uc3m | Universidad Carlos III de Madrid

University Degree in Computer Science and Engineering
2017-2018

*Bachelor Thesis*

# MI-KPI: MEASURING THE DEGREE OF INNOVATION IN HIGHER EDUCATION INSTITUTIONS

*Autor: Leire Litwin Echevarría*

*Tutor: José María Álvarez Rodríguez*

# ACKNOWLEDGEMENTS

I would like to thank my parents, María Lourdes Echevarría and Darek Jacek Litwin for all their constant support during the degree and for believing in me at all times. Especially my mother, for all the candles she has lit for me.

On the other hand, to my friends Ana López, Alicia Laguna, Lucía García and Ana Busto, for always being there, for all the library hours and for being my great discovery at the university, as well as all the friends that I have made in the FAM.

My classmates Juan Tafarello, Ignacio García, Alejandro Al Maullem, Alberto Rojas, Adrián Sánchez and Álvaro Santos, for all the projects done together, the long classes and all the challenges that we have overcome with effort, stress and laughter of all the moments during the degree.

But in this case, special thanks to my tutor, José María Álvarez Rodríguez, for helping me whenever I asked him during the bachelor's thesis and for the positive and cheerful attitude he has always had.

# ABSTRACT

This bachelor thesis is based on measuring the innovation in higher educational institutions by measuring the number of MOOCs and computing some metrics. In order to do so, the process of the project will be divided into three stages: data collection, data computation and data visualization. First, the data will be collected from some MOOCs providers. Then, the data will be cleaned and the metrics will be computed. Finally, a web application will be created and the data will be displayed by means of ranking tables and graphs.

# Table of Contents

# Index of Tables

# List of Figures

# 1. Introduction

## 1.1 Overview

Technology is the future and evidence of its advances are perceived everywhere. Its effects can be seen in work, communication, social media, leisure, health, even at home and in our daily lives. However, it is worth mentioning that one of the aspects that has innovated the most today's technology is education. With the use of the Internet, it is radically changing the methodology that is used to teach all over the world. Nowadays, more and more resources are available in the Internet, since learning a new language to learn how to create a circuit. There is literally a topic for almost anything you want to learn that can be found in the Internet, as it is connecting more and more devices from all around the world. People can post online their expertise and therefore, knowledge is worldwide connected.

Considering the great impact that the Internet has had in education, I decided to make this project, which will consist of measuring the innovation of most of the worldwide universities regarding to the number of MOOCs that have uploaded online in platforms like Coursera or edX.

What is it a MOOC? MOOC refers to Massive Open Online Course. It is a online program of open access in the Internet offered by the leading worldwide universities. MOOCs are often based on a degree that the university is offering to their students, therefore it is an opportunity to study courses from prestigious institutions, like Stanford, the Imperial College London, MIT, or even well-known enterprises like Google or museums like the MOMA. Teaching materials may include videos, set reading, problems to solve and student forums.

## 1.2 Motivation

The motivation of this project was that the current top universities of the world are mostly private or inaccessible to most part of the population. According to my research, the most well-known rankings of universities (Shanghai Ranking, QS Top Universities or The World University Rankings) are taking into account the following factors in their methodology:

1. Quality of Education (based on Nobel prizes, awards to students and surveys)
2. Quality of Faculty Staff (Researchers and Nobel prizes)
3. Research Output Papers
4. Per Capita Performance or Industry Income
5. International Student / Faculty ratio

Nevertheless, none of the rankings are taking into account the innovation of the universities and how much content they have available for students through the Internet. Thus, my goal with this project is to measure the innovation of this universities by comparing the number of online courses they have available and then compare the results with these other well-known rankings.

## 1.3 Objectives

Choosing a degree is a transcendental decision and even more by having this variety of universities around the world. It is not only about choosing a degree, but also about choosing the right university. The main objective to be achieved is to help students, teachers, doctors or any person related to the educational environment to have access to a different kind of ranking that shows progress and advances in universities.

This project meets the following objectives:

1) Collect data from several sources and cleaning the necessary data.
2) Measure the innovation of the universities with regard to technology and its advances (MOOCs) by creating some metrics.
3) Create a comparison between today's regular rankings and the one that measures the MOOCs according to the different metrics.
4) Display the top most innovative universities according to the metrics.
5) Filter the universities according to different criteria.
6) Compare two different universities.

# 2. State of Art

In the state of the art, it is established the problem of quality indicators in the higher education sector for rankings and traditional education. In addition, the question arises whether MOOCs serve as indicators of innovation at Universities.

Moreover, it is described a compilation of results of other researches that have been carried out on the selected research topic. It is about establishing what has been recently done on the selected topic.

According to the research about MOOCs, there is no Universities Ranking based on the number of MOOCs or measuring the innovation. There are some rankings about the best MOOC courses offered by prestigious universities [1] and there are some leading rankings of universities provided by the following organizations:

1) ARWU - Shanghai Ranking
2) The QS World University Rankings - Top Universities
3) Times Higher Education
4) Webometrics
5) US News National Ranking

Therefore, this section will analyze the world leading universities rankings and their methodologies to create rankings. From the five rankings above, the methodologies of the first four will be evaluated, since the last one is constituted just by universities from United States.

## 2.1 ARWU (Shanghai Ranking)

The Academic Ranking of World Universities (ARWU) was first published in June 2003. Nevertheless, in 2009 the Academic Ranking of World Universities (ARWU) has been published and copyrighted by Shanghai Ranking Consultancy. Today, ARWU uses six objective indicators divided in four categories to rank world universities. More than 1200 universities are ranked by ARWU every year and the best 500 are published. [2]

The ARWU Ranking is well-known for its scientific, stable and transparent methodology. While evaluating their methodology of 2017, it can be seen that the faculty quality and research output constitute up to 80% of the weight, ignoring indicators like internationalization or innovation.

| Indicator | Description | Weight |
|---|---|---|
| Education Quality | Alumni of an institution winning Nobel Prizes and Fields Medals | 10% |
| Faculty Quality | Staff of an institution winning Nobel Prizes and Fields Medals Award (20%). Highly cited researchers in 21 broad subject categories (20%). | 40% |
| Research Output | Papers published in Nature and Science (20%). Papers indexed in Science Citation Index-expanded and Social Science Citation Index PUB (20%). | 40% |
| Per Capita Performance | Per capita academic performance of an institution. | 10% |

*Table 1: ARWU methodology of 2017 (Shanghai Ranking)*

## 2.2 QS Top Universities

QS World University Rankings is an annual publication of university rankings by Quacquarelli Symonds. QS Top Universities evaluates more than 1000 universities and 1000 of them are published in their general ranking. They also provide rankings by subject, masters ranking and employability ranking. However, it is only going to be analyzed the general ranking according to the methodology described below.

Their methodology of 2017 is highly based in surveys and ratios calculation. They take into account Academic and Employer surveys, the ratios calculation, citations and internationalization. However, they lack of industry or innovation indicators. [3]

| Indicator | Description | Weight |
|---|---|---|
| Academic Reputation | Based on their Academic Survey regarding teaching and research quality. | 40% |
| Employer Reputation | Based on their QS Employer Survey, and asks employers to identify those institutions from which they source the most competent, innovative, effective graduates. | 10% |
| Faculty/ Student ratio | It assesses the extent to which institutions are able to provide students with meaningful access to lecturers and tutors. | 20% |
| Citations per faculty | It is calculated dividing the total number of citations received by all papers produced by an institution across a five-year period by the number of faculty members at that institution. | 20% |
| International student ratio | It provides an internationalized higher education sector. It measures the ratio of international students and staff according to the total number of students and staff members. | 5% |
| International staff ratio | | 5% |

*Table 2: QS Top Universities methodology of 2017*

## 2.3 Times Higher Education

Times Higher Education (THE) is an English magazine about news in higher education. THE also provides data for the world's research-led institutions and creates one of the most well-known world rankings in higher education. It is based in London and used to be a partner with QS, but in 2009 they broke their partnership and since then has signed an agreement with Thomson Reuters.

Similarly to QS, their methodology from 2018 is highly based in surveys and ratios calculation. However, they take into account not only teaching and research as the majority of rankings, but also Internationalization and Industry Income. Regarding industry, it is briefly considered innovation, weighting lower than a 2.5%. [4]

| Indicator | Description | Weight |
|---|---|---|
| Teaching | Reputation survey: 15%<br>Staff-to-student ratio: 4.5%<br>Doctorate-to-bachelor's ratio: 2.25%<br>Doctorates-awarded- to-academic-staff ratio: 6%<br>Institutional income: 2.25% | 30% |
| Research | Reputation survey: 18%<br>Research income: 6%<br>Research productivity: 6% | 30% |
| Citations | Capturing the average number of times a university's published work is cited by scholars globally. | 30% |
| International Outlook | International-to-domestic-student ratio: 2.5%<br>International-to-domestic-staff ratio: 2.5%<br>International collaboration: 2.5% | 7.5% |
| Industry Income | A university's ability to help industry with innovations, inventions and consultancy. | 2.5% |

*Table 3: Times Higher Education methodology of 2018*

## 2.4 Webometrics

The "Webometrics Ranking of World Universities" is an initiative ranking from Spain. It is carried out by the Cybermetrics Lab, a research group belonging to the Consejo Superior de Investigaciones Científicas (CSIC). Cybermetrics Lab uses methods to evaluate indicators to measure the scientific activity on the Web. Apart from that they created a Ranking of World Universities with those indicators.

The 2018 methodology of Webometrics is based on the scientific activity on the Web, opposite to other rankings, they do not measure Teaching or Research activities of the universities, but any uploaded content online or backlinks to the university's webpages. Their methodology is divided in five categories - presence, visibility, transparency and excellence.[5]

| Indicator | Description | Weight |
|---|---|---|
| Presence | Size (number of webpages) of the main web domain of the institution. It includes all the subdomains sharing the same (central or main) web domain and all the file types including rich files like pdf documents. | 5% |
| Visibility | Number of external networks (subnets) originating backlinks to the institution's webpages, after the values being normalized. | 50% |
| Transparency | Number of citations from Top authors according to the source. | 10% |
| Excellence | Number of papers amongst the top 10% most cited in 26 disciplines in 5 year period of time. | 35% |

*Table 4: Webometrics methodology of 2018*

## 2.5 Methodologies Outcome

What can be concluded from the evaluation of the previous methodologies is that there is no well-known ranking that has an indicator or parameter for measuring innovation. Most of the indicators regarding teaching, research and citations are necessary, although some of them are measured in an outdated way. Innovation and technological resources are not measured anyhow in higher institutions rankings, being one critical indicator of the universities evolution and adaptation to the future. Thus, in this project, it will be measured and compared with the Shanghai Ranking results.

## 2.6 Tableau Software

With respect to the visualization module, Tableau Software is a enterprise that produces interactive data visualization products. The software will query databases, OLAP cubes or spreadsheets and display the results into interactive graphs.

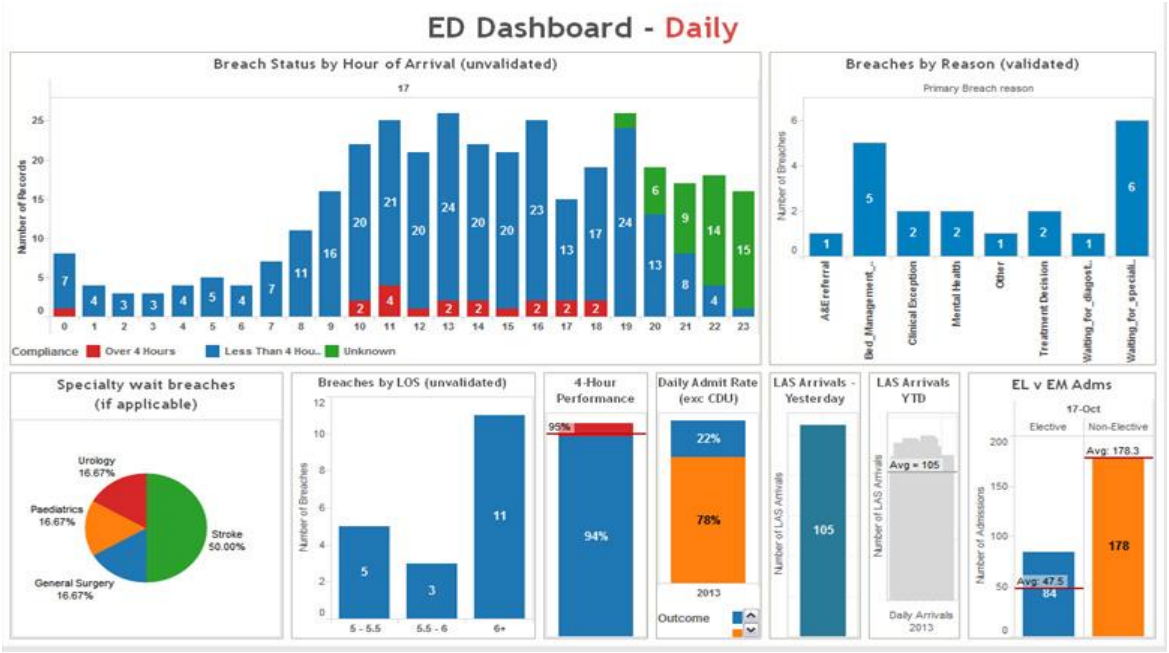



*Figure 1: Tableau example of graphs [6]*

This tool is extremely useful for companies and data analysis. However, it is still a canned product and a tool too powerful. Therefore, in order to improve the abilities and skills for development and user interfaces, it was implemented using Highcharts.

# 3. Analysis

In this section will be done the requirements analysis of the application, which is composed of multiple tasks to determine the needs and functionalities of the systems. In order to do it, first the use cases will be developed and then the system requirements, both functional and non-functional.

In this process of evaluating the requirements of the system, there will be three main processes to consider - Data Collection, Computation and Visualization. First, the data needs to be collected. Then, this data needs some cleaning, processing and the metrics have to be computed. Finally, the data needs to be displayed in the web application.

## 3.1 User stories

User stories are part of an Agile approach that helps to write requirements by focusing on the desired functionality a user wants, being a user centric approach. It also helps to determine and target the type of user that will use the application, in this case people related to education, for instance teachers and students. They have to be short, with a description of the feature the user would like the system to have. In this way, it helps the developers to set the context of the system, create the requirements and design it in a better way. Some user stories for the "MOOCs Ranking" web application are:

| Targeting User | As a user, I want... | So, the user shall be able to... |
|---|---|---|
| High-school student | to choose the most suitable university for me | see a ranking and filter the best universities by country and by ranking, to improve the speed of the search |
| Erasmus student | to compare universities between each other | compare two universities and display the different results in graphs |
| University teacher | to do a research in different universities of the world | filter the ranking table according to different metrics |
| Data Curator | to maintain updated the web application | collect and clean the data, compute the metrics and insert it into the database |

*Table 5: User Stories*

## 3.2 Use cases

A use case is a description of the activities that the user will be able to perform in the system or process that is intended to be developed. The user will be able to interact in several ways with the system represented in the following graphs:

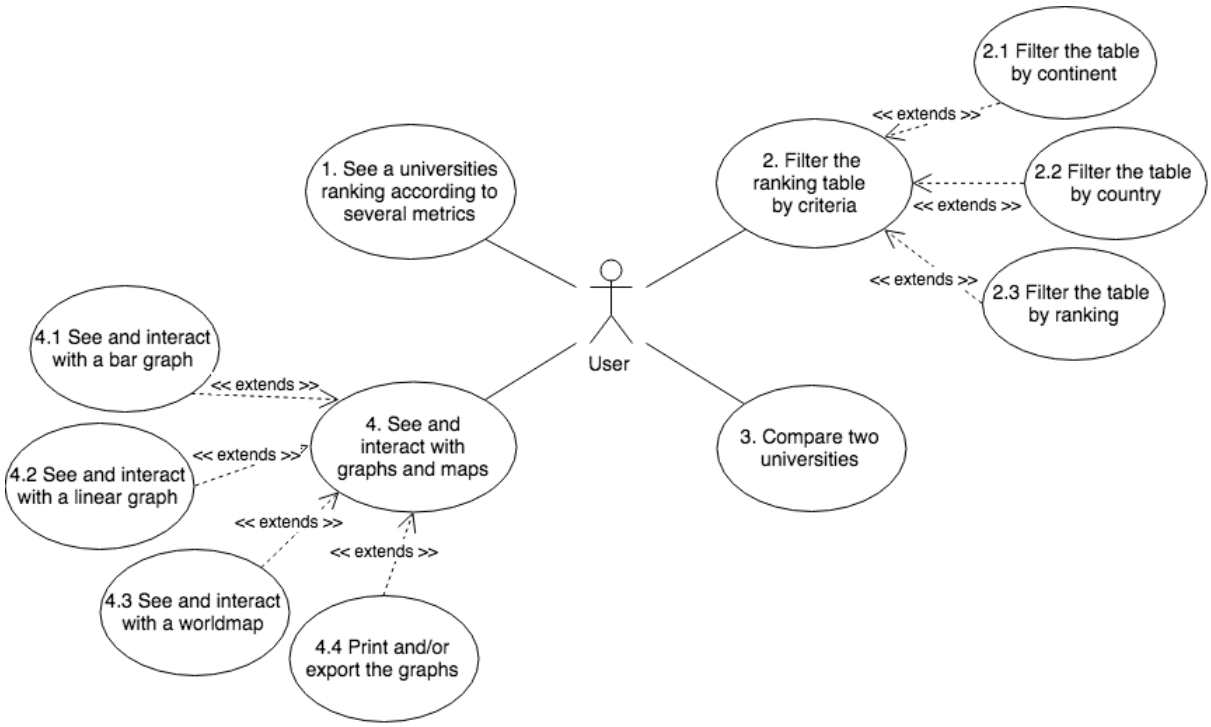



*Figure 2: Use cases for user*

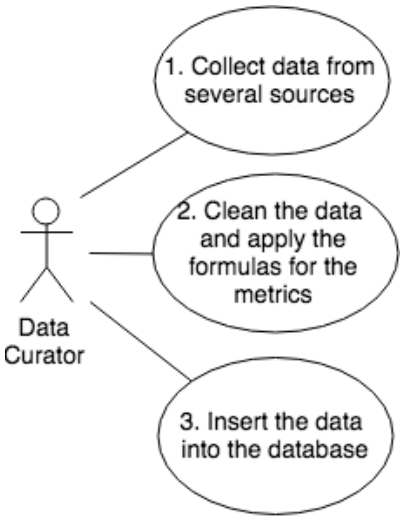



*Figure 3: Use cases for data curator*

## 3.3 Use Cases Specification

In this section, the aim is to specify how the use cases are going to be organized and the attributes that are going to be included in each table.

- Identifier: Unique identifier for the user cases, in the format UC-XX, being XX a number starting from 01, until the last one. Another identifier for admin cases, in the format AC-XX.
- Description: Short description of the use case.
- Precondition: The requisite that has to happen so the use case is executed.
- Postcondition: The consequence that will happen or action that will be taken when the use case is performed properly.

| | |
|---|---|
| **Use case identifier** | UC-01 |
| **Description** | The user shall be able to see a university ranking according to several metrics: metric 2, 9 and 10. |
| **Precondition** | The user shall load the data page in the website. |
| **Postcondition** | The user shall be able to interact with the ranking. |

*Table 6: Use Case 1 Specification*

| | |
|---|---|
| **Use case identifier** | UC-02 |
| **Description** | The user shall be able to filter the ranking table by criteria. These criteria can be by continent, country and/or ranking. |
| **Precondition** | The user shall see the table of the universities ranking. |
| **Postcondition** | The table shall change according to the parameters the user is using to filter it. |

*Table 7: Use Case 2 Specification*

| | |
|---|---|
| **Use case identifier** | UC-03 |
| **Description** | The user shall be able to compare two universities. |
| **Precondition** | The user shall enter the two names of the universities to compare in the dropdowns. |
| **Postcondition** | The page shall display some graphs comparing both universities. |

*Table 8: Use Case 3 Specification*

| | |
|---|---|
| **Use case identifier** | UC-04 |
| **Description** | The user shall be able to see and interact with graphs and maps. The graphs can be linear, pie and bar charts. |
| **Precondition** | The user shall enter the graphs page. |
| **Postcondition** | The page shall display some graphs according to the filtering parameters. |

*Table 9: Use Case 4 Specification*

| | |
|---|---|
| **Use case identifier** | AC-01 |
| **Description** | The admin shall be able to collect data from several sources. |
| **Precondition** | --- |
| **Postcondition** | The program shall download a file with all the data. |

*Table 10: Use Case 5 Specification*

| | |
|---|---|
| **Use case identifier** | AC-02 |
| **Description** | The admin shall be able to clean the data and apply the formulas for the metrics. |
| **Precondition** | The file shall be containing the downloaded data. |
| **Postcondition** | The file shall be computed, cleaned and in json/csv format. |

*Table 11: Use Case 6 Specification*

| | |
|---|---|
| **Use case identifier** | AC-03 |
| **Description** | The admin shall be able to insert the data into the database. |
| **Precondition** | The data shall be in a json/csv file format and cleaned. |
| **Postcondition** | The data shall be inserted into the database. |

*Table 12: Use Case 7 Specification*

## 3.4 System Requirements Specification

In this section, the aim is to specify how the system requirements are going to be organized and the attributes that are going to be included in each table. These system requirements are extracted from the use cases and atomizing as much as possible. The attributes that are going to be included in the tables are the following:

- Identifier: Unique identifier of the requirement, in the format FR-XX, being XX a number starting from 01, until the last one.
- Category: Functionality of the system.
- Description: Short description of the requirement.
- Dependency: Short description of another requirement if the actual requirement depends for its performance on this other requirement.

According to the ESA standards, the system requirements are going to classified in terms of functional and nonfunctional.

