UNIVERSITYBUDDY: A WEB APPLICATION FOR MENTORING AND HELP DURING ERASMUS AND OTHER STUDY ABROAD PROGRAMS

Calificación:

9.0

TRABAJO FIN DE GRADO
CURSO 2025-2026

AUTHOR
IAN ÁLVAREZ DE FRANCISCO

DIRECTOR
RAMÓN GONZÁLEZ DEL CAMPO RODRÍGUEZ BARBERO

DEGREE IN SOFTWARE ENGINEERING
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

UNIVERSITYBUDDY, A WEB APPLICATION FOR MENTORING AND HELP DURING ERASMUS AND OTHER STUDY ABROAD PROGRAMS

SOFTWARE ENGINEERING FINAL BACHELOR'S THESIS

AUTHOR
IAN ÁLVAREZ DE FRANCISCO

DIRECTOR
RAMON GONZÁLEZ DEL CAMPO RODRÍGUEZ

SEMESTER: FEBRUARY 2026

SOFTWARE ENGINEERING
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

8TH OF JANUARY, 2026

DEDICATIONS

To my family, who have supported me through everything and have always given me their unconditional support, through thick and thin.

To my friends and everyone who has been alongside me through everything, supporting me through it all over the last few years.

ACKNOWLEDGEMENTS

I want to thank everyone who made it possible to complete this project. Starting with my professors and my tutor who have given me all the tools necessary to create this.

I also want to thank everyone I've met since beginning this university degree, providing me with great friends and great support throughout everything, academic and non-academic alike.

I want to generally thank everyone who helped in one way or another, consciously or not, to help fulfil my goals and get me to this stage, it wouldn't have been possible without any of them.

ABSTRACT

UNIVERSITYBUDDY, A WEB APPLICATION FOR MENTORING AND HELP DURING ERASMUS AND OTHER STUDY ABROAD PROGRAMS

UniversityBuddy is an application designed to support students across their academic journey, from their university search onward. It allows students anything from user registration to profile management to the discovery and evaluation of universities as well as facilitating mentoring communication. The platform provides access to reviews, ratings, and community forums where students can share their experiences.

This application meets the need to centralize information and facilitate connections for students planning or experiencing international academic mobility. It provides a platform where future, current, and former Erasmus students can share experiences and rate universities across multiple dimensions, such as installations, university life, accommodation, academic level, and activities.

The system supports multiple user roles and includes features for mentorship groups, event scheduling, and availability management. It is not just an information repository, but an interactive community platform that bridges the gap between students at different stages of their Erasmus journey, fostering knowledge sharing and peer support across geographical and temporal boundaries.

Keywords

Web Application, Social Network, JavaScript, TypeScript, PostgreSQL, Project management, User friendly interfaces

Contents

FIGURE INDEX

Chapter 1- Introduction

1.1 Motivation

The first big question behind every project is "Why?" and behind this one the explanation is simple: because I would have loved to have access to a tool like this before and during my university experience, especially before, during and after my Erasmus Experience. The fact that there wasn't a unified platform where students could share their experiences in different universities and cities/countries baffled me. So logically the best option was to have this Bachelor's Thesis be based on creating an application which can offer a helping hand to those new students diving into new experiences.

Before being set on the idea, it was optimal to confirm that it wasn't a project inspired by personal bias. In order to get this feedback, conversations with some colleagues who had problems with the same situation were had, and this idea was pitched to them to get different perspectives from people in different stages of the process. These people were later interviewed individually so a deeper assessment of what the correct scope and definition of my web application would be.

Before starting the project this idea was discussed with the thesis tutor where he brought up an idea. He suggested that along with helping in the choice of university it could serve as an aid for the mentoring process which takes place in most universities where mentors often have a hard time organizing their mentoring groups. This was a great idea and it was decided to evolve this project into making this app a companion for your whole time at any specific university, a hub for student interactions and help during your studies through the use of forums with varied scopes.

After these discussions a general image of what the project would entail was generated: A hub for students who need input from other students whether they are new or not, centralizing mentorship organization to a single platform.

