Grado en Ingeniería Informática
Informatikako Ingeniaritzako Gradua

Proyecto fin de grado
Gradu amaierako proiektua
Auzonet: Diseño e implementación de una red social para comunidades de vecinos

Aitor Brazaola Vicario

Director/a: Diego López-de-Ipiña González-de-Artaza

Bilbao, mayo de 2016

Abstract

The lifestyle of today's society moves away the traditional contact between neighbors where is promoted the goods, services and information exchange between citizens. In addition, the growth of cities makes more difficult for councils and public institutions being aware of needs and concerns of the inhabitants.

Auzonet as digital platform focused on the residents, provides the perfect solution to neighbors providing a web platform to exchange relevant information and being a reliable data source for public institutions.

Keywords

OpenData, Web, App, Social network.

Contents

List of Figures

List of Tables

Algorithm Index

Chapter 6

1. INTRODUCTION

The aim of this project is create an online social platform for benefit different parties, on the one hand, the Bilbao's council citizens, building a meeting point in the community for posting warnings or relevant information with other people living in the same neighborhood, and publishing requests or offers to help among them.

On the other hand, the information generated by the platform usage can be a source of high-quality data for council's services improvement, allowing these institutions to take better decisions for improve the life quality of the city.

This project must be developed using resources of European project WeLive [**WeLive**] which DeustoTech [**DeustoTech**] MORElab [**Morelab**] belongs to, using their libraries and programming interfaces for obtaining Bilbao's council Open Data.

WeLive is a web platform to promote OpenData in institutions created by several European entities that provides a range of services for developers and public entities to facilitate Open Data usage in third-party applications.

The main challenges will tackle this project are:

- Create a social web platform easy to use to promote users collaboration.
- Add value to the platform using public datasets.
- Learn to develop a full-stack web and mobile application.

In addition, during the development may be necessary develop new reusable software components for allow WeLive custom integrations and make it available in the application repository of WeLive platform.

This project will need technical and interpersonal skills to achieve good collaboration between WeLive and Auzonet teams to build the best quality software.

To carry out major development of the project, will be used a web framework for building the web application with maximum code reuse and better efficiency, for developing mobile application, a portable version of the web will be embedded using multi-platform deployment tools.

The previously mentioned tools add the challenge of make a web design responsive for adapt all visual components to different smartphone's screen sizes and learn how to communicate applications via API's that requires WeLive to interact with public data.

All phases of development will put in manifest the knowledge acquired during the university degree and require new technical and theoretical skills.

This document describes the project, how has be planned, developed, the objectives and procedures to make good usage of public data released by institutions adding value to services offered by cities to citizens.

In the development chapter internal process will de explained in detail, reasoning the tools finally used and the why has been chosen, the project structure and possible functional divisions of software components, design and requirements.

Finally, the last chapter include conclusions found throughout the entire process of creation and improvements to the platform for future updates.

2. BACKGROUND AND RATIONALE

2.1. BACKGROUND

The aim of this project is create a social network for neighbors, using public data for avoiding error during users registration and making good usage of public datasets, but as can be seen in the following state of art study, this is not the first initiative taken in this field.

2.1.1 Social network for neighborhoods

Creating a social network for neighbors in which all members of a portal or community can interact with each other by sharing resources and information is not a new initiative, in other countries like Germany or United States similar projects [**larazon**] have been developed, internet has extended the creation of communities of all kinds of areas, however, the already existing proposals rarely get benefit of the public entities data, and in many cases, they are the most reliable source of information for building applications and services for citizens.
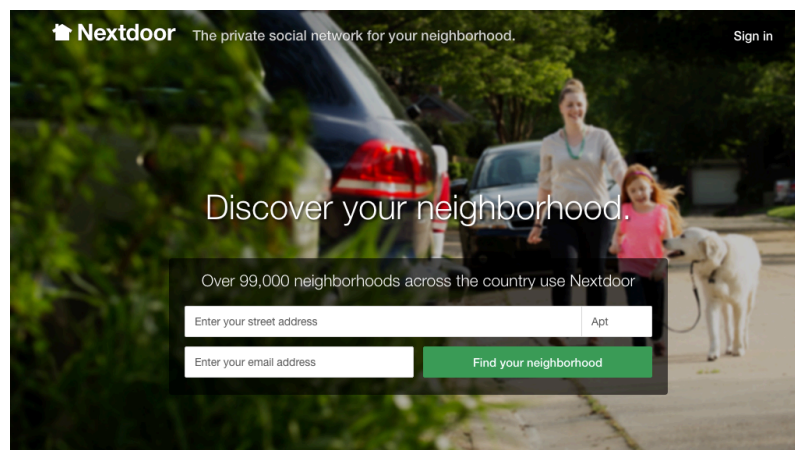
Figure 2.1: Nextdoor. Existing alternative in USA.

Nextdoor is a service only for USA neighborhoods created in California working since 2010, and the most remarkable features are:

- Meeting point for find possible babysitters.
- Chats of different topics in each neighborhood.

- Exchange point for goods or services.
- Shared garages management.

There is no revenue currently coming from the service and is completely free of charge, the project is supported by venture capital firms and they are still looking for a way of making revenue that benefits both users and creators not specified yet.

Nextdoor has been recommended by many famous publications like New York Times, CNN and Wall Street Journal, what definitely demonstrate the feasibility of these solutions for neighborhoods.

Recently, this platform has extended his coverage expanding to Netherlands, outside of the USA.
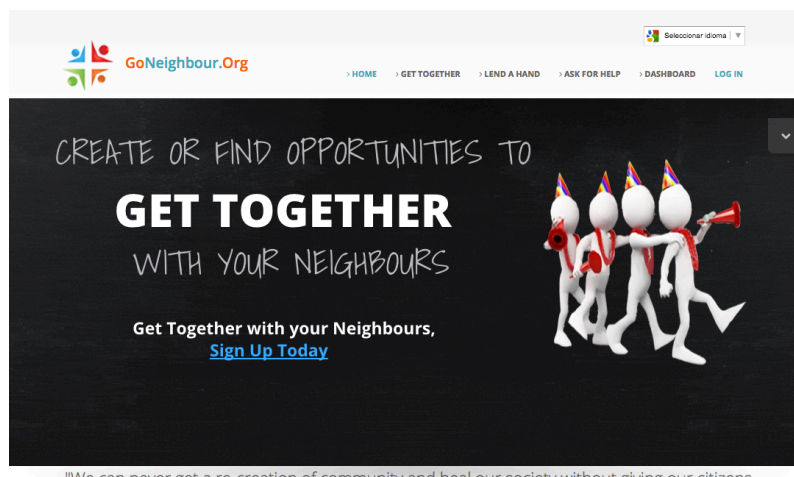
Figure 2.2: Goneighbour, Canadian alternative to Nextdoor

GoNeighbour is another social network with neighborhoods in mind, created in Canada with less market share, and similar features, trying to port the Nextdoor experience exclusive from USA to Canada, but making an emphasis on creating a market of goods by neighbors.

2.1.2 Open Data

As we have seen, there are a growing tendency of internet neighborhood communities, but all the previously mentioned use geolocation or has his own database of addresses, this is the essential difference with Auzonet. Auzonet takes advantage of data released by public organizations like Bilbao's council for offering data consistency and provide precise information.