## 3.5 Functional Requirements

Functional requirements (FR) defines the capabilities and functions of the platform system. More specifically, functional requirements will be grouped in three different categories according to the functionality of the system: data collection, computation and visualization.

| Identifier | Category | Description | Use Cases | Dependency |
|---|---|---|---|---|
| FR-01 | Data Collection | The system acquisition module shall be able to gather data from Coursera, EdX and the Shanghai Ranking. | AC-01 | --- |
| FR-02 | Computation | The system computing module shall be able to organize the collected data and compute the MOOCs Ranking according to different metrics. | AC-02 | FR-01 |
| FR-03 | Computation | The system computing module shall not compute nor insert any negative numerical values for the data in the database. | AC-02 | FR-01 |

| | | | | |
|---|---|---|---|---|
| FR-04 | Computation | The system computing module shall have the data in a JSON file or CSV file format. | AC-03 | --- |
| FR-05 | Computation | The system computing module shall insert the file data into a MongoDB database. | AC-03 | FR-01, FR-02, FR-03, FR-04 |
| FR-06 | Visualization | The system visualization module (server side) shall retrieve the data from the database. | --- | FR-05 |
| FR-07 | Visualization | The system visualization module (client side) shall request the data from the database. | --- | FR-06 |
| FR-08 | Visualization | The system visualization module shall display a dynamic table with the data in a website according to several metrics. | UC-01 | FR-06 |
| FR-09 | Visualization | The system visualization module shall display and interact with graphs in the website. | UC-04 | FR-06 |
| FR-10 | Visualization | The system visualization module shall filter the data when the user interacts in the website. | UC-02 | --- |
| FR-11 | Visualization | The system visualization module shall be able to compare two universities. | UC-03 | FR-06 |
| FR-12 | Visualization | The system visualization module shall be able to download and/or print graphs. | UC-04 | --- |

*Table 13: Functional Requirements Specification*

## 3.6 Non-Functional Requirements

Non-functional (NFR) requirements refer to the constraints of the platform system. In this case the attributes that are going to be included in the tables are differing from the functional requirements. There will be the following:

- Identifier: Unique identifier of the requirement, in the format NFR-XX, being XX a number starting from 01, until the last one.
- Description: Short description of the requirement.

- Priority: The urgency of the implementation of the requirement. May be High, Medium, or Low.

| | |
|---|---|
| **Identifier** | NFR-01 Response time |
| **Description** | The maximum response time of the system or any server request shall be 5 seconds. |
| **Priority** | Medium |

*Table 14: Non-Functional Performance Requirement Specification*

| | |
|---|---|
| **Identifier** | NFR-02 Device Platform compatibility |
| **Description** | The system shall be accessible and responsive using any type of devices - computer, tablets or mobile phones - independently of the operating system. |
| **Priority** | High |

*Table 15: Non-Functional Accessibility Requirement Specification*

| | |
|---|---|
| **Identifier** | NFR-03 Browser compatibility |
| **Description** | The system shall be compatible with most of the browsers, but shall definitely work properly with the following ones:<br>● Google Chrome<br>● Mozilla Firefox<br>● Safari<br>● Opera |
| **Priority** | High |

*Table 16: Non-Functional Compatibility Requirement Specification*

| | |
|---|---|
| **Identifier** | NFR-04 Server uptime |
| **Description** | The server shall be hosted in Heroku and uptime at least 99.9% of the time. If the dyno fails, there is no support dyno to keep the server on, but the percentage of failure is less than 0.1%. |
| **Priority** | High |

*Table 17: Non-Functional Deployment Requirement Specification*

| | |
|---|---|
| **Identifier** | NFR-05 Sharing data in JSON |
| **Description** | The API of the system shall be in JSON to support interoperability. |
| **Priority** | High |

*Table 18: Non-Functional Interoperability Requirement Specification*

| | |
|---|---|
| **Identifier** | NFR-06 Data Integrity |
| **Description** | The data shall be maintained accurate and consistent through the process of collecting it, cleaning it and computing the metrics. |
| **Priority** | High |

*Table 19: Non-Functional Data Integrity Requirement Specification*

| | |
|---|---|
| **Identifier** | NFR-07 User Interface Design |
| **Description** | According to design patterns, the system shall have an effective page layout, having the navigation menu at the top, consistent sidebars, dynamic responsive tables, clear layout and, in general, a high ease of use for the end user. |
| **Priority** | High |

*Table 20: Non-Functional Usability Requirement Specification*

| | |
|---|---|
| **Identifier** | NFR-08 Multilingualism |
| **Description** | The system shall have multilingualism supported - the website shall be displayed in English or Spanish. |
| **Priority** | Medium |

*Table 21: Non-Functional Second Usability Requirement Specification*

# 4. Design

In this section will be done the architecture design of the software, which is composed by high level structures representations of the web application and it will follow the 4+1 architectural model. It will be constituted by the system overview, the methodology, technological environment and then it will follow the 4+1 model. Finally, it will design the database model and the User Interface by means of Mockups.

## 4.1 System Overview

The project of universities ranking and metrics according to MOOCs will have the following design overview.

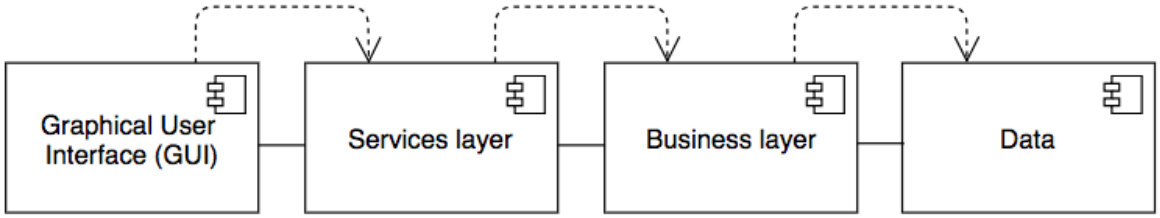



*Figure 4: System Overview diagram*

First, it will have the data layer, in which all the data will be collected. The business layer will process the data and compute the necessary parameters. Then, the services layer will be in charge of dealing with the web application core functionalities. Finally, the graphical user interface layer will be displaying all this collected, cleaned, processed and computed information to the user in an attractive way to the end user.

This section will also explain the connections between the web application and external entities that interact with it. In this case, the entities will not interact directly with the web application, but they are used in order to collect the data and compute the ranking for the web application. This “interaction” will be illustrated in the following diagram.

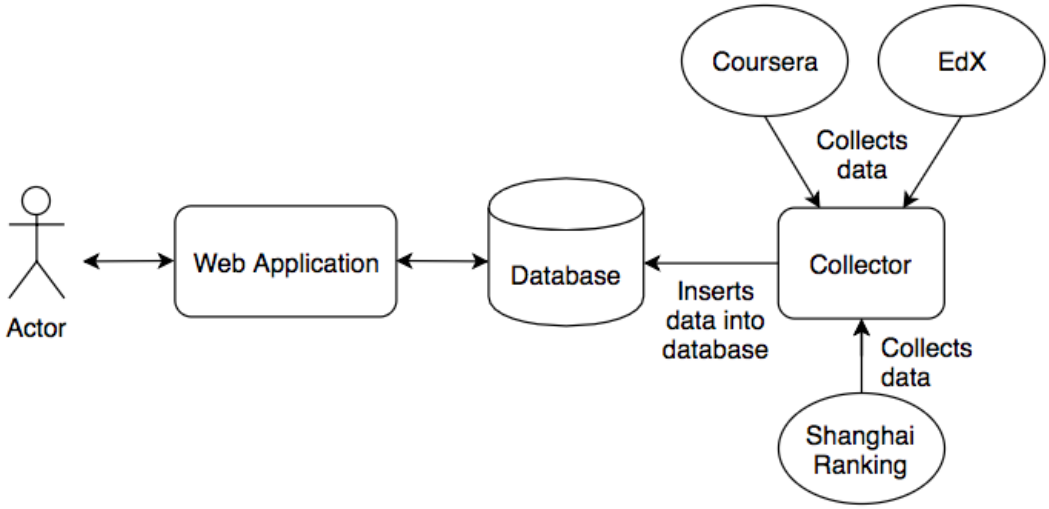



*Figure 5: System External Interactions diagram*

This web application connects to the server, which will go to the database in order to retrieve some information. This information was initially obtained by means of a program in python that collects the information from Coursera API and from EdX's RSS by parsing it. Nevertheless, Coursera deprecated the API, therefore a crawler was created to obtain the data.

## 4.2 Methodology

In this section, it will be described the methodology followed to design the system's architecture. It will be done using UML, which stands for Unified Modeling Language. UML is a graphic language that it is used for visualizing, designing and documenting a system. UML is the tool that will be used to describe the 4+1 model, which is an architectural model to describe a system based on four different views that will lead to describe the scenarios, which are the use cases. The views are logical, process, development and physical.

- **Logical:** The logical view will be describing the class diagrams and package structure. It represents the conceptual objects in the system.
- **Process:** The process view will represent the activities and workflow of the system. There it will be composed of sequence diagrams, showing the interactions inside the system.
- **Development:** The development view will represent the structure of the system. It will be described using a component diagram and then the components specifications.

- **Physical:** The physical view will specify the hardware of the system and how the components are mapped on the physical layer by using a deployment diagram.
- **Scenarios:** The scenarios refer to the use cases, which are the activities the end-user will be able to perform. In this system, on top of the end-user, there is also a data curator, which is an actor that will be in charge of updating the web application.

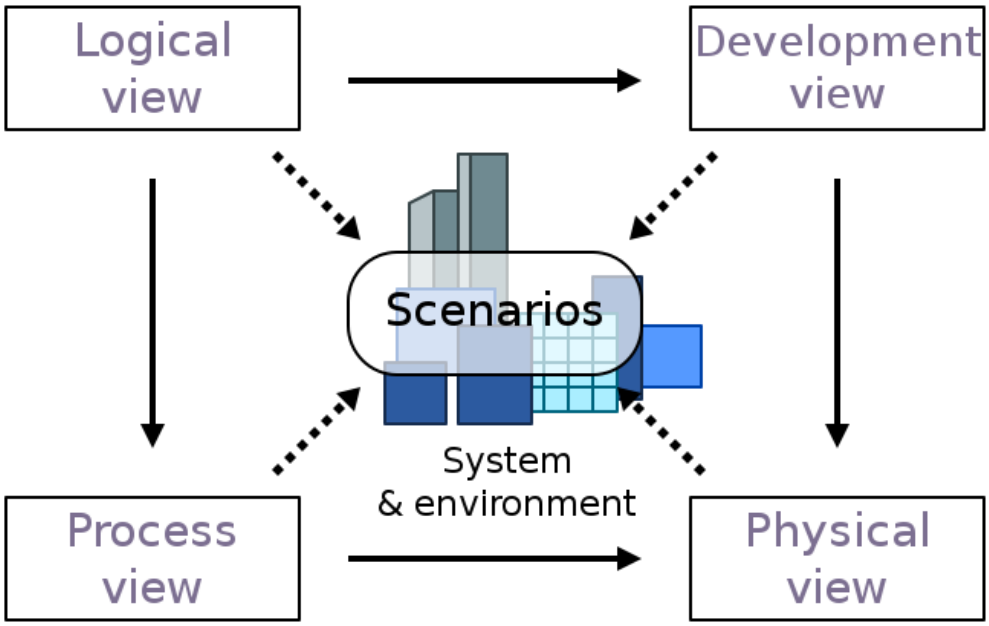



*Figure 6: 4+1 Architectural Model [7]*

## 4.3 Technological Environment

The architecture design for the application is similar to a MEAN Stack Architecture. What is a MEAN Stack Architecture? It is an open-source collection of JavaScript based technologies used to develop web applications. The MEAN acronym refers to MongoDB, Express, Angular.js and Node.js. MongoDB is a NoSQL database. Express is a web application framework for Node.js, which is designed for building web applications and APIs. Instead of using Angular.js, the application uses EJS, which is a simple templating language that generates HTML markup with plain JavaScript. Node.js is an environment that executes JavaScript code server-side. [8]

This diagram will explain better the architecture used for the system:

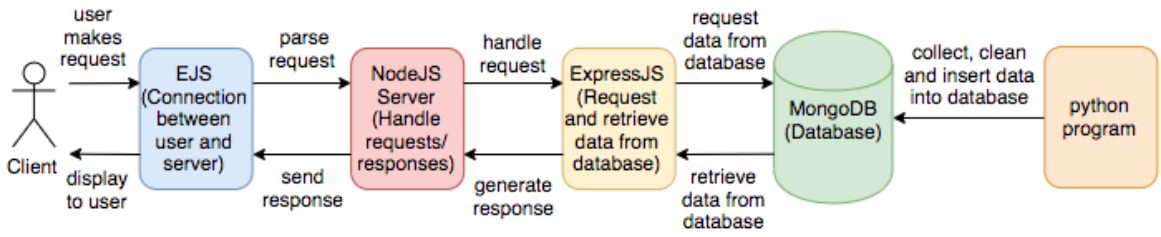



*Figure 7: Technological Environment diagram*

Essentially, this "MEAN" architecture will follow the Model-View-Controller (MVC) design. The model will describe the database design and is the responsible for maintaining the data, which in this case is handled by MongoDB. The view presents the model in a way in which the user will be see and interact with the data in the User Interface (UI). It is handled by EJS and the controller. The controller will establish and control the interactions between the model and the view, which in this case will be handled in NodeJS and ExpressJS.

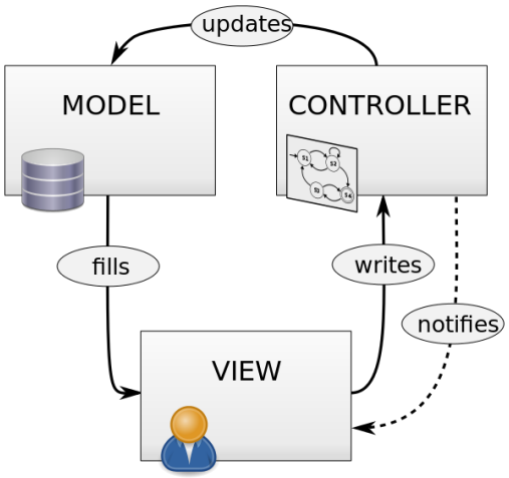



*Figure 8: Model-View-Controller diagram [9]*

## 4.4 Class diagram

The class diagram is the conceptual diagram to display the architecture of the system in terms of concepts or classes.

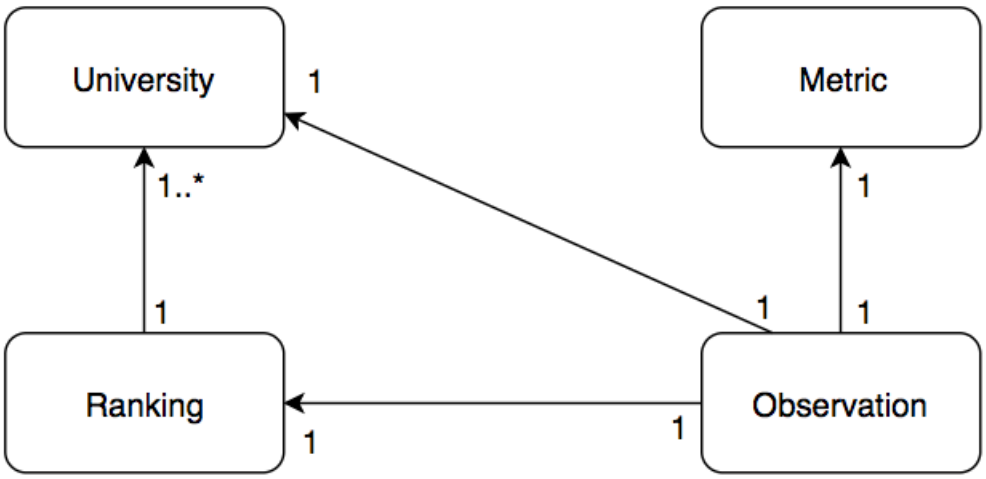



*Figure 18: Class diagram*

The previous defined concepts will be defined:

- **University:** This concept of University is defined by the university attributes, such as the name of the university, the size (number of students the university has), the number of MOOCs, the country and the continent of the university.
- **Ranking:** It is a concept that defines the position of a concept with respect to others, in this case the ranking is composed of a University and its position.
- **Metric:** Function that will result in a measurement or a set of measurements applying a certain formula in order to calculate these measurements.
- **Observation:** Constitutes a set of data composed by the information of the University combined with its ranking position and having applied the different metrics to it, in order to calculate some new ranking positions according to these several metrics.

As it is shown in the previous diagram, the observation should have a unique university and its position of the ranking. However, one Observation, which means one University and its ranking can have several metrics to measure the new ranking position.

## 4.5 Package Structure

The package structure will define the structure of the code, according to the following model. The structure is based on the Model-View-Controller (MVC) software architecture design, explained previously. The sections design will be similar to the one explained in the overview but slightly simplified, and the component diagram will follow it. First, the packages will be divided according to Graphical User Interface, Services and Database.

Inside the Graphical User Interfaces, there will be two different packages: views and public. The views will constitute the website pages - ejs - and the public, which will have all the files related to the website interface, for instance images, stylesheets and javascript files.

In the Services section, there will be just one package named app. Inside app, there will be three sub-packages: controllers, routes and models. The controllers will process the requests, retrieve the information from the database and send it to the routes. The routes will route the requests from the views to the controller and vice versa. The models will create the Schema structure inside the app for the database.

For the database, it will be just the config, in which it is made the connection to the MongoDB database.

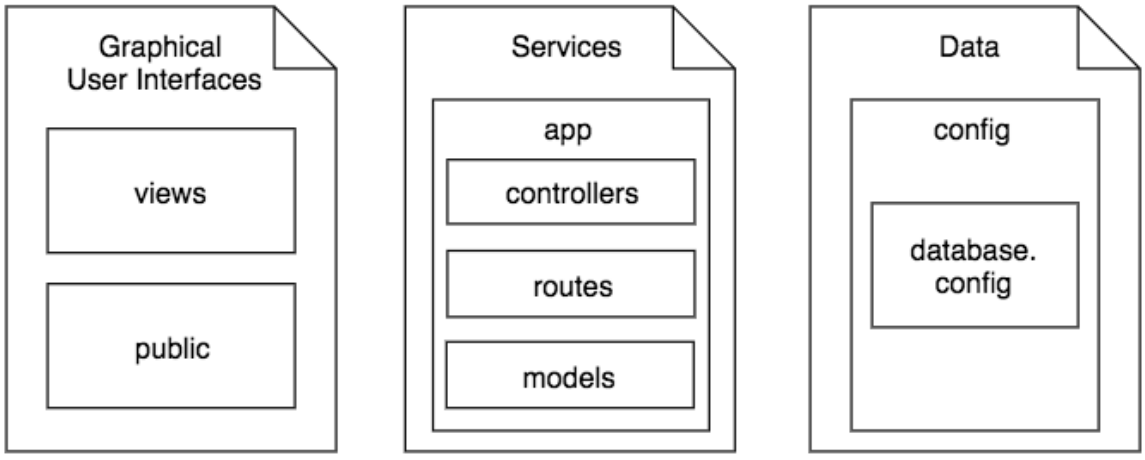



*Figure 19: Package structure*

## 4.6 Sequence diagrams for user

The sequence diagrams depict the process view and they will show the functions the system will perform in a time sequence when the user executes an activity. Each activity is linked with a Use Case; thus, the diagrams will be a time representation of the use cases.

### 4.6.1 Sequence diagram for seeing the universities ranking

In this diagram, the user will request the universities, then the route will ask to find all universities to the controller, which will retrieve the data from the model. Finally, the routes will render the results in the view and be displayed to the user.

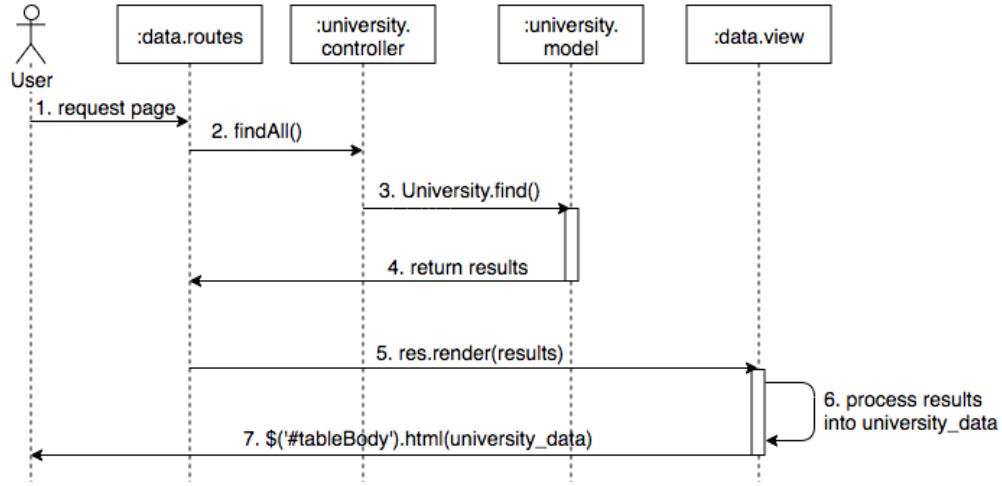



*Figure 9: Sequence diagram for seeing a university ranking*

### 4.6.2 Sequence diagram for filtering the table by continent

The user will filter by continent, so a request is sent to the routes. The controller will then retrieve data by continent from the model and the routes will render the results in the view.