1.2 Objectives

Now that the general concept of the application was defined, the next step was to specify what the application would offer the user. The main objective is building a user-friendly, intuitive web application which is easy on the eye and can be easily understood by anyone independent of their tech-savviness. Students must be able to easily find information about their (future/current) university as well as facilitate a hub for everything related to mentoring. To achieve this objective it is necessary to set some more specific objectives:

- **Create a personalized search function** to be used by students to search for universities through filters.
- **Create a mentorship hub** where mentors and students alike can easily access information about any events as well as participate in specific discussion forums.
- **Create a landing page for each University, city and country** where users can easily access information about those places and participate in dedicated forums.
- **Create a place where new students can connect with others in similar situations**
- **Design a modern web application** using a production-grade modern stack which is applied throughout the tech world, applying best practices while coding, maintaining a clean and easy to understand codebase.
- **Make the application easily accessible** by facilitating an easy architecture to execute whatever the system the user is on.
- **Follow an organized development methodology** to guarantee a quality product and a well-organized end result.
- **Run database, backend and frontend services separately** to guarantee a quality product and solid project structure.
- **Include tests** to ensure the application conforms to the specifications.

1.3 Workflow Methodology

During the development of this project the methodology used was based on applying agile methodologies to keep track of work done and help with setting short-term tasks as a part of the bigger picture. Whether these tasks are part of the planification process or the development of the app, they contained a small description of what was needed and they were assigned to weekly "sprints".

Each sprint followed the same format, as these sprints were weekly, I will refer to the days of the week during which each "activity" was planned. A strict format was used to maintain the workflow during the whole development process and help create a quality product through the continuity this process provides. The following workflow was used:

- **Monday:** In the case of having a development task, a visual representation of what was needed was created using *Figma*. Any necessary research on possible new technologies to be added would also be carried out on Mondays, sometimes extending throughout the week if necessary. If none of the current tasks were development ones, carrying them out would be Monday through Friday.
- **Tuesday-Friday:** During the week, the mass of the workload was carried out, whether it be developing new functionalities, tinkering with the look and feel of the frontend, or editing the database structure to adapt for changes, they were carried out in this timeframe.
- **Saturday-Sunday:** The vast amount of organization was carried out during the weekend. Firstly, any task that was deemed "Done" would be tested and considered finalized if the result was satisfactory. Secondly, the state of the project was analyzed to determine what tasks had to be included in the following week's sprint. Following this analysis, the goals for the following week were clear and the objectives set.

During this whole process, the most important factor was organization. To achieve this organization, it was key having a platform where all the information relating to these sprints was centralized. To achieve this, *Notion* was used as an organization tool. Inside *Notion* the specifications for the whole app were defined so it could be referenced during the development process. There was also a "Kanban Board" where all the tasks were shown together with the state they were currently in (Not Started, In Progress, Testing and Done), the type of task (Research, Planning, Development, Documentation) as well as what week that task started and when it was considered "Done".

This structure that was followed not only helped maintain a correct workflow and continuity leading to an organized development process. It also helped with the documentation of the overall process that was followed so it could later be easily represented.

Alongside the week-by-week workflow, there were a couple of key dates that were set beforehand to guarantee that a satisfactory version of the web application could be achieved by turn-in date, without the need to rush the development process to get to the finish line on time. In the following section, those key dates, the set objectives and the sprints will be explained. There will also be a recap of each sprint and what was carried out during each one.

1.3.1 Planification

Before the project started, there were three key objectives that were estimated before the project started. These three objectives were the following:

- **Define Project Specifications:** By the 15th of September the goal was to have a solid project specification. This means that users types, use cases and what stack would be used was to be defined.
- **Have a functioning MVP:** By the 13th of October the goal was to have a functioning MVP (Minimum viable product). What that MVP would contain was defined during the planification process, and in the case of this application, it meant a functional frontend where a user could search for universities by country and city filters. All of this while maintaining an

application design that would serve as a "backbone" for the rest of future functionalities to be easily built on top of this foundation.

- **Final Application built:** By the 8th of December the goal was to have a fully functional application in order to be able to document the process accordingly before the 8th of January turn-in date.

With these objectives set, the project was ready to begin, and with it came the first sprint. In the following section I will shortly resume every sprint that was carried out with its title and a few bullet points on what objectives were set for each one.

1.3.2 Workflow

- **Week 1(01/09-07/09): Research**
    - Research and decide which backend, frontend and DB technologies would be best for this project.
    - Decide who to interview and decide which questions are best for each case.
- **Week 2(08/09-14/09): Define Specifications**
    - Define project specifications, users and use cases.
    - Interview possible users of the application.
- **Week 3 (15/09-21/09): Design Database**
    - Design the database using a Relational Entity diagram.
    - Build the initial database structure using *Prisma*.
- **Week 4 (22/09-28/09): Build backend**
    - Create backend structure and project architecture.
    - Investigate necessary endpoints for MVP.
- **Week 5 (29/09-05/10): Finish backend and begin frontend development**
    - Create necessary endpoints for backend.
    - Begin creating landing page.