Firstly, is important to analyze how is categorized Open Data. Tim Bernsers-Lee, the inventor of the Web and Linked Data initiator, suggested a 5-star deployment scheme for Open Data [**ODStars**].

As we can see in the image 2.3, Open Data is classified by how data is accessible by human or computers. Along this project, the data source is categorized as three stars because is publicly available of Bilbao's council data web platform [**Bilbao Open Data**] in structured format without proprietary extension[1].

Although it is true that currently, many public entities make available this data in digital format only by the look of transparency and are unaware of the true value of themselves, is one of the

---

[1]Bilbao's council data is provided in CSV and later converted to JSON via WeLive API
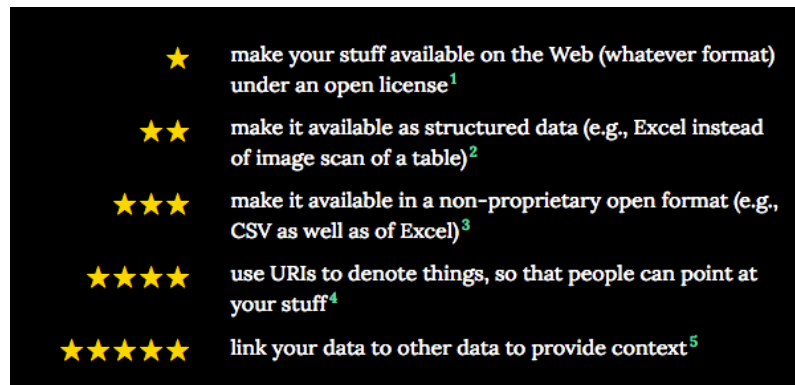
Figure 2.3: 5 Star of Open Data

motivation of this project, in fact, have been moments during the development in which the data found poorly formatted and have had to request modifications to the agencies involved.

The fact that most of datasets that are published by public organizations are not used in systems created by third parties, causes certain unconcern in the status and integrity of them.

WeLive Project

Figure 2.4: WeLive Project

WeLive is an EU H2020 funded project, addressing ICT-enabled open government – INSO1 topic, about a new concept of public administration based on citizen co-created mobile urban services.

The project is conceived to transform the current approach towards e-government by providing a new open model oriented towards the design, production and deployment of public services based in the collaboration of a quadruple helix, i.e. research organizations, companies, public administrations and citizens. WeLive will provide a novel We-Government ecosystem of tools (Live) built on the Open Data, Open Services and Open Innovation paradigms that is easily deployable in different public administrations and which promotes co-innovation and co-creation of personalized public services through public-private partnership and the empowerment of all the stakeholders to actively take part in the value-chain of a municipality or a territory. [**WeLive**]

The city of Bilbao, is one of the four pilot cities selected for the project, thanks to the process of urban regeneration addressing the physical decline of the city after being highly dependent of shipbuilding and steel industry.

The WeLive initiative in Bilbao is being carried out by several companies and research centers where University of Deusto and DeustoTech are in.

This project, attend one of the proposals made in smart cities context and tries to add value to Bilbao's citizenship experience.

2.2. RATIONALE

Internet social networks are quite common today, and neighborhood based ones are not an exception, this project aims to get the best advantages of social networks and Open Data, creating a platform to exchange goods, services or information among neighbors.

Furthermore, the project brings a high learning value of web applications development and also serve as an example of how based on WeLive project, can be created third party applications getting advantage of public data published by institutions and easily accessed via WeLive platform tools.

The project will face the challenge of creating responsive web applications easily usable on all mobile platforms like smartphones or tablets using modern web technologies and standards.

3.   GOALS AND SCOPE

3.1. PROJECT DEFINITION

Auzonet must meet the expectations of at least two types of users:

Citizen: The primary user of the system and beneficiary of all the features and advantages given by the platform.

Public institution: To analyze the data generated by the platform for statistics and promoting services to improve the quality of life in cities and people.

To achieve this objectives will be developed:

- A web application where users can register their neighborhood based on existing data published by the council of Bilbao, the rest of neighbors interested can join to it and get benefit from its functions.
- A mobile application that lets you interact with the main functions of the web from a smartphone.

3.1.1   Functionality

Users can search their portal trough an interactive form based on the existing neighborhoods and streets of Bilbao, for better confirmation of the place, a small map showing the exact situation of the door will be displayed, if the information is correct, a few setting should be configured like the privacy level of the community allowing to protect the access with a password or adding few lines of welcome message to new members via mail.

Each user can belong to more than one neighborhood, considering that some users may want to be aware of their usual community residence and the rented property one.

Each neighborhood community represented in the application has its own home page where users can see the pinboard with warnings or information notes published, and a lower table divided in two sections called Requests and Offers where will be displayed the posts created by members of that community.

From the home page of each community, the user can create a request for products, services or publish an offer specifying whether he wants to get paid or offer the service for free.

To ensure some confidence when users work with each others, a karma level system represented by a numeric value that is higher or lower by other users past reviews determine the trust level of each.

The home page also allow access to statistics about logged user in the platform, displaying charts with number of offers and request has published or attended and recover options for old publications to allow periodical posts of the same issue.

3.1.2 Limitations

Auzonet never going to manage payments among individuals, beyond the simple fact of interactions between requests and offers history among neighbors. Interested parties should agree on transaction methods by their own with complete responsibility for ensuring the successful transaction in a legal economic frame.

Will be the users themselves, who in case to use the platform will have to register their neighborhood once they sign up for the first time in the service to get benefit of the software features.

There are no plans to develop applications with smartphones native development kit of each mobile operating system for avoid multiple simultaneous developments during the creation of mobile application, only web technologies currently available will be used.

3.2. DESCRIPTION OF THE EMBODIMENT

3.2.1 Development methodology

The realization of Auzonet is separated in two main functional units, first will focus on finalizing the web application and then mobile app.

As can be seen in the figure 3.1 development phases are going to be the following:

Requirements: Analysis of the main functional requirements.

Design: Design data and logical structures needed for running the application and approach to the aesthetic design of the solution.

Web application development: Implementation process of the main application, with Welive project integration and "responsive" design for fitting on different screen sizes.

Mobile application development: Implementation process of mobile application that get benefits of the "responsive" design of the web application.

Tests: Different executions by end users and bug detection on the feature usage processes for debugging, It will try to involve collaborators and friends of the programmer for making real usage of the platform.

Being a project that involves only one developer will not be used agile methodologies that facilitate cooperation and teamwork, instead, a system of lists of tickets with different classifications will be used in a task manager software.

Tasks are classified as follows:

Table 3.1: Ticket categories.

Category	Description
Functional	Features that represent the core of the application.
Not functional	Features that enrich the user experience.
Bug	Faults to debug.
Improvement	Suggestions or new functionalities that add value to the platform.

Sometimes, certain tasks can come from external sources such as suggestions for improvement or major deficiencies found in a test, in that case, they are added to the list with the corresponding label and prioritized according to their importance.

3.2.2 Intermediate products

There are clearly two intermediate products clearly defined for this project:

Web application Is the main application developed along the project, should be accessible from all web browsers in the market and responsive design for fitting on all screen sizes.

Mobile application Is a wrapper of the main application, taking advantage of the responsive design, the mobile application should provide a native-like environment for iOS and Android.