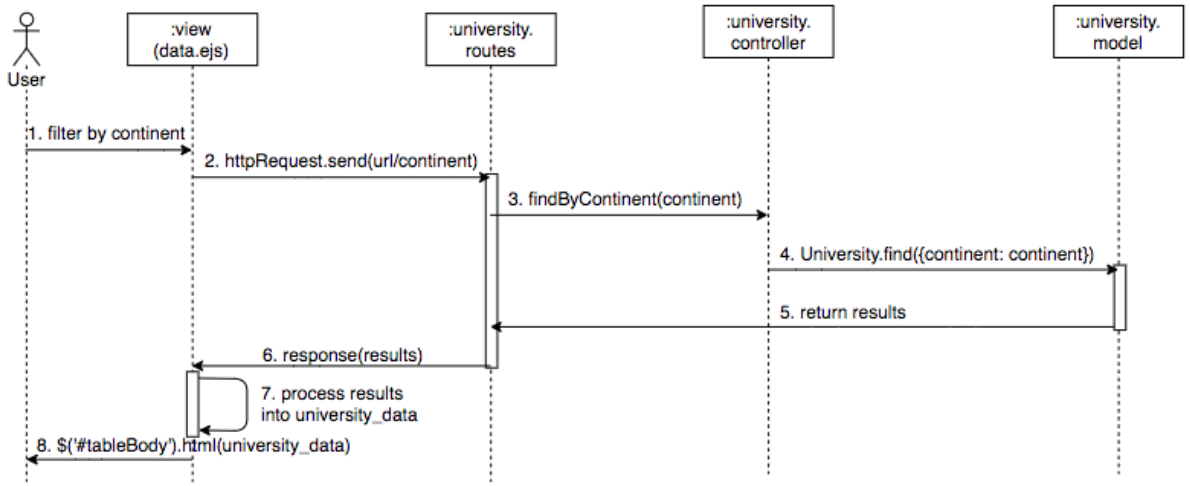



*Figure 10: Sequence diagram for filtering the table by continent*

### 4.6.3 Sequence diagram for filtering the table by country

The user will filter by country, so a request is sent to the routes to filter by country. The routes use the function findByCountry and the controller will then retrieve data by country from the model. Finally, the routes will render the results in the view.

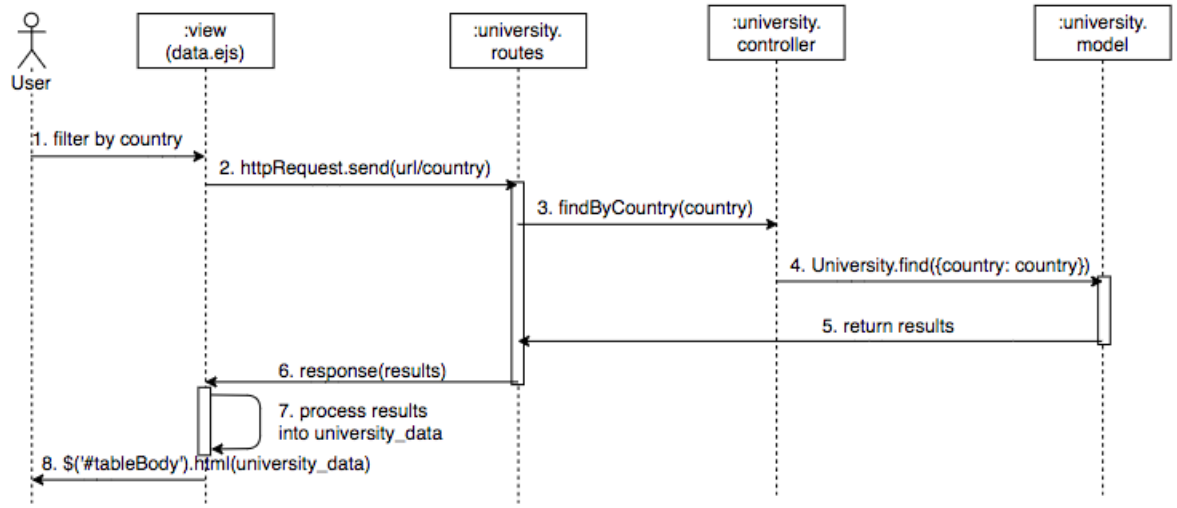



*Figure 11: Sequence diagram for filtering the table by country*

### 4.6.4 Sequence diagram for filtering the table by ranking

The user will filter by country, which is done in the view. To filter by ranking, first it will be checked the results for country and continent if they were filtered. Then, it is filter by the ranking position values and update the table.

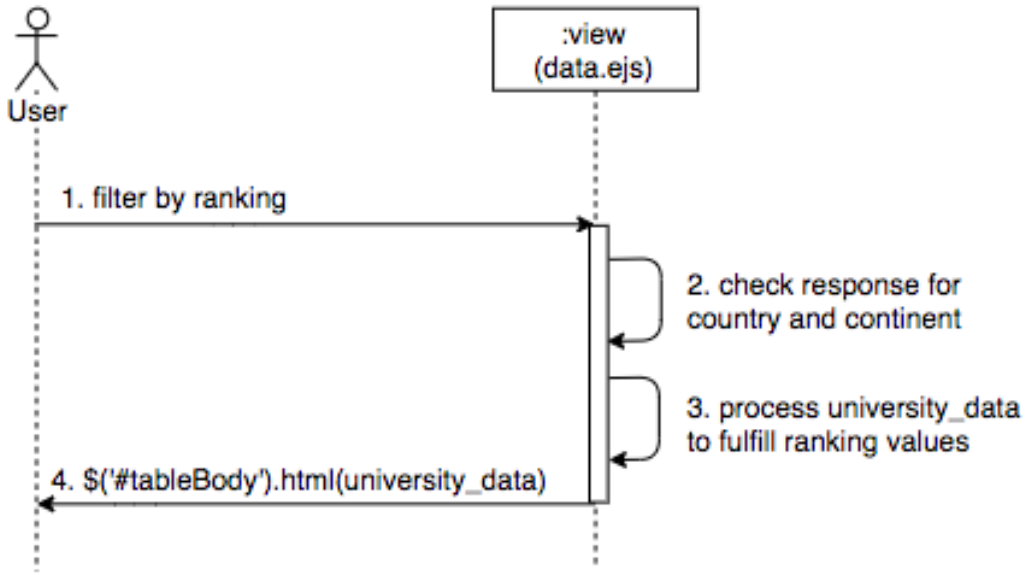



*Figure 12: Sequence diagram for filtering the table by ranking*

### 4.6.5 Sequence diagram for comparing two universities

In order to compare two universities, the user first requests the graphs page, which works the same as in sequence diagram 4.6.1 and it will retrieve all the data from the database. Then, the user selects two universities to compare and in the view, it will be filtered the retrieved data and displayed to the user.

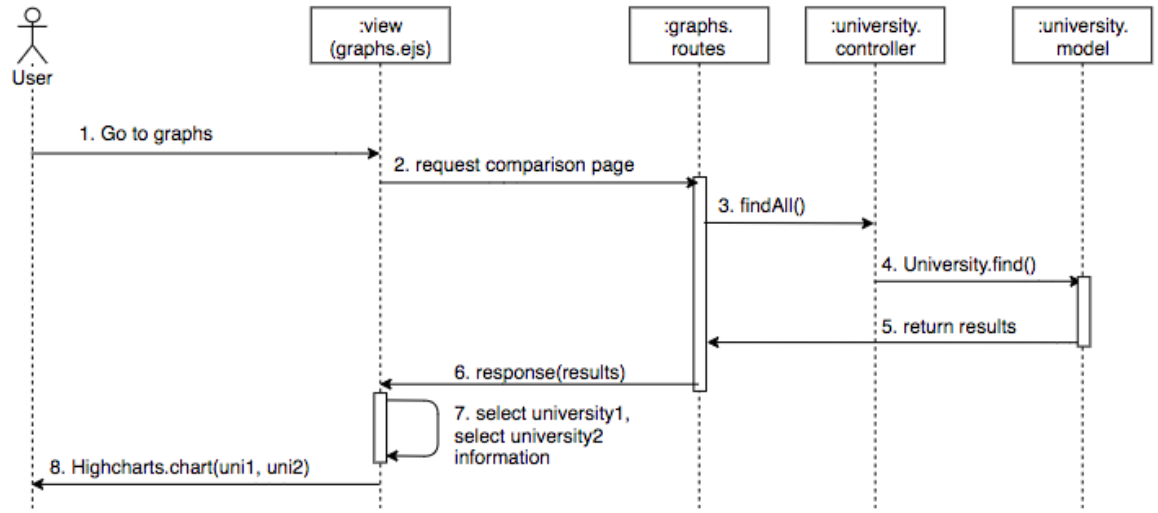



*Figure 13: Sequence diagram for comparing two universities*

### 4.6.6 Sequence diagram for seeing and interacting with world map

In this diagram, the user goes to maps section in graphs. The data will be retrieved as in the previous diagram and then the view will filter the data into the map to display the results in the map to the user.

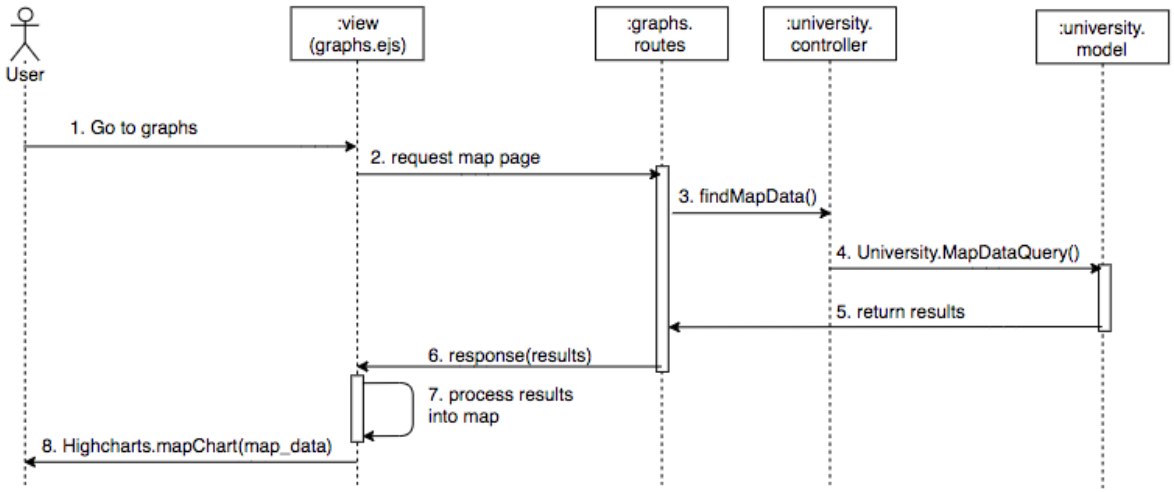



*Figure 14: Sequence diagram for seeing and interacting with world map*

### 4.6.7 Sequence diagram for seeing and interacting with bar graphs

In this diagram, the user goes to bar chart section in graphs. The data will be retrieved as in the previous diagrams and then the view will filter the data into the bar chart to display the results to the user.

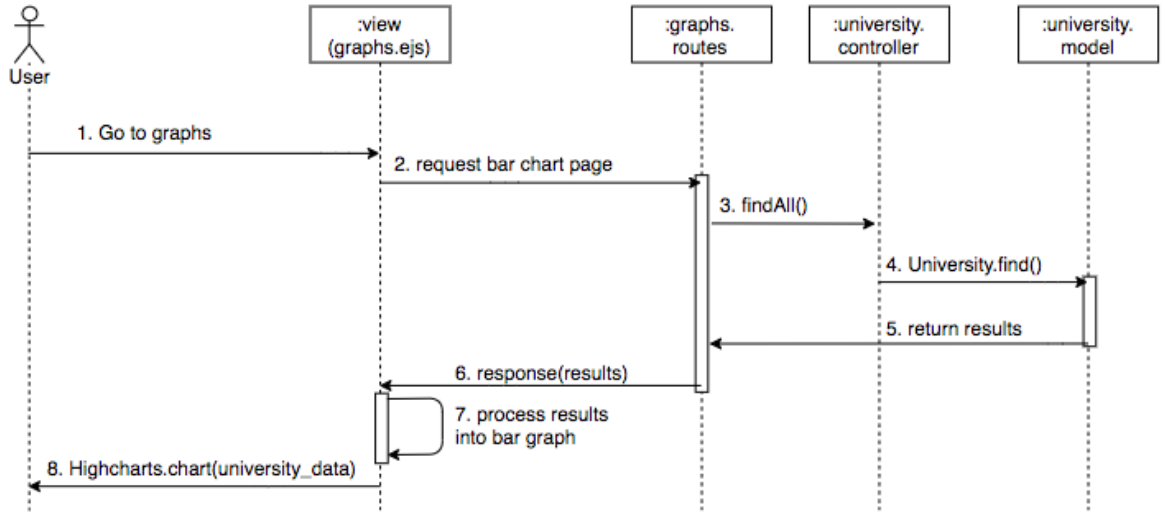



*Figure 15: Sequence diagram for seeing and interacting with bar graphs*

### 4.6.8 Sequence diagram for seeing and interacting with linear graphs

In this diagram, the user goes to linear graphs section. The data will be retrieved as in the previous diagrams and then the view will filter the data into the linear graphs to display the results to the user.

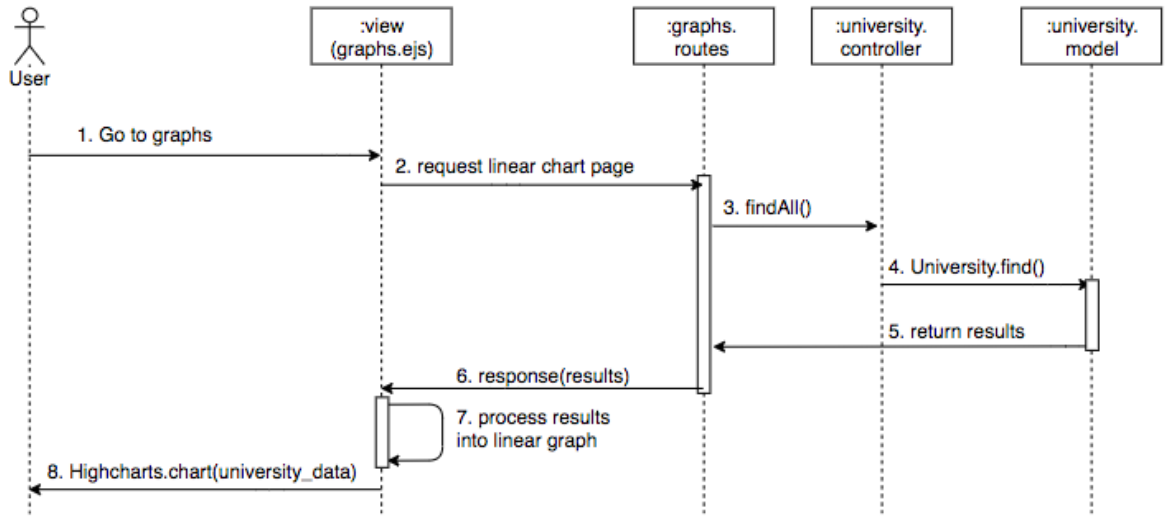



*Figure 16: Sequence diagram for seeing and interacting with linear graphs*

### 4.6.9 Sequence diagram for printing and/or export graphs

In this diagram, the user will click to print or export a graph. The data needs to be retrieved and inserted into the graphs as was represented in the previous diagrams. Then, the view will use the Highcharts script to print or export the graph.

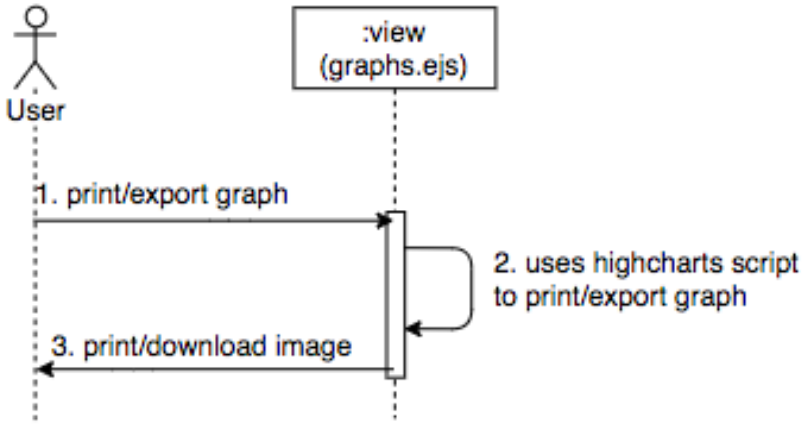



*Figure 17: Sequence diagram for printing and/or exporting graphs*

## 4.7 Component diagram

| ID | Component Name |
|---|---|
| C-1 | User Interface |
| C-2 | MOOCs Ranking Server |
| C-3 | Universities Management |
| C-4 | Filtering Engine |
| C-5 | MongoDB |

*Table 22: Components table*

The component diagram is part of the development view. It will display the different components of the system and how they relate to each other in terms of dependencies and interfaces. This diagram is used to model the organization of the system.

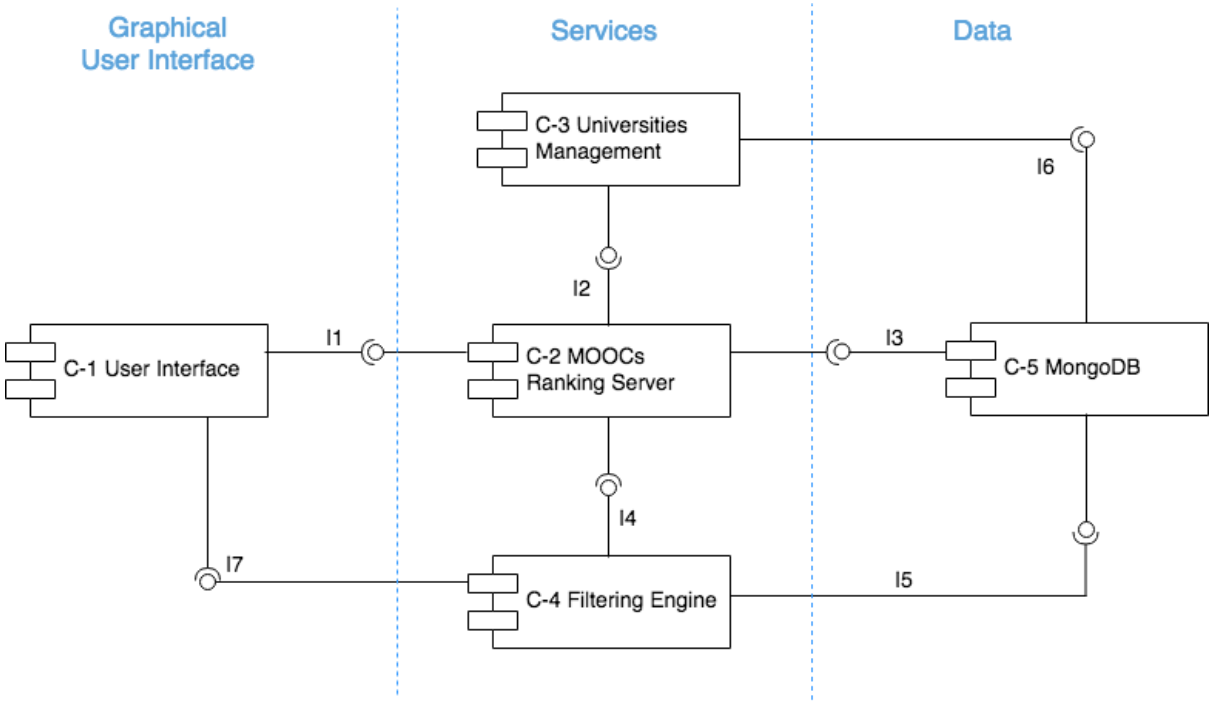



*Figure 21: Component diagram*

## 4.8 Component description

In this section, it is going to be described the components of the system. There will be subsections dealing with the aspects of which component, being those the identifier, type, purpose, function, subordinates, dependencies, interfaces, resources, processing and data.

### C-1 User Interface

| | | | |
|---|---|---|---|
| Identifier | C-1 User Interface | Type | Subsystem |
| Function | The main function of the User Interface is to interact with the user. It is the component that provides the communication within the server and the person using the web application. | | |
| Subordinates | | | |
| Dependencies | C-2 MOOCs Ranking Server, C-4 Filtering Engine | | |
| Interfaces | | | |
| Resources | Graphical Display | | |
| Processing | User sends request and display responses | | |
| Data | Input | User input | |
| | Output | Screen Display, Api calls | |
| Requirements | FR-07, FR-08, FR-09, FR-11 , FR-12, NFR-02, NFR-03, NFR-06 | | |

*Table 23: Component C-1 table*

### C-2 MOOCs Ranking Server

| | | | |
|---|---|---|---|
| Identifier | C-2 MOOCs Ranking Server | Type | Server |
| Function | | | |
| Subordinates | C-1 User Interface | | |
| Dependencies | C-1 User Interface, C-5 MongoDB Database, C-3 Universities Management, C-4 Filtering Engine | | |
| Interfaces | I1 | | |

| | | |
|---|---|---|
| Resources | Processing power | |
| Processing | Handle requests and create responses | |
| Data | Input | User request |
| | Output | Response according to request |
| Requirements | FR-06, NFR-01, NFR-04, NFR-05 | |

*Table 24: Component C-2 table*

### C-3 Universities Management

| | | | |
|---|---|---|---|
| Identifier | C-5 Universities Management | Type | Management |
| Function | Will handle everything related to universities, including creation and deletion. | | |
| Subordinates | C-2 MOOCs Ranking Server | | |
| Dependencies | C-5 MongoDB Database | | |
| Interfaces | I2 | | |
| Resources | Processor power, Model to create universities schema | | |
| Processing | Creates universities schema | | |
| Data | Input | | |
| | Output | Creates model Schema | |
| Requirements | FR-02, FR-03, FR-04, NFR-05 | | |

*Table 25: Component C-3 table*

### C-4 Filtering Engine

| | | | |
|---|---|---|---|
| Identifier | C-4 Filtering Engine | Type | Search |
| Function | Component that searches in the database according to user's request. | | |

| | | |
|---|---|---|
| Subordinates | C-1 User Interface, C-2 MOOCs Ranking Server | |
| Dependencies | C-5 MongoDB Database | |
| Interfaces | I4, I7 | |
| Resources | Processor power | |
| Processing | Search through the database | |
| Data | Input | User request for filtering |
| | Output | The information related to the request. |
| Requirements | FR-10 | |

*Table 26: Component C-4 table*

### C-5 MongoDB Database

| | | | |
|---|---|---|---|
| Identifier | C-5 MongoDB | Type | Database |
| Function | Store all data about universities. | | |
| Subordinates | C-2 MOOCs Ranking Server, C-3 Universities Management, C-4 Filtering Engine | | |
| Dependencies | C-2 MOOCs Ranking Server | | |
| Interfaces | I3, I5, I6 | | |
| Resources | Data storage and management. | | |
| Processing | Retrieve the data from the database according to requests. | | |
| Data | Input | Request data from database. | |
| | Output | The array object of the components that need to be in the response. | |
| Requirements | FR-01, FR-05 | | |

*Table 27: Component C-5 table*

## 4.9 Deployment diagram

The deployment diagram is the representation of the physical view. It will describe the hardware that it is needed in the system and how the components are divided into it. The illustration will be done using Nodes.