- **Week 6 (06/10-12/10): Finish MVP**
    - Create university search frontend.
    - Test all initial functionalities.
- **Week 7 (13/10-19/10): Add Authentication + Log in**
    - Investigate *bcryptjs* library for authentication
    - Implement authentication logic throughout the application.
    - Add Registration and Log in screens
- **Week 8 (20/10-26/10): Add University Rating feature**
    - Create a form to submit ratings and build the logic behind it.
    - Add stars to the main University page to make it feel more appealing to users.
- **Week 9 (27/10-02/11): Mentor hub + Event management**
    - Create a hub for mentorship events with a calendar and info for each event.
    - Add mentorship group logic with distinguished groups with their assigned mentors.
    - Add event logic with public and private events.
- **Week 10 (03/11-09/11): Search by ratings**
    - Add the possibility to filter university search by user ratings.
- **Week 11(10/11-16/11): City and Country Information**
    - Add dedicated pages for countries and cities with forums on each page.
    - Add map to city page.

- **Week 12(17/11-23/11): User Profile**
    - Create an intuitive user profile where all information is centralized
    - Add the possibility to edit Erasmus status
    - Integrate event Calendar
- **Week 13(24/11-30/11): Contact Search**
    - Add the possibility of looking for students in similar Erasmus situations who are open to contact.
- **Week 14(01/12-07/12): Admin Dashboard**
    - Create a dashboard for university administrators to assign students and mentors to mentorship groups.
    - Give administrators the possibility of adding new universities, cities and/or countries.

As was defined in the initial goals, by the 8th of December, there was an application that satisfied the expectations set before the development began, thus marking the beginning of the thesis writing section. This section spans exactly one month, up to the 8th of January. It all started with a meeting to discuss the state of the application and if the necessary requirements were met. After agreeing that the application was in a good state to turn in and the writing process could commence.

Chapter 2 - State of the Matter

This chapter contains a description of apps which provide similar functionality to that of the application being built. In order to understand why this application is necessary, it is also important to understand what tools students have at their disposal during these indecisive moments in their academic lives. This analysis consists of explaining what these different apps offer, separating advantages from disadvantages and explaining what *UniversityBuddy* could offer to improve the experience.

Apart from explaining what similar applications exist, this section will include interviews with possible users that find themselves in different stages of their academic lives and know what tools they had at their disposal to achieve their goals and know how *UniversityBuddy* could fill that gap that exists for students in these positions.

2.1 Similar Applications

In this section, a brief description on some similar applications will be given. These applications served as lessons as to what works and what doesn't, to get an insight into what should be the focal points when building the application.

2.1.1 RateMyProfessors

RateMyProffesors [1] is an application based in the United States where students evaluate their universities and professors. This application is widely used by students across the whole United States, although due to the way the American university system works, it is only used there and not anywhere else as it is fitted to the way these universities work. *UniversityBuddy* is heavily inspired by this concept, where students can decide what university interests them the most, not from an academic perspective, but based on other students' input. For the construction of *UniversityBuddy* the rating concept was introduced but only applied to universities. Along with that, as it is mainly directed to new students who only want to know about the university and not for current students looking for what classes to take it was considered to be a disadvantage to have such a large

constant user base which could clutter the forums with experienced students information and "scare off" new students which would contradict what the objective is.