Both applications are the final product, the idea is provide a full stack platform working from the first day on web and mobile, with same user experience and without losing features.

3. GOALS AND SCOPE
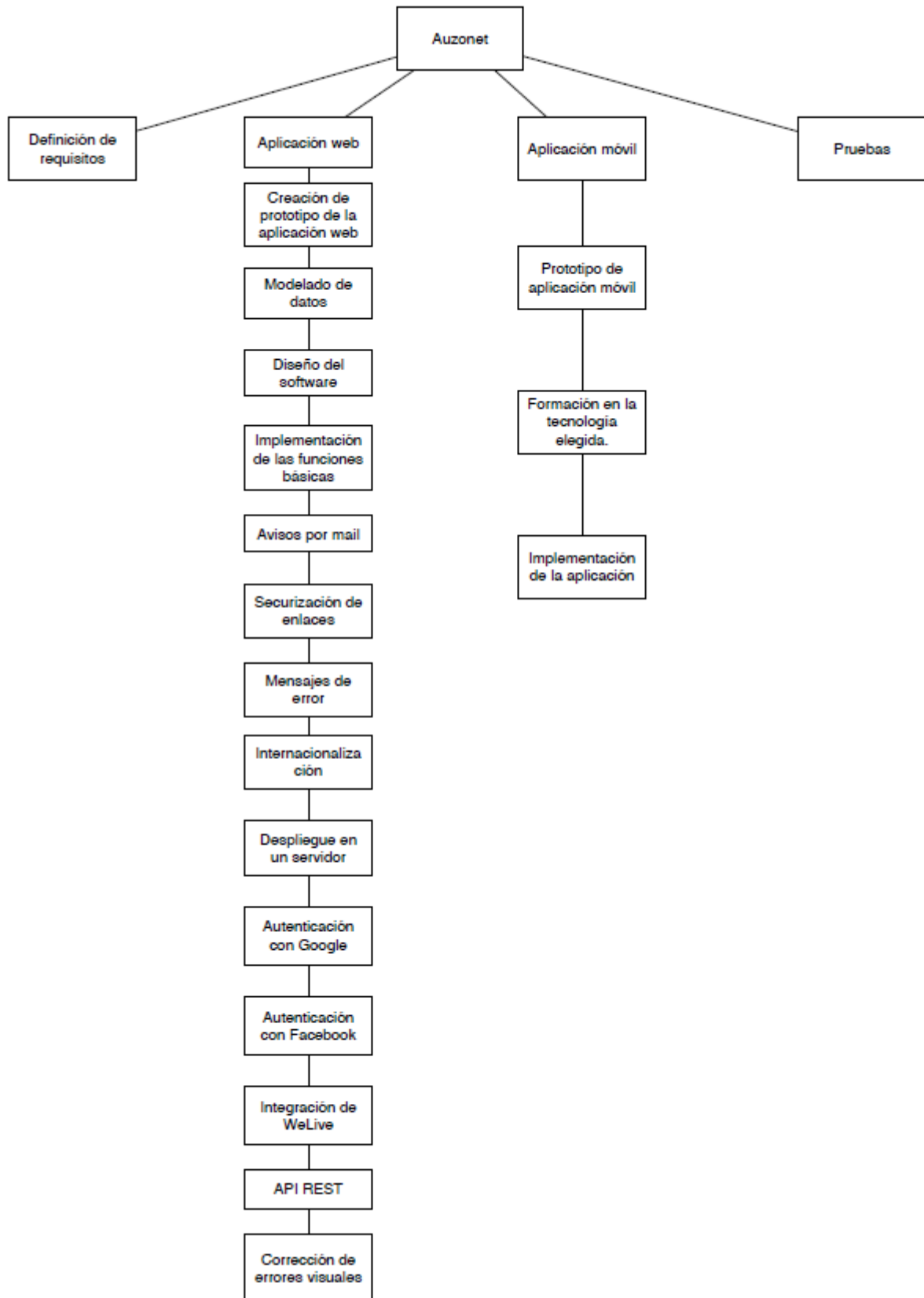
Figure 3.1: EDT

3.2.3   Main tasks

T1 - Requirement analysis

A detailed analysis will be performed for understanding all the features and limitations of the platform, how has to integrate Open Data sources and how might be look and feel for satisfy all user needs in a wide range of ages.

Web application

T2 - Web application prototype: Create a low fidelity prototype showing the final application look and feel.

T3 - Data modeling: Design how the data is being stored and relationships between tables.

T4 - Software design: Develop the logical base structure of all classes and packages that would be the architecture of the application on server side.

T5 - Core features implementation: Implementation of exchange features between users such as offering and requesting functions.

T6 - Mail notifications: Implement a notification system via email with custom link to actions with friendly look and feel.

T7 - Security assessment: Fix data leaks via url cross-site scripting or non allowed access to other users data changing url patterns.

T8 - Friendly error messages: Create user friendly error messages integrated on platform's design to provide user feedback.

T9 - Internationalization: Make application available in english and spanish.

T10 - Server deployment: Deploy the application on a server for allow the access via internet.

T11 - WeLive integration: Implement WeLive API calls to request Bilbao's Open Data required in community set up process.

T12 - Fix high density display bugs: Fix some visual error when the application is loaded on devices with high density displays, like retina MacBooks or iPads.

Mobile application

T13 - Mobile application prototype:

T14 - Research of technology selected:

T15 - Implementation:

T16 - Testing

The platform access will be distributed in a private beta testing to users from different ages to gather feedback of usability improvements and bug reporting.

3.3. ORGANIZATION

3.3.1 Organizational structure

The organization as can be seen in the figure 3.2, is composed by the Project manager, WeLive development team experts to help in WeLive integration process and the student who perform the roles of programmer and designer.
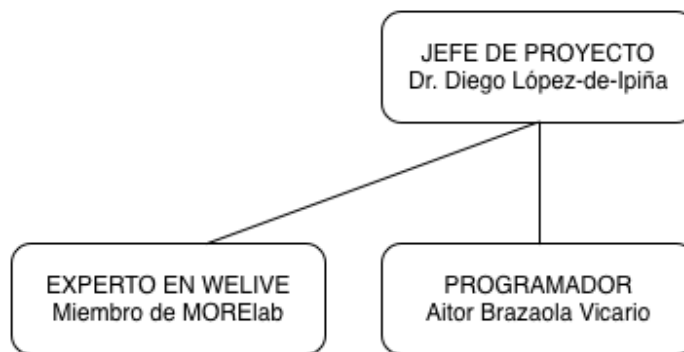
Figure 3.2: Organizational schema

Every month will take place a session for review the work done and determine the status of development, in this sessions it will be assess if are necessary to change some functionality or bugs.

The main assistants will be the programmer and project manager, details exposed will be gathered, and requested changes will be prioritized over the rest of features in the pipeline.

3.3.2 Human resources plan

The student will be the unique physical person in the development team but will be necessary the acquisition of various roles along the project, it is true that un punctual cases can be helped by other mates of the laboratory where is working on, but the most of the project is developed by him.

Following, there are listed the main roles played by the student:

Programmer: It is responsible for creating all the logic of the program and perform the different configurations in the devices responsible for the platform work. Its functions also cover the early stages of software design, data schema creation and conduct appropriate tests to detect errors.

