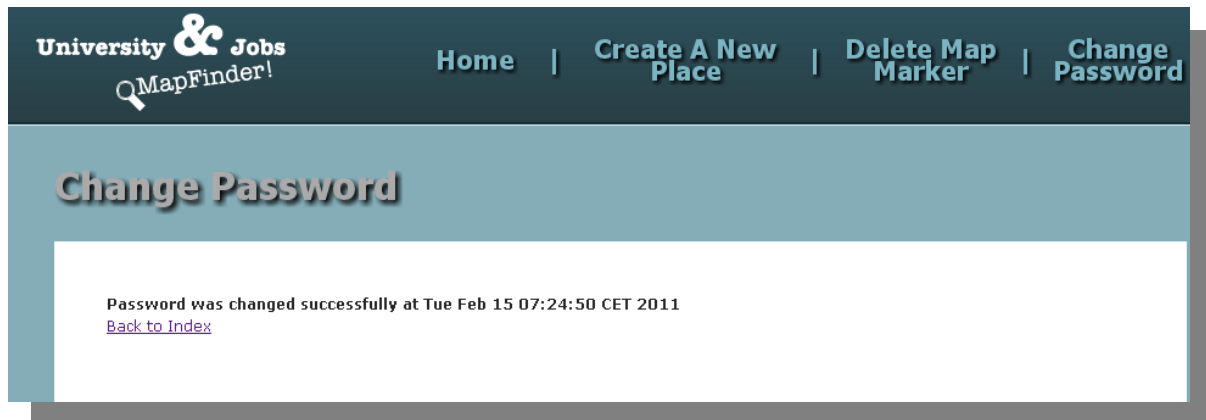
generaXML.jsp: generaXML receive the four parameters with the speciality, type, country and city chosen by the user and make a query to the data base to import the available information markers for that specification, in XML format.

admin.html: This is the page where the admin has to log in, introducing correctly the password of the system to get in the private Admin Menu.

adminMenu.jsp: If the password is correct, **the administrator** get in a private area to manage the site with the options: "Create a Marker", "Delete a Marker" and "Change the Password". If the password is wrong, **the user** will be redirected to the home-page.

changePass.html: The administrator can change the current password, if he write correctly the old password and write twice the new one.

newPass.jsp: If the administrator forget some field or it's written wrongly, the system will be notice him about wrong old password or new passwords are not the same. In successful case it will be notified.



insertData.jsp: Through this file the admin can register the new companies or universities which previously have sent the register-application and are interested in join to the web site.

insertDataSP.jsp: This file works equally that "insertData.jsp" with the difference that the administrator can register a new speciality if it isn't in the system.

saveInsertData.jsp: This file is responsible to check the new input data are correct, and none field have been forgotten. If some field were empty would happen this:

The image shows a screenshot of a web form titled 'Create Map Marker'. The form is set against a light blue background. It contains several input fields: 'Speciality' is a dropdown menu with 'Bussines' selected; 'Type' has two radio buttons for 'University' and 'Company', with 'Select again!' in red text to the right; 'Name' is a text box containing 'HTW'; 'Address', 'City', and 'Country' are text boxes, each with a red error message 'No empty field' to its right; 'Web' is a text box containing 'www.htw-berlin.com'; 'E-mail' is a text box containing 'info@htw-berlin.com'; and 'Comment' is a large text area with a red error message 'No empty field' to its right. A 'Submit' button is located at the bottom of the form.

saveInsertDataSP.jsp: This file does the same that "saveInsertData.jsp" with the modification that the speciality field is a text box and the administrator may write a new one. It is worth pointing out that the system remember the fields already are written, and as you can see, next to the forgotten fields write a error message.

deleteData.jsp: This is the responsible to offer to the administrator the possibility to delete some marker of the data base. There is a list with every name of the company or university and the administrator choose which want to delete it.

deleteOk.jsp: deleteOk just show the confirmation or error message if the deleted process was successful or not.

style.css: This is where are written the styles different of the parts of the web site.

CONCLUSIONS & PERSONAL OPINION.

Today, internet is a very good tool to do everything, from shopping at the supermarket until to get a job. That's why this project try to reflect the use that currently is having internet in the society.

The idea of can get an internship or to find the proper university to carry out studies are in the same line that the latest advances in web technology.

As everybody knows, everything is in internet, and in this case, if you are looking for an university or a company where to make an internship, in internet you can find the solution.

That's why I made this application, to give a chance to the people who has most complicate to find a first job or to decide where to study.

From logical view point, this project have been about research in the services that Google Maps offers to the developers, I could say that I have been more than 66% of the development time, looking for services, checking the Google Maps API, learning JavaScript, summarizing, **researching**. That's why I consider that I really did a Research Project about Google Maps and How to use it in a business context.

FUTURE WORK.

Well, there are several ways to improve in the future this application. Below are detailed some improving:

Creating User Profile:

Following the sense of a social network, the application could create user profiles where the user could detail his personal data and his curriculum vitae, with all the previous experience and his exact preferences about his study field. Thus, the application will gain effectiveness, and the companies be able find the proper candidates to finally to find the perfect new-graduate to contract him.

Application to create Curriculum Vitae:

Of course, If I would improve the application with a page where it can create a curriculum vitae online, with all the previous experience of jobs, studies and important information about the user.

Registering job offers:

This application would gain effectiveness if the companies could register and explain themselves the job offers and manage it. Thus, the new graduates would find easy and exactly the proper first job according to their preferences besides of quick.

Creating a forum:

it would be a good idea, to create a forum where the users can discuss about their preferences, companies and experiences exchange. As well it would be a way to help every user and company to choose the correct candidate.

Appendix I: Example Of Application.

This section I will show the options of this application, from a point of view graphic, with screen-shots and explanation about how to use the application and how to manage it.

The objective of this is to give to the user (or admin) of the application a easy guide of how to use it and to know all the features and sections that the web page has.

I will give a short description of the main actions and functions of this web page from the point of view of the user or the admin:

USER:

-Checking Places: Here is where the user can check the available universities and companies.

-Register-Application: This is about the process of register of some new university or company.