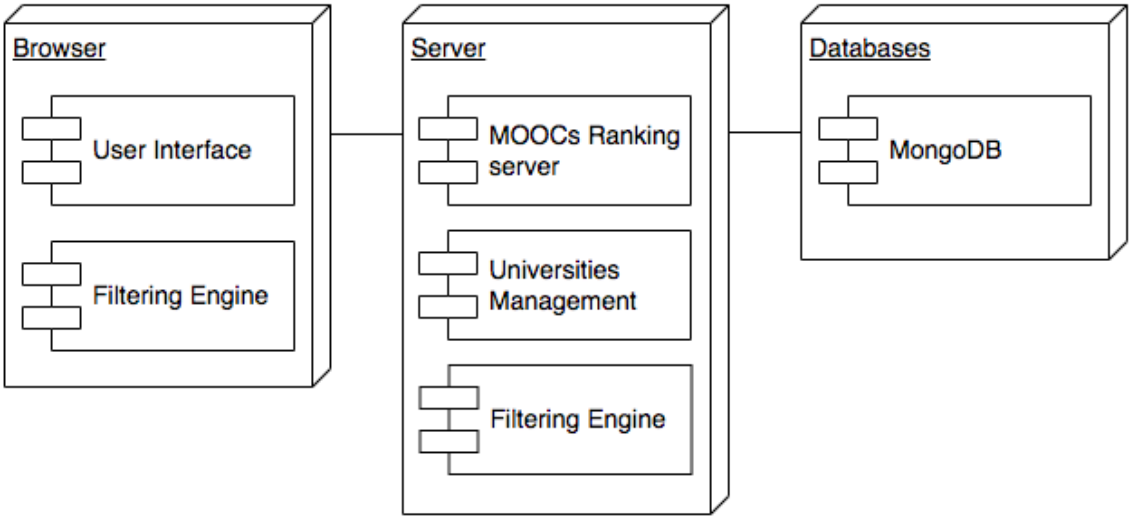



*Figure 20: Deployment diagram*

This deployment diagram will show how the architecture of the physical components are distributed. First, each user will use the browser in order to display the User Interface. When the user interacts with the web application, this will go to the Server, in which there are several components as it is the MOOCs Ranking Server, Universities Management and Filtering Engine. If it is needed to retrieve information it will go to the database.

## 4.10 Database Model Design

In this section will be described the database model design. First, there will be some tables representing each table in the database, or each collection when referring to MongoDB. The tables will specify the parameters name, type and a description of it. It will also specify the primary keys and the foreign keys. Finally, there will be a relational model in which it will be presented the relationships between the different collections.

### 4.10.1 University Model Schema

| Parameter Name | Parameter Type | Description |
|---|---|---|
| universityName | String | The name of the university. [PRIMARY KEY] |
| country | String | The country of the university. |
| countryCode | String | The country code of the university with 2 characters. |
| continent | String | The continent of the university. |
| universitySize | Number (Integer) | The number of students enrolled in the university. |
| universitySizeValue | Number (Integer) | A simplified calculation of the universitySize parameter that will take the following values according to the university size:<br>1 if universitySize < 5000<br>2 if universitySize 5001-15000<br>3 if universitySize 15001-25000<br>4 if universitySize 25001-35000<br>5 if universitySize > 35000 |
| courseraMOOCS | Number (Integer) | The number of Coursera MOOCs of the university. |
| edXMOOCS | Number (Integer) | The number of EdX MOOCs of the university. |
| totalMOOCS | Number (Integer) | The number of total MOOCs published by the university. |

*Table 28: Database University Model Schema*

### 4.10.2 Ranking Model Schema

| Parameter Name | Parameter Type | Description |
|---|---|---|
| rankingID | Number (Integer) | The number to identify the ranking in the ranking table. [PRIMARY KEY] |
| rankingName | String | The ranking name. In this case, there are 4: ShanghaiRanking, Metric2Ranking, Metric9Ranking and Metric10Ranking. |
| url | String | The url of the ranking page. |

*Table 29: Database Ranking Model Schema*

### 4.10.3 Metric Model Schema

| Parameter Name | Parameter Type | Description |
|---|---|---|
| metricID | Number (Integer) | The number to identify the metric in the metric table. [PRIMARY KEY] |
| metricName | String | The metric name. In this case, there are 3: Metric2, Metric9 and Metric10. |

*Table 30: Database Metric Model Schema*

### 4.10.4 Observation Model Schema

| Parameter Name | Parameter Type | Description |
|---|---|---|
| universityName | String | The university collection, united by the universityName. [PRIMARY KEY and FOREIGN KEY] |
| type | Number (Integer) | 0: Ranking Type<br>1: Metric Type<br>[PRIMARY KEY] |
| typeID | Number (Integer) | The ID for the Ranking or the Metric, according to the type parameter. [PRIMARY KEY and FOREIGN KEY] |
| year | Date (year) | The year the university was classified in that specific position according to that rankingID. [PRIMARY KEY] |
| value | Number (Float) | Value according the metric or ranking. |

*Table 31: Database Observation Model Schema*

### 4.10.5 Relational Database Model

The relationship between these 4 different model schemas is defined with the following diagram. This diagram will identify the collections name, parameters, the primary keys and the foreign keys, representing the relationships.

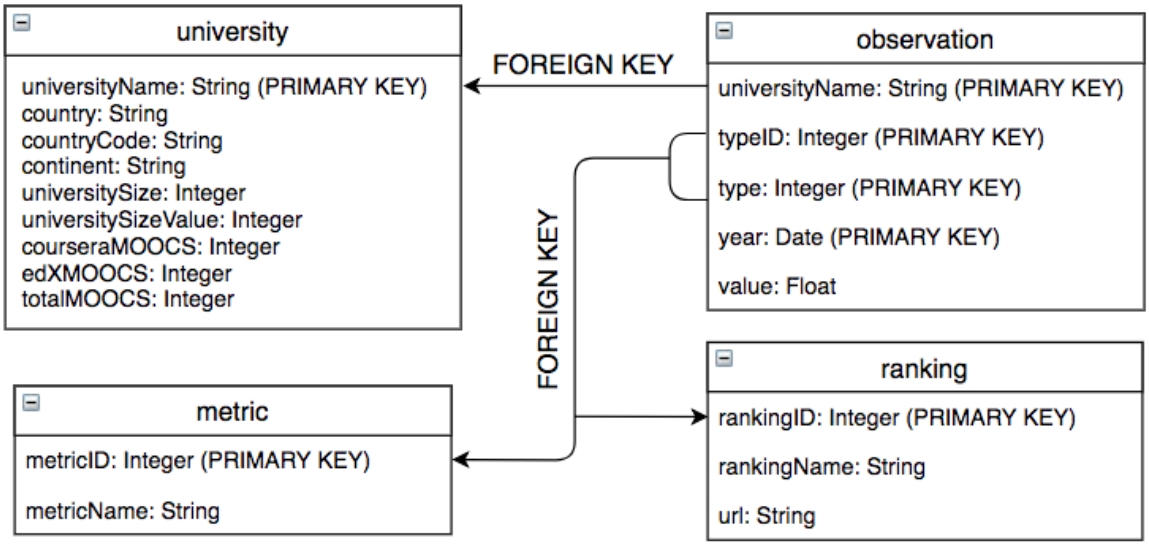



*Figure 22: Relational Database Model*

## 4.11 User Interface Mockup

This is the mockup page for the Home Section. When the user enters the webpage, this will be the first page appearing. Here there will be some general information about the project and the links to the sections of Data and Graphs.

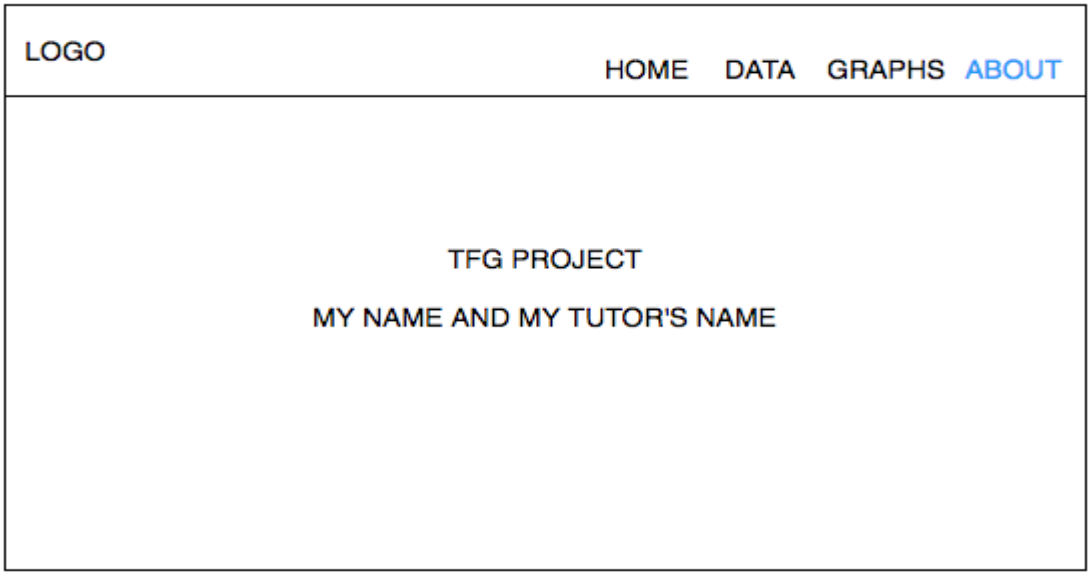



*Figure 23: Mockup page for Home*

This is the mockup page for Data. Here, there will be a dynamic responsive table with the university ranking. There is a sidebar displaying the filtering for the ranking table, regarding continent, country and ranking position.

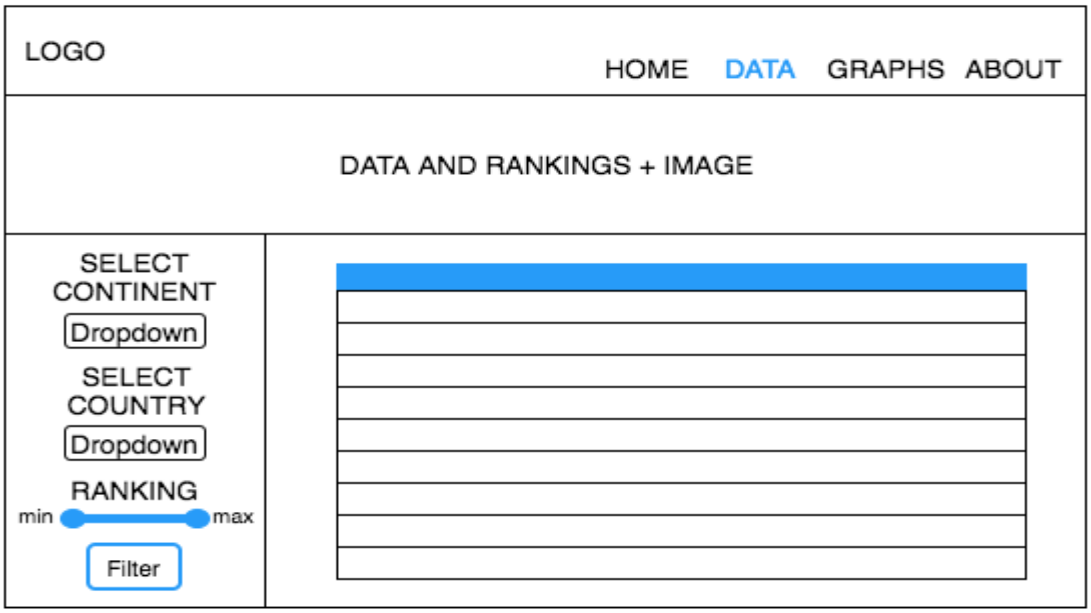



*Figure 24: Mockup page for Data*

This is the mockup page for graphs, in which there are 4 new sections - maps, linear graph, bar chart and comparison. There is also a filtering sidebar. The main page displayed will be the world map with the number of universities per country.

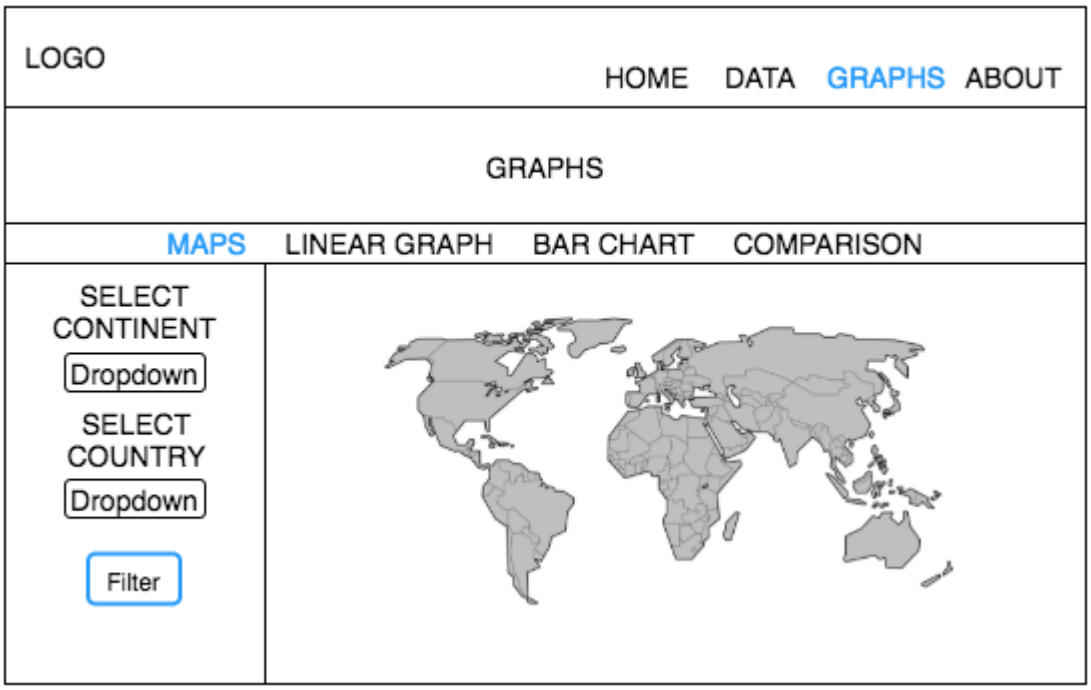



*Figure 25: Mockup page for Maps (Graphs)*

This is the mockup page for the section of linear graphs inside graphs.

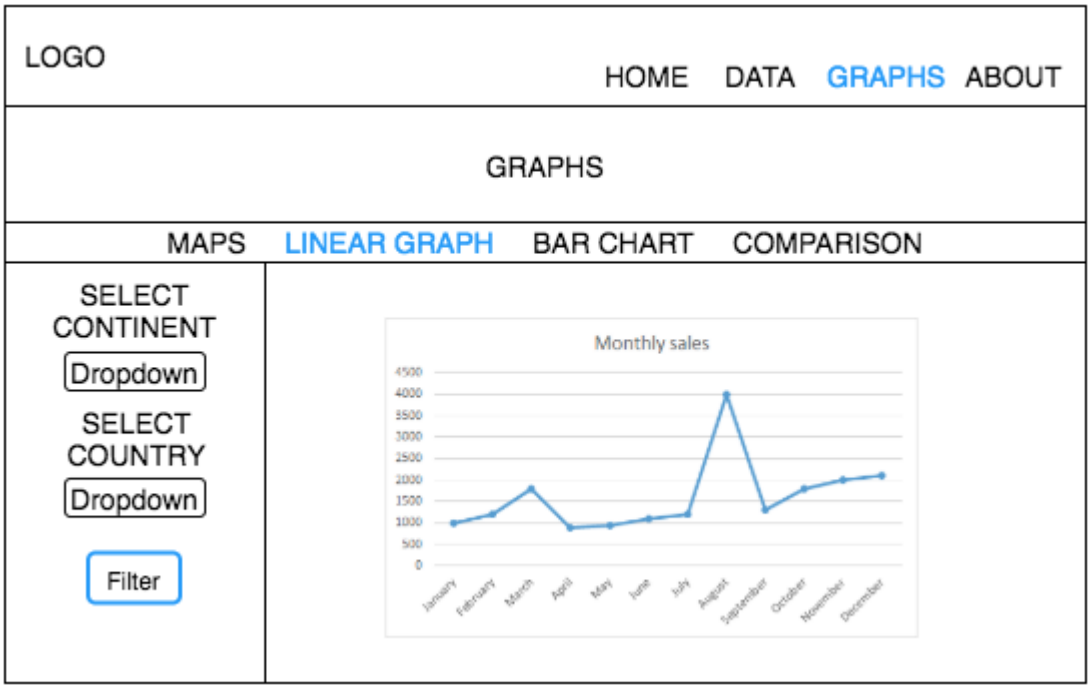



*Figure 26: Mockup page for Linear Graphs (Graphs)*

This is the mockup page for the section of bar chart inside graphs.

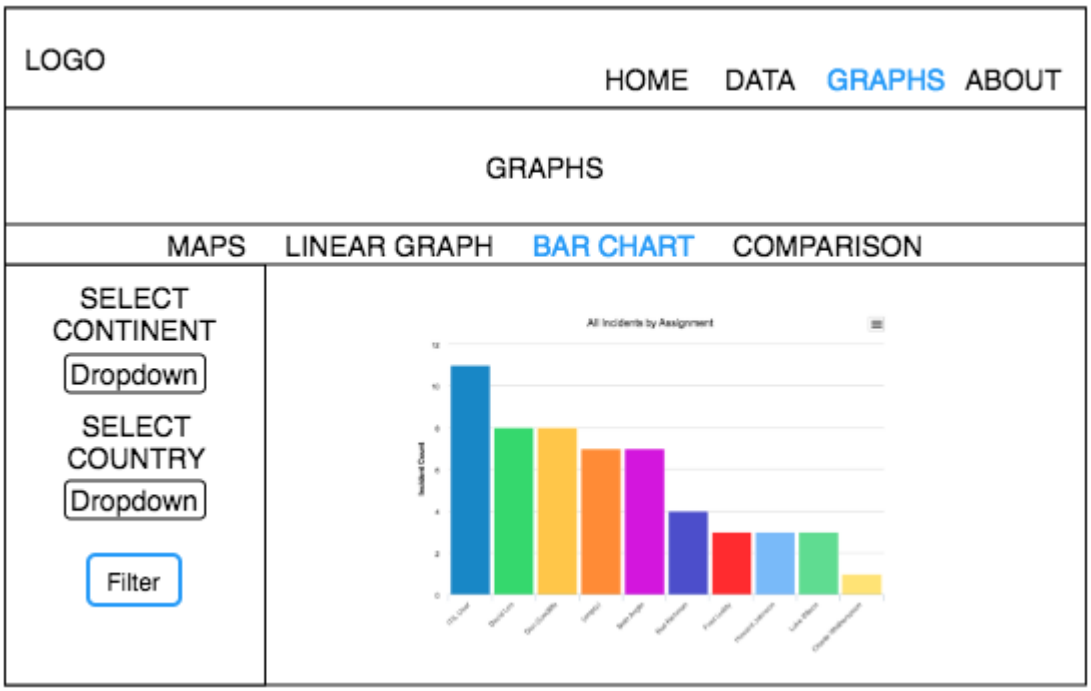



*Figure 27: Mockup page for Bar Charts (Graphs)*

This is the mockup page for the section of comparison inside graphs. There will be 2 dropdowns for selecting which universities to compare. Then, when clicking in compare some graphs comparing both universities will be displayed.

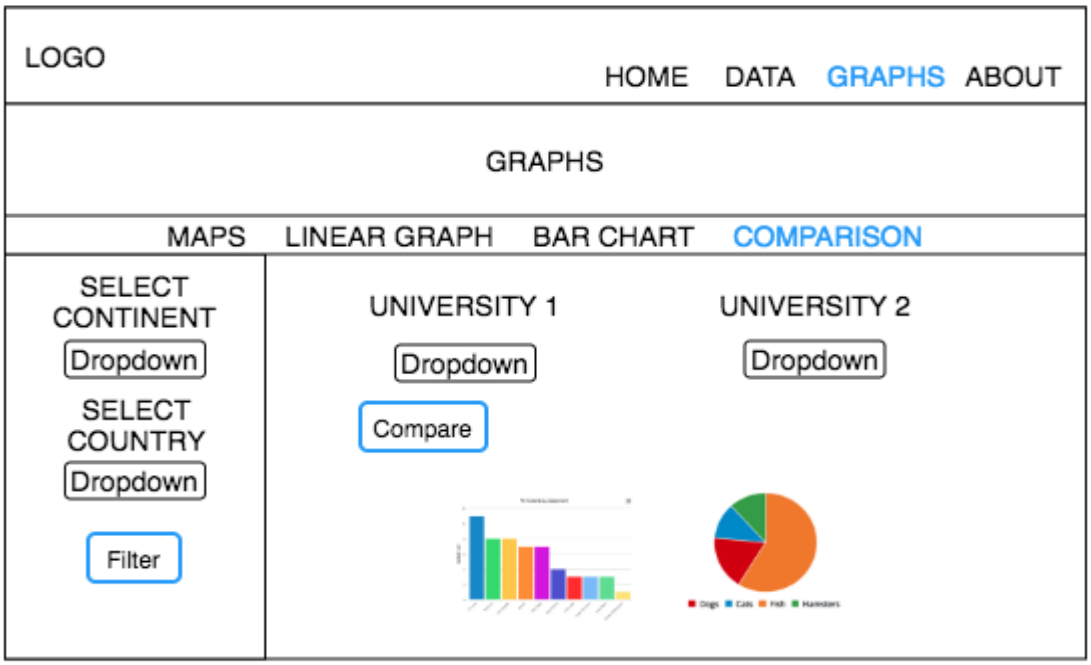



*Figure 28: Mockup page for Comparison(Graphs)*

This is the mockup page for the About Section. When the user clicks in about, there will be some information saying that this project is a TFG done by Leire Litwin Echevarría and supervised and tutored by José María Álvarez Rodríguez.