Designer: It is commissioned to design a user interface that fits all available screens and make the application accessible to people with disabilities and ensure a satisfactory user experience and consistent navigation between different sections of the web.

When WeLive API integration development will being taking place, programmer will have help from members of WeLive development working in the same DeustoTech laboratory, WeLive team collaboration is crucial for detecting failures on a real third party application and ensure an optimal implementation of this programming interface.

In addition, project manager is responsible for approving all changes and proposals considered by the student to improve the product. Project manager will be present at meetings and will be responsible for setting milestone dates.

4. PLANNING

4.1. PRECEDENCE DIAGRAM
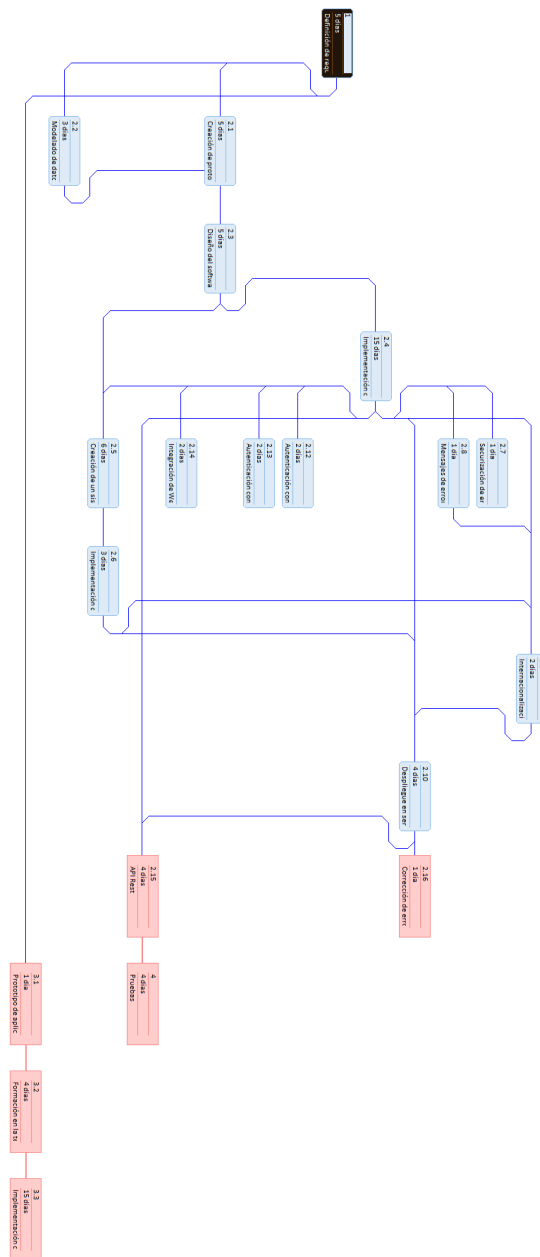
Figure 4.1: Precedence diagram

The project development is going to happen during the working time in DeustoTech MORElab of 4 daily hours, and the team will be formed by the following actors:

Programmer: Person in charge of programming the control structures of the platform.

Designer: Person in charge of the final appearance with the end user will interact.

Project manager: Person in charge of monitoring the progress of the project and its organization.

Experto en plataforma WeLive: MoreLab team member participating in the development of technical advice WeLive platform for the team.

4.2. GANTT DIAGRAM

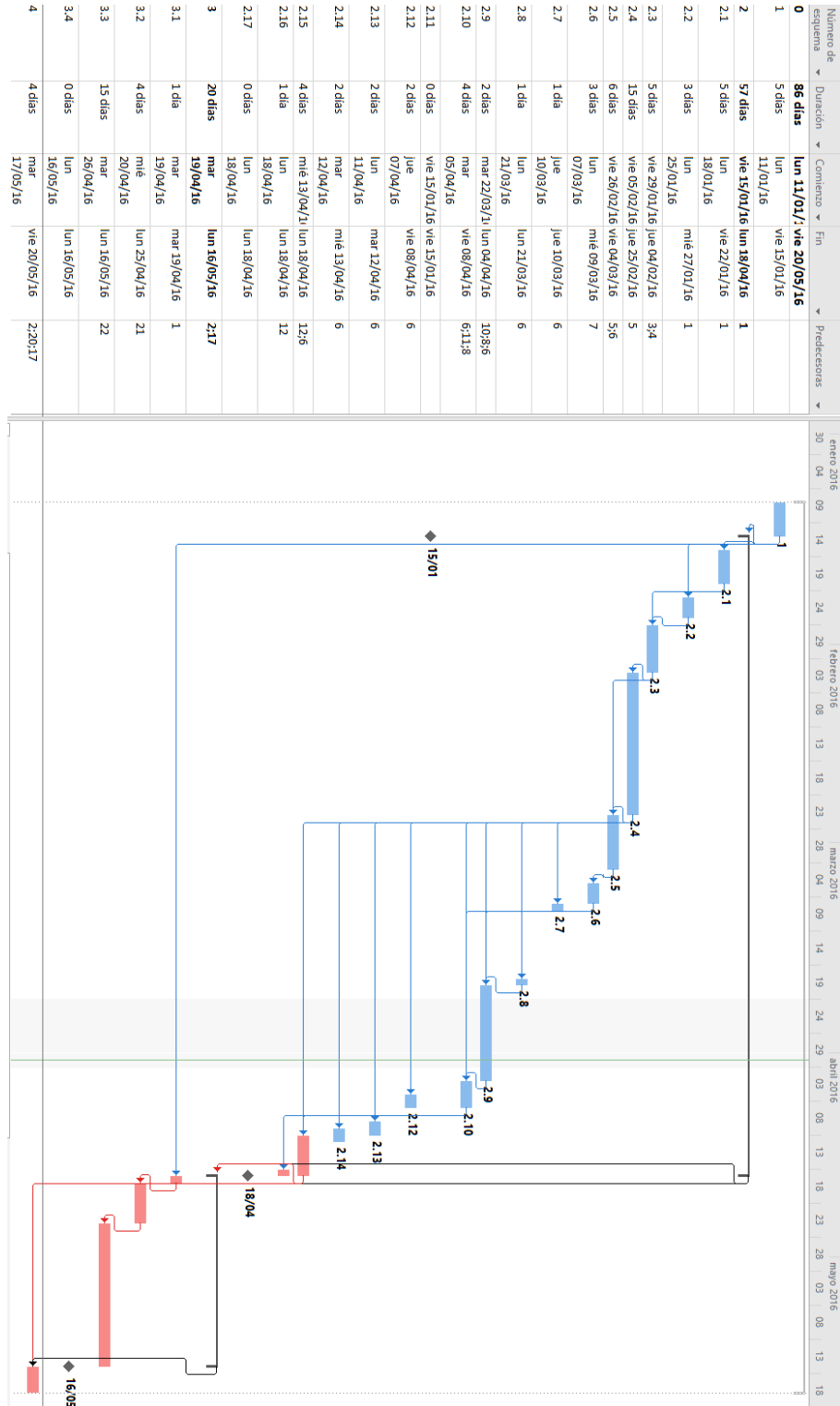| Número de esquema	Duración	Comienzo	Fin	Predecesoras	
0	86 días	lun 11/01/; vie 20/05/16			
1	5 días	lun 11/01/16	vie 15/01/16		
2	57 días	vie 15/01/16	lun 18/04/16	1	
2.1	5 días	lun 18/01/16	vie 22/01/16	1	
2.2	3 días	lun 25/01/16	mié 27/01/16	1	
2.3	5 días	vie 29/01/16	jue 04/02/16	3;4	
2.4	15 días	vie 05/02/16	jue 25/02/16	5	
2.5	6 días	vie 26/02/16	vie 04/03/16	5;6	
2.6	3 días	lun 07/03/16	mié 09/03/16	7	
2.7	1 día	jue 10/03/16	jue 10/03/16	6	
2.8	1 día	lun 21/03/16	lun 21/03/16	6	
2.9	2 días	mar 22/03/1	lun 04/04/16	10;8;6	
2.10	4 días	mar 05/04/16	vie 08/04/16	6;11;8	
2.11	0 días	vie 15/01/16	vie 15/01/16		
2.12	2 días	jue 07/04/16	vie 08/04/16	6	
2.13	2 días	lun 11/04/16	mar 12/04/16	6	
2.14	2 días	mar 12/04/16	mié 13/04/16	6	
2.15	4 días	mié 13/04/1	lun 18/04/16	12;6	
2.16	1 día	lun 18/04/16	lun 18/04/16	12	
2.17	0 días	lun 18/04/16	lun 18/04/16		
3	20 días	mar 19/04/16	lun 16/05/16	2;17	
3.1	1 día	mar 19/04/16	mar 19/04/16	1	
3.2	4 días	mié 20/04/16	lun 25/04/16	21	
3.3	15 días	mar 26/04/16	lun 16/05/16	22	
3.4	0 días	lun 16/05/16	lun 16/05/16		
4	4 días	mar 17/05/16	vie 20/05/16	2;20;17	