ADMIN:

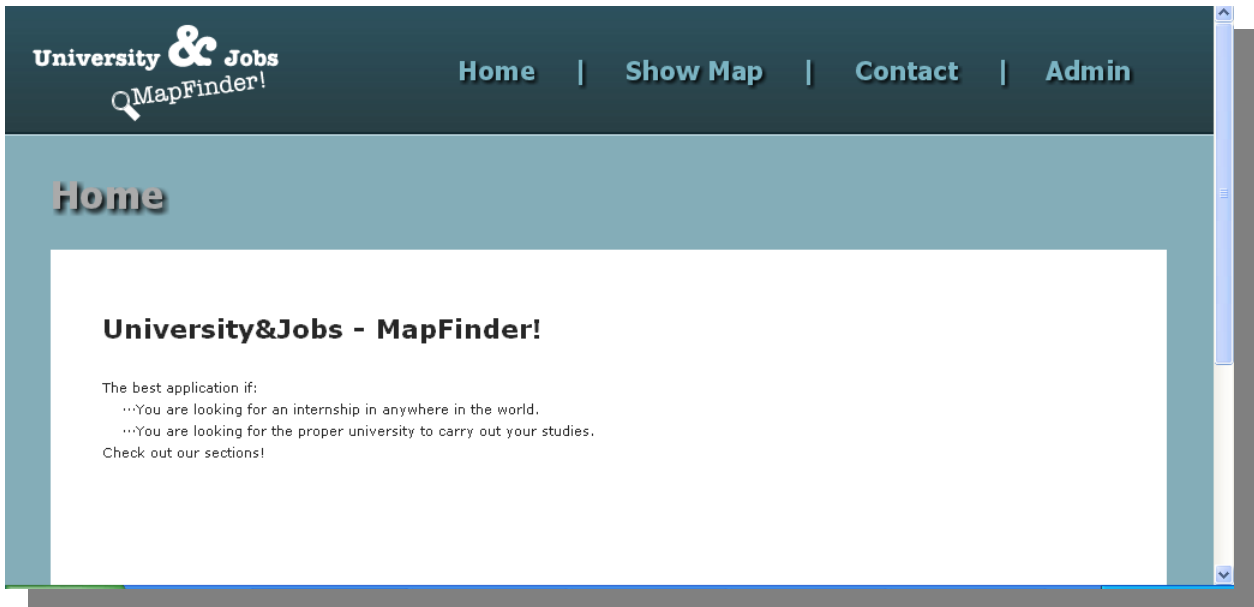
-Adding Places: Site from where the admin can register the new companies or universities which want to join the web page.

-Removing Places: Site from where the admin can delete the companies or universities which the admin added.

-Changing Password: This is the place where the administrator can change the password of the system.

Previous Steps.

In each case, we will have to have an internet connection, and through a browser as could be Mozilla Firefox or Google Chrome, get into the web page of this project and we will see this:



As we can see, it has a menu with several options:

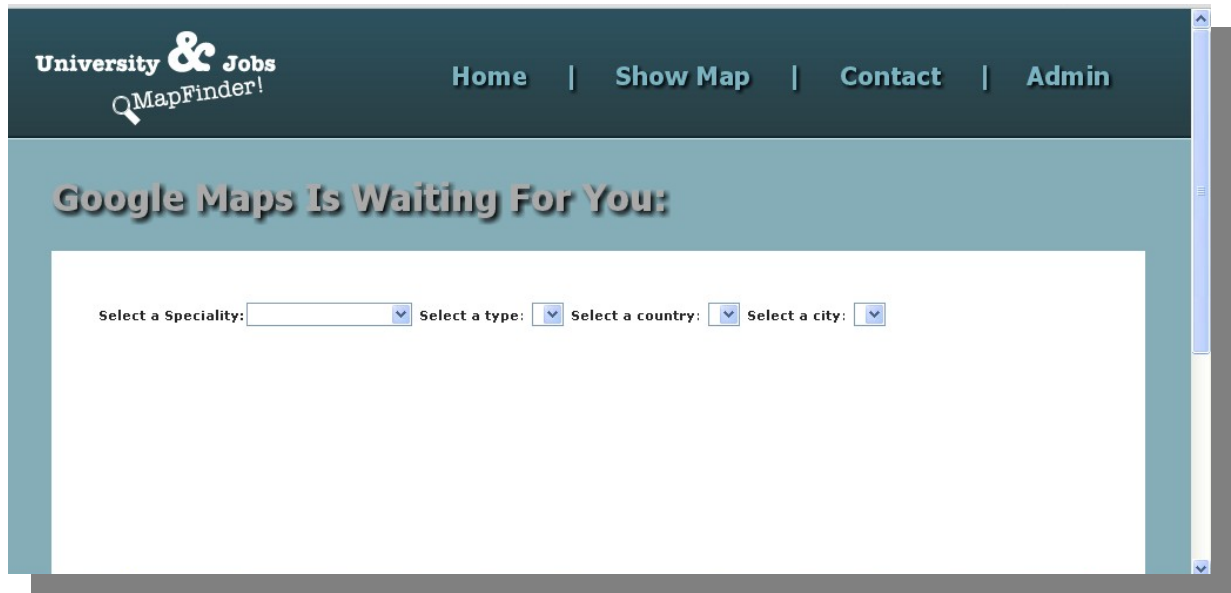
- Home**
- Show Map**
- Contact**
- Admin**

And a small explanation about the objective of the web page.

USER:

Checking Places:

The first, once we have get into the web site, if we click in "Show Map" we will see the next:



The screenshot shows the top navigation bar of the website with the logo 'University & Jobs QMapFinder!' and links for 'Home', 'Show Map', 'Contact', and 'Admin'. Below the navigation bar, the main heading reads 'Google Maps Is Waiting For You:'. Underneath this heading is a search form with four dropdown menus: 'Select a Speciality:', 'Select a type:', 'Select a country:', and 'Select a city:'.

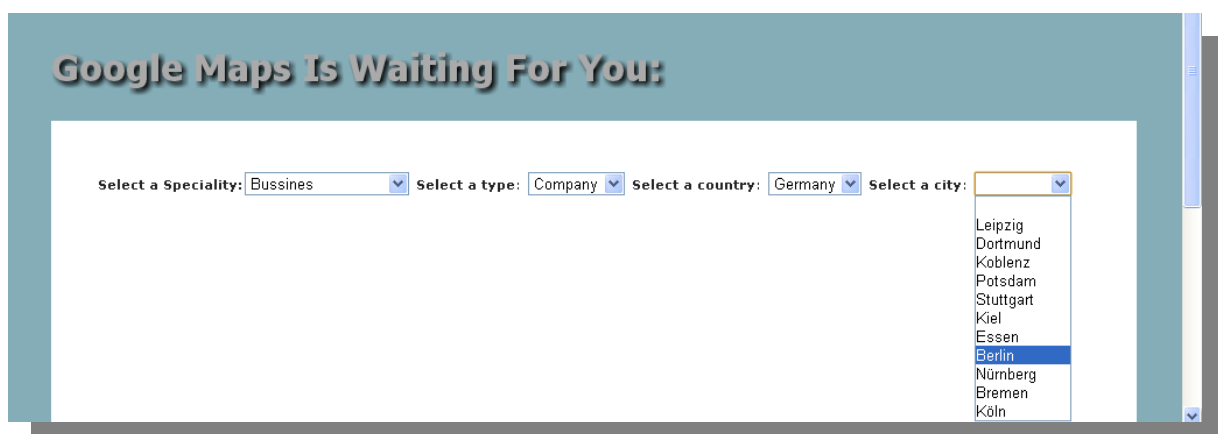
And we can see 4 combo box's:

-**Speciality:** Here, the system will show the available specialities that currently ara in the data base about companies and universities.