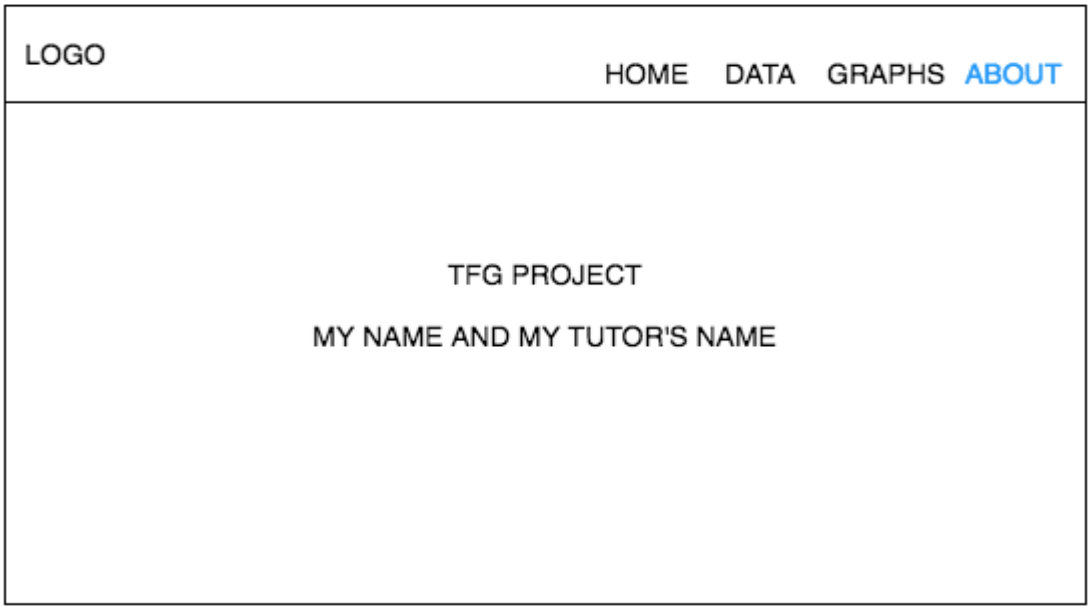



*Figure 29: Mockup page for About*

# 5. Implementation and Testing

## 5.1 Overview of Implementation

For the implementation of the project, it can be divided in four different stages - obtaining and computing the data, creating and introducing the data in the database, creating the app server, displaying it in a user interface (UI) and finally deploying the web application.

### 5.1.1 Methodology and Resources

In the implementation phase of the project, I used a SCRUM methodology. Scrum is a reference model that defines a set of practices and roles that is defined by the agile development frameworks for software development. SCRUM is based on dividing roles. In this particular case, my tutor, José María Álvarez Rodríguez, was the Product Owner and the Client. On the other hand, I had the role of full-stack developer, doing the research, analysis, design, implementation and testing. Moreover, it has been applied the Scrum concept of having an incremental development strategy and overlapping of different phases of development, instead of a sequential development. The Sprint defined in the SCRUM methodology is the period of time for the project. The sprints duration shall be constant; therefore, the milestones were defined every two weeks. Some of the advantages SCRUM provides are improvement of productivity, changes flexibility, better software quality and risk reduction.

One of the external resources that I have used while developing the application is Github. Github is an open platform to develop and host collaborative projects using Git version control system. Even though this project was individual, it has been used Github for several reasons. First, to keep track of the versions and commits in the development process. It is also a way of having a backup copy stored online and, if anyone would like to see the implementation process, it can be accessed online.

The IDE that it was used was IntelliJ IDEA since it supports Javascript, the main language used in the web application. It is developed by JetBrains, known also as IntelliJ. It provides features like debugging, autocomplete and refactoring.

### 5.1.2 Data collection and computation

For the first stage, obtaining and computing the data, I will use a program in python that downloads the data from Coursera and EdX. The data from Coursera was being taken from the API, but it is now deprecated. Now, the data from Coursera will be taken from a crawler, whereas EdX will use the RSS. These generate csv files. From there, the data shall be cleaned manually, using the same name convention for the universities and counting the number of MOOCs they offer. Once this data is collected, then the Metrics need to be calculated that will be used to compute new rankings.

In the computation of the new rankings, there will be three different metrics - Metric 2, 9 and 10 [10]. A Metric is a function that computes some measurements that in this case will be the indicators for the rankings. Metric 2 and 9 will take into account the total number of MOOCs divided by the university size, in other words, the number of enrolled students. Metric 2 uses the exponential function, whereas Metric 9 uses the inverse exponential. Therefore, these two metrics will measure the innovation by counting the total number of MOOCs with respect to the University Size, measured with the University Size Value. The results will be better when the size is smaller and the number of MOOCs is bigger. The value of the University Size (students number) is measured using the following criteria:

UniversitySizeValue = 1 if universitySize < 5000
UniversitySizeValue = 2 if universitySize 5001-15000
UniversitySizeValue = 3 if universitySize 15001-25000
UniversitySizeValue = 4 if universitySize 25001-35000
UniversitySizeValue = 5 if universitySize > 35000

The formulas for the Metrics 2 and 9 are the following ones:

$$\text{Metric 2} = \frac{totalMOOCs\ of\ University}{EXP(universitySizeValue)}$$

$$\text{Metric 9} = \frac{totalMOOCs\ of\ University}{EXP(-Metric\ 2)}$$

On the other hand, Metric 10 will take into account the number of total MOOCs with respect to the exponential of the Shanghai Ranking position (ARWU). In this way, it will take into account the parameters as quality of education and faculty, research output and per capita

performance. However, it will also use the number of MOOCs as an indicator for the university innovation performance. The formula for the Metric 10 is the following one:

$$\text{Metric 10} = \frac{totalMOOCs\ of\ University}{EXP(-Shanghai\ Ranking/500)}$$

The three Metrics will have the shape of a power law distribution. In this case, the number of MOOCs will be the y axis, thus the more the better, whereas in the x axis it will be the university size. This directly favors the universities with lower number of students.

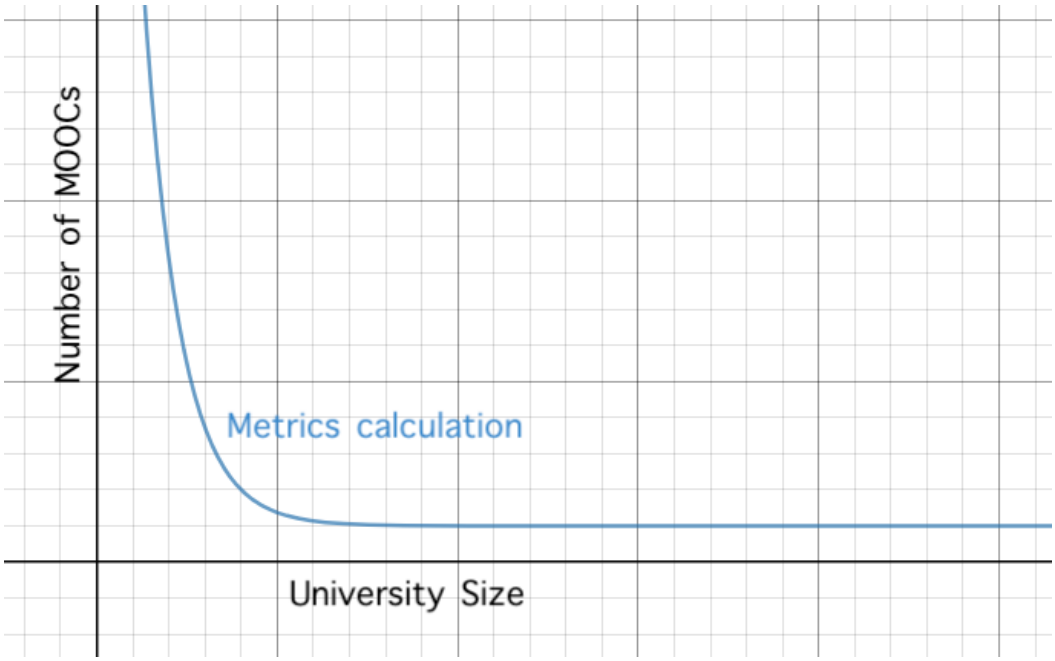



*Figure 30: Power Law Distribution for universities*

Then, once all this metrics and rankings are calculated, this data needs to be processed before inserting it into a database because there are some issues with duplicates and name conventions.

### 5.1.3 Database schema and insertion

Then, in the second stage, introducing the data into the database, I will use MongoDB. MongoDB is a free NoSQL program used for databases. Instead, MongoDB uses schemas that can be defined by means of a JSON or a CSV file. First, I created locally a database with my own user and password. There, I created a collection (table in MySQL) named "university" that will contain all the information regarding universities. Then, I will have

another collection for the “ranking”, another collection for the “rankingPosition”, another one for the “metric” and finally one for the “observation”, which will contain all the information merged together.

All these collections will be defined by a Mongoose Schema and then it will be exported by the module. This is an example of the “university” collection:

```
var mongoose = require('mongoose');

var UniversitySchema = mongoose.Schema({
    universityName: String,
    country: String,
    countryCode: String,
    continent: String,
    universitySize: { type: Number, min: 0 },
    universitySizeValue: { type: Number, min: 0, max: 5 },
    ranking: { type: Number, min: 1 },
    shangaiRanking: { type: Number, min: 1 },
    courseraMOOCS: { type: Number, min: 0 },
    edXMOOCS: { type: Number, min: 0 },
    totalMOOCS: { type: Number, min: 0 }
},{
    _id: false
});

module.exports = mongoose.model('university', UniversitySchema);
```

*Figure 31: Mongoose Schema for “university” collection*

In order to insert information, after doing all the computations, the generated excel (xlsx) file is exported into a csv file, which will be inserted into the database. In order to do so, first MongoDB needs to be running in the terminal, then the collections are inserted into the database. At first, this step was made locally, although after deploying the web application in Heroku, it needs to be imported it to the MongoDB database that Heroku is using as resources. The command to use can be found in the Admin Manual.

### 5.1.4 App Server creation

Then, I connected it with a server by using Express and Node.js. Express is a web application framework for Node.js, which is designed for building web applications and APIs. Node.js is a cross-platform run-time environment that executes JavaScript code server-side. I am using the version 5.6.0 of Express and the version 9.2.1 of Node.js.

After installing these two tools, the next step is to create the model of the database using Mongoose Schema, the university routes creating the end-points and the controller that will contain the functions of these end-points. First, it needs to be created some routes that should be setted up in the server.js file. This file will be the one creating the web application, requiring and using the routes, configuring the connection to the database and listening to the port number or the environment variable in case of the deployment in Heroku 9.11.1. Once the routes are configured, these will be used to render the views in the application, except for the university routes, which will connect the views and the controller.

The university routes contain the main REST API functions. Thus, there will be functions for getting, posting, putting and deleting information in the database. Even though, right now only the functions for GET and POST are being used, it might be useful in the future to have DELETE and PUT, so the end-points were created. To retrieve the data, the GET route will be used so that the call is routed to the find() function of the controller, which used to search into MongoDB.

Regarding the process for connecting the views with the web server, it will be done through the EJS and the routes, where the information will be rendered in the EJS page. In other cases, the user will use the view displayed using the EJS file to make calls to the back-end of the application. Then, this call will be routed and processed through a function in the controller that will retrieve the data from the database and then generate a response for the user. There are two figures for exemplifying of how the routes work.

GET:

```
router.get('/', function(req, res, next) {
  universities.findAll().then( function(dataRetrieved){
    //console.log(dataRetrieved);
    res.render('data', { title: 'MOOCs Ranking | Leire Litwin',
    data: dataRetrieved
    });
  }, function (reason) {
    console.log("argh"+reason);
  });
});
```

*Figure 32: Router get function*

POST:

```
// Retrieve all universities by country
app.post('/universities/countries/:country', function(req, res, next) {
    universities.findByCountry(req.params.country).then( function(dataRetrieved){
        res.status(200).json(dataRetrieved);
    });
});
```

*Figure 33: Controller post function for finding country*

### 5.1.5 User Interface

Regarding the User Interface used, everything is implemented using HTML 5, CSS version 4, EJS, Javascript and JQuery. For the homepage, I used a technique called parallax scrolling in order to create a sense of dynamism and so that the website was not so static. In the data section, there will be tables displaying the top-ranking universities. These tables are responsive and dynamic, so when changing the filter parameters, the table will get updated immediately.

This is an example of the Javascript code to update the table according to the filtering the user has requested and the response of the server. First, the user will be make a request. If the status is 200 (OK), which means the request has been processed correctly, then the response will be parsed. From there, the data of the response will be looped and it will be filtered by ranking before updating it into the table.

```
function updateTable(httpRequest) {
    if (httpRequest.readyState == XMLHttpRequest.DONE) {
        if (httpRequest.status == 200) {
            //console.log("reply received");
            //alert(httpRequest.responseText);
            var response = JSON.parse(httpRequest.responseText);
            //console.log(response);
            var university_data = '';
            for(var i=0; i<response.length; i++) {
                if(response[i].ranking >= parseInt($( '#slider-range ').slider("values")[0])){
                    if(response[i].ranking <= parseInt($('#slider-range').slider("values")[1])){
                        university_data +='<tr>';
                        university_data += '<td>'+ response[i].universityName +'</td>';
                        university_data += '<td style="text-align:center;">'+ response[i].ranking + '</td>';
                        university_data += '<td style="text-align:center;">'+ response[i].shangaiRanking +'</td>';
                        university_data += '<td>'+ response[i].country + '</td>';
                        university_data += '<td style="text-align:center;">'+ response[i].universitySize +'</td>';
                        university_data += '<td style="text-align:center;">'+ response[i].totalMOOCS +'</td>';
                        university_data +='</tr>';
                    }
                }
            }
            $('#tableBody').html(university_data);
            sortTablebyNumber(1);
        } else {
            alert('There was a problem with the response request');
        }
    }
}
```

*Figure 34: Javascript code for updateTable() method*

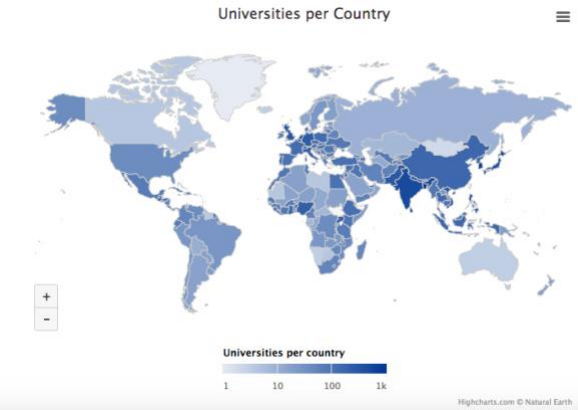
Regarding the graphs section, this will display graphs, charts and maps from Highcharts using my own data. Highcharts is a pure JavaScript library used to enhance web applications by adding interactive charting capability. My web application focuses on linear graphs, bar graphs, pie charts and maps to display the different regions. By using Highcharts, there is also a section called "Comparison", where the user will be able to compare two universities using charts views.

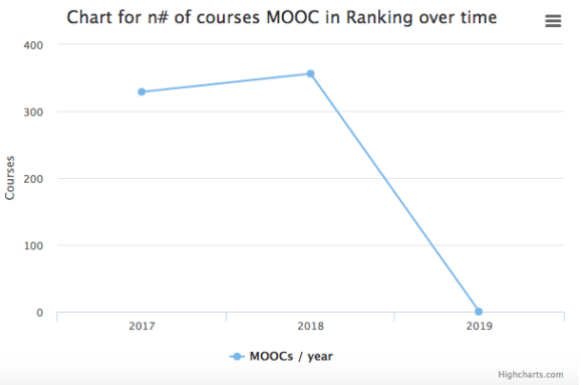
This is an example of the Highcharts code for a pie graph. Other types of graphs follow the same structure and slightly differ in some of the parameters.

```
Highcharts.chart('pie-graph-universities-1', {
    plotOptions: {
        pie: {
            allowPointSelect: true,
            cursor: 'pointer',
            dataLabels: {
                enabled: false
            },
            showInLegend: true
        }
    },
    chart: {
        renderTo: 'container',
        type: 'pie'
    },
    title: {
        text: 'Number of MOOCs per country'
    },
    series: [{
        name: 'Number of MOOCs per country',
        colorByPoint: true,
        data: countryMOOCsPie
    }]
});
```

*Figure 35: Highcharts Pie Chart example*

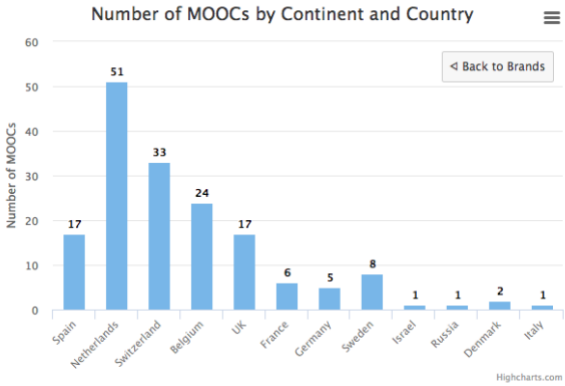
These are the different types of graphs that the MOOCs Ranking website will display:

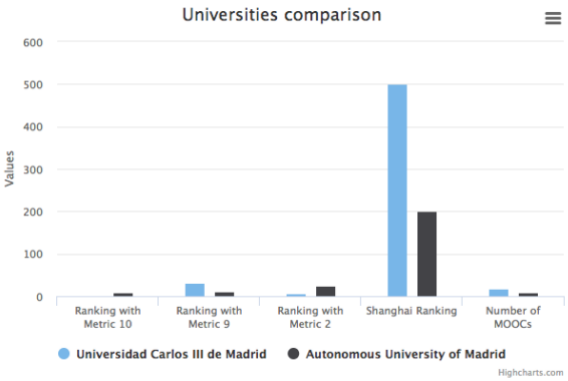


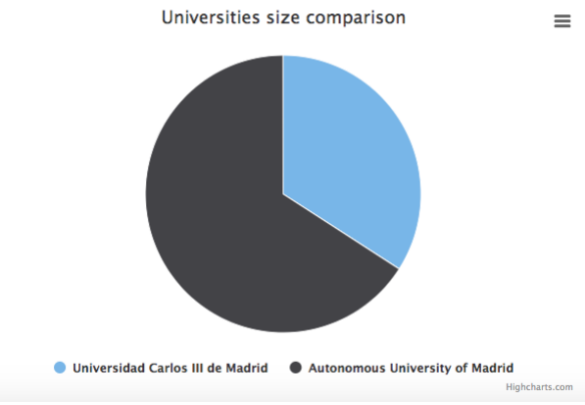

Maps: Distribution among physical space.

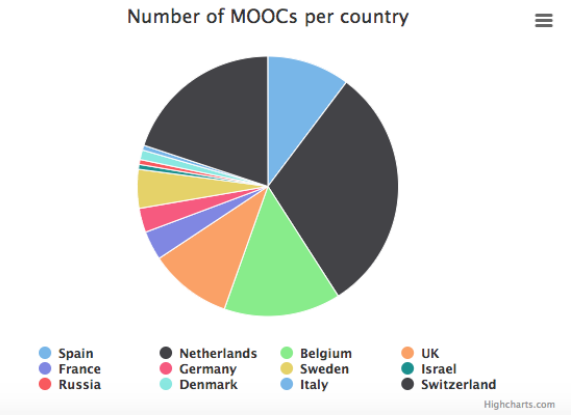
Linear: Comparison over time of many periods in a non-cyclical way.




Bar Charts: Used for comparison of items with one value each and few categories. When there are multiple categories, a horizontal bar chart shall be used.




Pie Graph: Static Composition. For comparing the university size (number of students) instead of bar chart, it was also used a pie chart for comparison because when the difference of values is significant, the bar graphs will look very unequal and the difference will not be seen clearly.

This diagram will display the different view pages and what they will display to the user in the website.

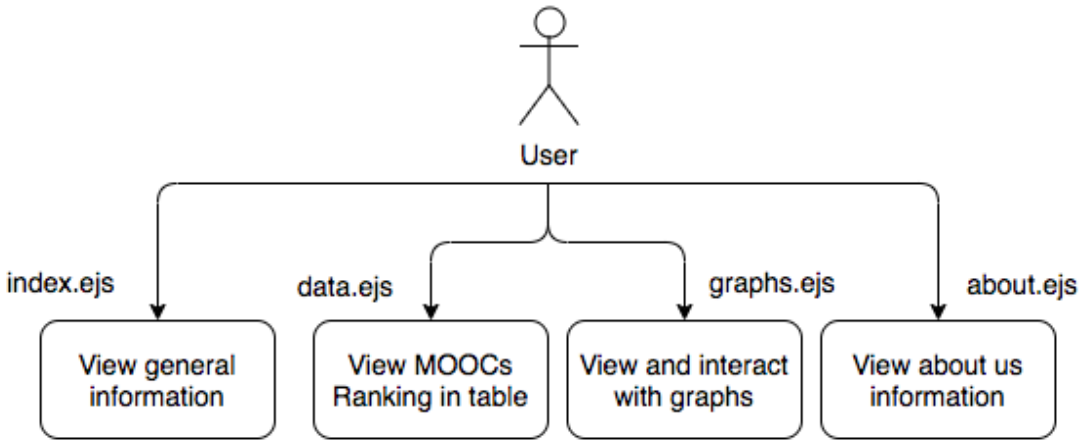



*Figure 36: User interaction with website ejs pages*

### 5.1.6 Heroku Deployment

As it was working correctly locally in the computer, the next step was to deploy it online through an application named Heroku. Heroku is a cloud platform as a service (PaaS) supporting Node.js that is used as a web application deployment model. The source code of my web application was deployed by using Heroku Git and pushing to master. Then, it was added a MongoDB to the resources of it.