Figure 4.2: GANTT diagram

4.3. WORKLOAD ESTIMATION BY PROFILE

The following estimation is based on the different profiles of the total 255 hours:

Programmer: 215 hours.

Designer: 22 hours.

Project manager: 15 hours.

WeLive platform expert: 3 hours.

- 1. Requirements definition: 15 hours.
- 2. Web application: 168 hours.
    - 2.1. Web application prototype: 15 hours.
    - 2.2. Data modeling: 9 hours.
    - 2.3. Software design: 15 hours.
    - 2.4. Implementation of the core features: 51 hours.
    - 2.5. Users system creation: 18 hours.
    - 2.6. Implementation of mail notifications: 15 hours.
    - 2.7. Link security checks: 3 hours.
    - 2.8. Friendly error messages: 3 hours.
    - 2.9. Internationalization: 6 hours.
    - 2.10. Server deployment: 12 hours.
    - 2.13. Visual customization and logo: 6 hours.
    - 2.14. WeLive API integration: 12 hours.
    - 2.16. High density displays visual fixes: 3 hours.
- 3. Mobile application: 60 hours.
    - 3.1. Mobile application prototype: 3 hours.
    - 3.2. Learning the selected technology for development: 12 hours.
    - 3.3. Mobile app implementation: 45 hours.
- 4. Tests: 12 hours.

5. BUDGET

This section list all devices and human resources involved in project development, all software used is Open Source or with student licenses free of charge, so there are no costs related with them.

Internet connection, power usage and server architecture is provided by DeustoTech without specific cost for the project.

5.1. HUMAN RESOURCES

Table 5.1: Human resource budget.

Profile	Workload	Cost	Price
Programmer	255 h.	9€/h.	2295€
Designer	22 h.	9€/h.	198€
Project manager	12 h.	50€/h.	600€
WeLive platform expert	3 h.	6€/h.	18€
TOTAL			3.111€

5.2. HARDWARE

Table 5.2: Hardware budget.

Hardware	Price	
Apple MacBook Pro	1.119€	
Apple iPhone 6	639€	
Lenovo ThinkVision T2224pD	165€	
TOTAL		1.923€

6. DEVELOPMENT

The main section of the development is going to be for the web application, the part of the project with more effort, and technologies involved in development, then, the mobile application as a derivative product from the web app will be explained in detail reasoning the technologies used with their advantages and drawbacks.

Both products has been submitted to testing with collaborators and the process of the tests will be detailed, finally a user manual will be included for helping to use all functionality implemented by the platform.

6.1. WEB APPLICATION

6.1.1 Software and Hardware requeriments

The application must have a good user interface, allowing users from all ages for using it with less effort, furthermore, the application must make a good usage of the public data of the Bilbao's council provide using WeLive platform.

Software requirements

- All communities existing on Bilbao must be available for register in the platform.
- The communities must have a common place to exchange information among members.
- The users information must be protected with security measurements for avoiding data stealing or spam.
- Users can post or attend offers among them.
- Users can post or attend requests among them.
- Users can set an access password to communities for avoid external viewers.
- Users can belong to more than once community.
- There have to exist a way for measure the trustfulness of each user in the community, a reputation system.
- There have to exist a mail notification system in real time for the activity in each community.
- The look and feel of the platform should fit to different sizes of screen.

Hardware requirements

- The application must run on MORElab servers.

- The performance of the application must be smooth and fast on every web browser.
- The platform must have a smartphone application.
- The platform must be able to manage a huge quantity of request concurrently with no fault.
- No new hardware than the already existing on MORElab must be need.

6.1.2 Hardware specification

For fulfilling the previous requirements and taking advantage of the web server technology already available on MORElab, the server where the application is deployed belongs to the laboratory and the same domain of MORElab applications is used. The application is accessible on http://apps.morelab.deusto.es/auzonet

The hardware requirements are going to fit the existing hardware shared for other applications developed on MORElab, specifically, in the server called "Olentzero" with the following specifications:

- Intel(R) Xeon(R) CPU E5-2430 @ 2.20GHz
- 16 GB RAM
- 1TB Hard disk

Deploying here the application where already have available a complete software environment of web dependencies save time reusing already existing modules.

6.1.3 Software specification

The web application running on "Olentzero" server must have the following software installed for correct performance:

Ubuntu Server 14.04.4 LTS: Open Source operating system running on "Olentzero" server.

Python 2.7 or newer: Python is a widely used high-level, general-purpose, interpreted, dynamic programming language.

Django framework 1.9: Develop the web application

gunicorn: Python WSGI HTTP server for UNIX to serve the web application

nginx: A free, Open-Source, high-performance HTTP server and reverse proxy for serve the static and media files.

pip: A python package manager for install python packages in the virtual environment.

supervisord: A client/server system to monitor and control processes on UNIX-like operating systems for monitoring the gunicorn instance that serves the web application.

virtualenv: A tool to create isolated python environments for create an isolated environment for developing and deploying purposes, without library conflicts.

virtualenvwrapper: A set of extensions to virtualenv for ease the use of virtualenv through the command line.

MySQL: For storing the data.

Django is a modular web framework extensible via applications, which are basically plugins that add to the core functionality of Django special features, in Auzonet development, has been installed the following ones:

django-bootstrap: For native integration of front-end library Twitter Bootstrap.

django-bower: For manage all the front-end packages and libraries.

django-favicon: For generating and inject the appropriate sizes of favicons in different browsers.

Mobile application

Mobile application has been designed as a wrapper of web version, this decision is because for developing native applications for all mobile operating systems require a Restful API and independent developments that could not fit in project's calendar.