-**Type:** Type means "University" or "Companies". According to the preferences of the user, he will want to know information about a company or university.

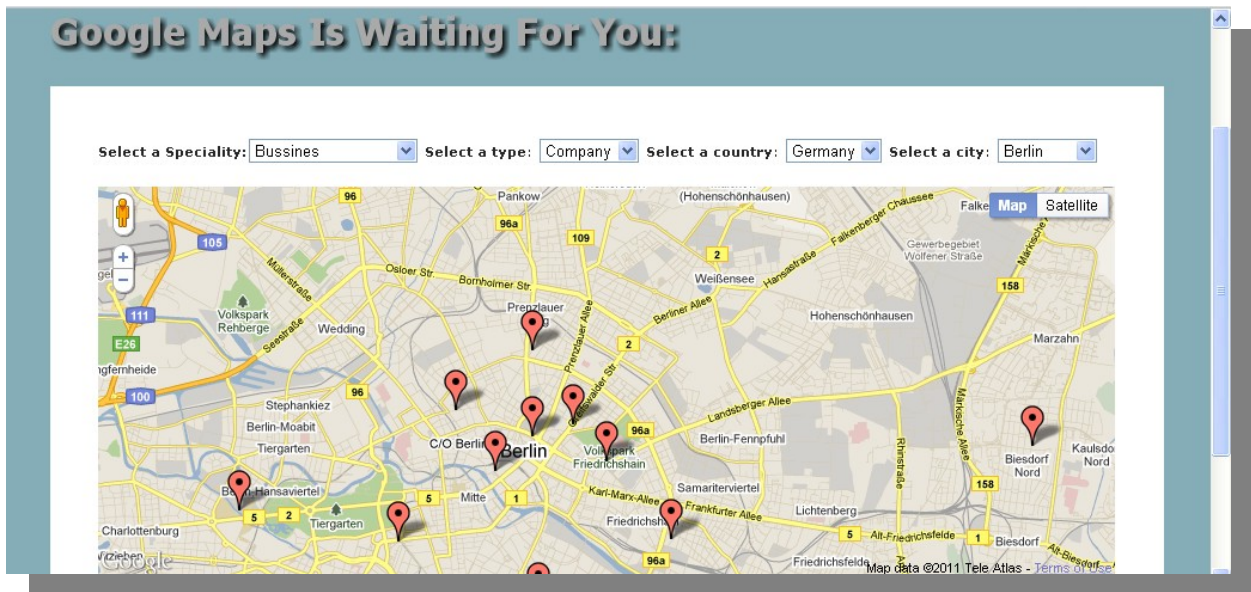
-**Country:** As you can imagine, in this combo box will appear the countries available for the type and speciality previously selected.

-**City:** And in this combo box, it will appear the available cities in the country selected with the type and speciality chosen.



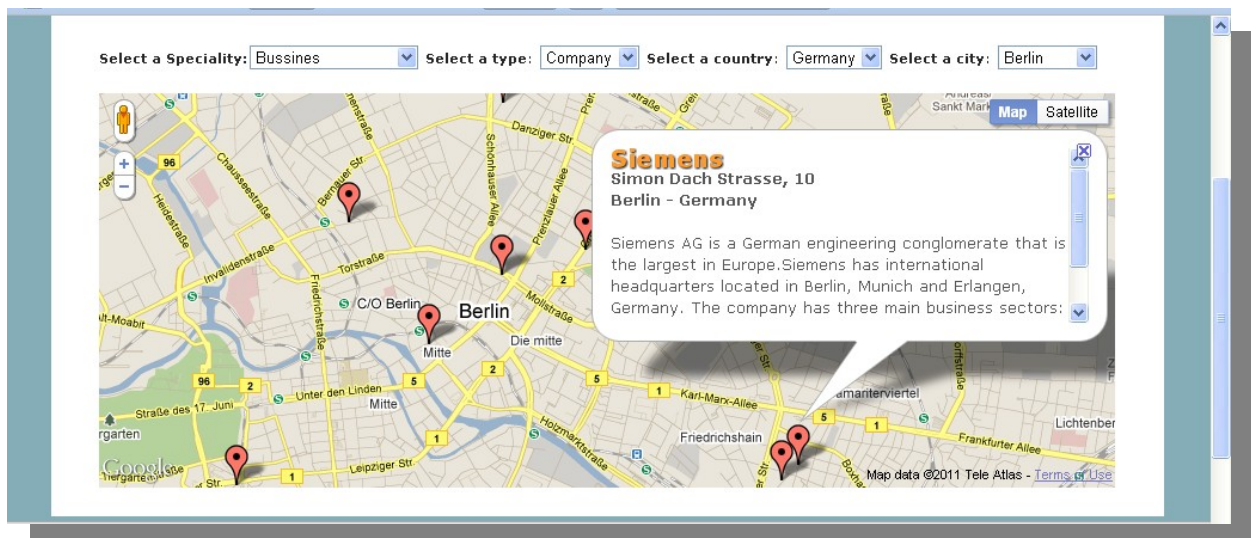
This screenshot shows the search form with the following selections: 'Bussines' in the 'Select a Speciality:' dropdown, 'Company' in the 'Select a type:' dropdown, and 'Germany' in the 'Select a country:' dropdown. The 'Select a city:' dropdown is open, displaying a list of cities: Leipzig, Dortmund, Koblenz, Potsdam, Stuttgart, Kiel, Essen, Berlin (highlighted), Nürnberg, Bremen, and Köln.

As you can see above of this, just when the user click in the city that he has interested, it will charge a map of Google with the information of the companies which has job offers for new graduates of, in this case, Bussiness in Berlin (Germany):



Currently, in Berlin there are more than 11 Companies interested in get some new-graduate!

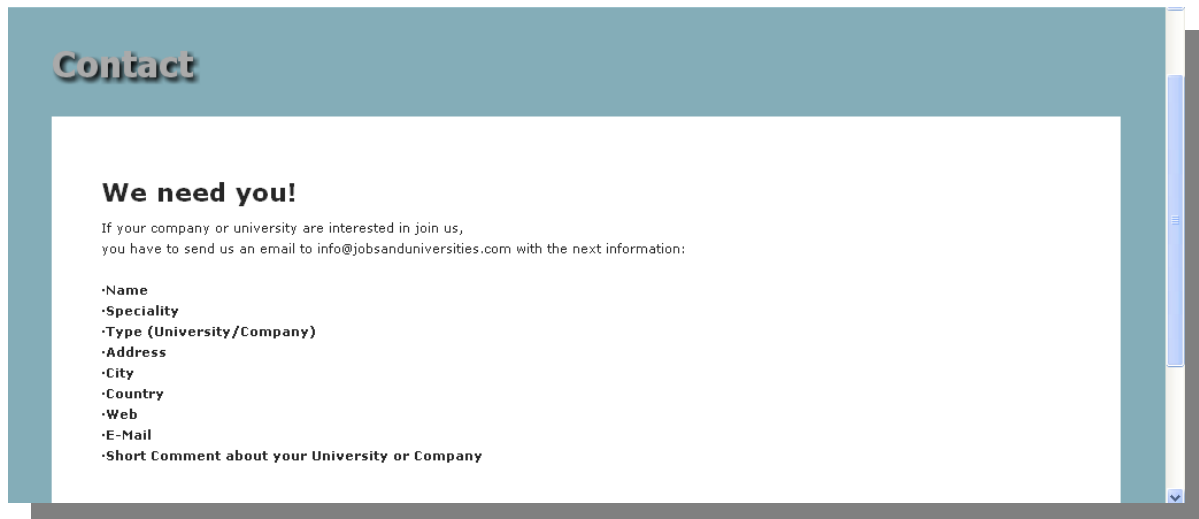
If we click in one of them:



We can see all the information about this company and get in touch with them if we are interested in work with them, thanks of the link at the lower part of the bubble.