Once the database was created and attached to Heroku, it was created a username and password. All the collections were inserted into the database, by running again the command of the Admin Manual. This time the database name is: ds159866 and the port number: 59866.

Finally, the web application, with all the data inserted, is running properly in Heroku in the following link:

https://moocs-university-ranking.herokuapp.com/

## 5.2 Testing

### 5.2.1 Results

These are the results according to Metric 2, which is dependent on the number of MOOCs and the university size.

| University Name | Ranking | Shanghai Ranking | Country | Size | MOOCS |
|---|---|---|---|---|---|
| Moscow Institute of Physics and Technology | 1 | 501 | Russia | 3319 | 39 |
| University of Pennsylvania | 2 | 17 | United States | 10019 | 54 |
| Duke University | 3 | NaN | United States | 14832 | 44 |
| Ecole Polytechnique Federale de Lausanne | 4 | NaN | Switzerland | 10536 | 37 |
| IE Business School | 5 | NaN | Spain | 547 | 13 |
| Rice University | 6 | NaN | United States | 3893 | 13 |
| Swiss Federal Institute of Technology Lausanne | 7 | 92 | Switzerland | 10124 | 33 |
| Wesleyan University | 8 | NaN | United States | 3213 | 12 |
| HEC Paris | 9 | 501 | France | 4000 | 11 |
| University of Geneva | 10 | 53 | Switzerland | 14489 | 27 |

*Figure 37: Metric 2 results table*

These are the results according to Metric 9, which is also dependent on the number of MOOCs and the university size.

| University Name | Ranking | Shanghai Ranking | Country | Size | MOOCS |
|---|---|---|---|---|---|
| Moscow Institute of Physics and Technology | 1 | 501 | Russia | 3319 | 39 |
| University of Pennsylvania | 2 | 17 | United States | 10019 | 54 |
| Duke University | 3 | NaN | United States | 14832 | 44 |
| Ecole Polytechnique Federale de Lausanne | 4 | NaN | Switzerland | 10536 | 37 |
| IE Business School | 5 | NaN | Spain | 547 | 13 |
| Rice University | 6 | NaN | United States | 3893 | 13 |
| Swiss Federal Institute of Technology Lausanne | 7 | 92 | Switzerland | 10124 | 33 |
| Wesleyan University | 8 | NaN | United States | 3213 | 12 |
| HEC Paris | 9 | 501 | France | 4000 | 11 |
| University of Geneva | 10 | 53 | Switzerland | 14489 | 27 |

*Figure 38: Metric 9 results table*

These are the results according to Metric 10, which is dependent on the number of MOOCs and the Shanghai ranking.

| University Name | Ranking | Shanghai Ranking | Country | Size | MOOCS |
|---|---|---|---|---|---|
| Moscow Institute of Physics and Technology | 1 | 501 | Russia | 3319 | 39 |
| University of California, Berkeley | 2 | 5 | United States | 33000 | 83 |
| Delft University of Technology | 3 | 151 | Netherlands | 19613 | 51 |
| Saint Petersburg State University | 4 | 301 | Russia | 32400 | 36 |
| National Research University Higher School of Economics | 5 | NaN | Russia | 31000 | 59 |
| Harvard University | 6 | 1 | United States | 22000 | 57 |
| University of Pennsylvania | 7 | 17 | United States | 10019 | 54 |
| National Taiwan University | 8 | 151 | Taiwan | 32168 | 41 |
| Peking University | 9 | 71 | China | 30248 | 45 |
| University of Illinois Urbana-Campaign | 10 | 37 | United States | 44087 | 47 |

*Figure 39: Metric 10 results table*

These are the results obtained by the three metrics. As it can be observed, the measurements vary from the different metrics. Metric 2 and 9 have the same top ten universities in the ranking, since both of them are dependent on the university size. However, Metric 10 depends on the Shanghai Ranking.

Metric 2 and 9 have their top ten composed of 4 universities in United States, 3 in Switzerland and 1 in France, Spain and Russia. This means that the universities in Europe, Asia and United States have numerous MOOCs with respect to the size of the university. The universities of Africa, Oceania and South America are excluded in the top list of these two metrics.

Contrarily, Metric 10 has its top-ten composed of 4 universities in United States, 3 in Russia and 1 in Netherlands, China and Taiwan. The conclusion that can be obtained with this is that United States has again the majority of the best innovative universities. Asia is represented also in a big percentage and now Europe is just represented by Netherlands. Oceania, South America and Asia are again excluded in the top list of this metric.

To conclude, United States has the vast majority according to all metrics, being the country with more MOOCs, precisely 800 courses. Russia has the second place with approximately 200 and then the European and Asian universities are slowly innovating in their universities.

### 5.2.2 Test cases

The test cases will be tested against the use cases of the system. The following table will represent the use case ID, the test case ID and description, the expected result and the obtained result.

| Test Case ID | Use Case ID | Test Case Description | Expected Result | Obtained Result |
|---|---|---|---|---|
| TC-01 | UC-01 | - Download data<br>- Compute metrics<br>- Insert data into database<br>- Display data into webpage tables<br>- Change metric | The table shall change accordingly to the metric selected | The table changes accordingly. |
| TC-02 | UC-02 | - Have data in the database<br>- Display data into webpage tables<br>- Filter continent<br>- Filter country<br>- Filter ranking | The table shall change according to the selected continent, country and ranking. | The table changes accordingly. |
| TC-03 | UC-03 | - Have data in the database<br>- Select two universities to compare in the dropdowns<br>-Click “compare” | A bar chart and pie graph shall be displayed comparing the two universities selected. | The graphs appear and compare the universities. |
| TC-04 | UC-04 | - Have data in the database<br>- Display data into graphs<br>- Filter the graphs according to continent | The graphs shall be displayed and change according to filtering criteria. | The graphs are displayed and change. |
| TC-05 | AC-01 | - Run the python program for Coursera<br>- Have a file with Coursera data<br>- Run the python program for EdX<br>- Have a file with EdX data | There shall be two csv files with all the universities data. | The files are downloaded with the data. |

| | | | | |
|---|---|---|---|---|
| TC-06 | AC-02 | - Have several csv files with all data<br>- Merge files and remove conflicts<br>- Calculate metrics<br>- Check all parameters are positive and correct | Have a csv file merged, with no conflicts and with the metrics and ranking positions | The file is obtained correctly with all the parameters. |
| TC-07 | AC-03 | - Have a csv file with all data cleaned and merged together<br>- Run the MongoDB terminal command<br>- Insert data<br>- Check the MongoDB database | The database shall have all the data inserted properly into the collections. | The collections shall have the data inserted. |

*Table 32: Test Cases according to Use Cases*

# 6. Budget and Planning

In this section of budget and planning, there will be a table regarding the project hours by activities, a table regarding the activities and its budget, and a Gantt diagram for the planning of the project.

## 6.1 Project Hours by Activities

In this section, there will be a table describing different categories, the activities regarding each category and the number of hours spent on them.

| Categories | Activities | # hours | # hours |
|---|---|---|---|
| Proposal | Proposal for the project and decision-making | 3 | 3 |
| Research | Research in Rankings | 5 | 32 |
| | Research in MOOCs | 4 | |
| | Research in APIs for Coursera, EdX and Shanghai Ranking | 5 | |
| | Research in legal conditions | 3 | |
| | State of Art | 15 | |
| Analysis | User stories | 1.5 | 16 |
| | Use cases | 2.5 | |
| | Use cases Specifications | 4 | |
| | System Requirements | 8 | |
| Design | Sequence diagrams | 5 | 25 |
| | Class diagram | 6 | |
| | Component diagram | 6 | |
| | Deployment diagram | 5 | |
| | Mockup | 3 | |
| Implementation | Create a basic website for testing the web server | 2 | 118 |

| | | | |
|---|---|---|---|
| | Download, clean and compute data from Coursera and edX | 8 | |
| | Create the web server (NodeJS and Express) | 25 | |
| | Create the database (MongoDB) | 4 | |
| | Insert data into the database | 2 | |
| | Connect the web server to the database | 1 | |
| | Create server-database endpoints | 5 | |
| | Connect the web server to the website | 3 | |
| | Improve the website design | 20 | |
| | Connect the pages of the new website with the information of the database by means of EJS | 18 | |
| | Create the filtering feature in website | 13 | |
| | Implement the graphs in the website | 6 | |
| | Implement the map in the website | 7 | |
| | Deploy the web application | 4 | |
| Testing | Test cases against use cases | 40 | 40 |
| Documentation | Creating a table of contents | 1 | 59 |
| | Time spent doing research | 10 | |
| | Time spent writing | 38 | |
| | Time applying tutor suggestions | 10 | |
| Meetings | Meetings with tutor teacher | 4 | 4 |
| **TOTAL (h)** | **Number of total hours spent on the project** | **297 hours** | **297 hours** |

*Table 33: Project Hours Estimation*

## 6.2 Project Budget Breakdown

In this section, the costs of the project will be calculated. In order to determine them, the budget will be divided into Direct Costs (Human Resources, Equipment, Licenses and Printing), Indirect Costs and the total cost of the project.

### 6.2.1 Human Resources Costs

The Human Resources costs are divided in activities which are the same as the categories of the project hours table. Since I am supposing the human resources hours will be paid as 23€/h, then the sum of the hours per category will be multiplied for that value.

| Activities | Human Resources Costs (€) |
|---|---|
| Proposal | 69,00 € |
| Research | 736,00 € |
| Analysis | 368,00 € |
| Design | 575,00 € |
| Implementation | 2.714,00 € |
| Testing | 920,00 € |
| Documentation | 1.357,00 € |
| **TOTAL (€)** | 6.739,00 € |

*Table 34: Human Resources Costs*

### 6.2.2 Equipment Costs

The equipment cost will be calculated using the following formula to consider the wear of the computer:

$$Equipment\ cost = \frac{computer\ price * (25\%\ wastage/year)}{(12\ months/year)} * project\ duration\ in\ months$$

$$Equipment\ cost = \frac{1200€ * 0.25}{12} * 9 = 225{,}00€$$

### 6.2.3 Direct Costs

The direct costs will be the sum of Human Resources, Licenses, Equipment and Printing costs. The total will be 6.969,00€.

| | HR | Licenses | Equipment | Printing | TOTAL (€) |
|---|---|---|---|---|---|
| **TOTAL (€)** | 6.739,00 € | 0,00 € | 225,00 € | 5,00 € | 6.969,00 € |

*Table 35: Direct Costs*

### 6.2.4 Indirect Costs

The indirect costs usually refer to the costs that are not directly accountable, like the electricity, security and the office. In order to do an estimation, it will be considered as 20% of the direct costs, so the total will be 1.393,80€.

| | Direct Costs | Percentage | Indirect Costs |
|---|---|---|---|
| **TOTAL (€)** | 6.969,00 € | 20% | 1.393,80 € |

*Table 36: Indirect Costs*

### 6.2.5 Total Costs

The total costs are calculated by adding direct and indirect costs. Then, I will calculate the profit as a 10% and add the IVA (21%) to the sum of the costs plus the profit. The final sum of all the costs + profit + IVA will constitute the total cost of the project, being 11.130,89€.

| | TOTAL (€) |
|---|---|
| **Direct Costs** | 6.969,00 € |
| **Indirect Costs** | 1.393,80 € |
| **Total Costs** | 8.362,80 € |
| **Profit (10%)** | 836,28 € |
| **Costs + Profit** | 9.199,08 € |
| **IVA (21%)** | 1.931,81 € |
| **TOTAL + IVA (€)** | **11.130,89 €** |

*Table 37: Total Costs*

## 6.3 Gantt diagram

The Gantt diagram is a graphical tool whose objective is to determine the planned time for different tasks and establishing milestones for them.

| | | | Sept | | Oct | | Nov | | Dec | | Jan | | Feb | | Mar | | Apr | | May | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Categories** | **Activities** | **#h** | **M 1** | **M 2** | **M 3** | **M 4** | **M 5** | **M 6** | **M 7** | **M 8** | **M 9** | **M 10** | **M 11** | **M 12** | **M 13** | **M 14** | **M 15** | **M 16** | **M 17** | **M 18** |
| **Proposal** | Proposal for the project and decision-making | 3 | | | | | | | | | | | | | | | | | | |
| **Research** | Research in Rankings | 5 | | | | | | | | | | | | | | | | | | |
| | Research in MOOCs | 4 | | | | | | | | | | | | | | | | | | |
| | Research in APIs for Coursera, EdX and Shanghai Ranking | 5 | | | | | | | | | | | | | | | | | | |
| | Research in legal conditions | 3 | | | | | | | | | | | | | | | | | | |
| | State of Art | 15 | | | | | | | | | | | | | | | | | | |
| **Analysis** | User stories | 1,5 | | | | | | | | | | | | | | | | | | |
| | Use cases | 2,5 | | | | | | | | | | | | | | | | | | |
| | Use cases Specifications | 4 | | | | | | | | | | | | | | | | | | |
| | System Requirements | 8 | | | | | | | | | | | | | | | | | | |
| **Design** | Sequence diagrams | 5 | | | | | | | | | | | | | | | | | | |
| | Class diagram | 6 | | | | | | | | | | | | | | | | | | |
| | Component diagram | 6 | | | | | | | | | | | | | | | | | | |
| | Deployment diagram | 5 | | | | | | | | | | | | | | | | | | |
| | Mockup | 3 | | | | | | | | | | | | | | | | | | |
| **Implementation** | Create a basic website for testing the web server | 2 | | | | | | | | | | | | | | | | | | |
| | Download, clean and compute data from Coursera and edX | 8 | | | | | | | | | | | | | | | | | | |
| | Create the web server (NodeJS and Express) | 25 | | | | | | | | | | | | | | | | | | |

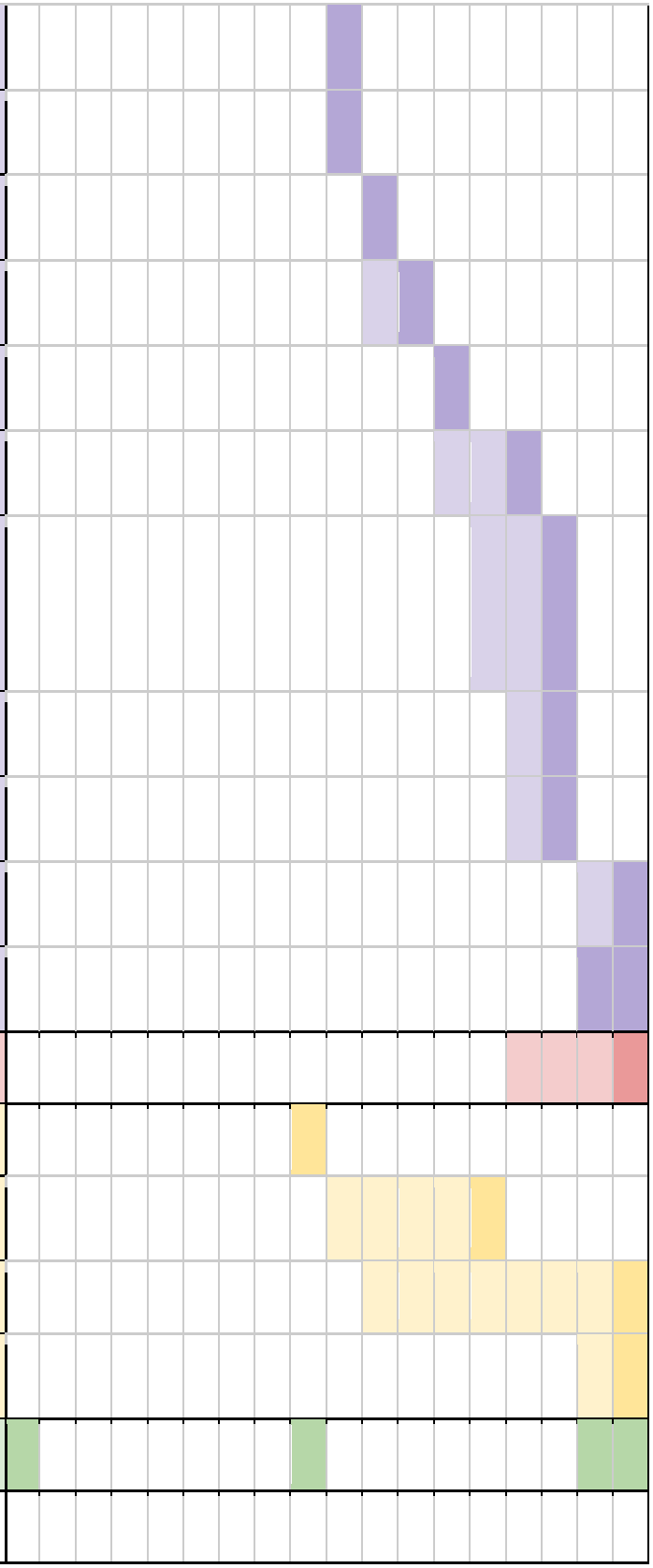
| | | | |
|---|---|---|---|
| | Create the database (MongoDB) | 4 |  |
| | Insert data into the database | 2 | |
| | Connect the web server to the database | 1 | |
| | Create server-database endpoints | 5 | |
| | Connect the web server to the website | 3 | |
| | Improve the website design | 20 | |
| | Connect the pages of the new website with the information of the database by means of EJS | 18 | |
| | Create the filtering feature in website | 13 | |
| | Implement the graphs in the website | 6 | |
| | Implement the map in the website | 7 | |
| | Deploy the web application | 4 | |
| **Testing** | Test cases against use cases | 40 | |
| **Documen-tation** | Creating a table of contents | 1 | |
| | Time spent doing research | 10 | |
| | Time spent writing | 38 | |
| | Time applying tutor suggestions | 10 | |
| **Meetings** | Face-to-face meetings with tutor teacher | 4 | |
| **TOTAL(h)** | Number of total hours spent on the project | **297** | |

*Table 38: Gantt diagram*

# 7. Legal and Socio-economic environment

## 7.1 Legal Framework

In this section, it will be explained the legal conditions of Coursera, EdX and Shanghai Ranking about the use of their data. Moreover, it will also be discussed the deployment conditions of Heroku and the logo trademark of Highcharts and Heroku.

### 7.1.1 Coursera legality

Regarding Coursera, I explicitly contacted their Coursera Support Team since their API is deprecated. Regarding their terms and conditions, there is not a problem to use the data obtained of the enterprise as long as it is properly mentioned the source of the information and the data does not violate the privacy or copyright conditions.

*"Coursera is committed to advancing the science of learning and teaching, and records of your participation in courses may be used for education research. In the interest of this research, you may be exposed to variations in the course content. Research findings will typically be reported at the aggregate level. Your personal identity will not be publicly disclosed in any research findings without your express consent."* [11].

Therefore, no information of a user's identity should be revealed. Furthermore, it is also forbidden to share content that you do not have the right to share or misappropriate someone else's intellectual property. For that reason, it should be always attributed the data or materials used to the original copyright owner. However, the data collected and used by MOOCs Ranking will not affect copyright conditions because does not use the user's identity and will only be used for research purposes.

### 7.1.2 EdX legality

Opposite to Coursera, EdX has an API that provides information about the courses. Nevertheless, this API is related to the MOOC courses information and its users, not about the universities [12].

According to their website, *"You agree to retain all copyright and other notices on any content you obtain from the edX Site. All rights in the edX Site and its content, if not expressly granted, are reserved."* [13].

Therefore, I contacted EdX Contact Support in order to ask if I could use the data from their RSS in my research and if there were any legal issues to that, being their answer that their information is free for use and with no legal implications.

### 7.1.3 Shanghai Ranking legality

Regarding Shanghai Ranking, I explicitly contacted their email since their website has no information about their legal conditions. In their reply, they said that I could use any data that is published on their website.

### 7.1.4 Heroku legality

*"The Content displayed and/or processed through your Application or other web site utilizing the Service shall not contain any of the following types of content: Content that infringes a third party's rights (e.g., copyright) according to applicable law."* [14]. According to this statement, in order to not infringe the copyright conditions, I need to mention the data has been obtained from the Shanghai Ranking website, and the Coursera and EdX applications and give credit to them.

### 7.1.5 Other Legal Conditions

There is a new normative in the EU (European Union) about the privacy in web named General Data Protection Regulation (GDPR). However, it does not apply because this web application has no citizen's data, but organizations data, thus personal information is not collected. [15]

According to legal conditions of Heroku and Highcharts, the logo must remain in the web application without any modifications to it. [16]

## 7.2 Digital Transformation in Higher Education

Education plays a major role in today's society. Individuals learn to think for themselves, learn about how the world works and how to interact with it. Therefore, higher education has a critical function because it helps people to specialize and learn in-depth content about a specific topic. In a sense, individuals become specialists about a profession. However, how should education be improved? Are universities innovating or are they still using traditional methods for teaching?

With respect to the socio-economic environment, it should be mentioned the importance of digital transformation in the education sector. Today, technology has revolutionized the way we learn, we connect and we share experiences and knowledge with people from other countries and continents. This means that, due to technology, the barrier of distance and time has been broken. Now, anyone can learn online without the need to be in person at a certain place and time. Therefore, little by little, the educational model is migrating to a fusion of traditional and non-classroom learning. Non-face-to-face education will continue to advance and will be positioned as an essential pillar in the education system, creating a hybrid model.

This model will have consequences regarding the interaction between teachers and students. Before the education was focused on what the teacher taught the student, while now the student is becoming autonomous and uses the tools that the teacher provides to them in order to learn on their own. This creates a comfortable and equitable study and work environment. Therefore, some pedagogical models have been established, for instance the concept of flipped classroom. This model transfers the work of learning out of the classroom and uses class time, together with the teacher's experience, to facilitate, practice and enhance knowledge.