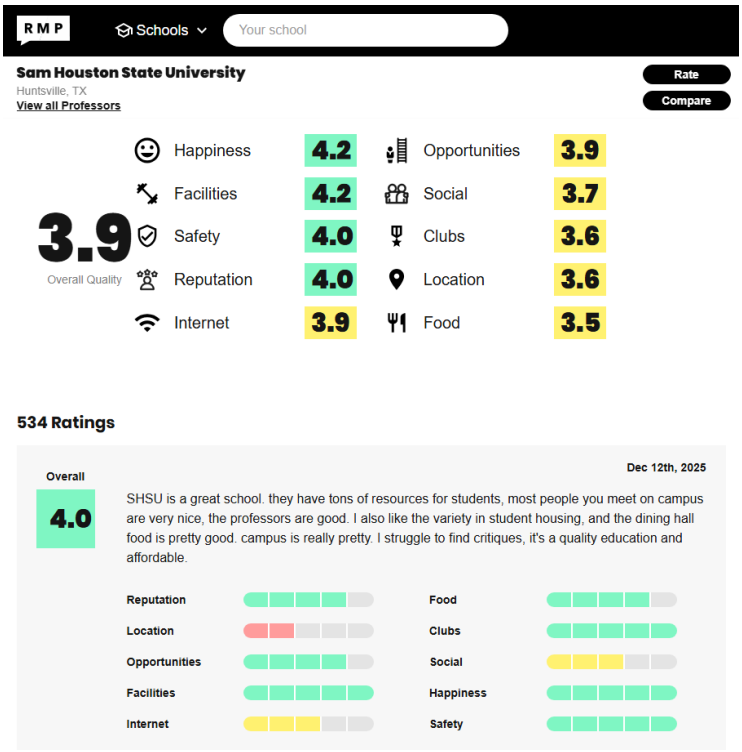

Figure 2-1. University Evaluation Screen on RateMyProffessors

2.1.2 Educaweb

Educaweb [2] is a Spanish platform developed with the sole purpose of helping Spanish students know what grade they might need to get in their university access exams to be able to realistically opt in to the degree they want to study. This application offers enormous information and very specifically, making it the best centralized platform where students can learn about their options and how realistic these are. This platform contains all the necessary information for anyone looking to study in Spain, as it contains information about prices, access grades, study plans, etc. *UniversityBuddy* is conceived as an additional tool to add to this one during your search because of the lack of user input contained in this application as it is based solely on academic and administrative information. Another downside that *UniversityBuddy* seeks combat is the fact that it is limited to education in Spain, while the reality is a large amount of students from Spain

(and other countries) seek different locations outside of their home country where they would like to study and knowing what options are available is something useful that this application doesn't provide.

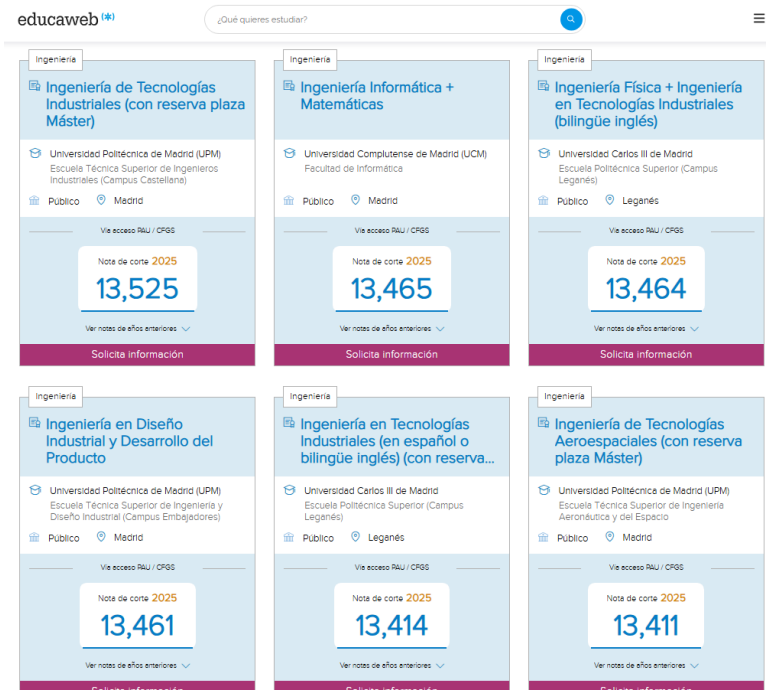

Figure 2-2. University Search on EducaWeb

2.1.3 Uniranks

Uniranks [3] is another application like *Educaweb* but at a larger scale, including countries from all over the world. The major downside for students looking for information on this webpage is that it isn't orientated solely towards students. This application ranks universities based on academic level and metrics such as "Awards & Recognitions" and "Academic & Research" which aren't the most influential metrics to most students. Apart from that, the ratings aren't transparent and their statistics for most universities appear as empty, showing that their user base doesn't seem interested due to the nature of the information provided by them. Although there are major downsides, this application served as a large inspiration for the construction of *UniversityBuddy*'s frontend, as the

design was simple, easy to navigate and all the information that was deemed relevant was easy to understand.