Register-Application:

To start the web page's feedback, the website needs companies and universities interested in attract new workers and students. So, once we are got into the web page, if we click in "Contact", we will see this:



The image shows a screenshot of a web page titled "Contact". The page has a teal header with the word "Contact" in a bold, sans-serif font. Below the header, the main content area is white and contains the following text:

We need you!

If your company or university are interested in join us,
you have to send us an email to info@jobsanduniversities.com with the next information:

- Name
- Speciality
- Type (University/Company)
- Address
- City
- Country
- Web
- E-Mail
- Short Comment about your University or Company

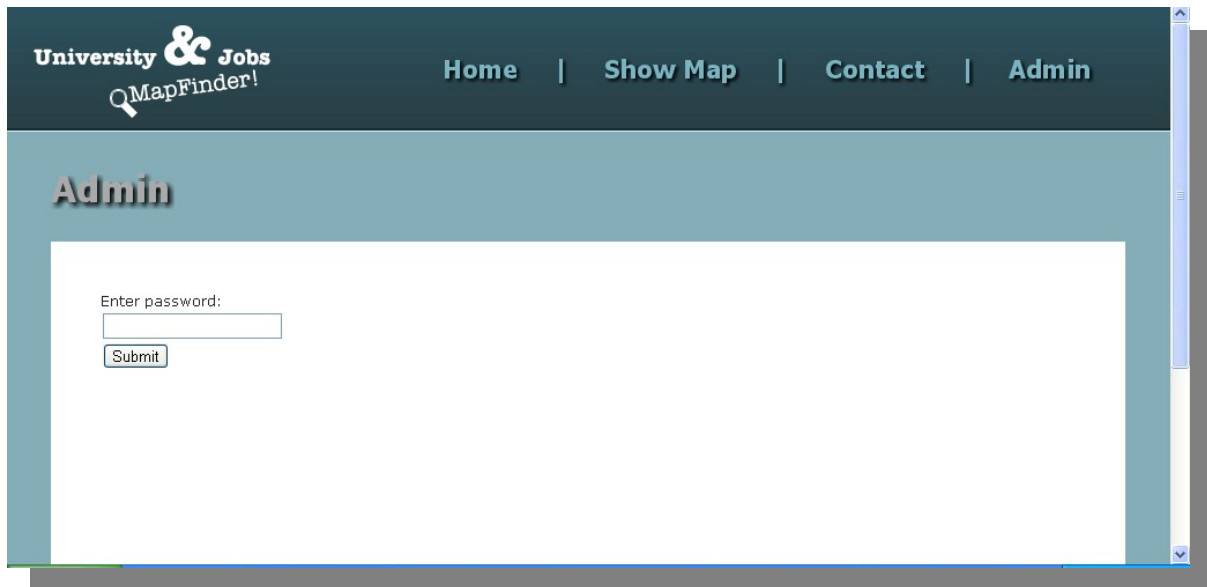
The page is displayed in a browser window with a vertical scrollbar on the right side.

Then, the companies and universities interested in appear in the web page, will have to send an email with all the data required. And then, the administrator will reply them telling if they have been admitted or not.

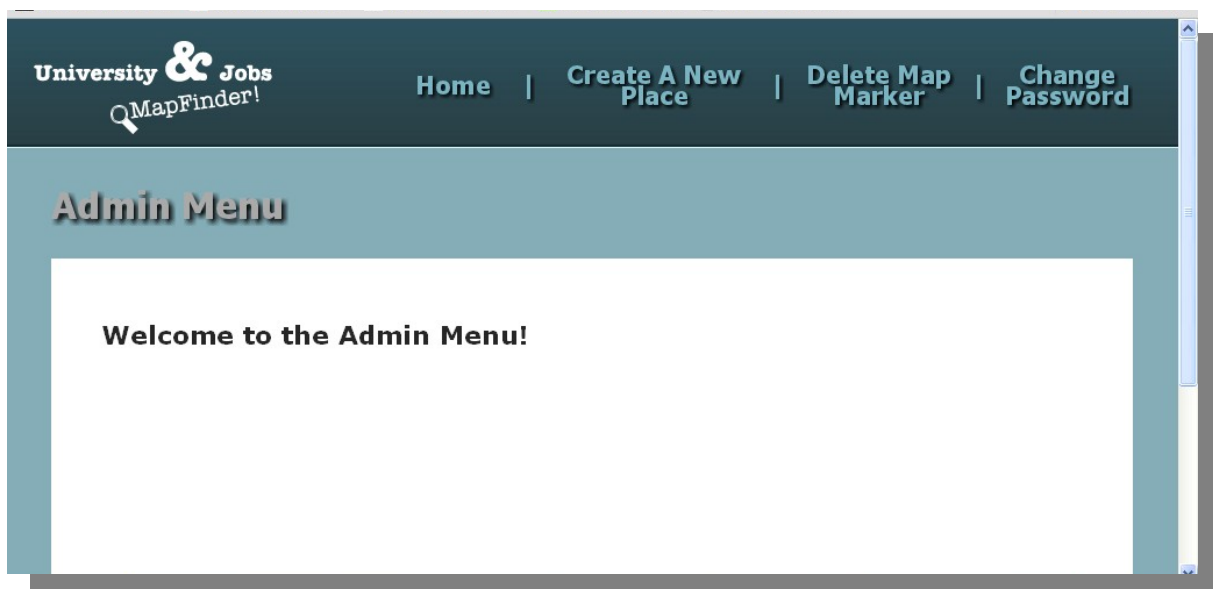
ADMIN:

Previous Steps:

Once we have got into the web site, if we click in "Admin" we will see this:




The user has to enter the password of the system to convert him in Admin. If the user write properly the password, he will see the admin part of the web site, with another private functions:



As you can see, now we have access to "Create A New Place", "Delete Map Marker" and "Change Password" , which I will explain below of this.

Create a New Place.