Another of the new learning tools that has emerged in recent years are MOOCs. As explained above, a MOOC is a Massive Open Online Course, which is an online course offered usually by a university in order to teach a diverse audience about a certain topic. It started in Canada, but it was officially settled in 2011, when a professor in Stanford uploaded the first MOOC about Artificial Intelligence - more than 150.000 people and 190 countries students signed up for his course.

MOOCs provide a more engaging and accessible education to everyone. It creates an interactive community of highly motivated learners. However, what is the role MOOCs will

have in the future of education? There will be targeting three main areas: high schools, universities and the job market [17]. Some of the goals that MOOCs are targeting to achieve are:

1. Create an educational transformation using the flipped pedagogy.
2. Teach students at University.
3. Support lifelong learning, specialization, improvement of professional skills.
4. Provide education for people who cannot access higher education.
5. Connect students from all around the world at bachelor or master level to share their expertise.

MOOCs could also serve to guide high school students to decide what they want to pursue their higher education. According to a study, more than 30% of students change or leave the degree in the first year of University. It turns out that 40% of students shuffle up to three possible degrees, while 33% doubt between two and 6% are lost [18]. This means that only 21% is clear about what to study. Therefore, complete some MOOC courses before making the decision of what degree to choose could be beneficial for students.

Another of the greatest benefits of the impact of technology on society is to provide an open, free and accessible education. Not everyone has the money or resources for a higher education, thus universities offer MOOCs - free courses from which the population can benefit and improve their knowledge in the specific area.

In 2018, the European Commission has adopted a Digital Education Action Plan consisting of 11 initiatives to support the development of digital skills and the use of technologies in education. The three priorities established pretend to make better use of digital technology for teaching and learning, develop the digital skills and improve education through better analysis.

One of these initiatives include: "*The European Commission facilitated moreover the Teacher Academy57, which offers to all teachers in Europe the possibility to pursue free professional development in the form of MOOCs.*" [19]

Nevertheless, it also needs to be analyzed the economic sector. There can be distinguished three different business models: United States, Europe and Africa/Asia.

In United States, having mostly private universities, MOOCs show an ongoing direction. It is already being implemented to mix online and on campus courses in order to decrease the students' fees and address the courses to a larger number of students.

On the contrary, the European Universities are obtaining grants from national and European agencies and the level of fees is rather limited. Therefore, there is no clear business model since funding will be a limitation in most institutions. The solution is to deliver ECTS when completing MOOCs, although these ones still are not transferable.

The last business model is the one of Africa and Asia. Most countries do not possess the number of required teachers and their training will take years. They do not have the capacity, the human resources or the capital to invest in building classical universities. Teaching through specialized MOOCs will lower the number of required teachers and the investments, teaching a diverse and large number of students. [20]

In one way or another, MOOCs represent a revolution and are the only realistic substitute to educate large masses of students. However, it is still an innovative project and therefore needs a strategy and previous study, as well as marketing. For a MOOC to be transcendental, it needs to be done at the right time and the right content. It needs also to have applied quality control and to be adapted to target a diverse population - people with a degree, postgraduate and without some studies. It will help to create a highly educated middle class, extend the area of expertise beyond a degree and change the way society educates itself. Nonetheless, in order for them to work, MOOCs needs validation from Universities and other institutions, being validated as university credits or as a certificate.

# 8. Manuals

In this section, there will be a description of two manuals - the user manual and the data curator manual.

## 8.1 User manual

The is the manual for the end-user, in order to explain how the web application can be used. First, the user will enter in the "Home" page, where the general information of the project will be explained, as well as the three different metrics that are being used. This is a screenshot of the homepage. Below in the website more information will be included.

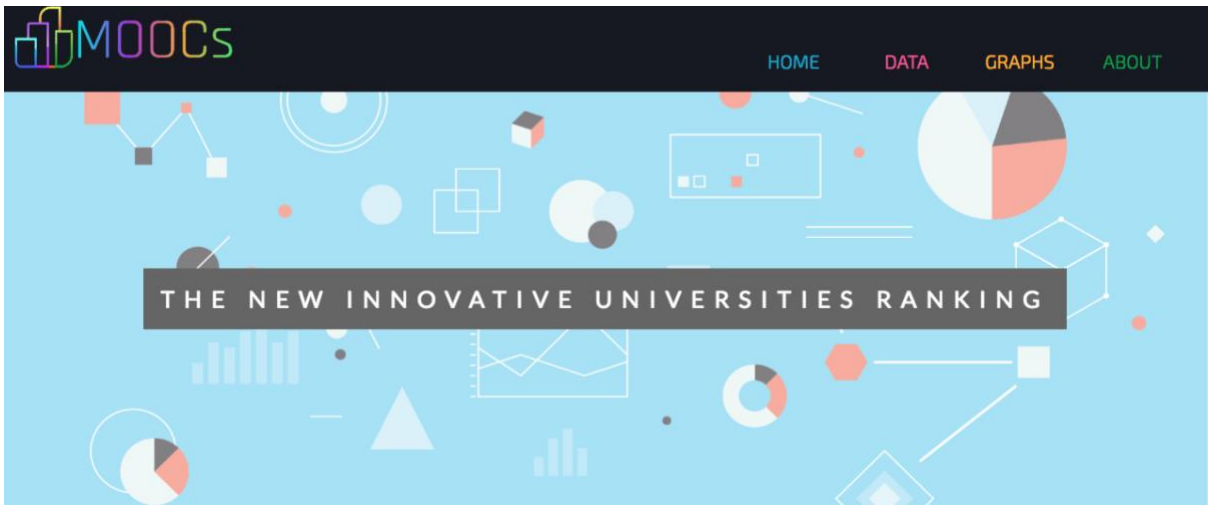



*Figure 40: Homepage screenshot*

In the data section, there will be a responsive table with all the data and some filtering to do. According to the screenshot below, there are two filtering ways.

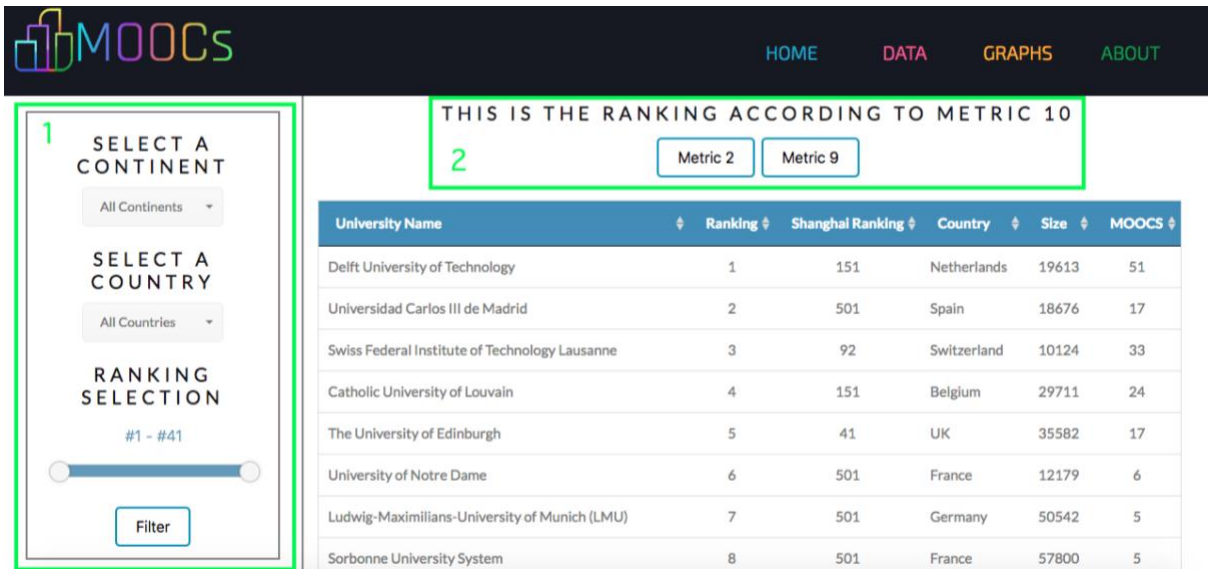



*Figure 41: Data page filtering methods screenshot*

The first is the filtering in the side, which will filter the table according to metric that is selected by continent, country and ranking position. The other filtering is placed at the top - it will change the table according to the metric that is being selected and the ranking will be completely different.

Now, in this new screenshot, it is still using metric 10, but selecting the country "Spain" and the ranking of the first 10 positions, so the table will be filtered with those inserted parameters in order to provide a more efficient search.

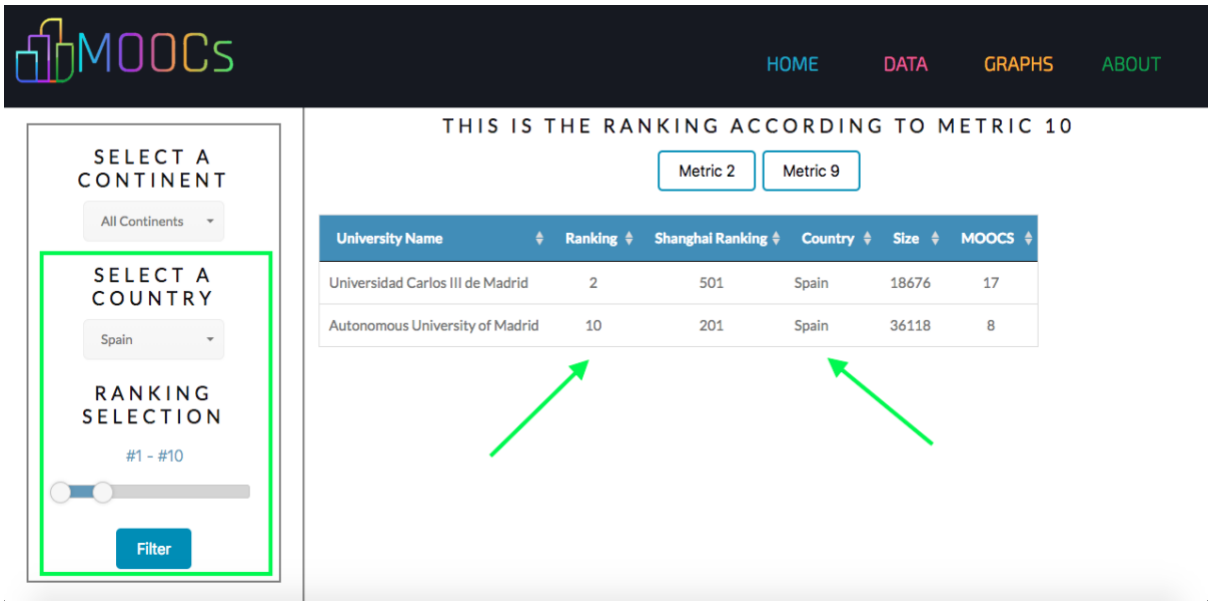



*Figure 42: Data page searching results screenshot*

As it is displayed in the screenshot, there are 3 different metrics: Metric 2, Metric 9 and Metric 10. When click in the button of another metric, it will change the ranking table. Now, there will be an example:

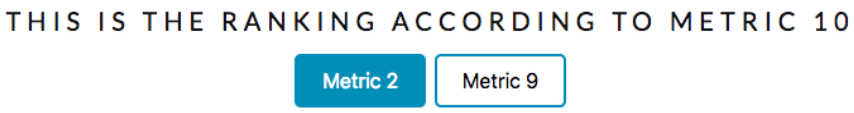



*Figure 43: Data page metric buttons screenshot*

In the example, it will be chosen "Metric 2", and the table will be filtered using the same parameters as in the example above: "Spain" and ranking position from 1 to 10.

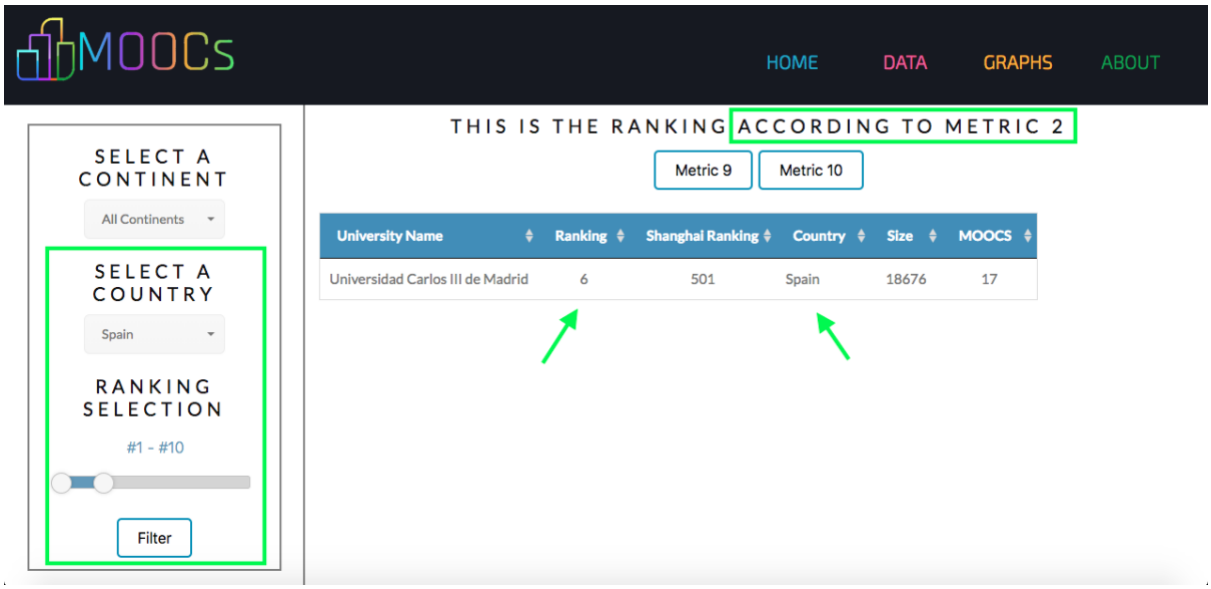



*Figure 44: Data page searching results in metric 2 screenshot*

The result in this case is completely different as in the previous example. Now, there is only 1 university according to this metric and it will be in position 6 instead of position 2. The explanation of each metric will be found in the homepage.

Regarding the graphs section, there is 4 different sections: Maps, Linear Graphs, Bar Chart and Comparison.

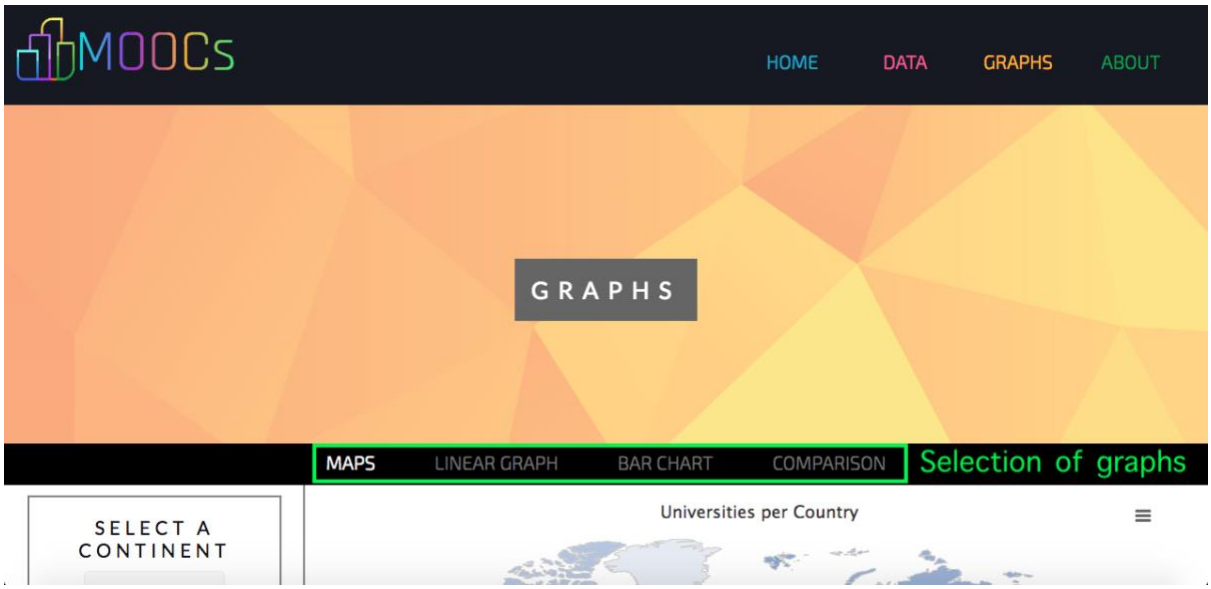



*Figure 45: Graphs page menu of graphs screenshot*

In the first section, “maps”, there is going to be a world map that displays the number of universities that every country has, inside the ranking I have made. Every time the user hovers or click in a country, this one will appear in green and it will be displayed the number of universities that it has. Here is a screenshot:

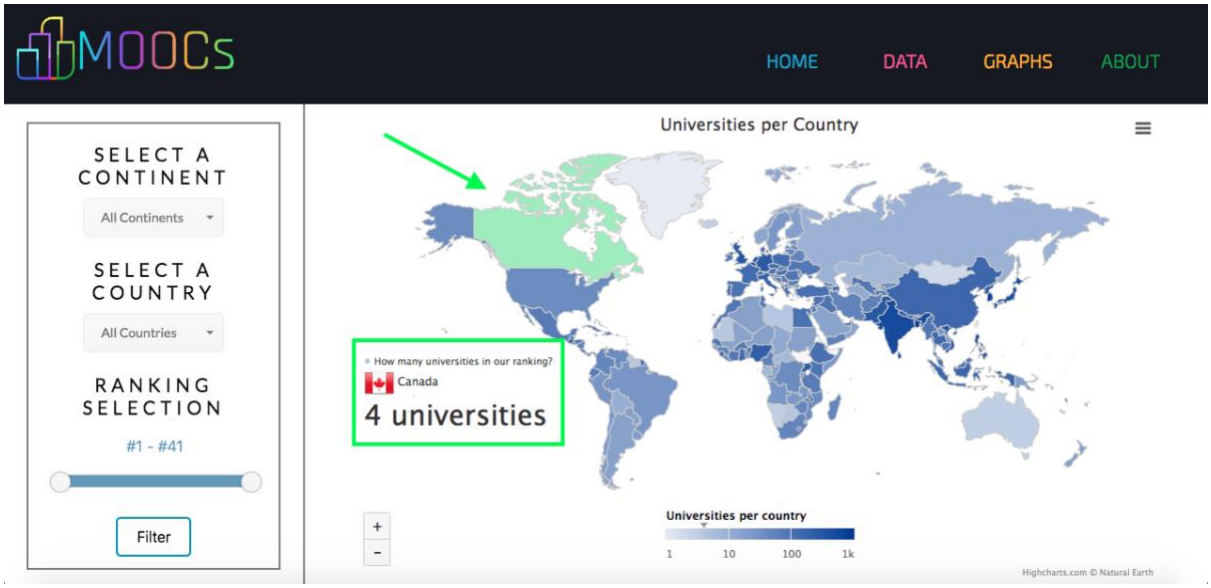



*Figure 46: Graphs page (maps) screenshot*

In the next section, “linear graph”, there will be a temporal display of how many MOOCs are being added per year. However, as I just started the project this year, the number of MOOCs remains the same, so this is a section to improve for the future.

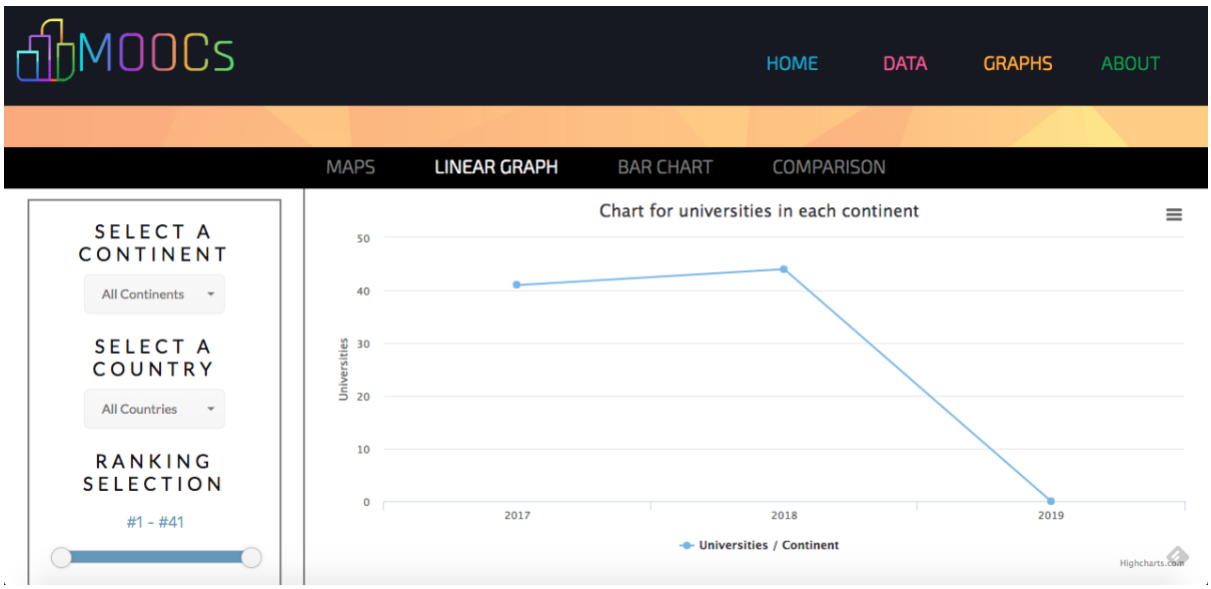



*Figure 47: Graphs page (linear graphs) screenshot*

In the section of "bar chart", there will be a chart displaying the number of MOOCs by Continent and by Country.