For performing this conversion, Apache Cordova solution has been used, being the perfect choice thanks to the Apache Cordova Open Source license.

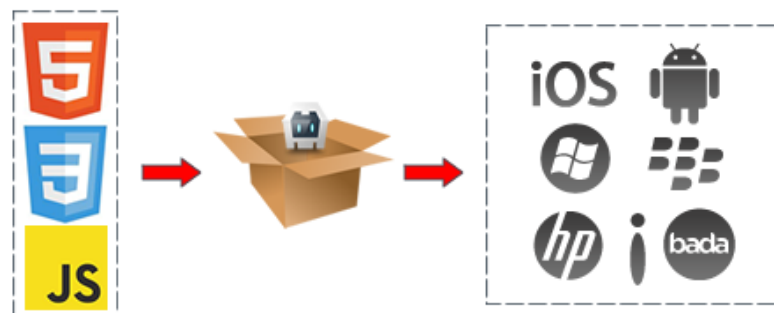The strucutre of an application made with Apache Cordova is the represented in figure 6.1

Figure 6.1: Apache Cordova structure

Apache Cordova allow access to native hardware features of the device from web technologies and provide customization of the application environment such as loading screen and native icon.

TODO: A schema of how the software layers are on top o the other.

Back-end

Auzonet, has some administration functionality on a separate web in the same server, accessible in the URL http://apps.morelab.deusto.es/auzonet/admin with an admin account, there are the management console in which is possible to manage all tables on the database schema.

Some features considered advanced, are only possible to perform here, contacting with the admin of the site. For example:

- Setting new categories for requests and offers.
- Change details of already registered users and communities.

• Remove user accounts.
• Resetting passwords.

In the last chapter are included ways to implement some of the mentioned options in the public user interface, but by the moment, only the features that has been considered essential has been implemented.

User registration process

When a new user is registered in Auzonet, a tiny wizard starts asking him if is searching an already created community or wants to register one for the first time, in case of registering a new one, the form displayed will gather data directly from Bilbao's council public data for showing the neighbors selection drop downs.

A set of subsequent AJAX calls to WeLive REST API are performed for obtaining the data.

```
1  oˆˆnIoˆˆnIoˆˆnIint nmainp(){
2  oˆˆnIoˆˆnIoˆˆnIprintfp(l+s"hello world"p);
3  oˆˆnIoˆˆnIoˆˆnIreturn l+m+mi0p;
4  oˆˆnIoˆˆnIoˆˆnIp}
```

Algorithm 6.1: "Hola mundo" en C.

```
kprint l+s+s2l+s+s2hola mundo!l+s+s2
```

During the registration process, an email is sent to the user for confirming the set up of the new account and another one is sent when the community process creation has finished and the user belongs to it automatically.

After the portal selection process, the user can set a password to the community, that password is stored in the community table of the database and is necessary for allowing others to join into community.

From the user perspective that joins into a existing community, can select the community at the beginning of the process after registering from a drop down showing all registered ones, if the community is private, a password will be prompt, once the join process is finished, will receive an email with a custom welcome message set by the creator of the community.

TODO: Add a chart of the registration flow process.

6.1.4 Deployment considerations

As any other Django developed application, some steps have to be followed for setting a clean installation of Auzonet in a server

TODO: Put a network map

During the development on the student's laptop, the application has been using an SQlite database running on a local server, this type of database is inadequate because the technical limitations of performance, for ensuring a fast and reliably data access a MySQL database.

"Olentzero" as previously mentioned already have installed typical software used by Web Applications, and there is no need to install a new instance of it, only creating an specific schema for Auzonet and configure the settings file for accessing it is required.

6.1.5 Issue management

All changes and requests received during the development process of the project will be managed through the following procedure:

1. Communication via electronic support of the requested modification.
2. Request meeting with the project manager for decide if the change has to be implemented.
3. If so, assess the technical changes required on the platform before deploying.
4. If the changes are technically feasible and fit the budget and schedule, proceed to implementation.
5. Modify the work plan and budget.

6.2. TECHNOLOGY

6.2.1 Software

The software involved in the development of this project is structured in the following blocks:

- Web application
  - Front-end
  - Back-end
- Mobile application

Web application: Front-end

HTML5

One of the requirements of the project was guarantee the maximum compatibility with the most common web browsers, because of that, for writing the views of the user interface, the last version of the HyperText Markup Language has been selected, HTML5 contain several improvements in the syntax that makes easier to develop and better for indexing in search engines, performance in rendering rich elements like videos and more.

HTML5 also come with more improvements that makes the template code cleaner easier to maintain, and provide certain characteristics that improve the smartphone page viewing, specially in this project, because of the technologies used for deploying in mobile devices, ensuring the best performance on smartphones is priority.

CSS3

For achieving a clear structure in the front end code, taking account the importance of getting an attractive design, the code responsible of all the visuals has separated in different external CSS files, actually, the most part of the CSS classes belongs to Twitter Bootstrap front end framework,

but in some parts of the web like the landing page, it was necessary to develop custom classes in custom files.

The decision to select version 3 of CSS is because as with HTML, this project also serve as learning proposal of the latest technologies for developing the future web, although is already know some feature incompatibilities in some browsers, the advantages in number of lines of code saving and new effects has taken more in account.

JavaScript

Being the language most extended on web development, the election of this technology ensure maximum compatibility with almost every device on the market. All the logic placed on client side is programmed with this language, from the AJAX calls, to controlling the user interface animations along the application.

JavaScript makes easy to manage dynamically any element on the pages, and thanks to dynamic panels fired by JavaScript events has been saved a handle of HTML pages on this project.

The following JavaScript libraries has been used for improve the language capabilities:

jQuery To simplify most used web development JavaScript operations.

jQuery UI For adding a toolkit of user interface elements.

jQuery numeric For helping to avoid non-numeric input in some forms.

Chart.js For drawing charts.

Twitter Bootstrap

Twitter Bootstrap is an easy way to provide good user experience through good user interface already built in CSS classes already created by experts in design, moreover, all their CSS classes are responsive and prepared for represent contents in wide variety of screen sizes using media queries.

Some parts of the application are fully designed by the student, like the landing page or logo, but getting benefit of all the CSS classes designed by professionals in Twitter Bootstrap framework, help to get a more sophisticated and advanced look to the application.

Figure 6.2: Twitter Bootstrap

Bower

Today, the amount of libraries involved in a web development is huge, and keep the latest versions of each is a nightmare for programmers, with Bower, the libraries can be managed from repositories, and be installed, updated or removed with one command line.

As similar UNIX tools like apt-get on Linux, Bower unify the package dependence for web developments providing a same file structure for avoiding library reference errors in the HTML pages and keep a detailed log of all the operations related.

Figure 6.3: Bower logo

Web application: Back-end

Unlike the front-end there are high variety of programming languages and frameworks for server side, the most extended is PHP but the learning proposal of the project was learn building a real application a new language and framework for web development, PHP is a language with the student already had used during the degree and researching on internet, others with high capabilities are growing.

Along the research, the following technologies were taken account:

- Python with Django
- JavaScript with AngularJS
- Ruby with Ruby on rails
- Java with Spring

Finally, Python was selected because the clean syntax and the philosophy of Django framework to avoid repetition in the most used web tasks and their solid community.