At this moment, the admin have to have a list with the companies and universities which want to join to the website, so, he will have to click in "Create a New Place" and he will see the next:

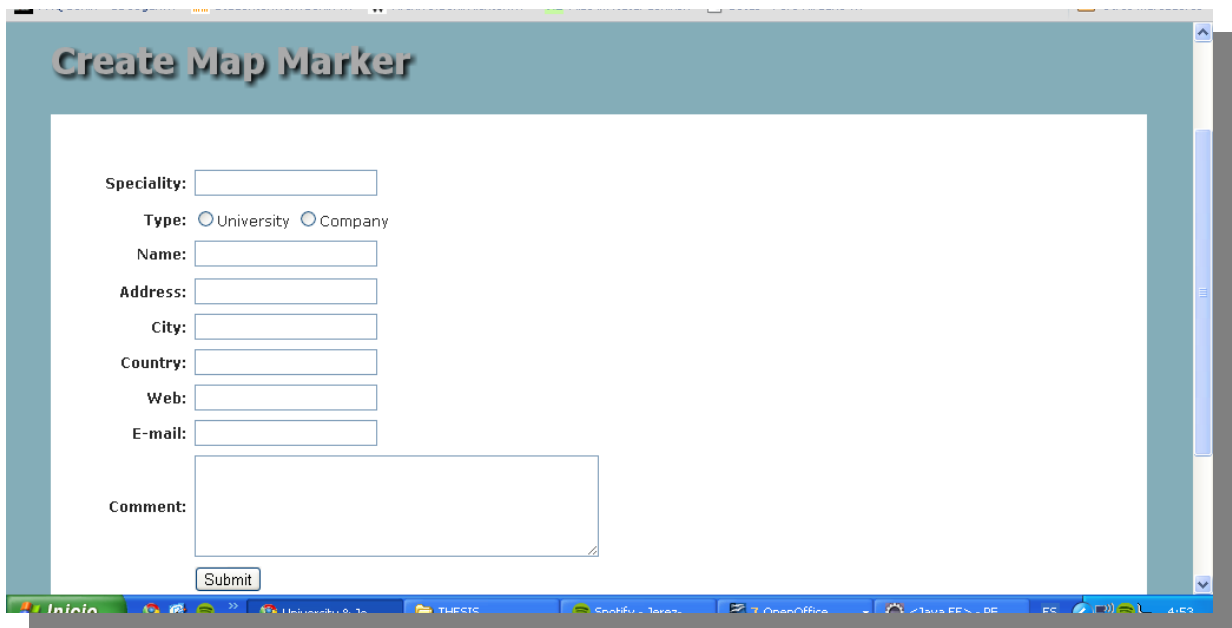


The screenshot shows a web form titled "Create Map Marker". The form includes the following fields and options:

- Speciality:** A dropdown menu currently showing "Sports" and a link for "New Speciality?".
- Type:** Radio buttons for "University" and "Company".
- Name:** A text input field.
- Address:** A text input field.
- City:** A text input field.
- Country:** A text input field.
- Web:** A text input field.
- E-mail:** A text input field.
- Comment:** A large text area.
- Submit:** A button at the bottom of the form.

We can see a form, where the admin have to write the info of every university or company to register them.

There is a combo box with the specialities already stored in the data base, but if the admin would need write a new one, clicking in "New Speciality?" we would remove the combo box and it would appear a text box where we could write a new speciality that it isn't in the system:



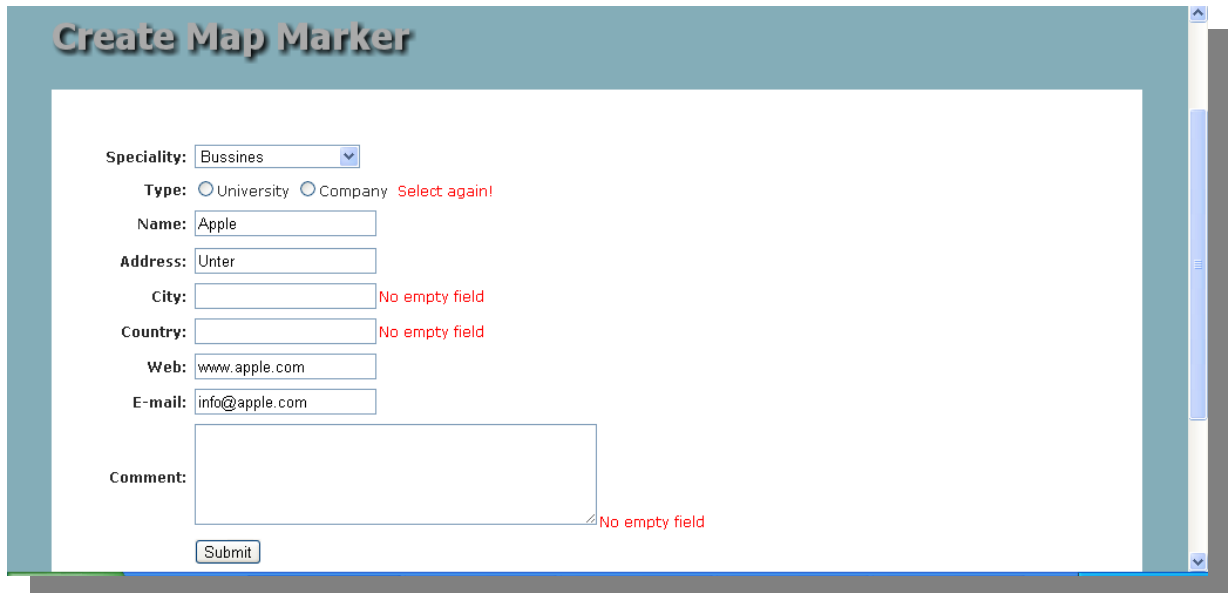
This screenshot shows the same "Create Map Marker" form, but the "Speciality" field is now a text input box instead of a dropdown menu. The rest of the form structure remains identical to the previous screenshot.

The form includes the following fields and options:

- Speciality:** A text input field.
- Type:** Radio buttons for "University" and "Company".
- Name:** A text input field.
- Address:** A text input field.
- City:** A text input field.
- Country:** A text input field.
- Web:** A text input field.
- E-mail:** A text input field.
- Comment:** A large text area.
- Submit:** A button at the bottom of the form.