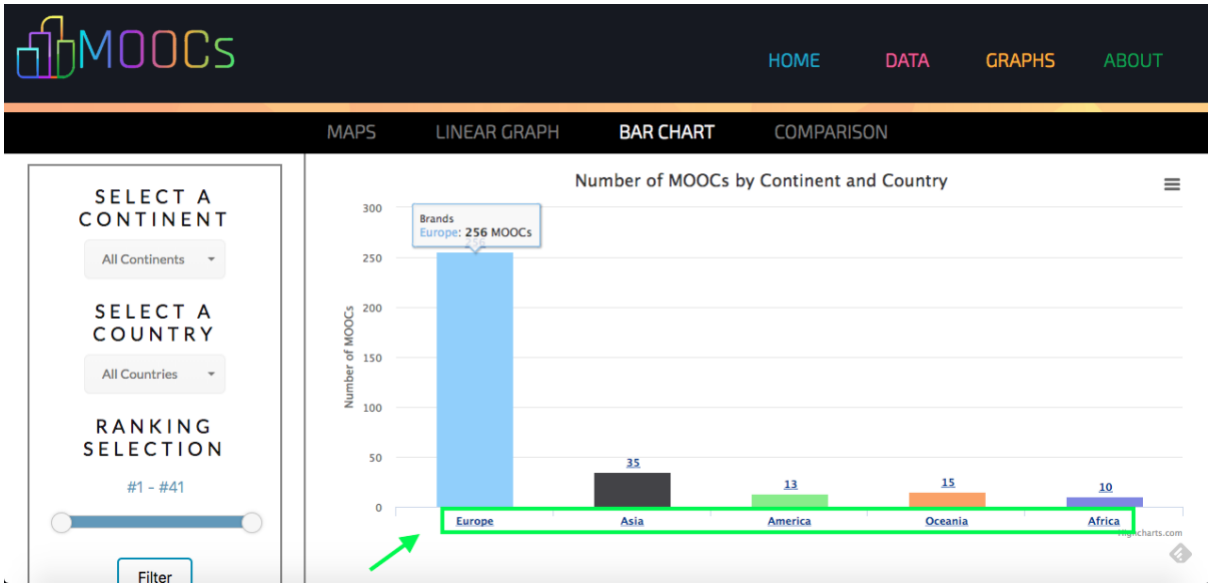



*Figure 48: Graphs page (bar chart by continent) screenshot*

When clicking in any of the bars of the continents, the graph will be modified and it will contain the countries of that continent and the number of MOOCs they have.

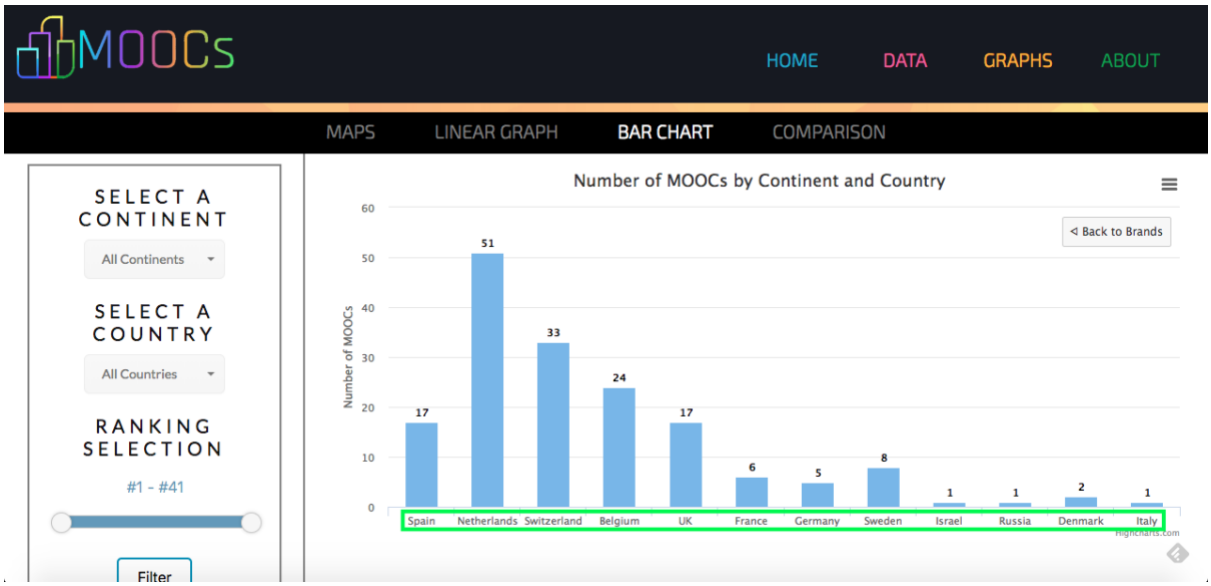



*Figure 49: Graphs page (bar chart by country) screenshot*

Finally, the last section is the "Comparison". The user will select 2 universities to compare in a dropdown list. When clicking on "Compare", then some graphs will appear displaying the results of the following categories: ranking according to metric 2, ranking according to metric 9, ranking according to metric 10, Shanghai Ranking and total number of MOOCs. Also, there will be a pie chart comparing both universities size.

In this first screenshot, it will be displayed the dropdowns are dependent from each other. When modifying the continent or country in the sidebar, then the university selection dropdowns will also be modified.

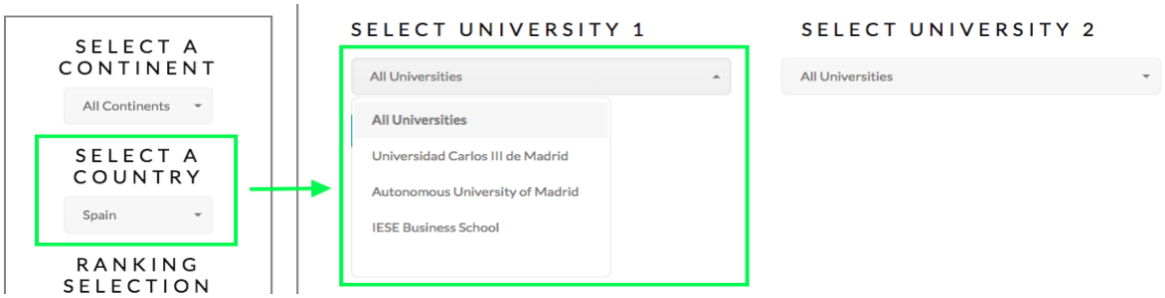



*Figure 50: Graphs page comparison dropdowns screenshot*

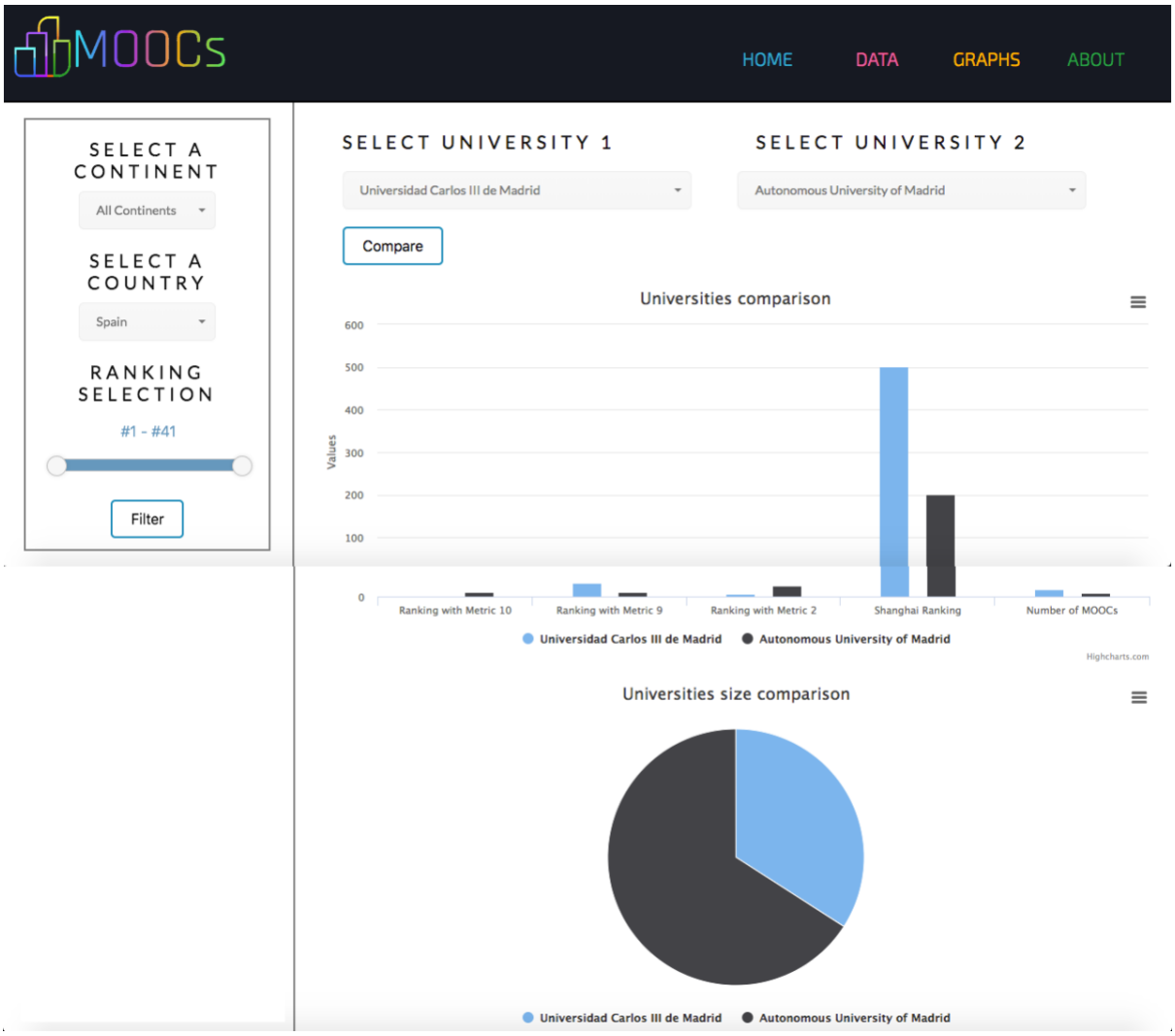



*Figure 51: Graphs page (bar and pie charts) screenshot*

Finally, "About" will display my name and my tutor name for this project.

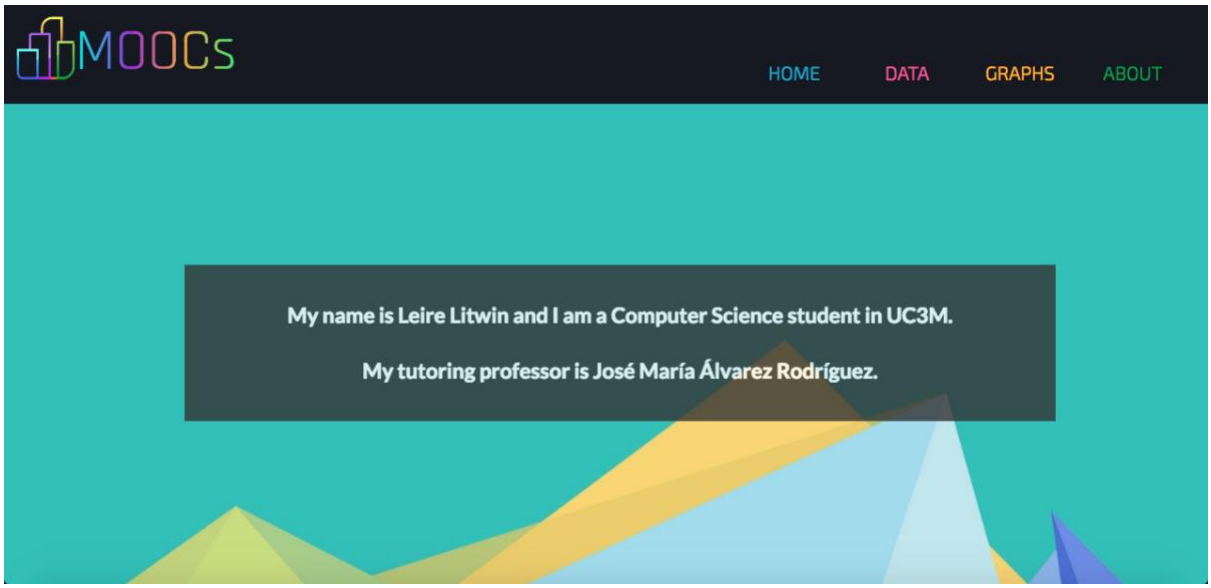



*Figure 52: About page screenshot*

## 8.2 Data Curator manual

In this part, it will be explained what the admin should do in order to maintain the web application running and updated. There are several tasks the admin should complete. First, the data collection. In order to complete this task there is 2 different programs: coursera-extractor.py and edx-extractor.py that will extract the MOOCs and universities information. In order to run them, it is only necessary to download the xml file previously and then to do in the terminal the following command inside the directory with the file in it:

```
python2 edx-extractor.py
python2 coursera-extractor.py
```

Then, the next task the admin shall complete is to clean the data, since many universities name do not coincide even though they refer to the same university. Some information, as the name of the courses, is not necessary for us, but interesting to have. Therefore, the admin shall delete the unnecessary data and remain the important one: universityName, country, countryCode, continent, universitySize, universitySizeValue, shanghaiRanking, courseraMOOCS, edXMOOCS, totalMOOCS, M2, ranking2, M9, ranking9, M10 and ranking10. I previously explained how to calculate these metrics parameters (M2, M9 and M10) and the ranking will be computed according to them.

Finally, in order to insert them into the database, the data should be divided in different tables, which are explained in the section 5.1.2. There should be different collections for each year, for instance: universities2017. Then, these tables should be exported into a csv file. In order to insert the several files into the database, it is only necessary to do:

```
mongoimport -h ds159866.mlab.com:59866 -d
heroku_8356kf3j -c collection_name -u myusername -p
mypassword --file file_path --type csv --headerline
```

Then, the application will do the rest of the tasks and the database will be updated with new data, so the application will also be updated according to the new information.

# 9. Conclusion and Future Work

The conclusion will be divided in three main points - the results obtained, the technical challenge and the personal achievement. Regarding the results, the answer to the question arises whether the number of MOOCs can be used as a valid indicator to measure innovation.

On the one hand, the education sector is key in terms of technology innovation. But do any of the existing rankings measure this innovation? Some of them take some innovation indicators based on a university's ability to help industry with innovations, even though it was found in just one ranking and weighted 2.5% - counting this measurement as practically nothing. The rest of them do not seem to reflect the innovation anyhow. For that reason, it would be interesting to use it as a transcendental weighted parameter.

Notwithstanding, I concluded that MOOC-related activities can be used as an indicator of innovation inside a ranking, but there are some considerations to take into account. Even though Metric 10 takes into account the Shanghai Ranking, the metric calculation of innovation should also consider other factors and indicators than the total number of MOOCs. In order for a ranking to be complete, it should measure innovation, but also parameters like doing faculty and students surveys, universities research and internationalization, maintaining also the innovation indicator. The business model also varies dependently on the continent - United States, Europe, Asia and Africa will not have the same business model. Therefore, the innovation in universities would work slightly differently. Finally, as concluded in the socio-economic environment, MOOC activities should be rewarded in order to be considered as a revolution. The gratification could be in several ways - either in ECTS validated by the university or in a certificate form.

With regard to the technical part, my project is also divided into different challenges. On the one hand, there is the difficulty of obtaining data and processing the data. However, today there are many technologies that facilitate these two processes. In this case, I learned how to make a python crawler to download information from MOOCs. This was a critical obstacle because at the beginning I had a program for the Coursera API. Nevertheless, recently Coursera deprecated the API and there was no way to access the data. For that reason, I had to create the crawler myself to download the data.

On the other hand, there is the entire process of creating a web application - the database, the server and the webpage. Nowadays, there are tools that facilitate the process of creating a web application, for instance Express and Node.js. This application is also deployed in Heroku, which means that it is actually working properly and it is ready to be used. It is not only a prototype or a project, but an actual application that people can use. Personally, regarding the technical solution, I have efficiently learnt in the development of the project and I believe it can be extremely useful for the future.

In conclusion, considering personal achievements, I feel satisfied of the application that I have built. In the implementation of the bachelor thesis, I have ascertained that what I most enjoy developing is front-end, human-computer interaction, the visual and interactive part of computer science and engineering. Personally, that was the most crucial part of doing this project, thus I would like to continue learning this field in a more specialized way and do a master's degree related to design, user interfaces and HCI.

## 9.1 Objectives achievement

All the objectives described in the section 3 of the Introduction part have been achieved when creating the web application.

| Objective nº | Description | Result |
|---|---|---|
| 1 | Collecting data from several sources and cleaning the necessary data. | Achieved |
| 2 | Measure the innovation of the universities with regard to technology and its advances (MOOCs) by creating some metrics. | Achieved |
| 3 | Creating a comparison between today's regular rankings and the one that measures the MOOCs according to the different metrics. | Achieved |
| 4 | Displaying the top most innovative universities according to the metrics. | Achieved |
| 5 | Filtering the universities according to different criteria. | Achieved |
| 6 | Comparing two different universities. | Achieved |

*Table 39: Objectives Achievement*

## 9.2 Future work

Some of the future work I would like to achieve is to continue measuring innovation in worldwide Universities in order to obtain a linear graph timeline and analyze the progress and innovation of the top universities around the world. In this way, it can also be evaluated the success of the MOOCs in the next years and if they really caused the revolution that was being expected.

Moreover, I believe it can be interesting to make a comparison with other top rankings, as can be “Top Universities”, “Times Higher Education” or “Webometrics”. Nevertheless, in order to do so, the methodology of the innovative ranking shall be improved and take into account other factors as the Teaching, Research and Internationalization of the University, although it shall still maintain a high weight for the innovation indicator.

Finally, the last goal to achieve for the future would be to automate all the process of obtaining and cleaning the data with a script that could automatically run every week. Thus, there would be no need to have an administrator to maintain the web application up-to-date.

To conclude, as the web application is built, the future work will be based on extending and perfect the project, adding new functionalities to the system and updating the data every week.

# 10. Bibliography

[1] The 50 Most Popular MOOCs of All Time. (2018, February 16). Retrieved from April 23, 2018, from https://www.onlinecoursereport.com/the-50-most-popular-moocs-of-all-time/

[2] About Academic Ranking of World Universities | About ARWU. (n.d.). Retrieved April 23, 2018, from http://www.shanghairanking.com/aboutarwu.html

[3] QS World University Rankings – Methodology. (2018, June 06). Retrieved April 23, 2018, https://www.topuniversities.com/qs-world-university-rankings/methodology

[4] World University Rankings 2018 methodology. (2018, April 03). Retrieved April 24, 2018, from https://www.timeshighereducation.com/world-university-rankings/methodology-world-university-rankings-2018#survey-answer

[5] 2018. Ranking Web of Universities. January New Edition. (n.d.). Retrieved April 25, 2018, from http://www.webometrics.info/en/node/200

[6] Tableau Software Dashboard [Digital image]. (n.d.). Retrieved June 12, 2018, from https://cdnl.tblsft.com/sites/default/files/dashboard-2---ed.jpg

[7] Dekker, M. D. (2016, July 14). 4 1 Architectural View Model [Digital image]. Retrieved June 14, 2018, from https://upload.wikimedia.org/wikipedia/commons/e/e6/4%2B1_Architectural_View_Model.svg

[8] E. (2018, May 25). MEAN Stack Architecture: AngularJS, NodeJS, ExpressJS and MongoDB. Retrieved May 2, 2018, from https://evincedev.com/blog/mean-stack-architecture/

[9] Surrel, G. (2013, May 27). MVC Diagram (Model-View-Controller) [Digital image]. Retrieved May 23, 2018, from https://commons.wikimedia.org/wiki/File:MVC_Diagram_(Model-View-Controller).svg

[10] Alvarez-Rodríguez J.M., de Amescua A., García-Guzman J., Sánchez-Segura M. Measuring the Degree of Innovation in Education Through the Implementation of an Indicator Based on Analyzing MOOC-Related Activities, pages 123–133. Springer International Publishing, Cham, 2017. DOI: http://dx.doi.org/10.1007/978-3-319-59044-8_14

[11] Coursera. (n.d.). Terms of Use. Retrieved May 3, 2018, from https://www.coursera.org/about/terms

[12] EdX API. (n.d.). Open edX ReST APIs. Retrieved May 4, 2018, from https://open.edx.org/open-edx-rest-apis

[13] EdX. (2018, April 19). EdX Terms of Service. Retrieved May 4, 2018, from https://www.edx.org/edx-terms-service

[14] Acceptable Use Policy | Heroku. (n.d.). Retrieved May 9, 2018, from https://www.heroku.com/policy/aup

[15] 2018 reform of EU data protection rules. (2018, June 08). Retrieved June 10, 2018, from https://ec.europa.eu/commission/priorities/justice-and-fundamental-rights/data-protection/2018-reform-eu-data-protection-rules_en

[16] Highsoft Standard License Agreement 4.0. (n.d.). Retrieved June 11, 2018, from https://shop.highsoft.com/media/highsoft/Highsoft-Standard-Terms-and-Conditions-4.0.pdf

[17] T. (2015, February 20). Massive Open Online Courses (MOOCs) and the next generation | Alex Cui | TEDxYouth@UTS. Retrieved May 14, 2018, from https://www.youtube.com/watch?v=gbkeWebvW1M

[18] Alías, M. (2017, November 22). Selectividad 2017 - Profesiones truncadas: El 34,5% de los estudiantes abandona la carrera que elige. Retrieved June 8, 2018, from https://www.vozpopuli.com/altavoz/educacion/carrera-universidad-abandono-fracaso-grado-Selectividad-nota_0_1041196454.html

[19] Plan de Acción de Educación Digital - Educación y formación - European Commission. (n.d.). Retrieved June 10, 2018, from https://ec.europa.eu/education/initiatives/european-education-area/digital-education-action-plan_es

[20] Epelboin, Yves. Professor at UPMC. (2016, June 8). MOOCs: searching for a viable business model. Retrieved June 10, 2018, from http://www.eunis.org/eunis2016/wp-content/uploads/sites/8/2016/02/EUNIS2016_paper_12.pdf