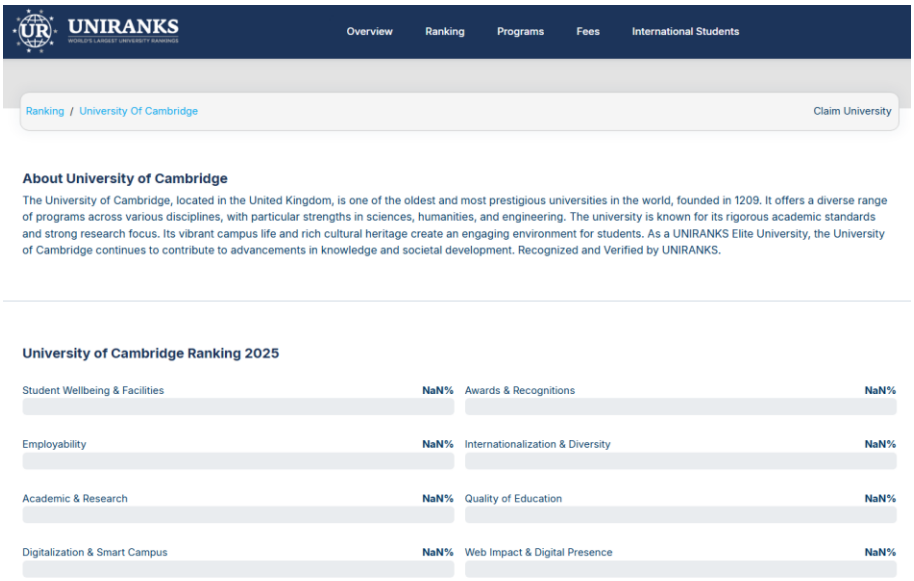

Figure 2-3. Uniranks University Screen

2.1.4 UCM "Experiencias Erasmus"

In the Universidad Complutense de Madrid's official webpage you can find a section dedicated to showing different students' experiences in different Erasmus cities with a short text describing them [4]. Apart from that, before a student begins their Erasmus experience, Universidad Complutense facilitates the student with the contact of any students with the same destination and the contact of some previous students who went to the same destination. The section showing some experiences is good for students looking for some insight into what their experience may look like without going too in depth. The way of providing contacts by the university was a big inspiration during the development of this app as well. The fact that these contacts with similar experiences who can help giving you some guidance before or during your stay, can be an amazing piece of help.

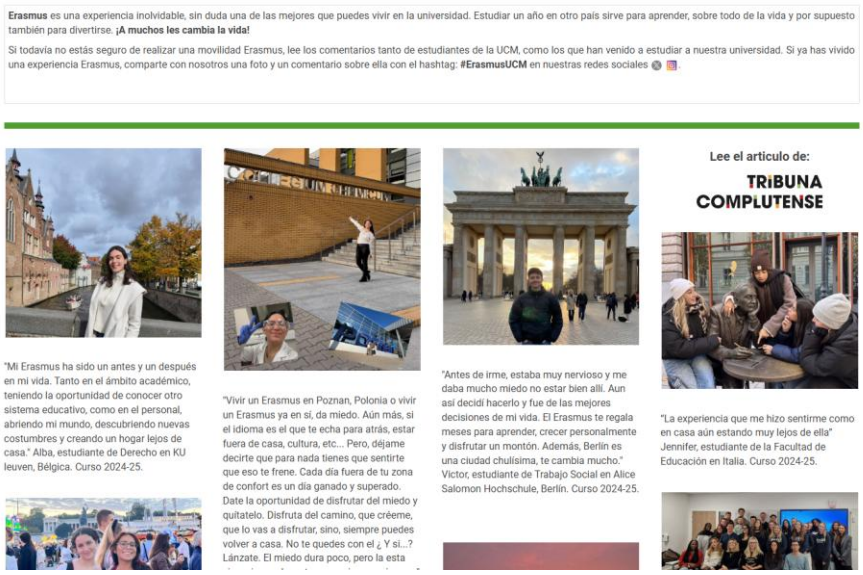

Figure 2-4. Experiencias Eramus section of UCM webpage

2.2 Interviews

Before conducting these interviews there was a planification stage, where the questions and interviewees would be decided. The goal was to get points of view from students in different stages of their international exchange program and from mentors, which would be the majority of this application's user base. For the different points of view from people that participated in international exchange programs there were three main points of view which seemed the most relevant: during the destination election process, during their exchange and after their exchange. In terms of mentors there were two mentors who had participated in the mentorship programs the "Facultad de Informática de la Universidad Complutense de Madrid" as mentors.