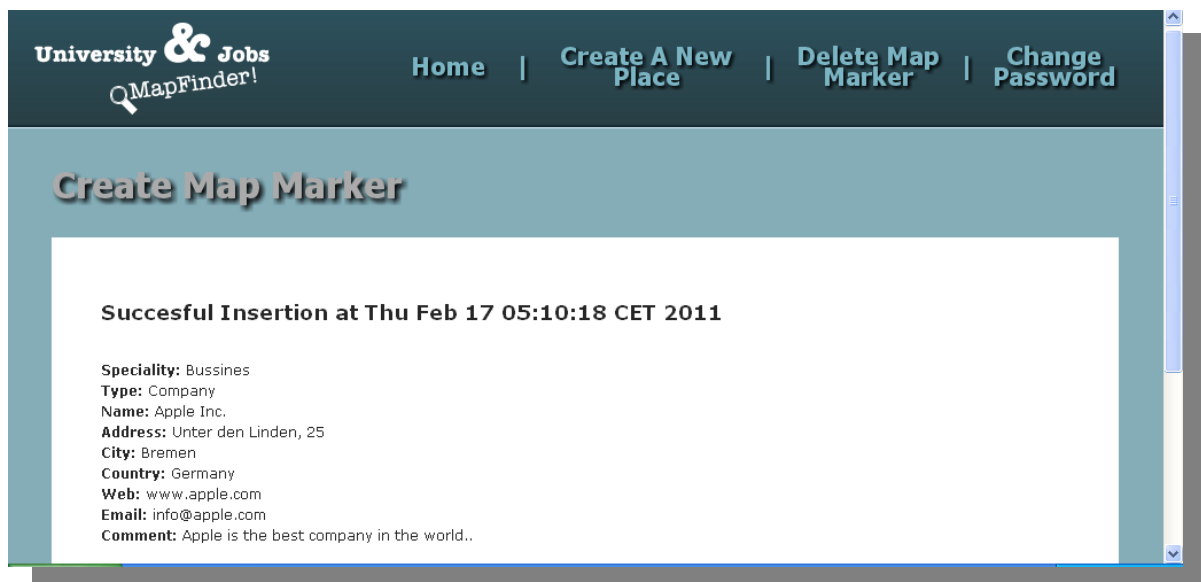
Now, we should write all the information about companies or universities, and click in 'submit' to store it in the data base and can be shown in the web site automatically.

But maybe, we can forget to fill some field, and the system will notice us, which field isn't filled and will remember with the data the field properly filled:



The screenshot shows a web form titled "Create Map Marker". The form contains several input fields: "Speciality" (a dropdown menu set to "Bussines"), "Type" (radio buttons for "University" and "Company", with "Company" selected and a red error message "Select again!"), "Name" (text input with "Apple"), "Address" (text input with "Unter"), "City" (text input with "Bremen" and a red error message "No empty field"), "Country" (text input with "Germany" and a red error message "No empty field"), "Web" (text input with "www.apple.com"), "E-mail" (text input with "info@apple.com"), and "Comment" (a large text area with "Apple is the best company in the world.." and a red error message "No empty field"). A "Submit" button is located at the bottom of the form.

Well, once we fill every field properly and clicking in "Submit", we should see a confirmation of satisfactory addition:



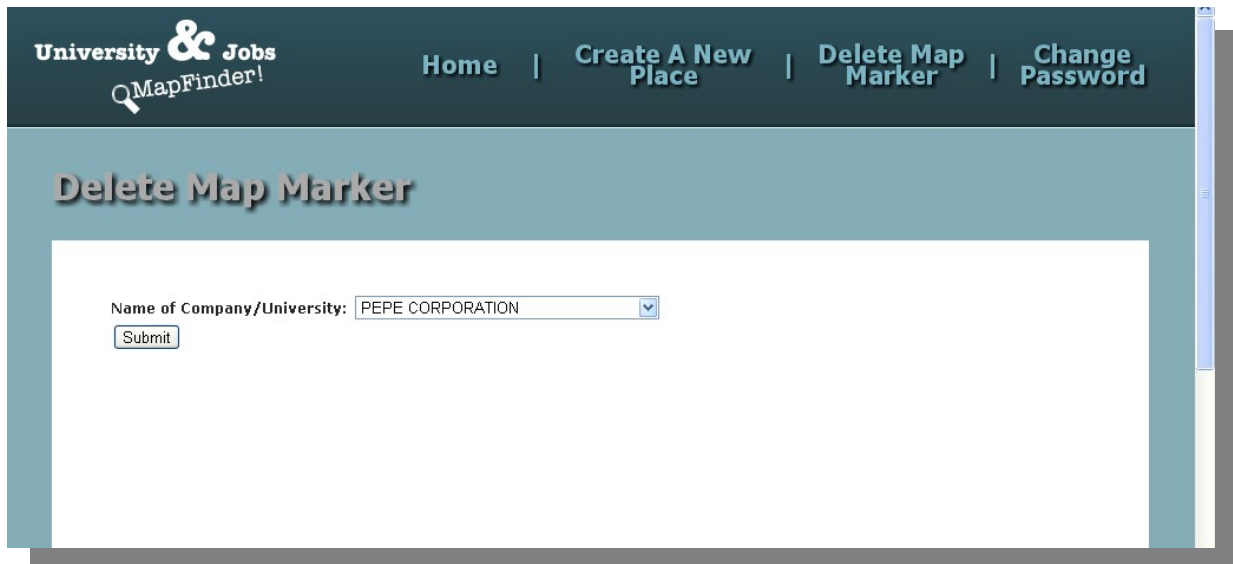
The screenshot shows the same "Create Map Marker" form, but now it displays a confirmation message: "Successful Insertion at Thu Feb 17 05:10:18 CET 2011". Below the message, the form fields are populated with the data from the previous screenshot: "Speciality: Bussines", "Type: Company", "Name: Apple Inc.", "Address: Unter den Linden, 25", "City: Bremen", "Country: Germany", "Web: www.apple.com", "Email: info@apple.com", and "Comment: Apple is the best company in the world..". The navigation menu at the top includes "University & Jobs QMapFinder!", "Home", "Create A New Place", "Delete Map Marker", and "Change Password".

And automatically, it will be ready to show to the users!

Removing Places.

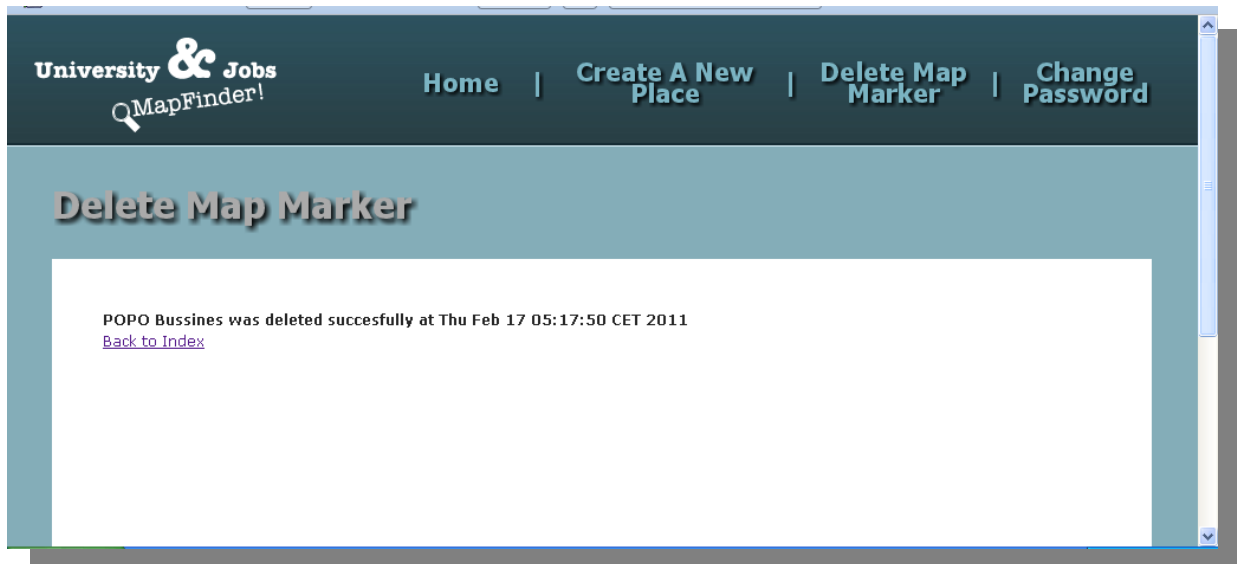
In the same way that the administrator can add a place, as well, it can delete an added place.