Figure 6.4: Django and Python logos

The documentation site of Django framework is clear and well structured and provides API's for

all the common needs for every web development like an ORM for database mapping or security features, moreover, is possible to build your own custom functionalities on top of them.

Mobile application

For providing a native environment to Auzonet in mobile platforms the native SDK of each platfor was desestimated due the limited resources for developing the project, thanks to web technologies and the responsive user interface designed in the web application, Apache Cordova framework has been used for provide a native look and feel on the common mobile operating systems.

Apache Cordova is a framework for create mobile applications using HTML, CSS and JavaScript gaining access to native features of the hardware like accelerometer or camera, as project from Apache Foundation, is licensed under Apache license that authorizes to use free of charge the software.

Figure 6.5: Apache foundation logo

Apache Cordova allows to get advantage of all the existing software written in the web application and port the experience to a native application distributable via mobile stores and porting to Android, iOS or Windows Phone.

For creating the application, has been necessary create artworks for different screen sizes and device orientations of the icon and splash screen.
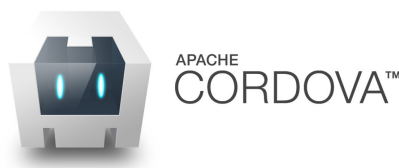
Figure 6.6: Apache Cordova logo

6.3. TOOLS

6.3.1 Code editor

The code editor used along the project for writing Python code and HTML front-end is Atom by GitHub.

Figure 6.7: Atom editor logo

Atom is an application built with web technologies and open source that allow to be modified by the community, because of that, the wide variety of plug-ins such as language syntax support or Git integrations is huge.

Before this project, the student had experience with other privative tools like Sublime Text but at the beginning of the development, the student made a small research for a new tool to learn for improve productivity in the programming phase.

Atom features like embedded browser and terminal, has become a huge productivity boost during the development.

6.3.2   Image editor

Though most of the visual assets are provided by Twitter Bootstrap, others like logo and mobile icons and splash screens has been created using a image editor.

Figure 6.8: Pixelmator logo

Pixelmator is a Mac application based on the popular Adobe Photoshop with less features but enough to design and prepare the visual assets used in Auzonet. Was selected because the student already owned a license and is easy to use.

6.3.3   Operating system

All the development process has take place on an Apple MacBook Pro of early 2011, property of the student with OS X El Capitan version 10.11, the UNIX solid architecture and good user experience became the perfect choice for being the main computer used for programming.

Figure 6.9: OS X El Capitan logo

The UNIX-like capabilities of this operating system has become an advantage for running almost every tool involved in a web development process.

6.3.4 Version control

Figure 6.10: GIT logo

Git has been the software responsible of keep version control of all the files of the application. Git is an open-source software developed by Linus Torvalds widely used in the industry, allows to keep a central repository and create branches separated from the main development code.

Figure 6.11: GitHub logo

GitHub is a service created for hosting Git repositories and allow store public and private code, thanks to the student pack offered by the company, code has been hosted in a private repository for free that, otherwise would be needed a paid subscription.

32

6.3.5 Testing

All the tests performed has been focused y the end user perspective, along two weeks the application was released under beta status to a closed group of people closer to the student for making a simulation of real usage.

The proposed tests starts with the creation of a new account by the user without any assistance watching the usability problems that the user have, and taking notes of how the user navigate through the app.

The proposed workflow for the test are always the same:

1. Register a new account.
2. Register a new community with password.
3. Create a new offer in the community.
4. Create a new request in the community.
5. Post a new advice on the community board.
6. Post a new info message on community board.
7. Logout
8. Create another account
9. Joining to the previously created community
10. Hire the offer created by the previous user and finish the agreement voting each other
11. Attend the request created by the previous user and finish the agreement voting each other
12. Enter in profile view and check the recent activity and karma.
13. Logout.

Same process with the mobile application.

All faults and user experience suggestions were gathered in the task manager software and tagged for fixing as soon as possible.

7. CONCLUSIONS AND FUTURE WORK

This project has developed a full-stack platform for citizens using Open Data very similar to already existing, well-known applications of every day life, in this chapter some conclusions and future lines of work will be presented.

7.1. DATA MINING

7.2. CONCLUSIONS

Today, we are surrounded by services that centralize the logic and data on cloud servers, smartphones quickly became the angular piece for interacting with them, having the knowledge of creating the full-stack technologies involved in these solutions is essential for every computer engineer.

The aim of learning a full stack technologies creating a real application based on social interaction and Open Data has achieved successfully, today, more than ever, we have a huge variety of technologies available, select the appropriate ones for a service a provide a good user experience is more difficult.

As can be seen, this project ended successfully achieving all the major objectives proposed from the beginning and implemented all the MORElab requested requirements with WeLive platform. A mobile application has been created ate the same time, bringing all the core functionality to different mobile operating systems and only latest web technologies has been used for proving the maturity of these solutions.

Both applications, mobile and web are fully functional, running on servers hosted in DeustoTech and ready to deploy in bigger environment with small tweaks. In the other hand, data mining purposes of user's data could not be tested or developed, due time limitations but the software architecture is totally prepared for future data analytics features.

7.3. FUTURE LINES OF WORK

Although the application is fully functional and complete, there are some features that could be interesting to develop in further releases:

- User profile customization by the end user.
- Facebook, Twitter and Google login integration.
- Native mobile apps.
- Public REST Api for allowing third party integrations.

7. CONCLUSIONS AND FUTURE WORK

- Full featured data analysis portal for public institutions.
- Automatic community detection using HTML5 geolocation API.
- Implement a search field for offers and requests.
- Add support for more cities and extend the scope of the posts.

All these features are really plausible to implement due the actual architecture of the platform.

8.   BIBLIOGRAPHY

9. ACRONYMS

label description

10. ACKNOWLEDGMENTS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum condimentum lectus et mauris pellentesque, non porta elit congue. Sed mattis sodales lobortis. Interdum et malesuada fames ac ante ipsum primis in faucibus. Nullam imperdiet lectus eget iaculis vestibulum. Curabitur nunc orci, fringilla nec pellentesque vitae, luctus placerat lorem. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Maecenas luctus tincidunt libero, eu consectetur ipsum viverra ut. Nam aliquam varius nulla sit amet pretium. Proin sollicitudin orci id enim commodo auctor. Sed eget lectus neque.

Proin suscipit consectetur semper. In id ipsum elit. Quisque nulla ex, consequat ac consectetur posuere, auctor eget tellus. Pellentesque pharetra sodales ornare. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Donec porttitor pretium blandit. Nulla facilisi. Duis pellentesque est sit amet risus posuere, vel consectetur nulla sodales. Donec porttitor interdum risus, ac convallis risus consequat quis. Integer sodales nunc nec neque efficitur luctus. Morbi tempor vestibulum iaculis. Aenean hendrerit nibh libero, id efficitur enim scelerisque non. Fusce ut quam sem. Praesent ut molestie orci, in mattis ligula. Aliquam et leo lacinia, ultrices purus nec, dictum est.