2.2.1 International Exchange Interviews

These interviews were all very similar, with an incrementing number of questions for those students who were more advanced in the exchange process. This enabled the different perception on the same question from those who had already come back home after exchange with those who hadn't yet chosen where to go and those who were currently living in their selected location. The interviewees were the following: Pedro Coll (Currently seeking Erasmus destinations), Miguel Fernández-Pacheco Portillo (currently on international exchange), Arturo Navarro Muñoz (Completed an

international exchange program). These will be represented as P,M and A respectively. The following section contains the questions and answers from these interviews:

Were you able to speak with other students who had been on exchange in the same location? If so, where did you obtain their information?

- P: More or less, I spoke with a friend who was in Poland and she recommended a different city than the one she went to. I also spoke with another girl from my class who's in a third city in Poland. I don't know anyone who has studied at most of the universities I applied to, but I do have some feedback about the countries and cities.
- M: Yes, the university website asked for my permission to share my data with people from my program who had been to the same destination before. That way I was able to contact them, and they gave me the information I needed.
- A: No, I couldn't find anyone

What was the main source of information about the cities/countries you were interested in?

- P: Friends, I was too lazy to search online, although I did check several forums. But almost all the information came from friends. First, I decided on the countries (based on prices, university difficulty, and ease of travel) and then the cities/universities. The first thing I did was rule out France and Germany, then I also considered the food. In any case, I obtained almost all the information from people who had been there before.
- M: Mainly my prior knowledge about each country, I wanted a destination in Central Europe and from there I asked acquaintances who had been on Erasmus in that area to give me their opinion.
- A: The PDF they gave us about the destinations, and then online Wikipedia or using Google Maps. I would go to the websites of the universities themselves, and I would also look at people's opinions on different websites.

How confident do you feel on your decision based on the amount of information available to you?

- P: Very little. That's why I preferred to rely more on curiosity than on knowing things because I can't really trust it since I don't have much information. Also, I left it until the last day and kind of did it randomly.
- M: Before going to study abroad, I wasn't very confident and had many doubts about the information I had. Now that I'm here, I know I made the right decision.
- A: I honestly feel like it was a bit of a blind choice, and if it happened, great.

Did you feel there was a lack of information at your disposal before your exchange?

- P: Yeah, it would be key to have a forum or some platform where people could leave reviews because it doesn't cost anything to leave a comment and it would help the next ones a ton.
- M: It would be good to have more talks or information points about the destinations, although I understand that if there are many, it can be somewhat complicated.
- A: It wasn't a lack of information, but rather misinformation. I remember that my destination information said I was in a city when in reality the destination was somewhere else, an hour away by bus.

Once on exchange, did you receive any help adapting to the city/country? Who gave you that help?

- M: Yes, the host university held talks and activities during the first week to learn about the city and topics related to the degree program; they were very helpful.
- A: Yes, the student association at the host university was very helpful. they had people called buddies, and each person had one assigned to them. They helped us a lot with anything we needed.

Do you feel like if you had more information you would have selected a different location?

- M: No, I've ended up in a place I really enjoy
- A: Yes, probably

2.2.2 Mentor Interviews

For these interviews, the intention was to get a broad view of how mentorship works in the *Universidad Complutense de Madrid* to get a feel on how integrating mentorship capabilities into the application could work. This would provide a broader scope for the application as university students who are seeking to apply for university will already have knowledge on the application and will be able to use the same one for mentorship activities in their first year, without the need for changing the place where they obtain university related information. The two tutors interviewed were Amaia Echenagusia Muñoz (referred to as "A" in the interviews) and Raúl Hernández Alvarado (referred to as "R" in the interviews).

Which platforms did you use to organize your mentorship group?

- A: WhatsApp and Google Forms
- R: I used Gmail; the mentoring coordinator at our faculty gave us the Gmail addresses of all the students, and then I asked them for their WhatsApp numbers so I could talk to them more easily that way.

Do you think a centralized application would be of any help?

- A: Yes, it would be much easier to organize the availability of the entire mentoring group, as well as the interests and doubts of each member of the group.
- R: It's possible, although I don't know to what extent. For example, for me, in my fourth year at university, yes, but a computer science student who has just started might find another platform or website overwhelming, considering Gmail, the campus website, the courses, the practical assignments...

Which were the main difficulties you encountered being a mentor?