If the admin clicks in "**Delete Map Marker**" we will see a window like that:



The screenshot shows a web browser window with a dark blue header. On the left is the logo "University & Jobs MapFinder!". On the right are navigation links: "Home", "Create A New Place", "Delete Map Marker", and "Change Password". The main content area has a light blue background with the title "Delete Map Marker" in bold. Below the title is a white form box containing a label "Name of Company/University:" followed by a dropdown menu showing "PEPE CORPORATION" and a "Submit" button.

Here, we can see a combo box with all the companies added to the system by the admin, and we will have to click in the combo box and select one of them and click in submit:

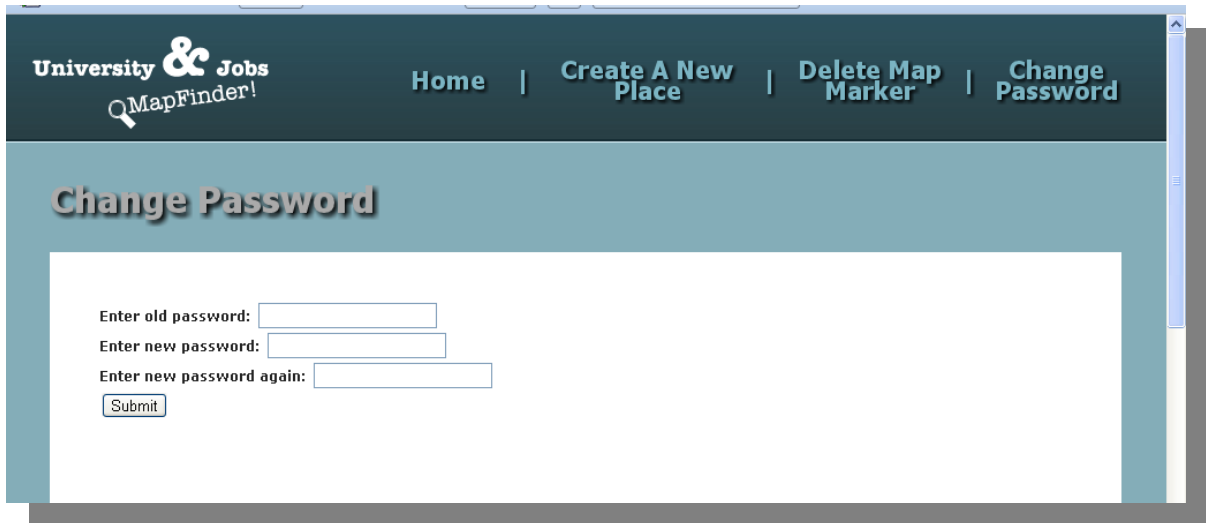


The screenshot shows the same web browser window as the previous one. The main content area now displays a message: "POPO Bussines was deleted succesfully at Thu Feb 17 05:17:50 CET 2011" followed by a blue link "Back to Index". The navigation links and logo remain the same.

If the deleting process have been successfully completed, we will see a similar message that above of this it can see, and the company/university selected will be removed completely.

Changing Password.

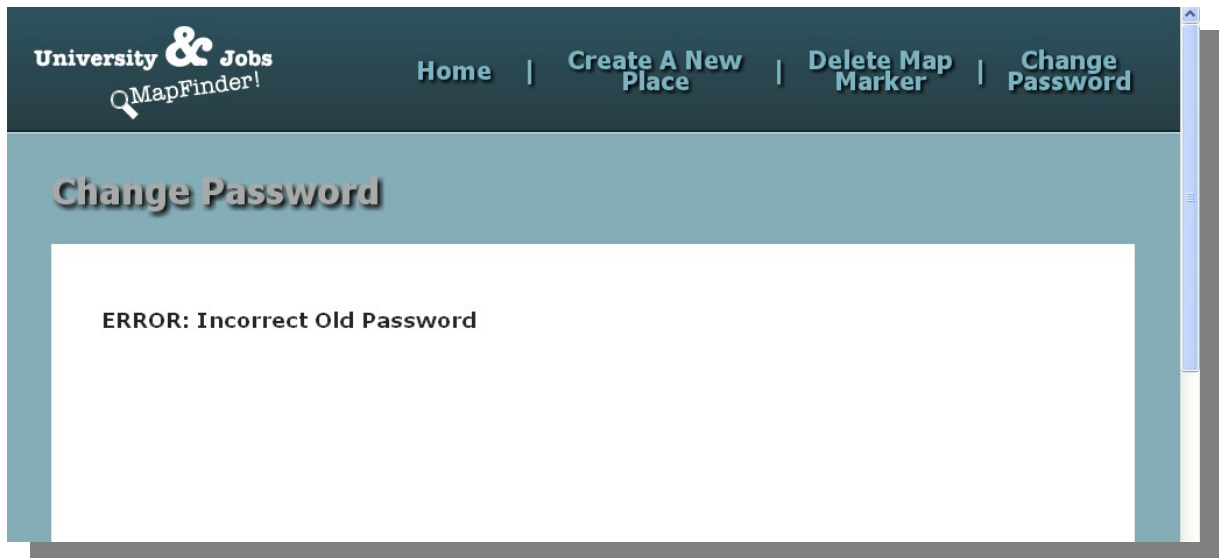
Sometimes it can be necessary to set a new one password, for this, we have to click in "Change Password" and we will see this:



The screenshot shows the 'Change Password' page of the University & Jobs QMapFinder! website. The page has a dark blue header with the logo and navigation links: Home, Create A New Place, Delete Map Marker, and Change Password. The main content area is light blue and contains the title 'Change Password' in bold. Below the title is a white form with three input fields: 'Enter old password:', 'Enter new password:', and 'Enter new password again:'. A 'Submit' button is located below the third field.