Ut cursus velit nunc, et tempor mi blandit laoreet. Fusce condimentum ipsum in arcu auctor, et efficitur massa luctus. Ut sapien libero, venenatis a ante eu, dignissim posuere ligula. Phasellus in purus accumsan, imperdiet nunc vitae, aliquam odio. Vestibulum dui orci, consequat eu faucibus imperdiet, faucibus a lectus. Cras imperdiet quam vel placerat eleifend. Morbi placerat gravida leo a tincidunt. Sed eu pulvinar risus, ac fermentum odio. Vivamus id dui magna. Fusce pellentesque turpis ex, ac blandit augue iaculis in. In sollicitudin bibendum dolor, in maximus velit maximus ac.

Cras venenatis vel sapien id volutpat. Sed pharetra turpis sed dui euismod sagittis. Vivamus lobortis eros ut risus ultrices tempus. Etiam sodales velit eu accumsan consectetur. Duis suscipit nisl a lacinia volutpat. Phasellus congue purus ex, nec bibendum leo ultrices lacinia. Praesent vel augue luctus, placerat ex nec, semper nisl.

Nunc congue enim in placerat hendrerit. Aliquam erat volutpat. Donec mollis commodo neque. Vestibulum at viverra ante. Aenean congue non urna quis consequat. Curabitur consequat, lacus et vestibulum euismod, ex odio placerat orci, vel lobortis tortor est in nunc. Sed rutrum mi non urna iaculis, vitae fermentum lectus iaculis. Pellentesque ornare leo eu risus ultricies, at sagittis odio interdum. Maecenas eu mattis tellus. Curabitur pellentesque accumsan nisl. Pellentesque eleifend sit amet arcu sit amet ornare. Phasellus mollis, turpis quis consequat interdum, justo tortor finibus leo, sit amet ultrices purus lacus et augue. In non aliquam dui. Vestibulum id sagittis est, et accumsan eros. Praesent eleifend imperdiet felis, sit amet mollis neque fermentum sit amet.

11. AGRADECIMIENTOS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum condimentum lectus et mauris pellentesque, non porta elit congue. Sed mattis sodales lobortis. Interdum et malesuada fames ac ante ipsum primis in faucibus. Nullam imperdiet lectus eget iaculis vestibulum. Curabitur nunc orci, fringilla nec pellentesque vitae, luctus placerat lorem. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Maecenas luctus tincidunt libero, eu consectetur ipsum viverra ut. Nam aliquam varius nulla sit amet pretium. Proin sollicitudin orci id enim commodo auctor. Sed eget lectus neque.

Proin suscipit consectetur semper. In id ipsum elit. Quisque nulla ex, consequat ac consectetur posuere, auctor eget tellus. Pellentesque pharetra sodales ornare. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Donec porttitor pretium blandit. Nulla facilisi. Duis pellentesque est sit amet risus posuere, vel consectetur nulla sodales. Donec porttitor interdum risus, ac convallis risus consequat quis. Integer sodales nunc nec neque efficitur luctus. Morbi tempor vestibulum iaculis. Aenean hendrerit nibh libero, id efficitur enim scelerisque non. Fusce ut quam sem. Praesent ut molestie orci, in mattis ligula. Aliquam et leo lacinia, ultrices purus nec, dictum est.

Ut cursus velit nunc, et tempor mi blandit laoreet. Fusce condimentum ipsum in arcu auctor, et efficitur massa luctus. Ut sapien libero, venenatis a ante eu, dignissim posuere ligula. Phasellus in purus accumsan, imperdiet nunc vitae, aliquam odio. Vestibulum dui orci, consequat eu faucibus imperdiet, faucibus a lectus. Cras imperdiet quam vel placerat eleifend. Morbi placerat gravida leo a tincidunt. Sed eu pulvinar risus, ac fermentum odio. Vivamus id dui magna. Fusce pellentesque turpis ex, ac blandit augue iaculis in. In sollicitudin bibendum dolor, in maximus velit maximus ac.

Cras venenatis vel sapien id volutpat. Sed pharetra turpis sed dui euismod sagittis. Vivamus lobortis eros ut risus ultrices tempus. Etiam sodales velit eu accumsan consectetur. Duis suscipit nisl a lacinia volutpat. Phasellus congue purus ex, nec bibendum leo ultrices lacinia. Praesent vel augue luctus, placerat ex nec, semper nisl.

Nunc congue enim in placerat hendrerit. Aliquam erat volutpat. Donec mollis commodo neque. Vestibulum at viverra ante. Aenean congue non urna quis consequat. Curabitur consequat, lacus et vestibulum euismod, ex odio placerat orci, vel lobortis tortor est in nunc. Sed rutrum mi non urna iaculis, vitae fermentum lectus iaculis. Pellentesque ornare leo eu risus ultricies, at sagittis odio interdum. Maecenas eu mattis tellus. Curabitur pellentesque accumsan nisl. Pellentesque eleifend sit amet arcu sit amet ornare. Phasellus mollis, turpis quis consequat interdum, justo tortor finibus leo, sit amet ultrices purus lacus et augue. In non aliquam dui. Vestibulum id sagittis est, et accumsan eros. Praesent eleifend imperdiet felis, sit amet mollis neque fermentum sit amet.
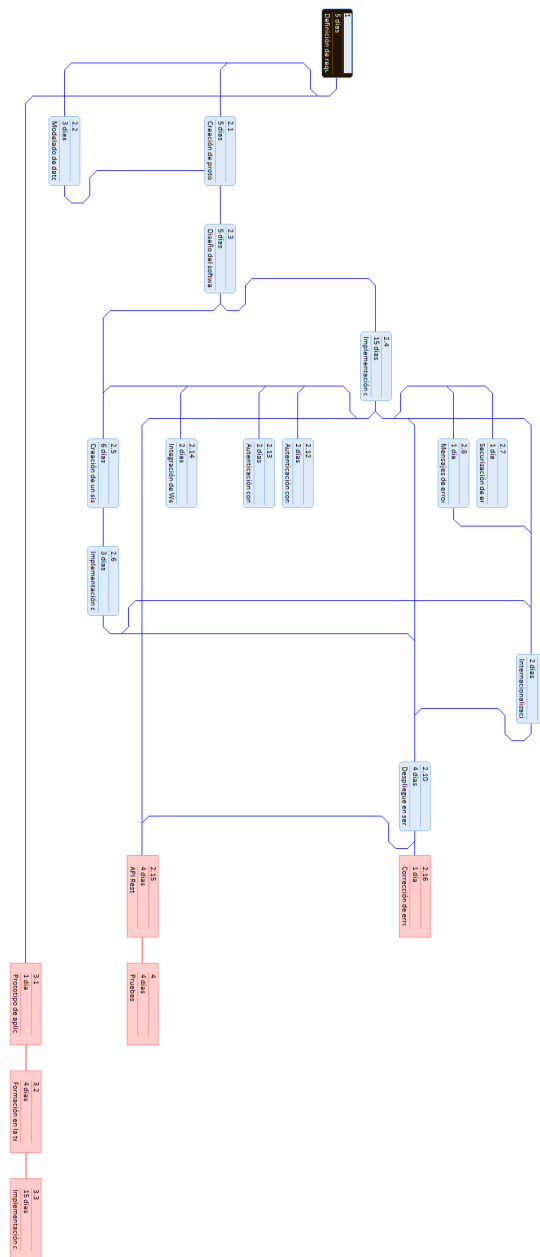
A.1. PRECEDENCE DIAGRAM2

Figure A.1: Precedence diagram