- A: The organization of the topics to be covered, not knowing completely which things were more important to talk about and which were not of such interest at that moment, since they are first-year students
- R: The students weren't very enthusiastic, and the schedules were bad too. They gave me students that didn't fit my schedule, nor theirs. Ideally, we would all have

the same schedule, all in the afternoon or all in the morning, so we can all meet up, and so they can get to know each other and support each other.

Did you have meetings? If so, how were they organized and how was attendance managed?

- A: We had weekly or bi-weekly meetings, organized with a survey on the WhatsApp group, and we knew in advance who would attend based on the survey responses, although even so, someone always failed to show up.
- R: If I had meetings, I would organize them by email or WhatsApp, although there came a point where it was no longer necessary and to avoid wasting both of our time, we would clarify everything by WhatsApp or Gmail.

Who assigns the mentorship groups?

- A: The mentor Coordinator from the university.
- R: The mentor Coordinator from the university.

How big was your mentorship group?

- A: Normally there are 5 students per group but in my case there was an exception and there were 10 students.
- R: 4 students

Are mentors only accessible to first year students?

- A: They are aimed at first-year students because they are more lost, but participation was not prohibited to anyone.
- R: For people entering the university, for example, a Palestinian student entered mid-course and also gained access to mentoring.

What would you improve from this program?

- A: It would have been great to have a website with a more specific guide on what to cover in each meeting and where to find more information on the most important topics, because the training we received was more focused on group management and some general information about the university, but it lacked a

bit more explanation on how to organize the whole program and how to find the right information.

- R: Yes, although it's difficult, the important thing is that the students are willing, and to be honest, that's difficult because I understand that they're turning 18 and the last thing they want to do is meet up with me and all that.

2.2.3 Conclusions

After conducting these interviews, many conclusions were reached. In terms of international exchanges, it's clear more information is needed and, in some cases, it can result in students being unhappy with their decision largely due to the lack of accessible key information. What's also evident is that students who went on international exchange programs are more than happy to help future students. Not only do students care about other students' opinions, but they also want information about the city and country, not only on what the university may offer them. This is why *UniversityBuddy* seeks to centralize all this information in one place. Most students had some contacts given to them by the university, as was my case too, but this is a limited pool, as some destinations are rarer and not visited every academic year, limiting the amount of contacts that the university can provide. This led to the idea that having the possibility of contacting more people who went there and are not necessarily from your university but from your city or country could widen that user pool and provide better contacts for these uncertain students.

In terms of mentorship, it's clear that more platforms to keep track of aren't helping students at all. This is why having the same platform, where students can read others' opinions before applying and once enrolled, access forums about university life to help guide them, would be very useful. If that same application also includes a portal where mentorship information is contained, it would help reduce this agglomeration of tools and help reduce that overwhelming feeling for these new students.

From these interviews, not only did many of the functions of the application arise, but it also helped develop a broader image of what it can provide students. It can serve as a tool that accompanies students throughout their whole university life. It starts with seeking user feedback on universities, then helping throughout the adaptation process, then helping other students and in the case of wanting to participate in an exchange

program, it provides information about possible exchanges and provides relevant contacts that could be helpful.

Chapter 3 – Technologies/Tools Used

During the development of this project many different technologies were used. Which technologies would be used were mostly decided before any lines of code were written, based on both personal experience and through research. Most of this research was done by searching for modern web application stacks and how they are organized, as well as help from various videos made by developers explaining simple concepts and showing how to begin a project with a similar architecture.

To organize this information, it will be provided in three different categories. Frontend, for those which were helpful in the construction of the user interface. Backend, for those technologies related to interacting with the database and managing user data, as well as managing the application logic. Lastly there is a general section, where technologies and tools that apply to the entire creative cycle will be mentioned.

3.1 Backend

3.1.1 PrismaDB

For the construction and development of the database, Prisma was used as the ORM. An ORM is a layer that maps database tables and rows to objects and methods in your programming language. The choice was made to use Prisma because Prisma Client can be used in *any* Node.js or TypeScript backend application [5]. As the application backend is built with Node.js this was an easy choice, and it helped the development process. Since Prima is oriented towards Node.js applications, and integrates very well with JavaScript, it made interaction with the database simple and effective. It also served a large utility when testing as Prisma also comes with a mockable API which can be used to verify the applications functionality against expected database interactions without having to interact with the database.