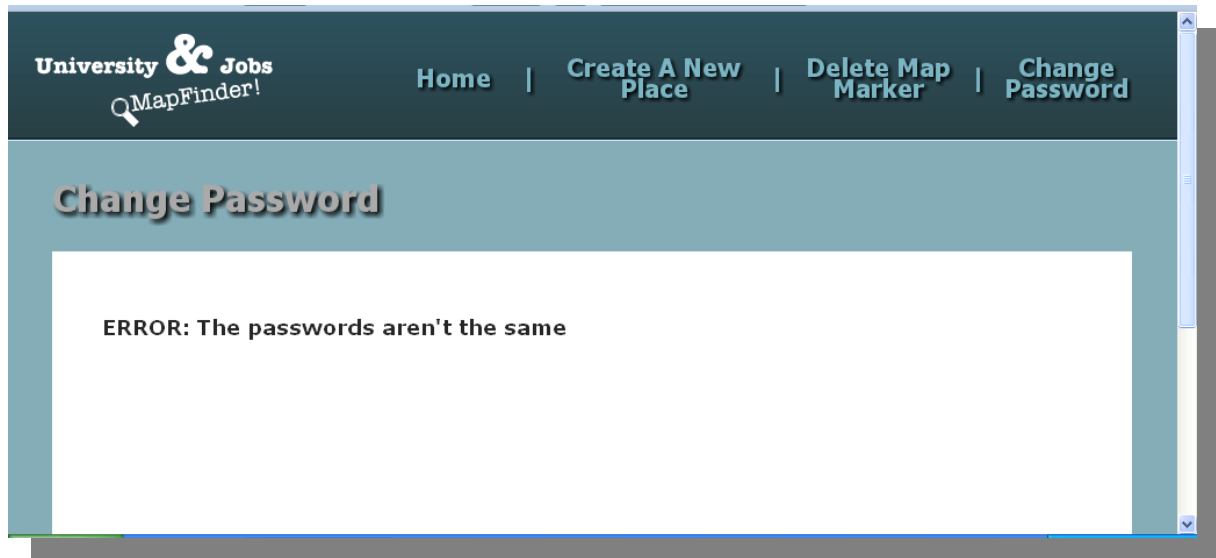
As you can see, for safety, it will be compulsory to enter the old or current password, and to write twice the new password.

If we forgot to fill some field or we write wrongly the current password we will see something like that:

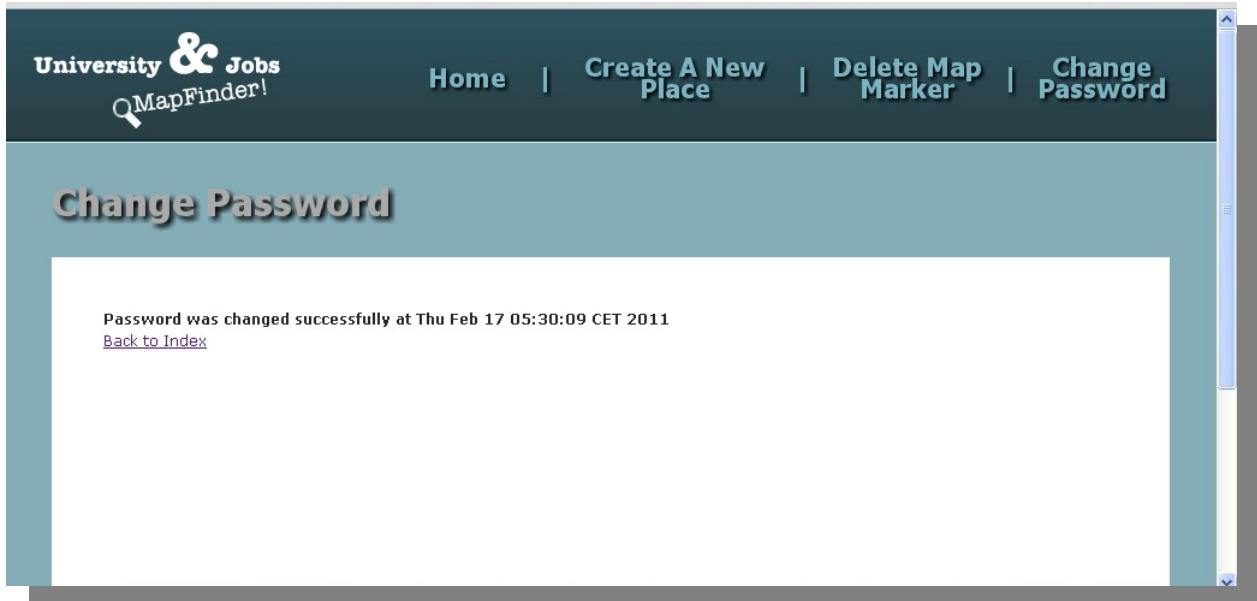


The screenshot shows the 'Change Password' page of the University & Jobs QMapFinder! website. The page has a dark blue header with the logo and navigation links: Home, Create A New Place, Delete Map Marker, and Change Password. The main content area is light blue and contains the title 'Change Password' in bold. Below the title is a white form with the error message 'ERROR: Incorrect Old Password' displayed in the center.

Or if we write rightly the current password but the we don't write correctly the new ones password we will see this:



And finally, if we write everything properly, we be able to change the password and we will see this:



BIBLIOGRAPHY.

INTERNET:

****References about General Programming:**

JAVASCRIPT: <http://www.librosweb.es/javascript/index.html>

GOOGLE MAPS API v.3: <http://code.google.com/intl/es-ES/apis/maps/documentation/javascript/reference.html>

CSS: <http://www.librosweb.es/referencia/css/index.html>

AJAX: <http://www.librosweb.es/ajax/index.html>

pgAdminIII: <http://www.arpug.com.ar/trac/wiki/PgAdmin>

All the theoretical definitions (THEORETICAL FRAMEWORK) have been extracted of Wikipedia Project [www.wikipedia.org]

****References about doubts and troubles:**

Google Maps Develop: (Markers, infoWindows, Overlays, XML to data base..)

<http://es.efreedom.com/Question/1-3821506/Problema-inicializar-el-mapa-de-Google-con-varios-marcadores>

<http://www.forosdelweb.com/f26/generar-xml-con-jsp-482589/>

http://books.google.es/books?id=dFc27ogrYg8C&printsec=frontcover&dq=XML+from+JSP&source=bl&ots=wIfeklkyp6&sig=CAwvNrVTckJImog3pB1m7GRjGLY&hl=es&ei=uVEGTYj1CIOTswb2mJSHCg&sa=X&oi=book_result&ct=result&resnum=10&ved=0CGUQ6AEwCQ#v=onepage&q&f=true

<http://www.forosdelweb.com/f13/google-maps-multiples-marcadores-866034/>

<http://www.svennerberg.com/2009/09/google-maps-api-3-infowindows/>

<http://stackoverflow.com/questions/1544739/google-maps-api-v3-how-to-remove-all-markers>

<http://code.google.com/intl/es-ES/apis/maps/documentation/javascript/examples/index.html>

BOOKS:

Requirements Specification Reference.

Raymon Turner. "The Foundations of Specification": Journal of Logic and Computation, Vol. 15. 2005.