Figure 3-1. Prisma Logo

3.1.2 PostgreSQL

The choice to use PostgreSQL is directly linked to the choice to use Prisma as an ORM. This made database changes and migrations easier to perform as well as maintaining the relational structure that was intended. Although it didn't affect the application in its current development state, PostgreSQL was the best option when considering how scalable this application could become and its optimizing capabilities were an important factor in this decision.

Figure 3-2. PostgreSQL Logo

3.1.3 Node.js with Express Framework

As said in the description of why *Prisma* was used, Node.js and Prisma go hand in hand, and these integration capabilities facilitated the decision to use Node.js with the *Express* framework. Node.js also offers high scalability for the future of the application and provides a solid backbone for a larger application that could come as a result of this one. The other main advantage was the *npm* ecosystem, which offers a very large variety of tools and libraries that can help in the development process, some of which were used in this application, with the main ones being described in the following subsections.

Figure 3-3. Node.js Logo

3.1.4 Jest

Jest is a JavaScript testing framework which can be widely used for testing applications developed with Node.js. In this project it was used to test the backend with

automatic tests. In this project it was used for validating API routes and services while mocking the Database using Prisma's mocking capabilities to test these database interactions in isolation. The reason this library was chosen was because of how easily it integrates with Node.js and how easily it was to execute these tests without the need to deploy the application.

Figure 3-4. Jest Logo

3.1.5 bcrypt.js

Bcrypt.js is another *npm* library that was used in this project. This library provides password hashing and validation, which was very useful when developing user authentication and then made validation for many API requests easier as the baseline that was used was already developed when implementing user validation. As for earlier components, the choice to use this technology was easy due to its integration with *Node.js* that made it easier to implement.

3.1.6 Docker

Docker was used in this application to provide lightweight containerized environments. Although this was not necessary for the current application, it provides a good backbone for a production-grade web application, facilitating a possible deployment automation mechanism. It also provided ease in developing this application on different machines, due to the fact it isn't necessary to install the libraries needed locally but instead when building these containers these libraries and frameworks are installed on the containers, facilitating installation and execution of the application. One container was used for the database and another for the backend, meaning these could be deployed independently and any database updates don't affect the backend at runtime and vice versa.

Figure 3-5. Docker Logo

3.2 Frontend

3.2.1 React & Next.js

React was used in this project to build the user interfaces. The main drawing point was the component-based nature of this framework which was heavily taken advantage of during the development of this application. Creating re-usable components helped maintain a constant look and feel to the application while reducing the amount of copied code that needed to be used when implementing these different user interfaces.

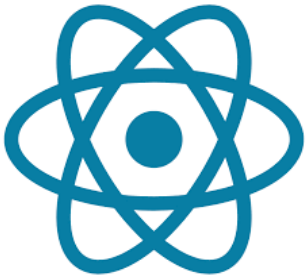

Figure 3-6. React Logo

In the case of this project, React is used alongside Next.js which offers server-side rendering capabilities and dynamic routing, where the configuration of routes is directly reflected in how the file structure is shown in the project. It also provides state management to manage the state of variables when using interactive features such as forms and API calls, reducing the amount of unnecessary reloads that might have been present with other frameworks.

Figure 3-7. Next.js Logo

3.2.2 Tailwind CSS

Tailwind offered this application styling facilities without the need for too many CSS files, offering global styling and utility classes that are made available throughout the whole project. Component styling was also used heavily by defining certain styles needed through a components' "className" instead of having to resort to creating CSS entries for unique components. All these reasons made the use of *Tailwind* very justifiable, augmented by the fact that it is very well integrated with *React* offering an easier development process.

Figure 3-8. Tailwind CSS Logo

3.2.3 Axios

Axios is the choice of HTTP client which was used in the frontend of this project. Every request to the backend API endpoints were carried out using this *npm* library. The reason *Axios* was used was, similar to most technologies in this project, to how well integrated it is within a *Next.js* application, offering a sure-fire solution to making HTTP requests from the frontend.

Figure 3-9. Axios Logo

3.3 General

3.3.1 JavaScript & TypeScript

JavaScript and *TypeScript* were the two programming languages used throughout this application. *JavaScript* is mainly used in the backend, due to the nature of *Node.js* and how well it integrates with *JavaScript*. Meanwhile *TypeScript* is used mostly in the frontend, although it could have been implemented in *JavaScript*, *TypeScript* offers a better structure and more reliability, as well as offering more scalability, which would be very beneficial in the case that this application had a large concurrent user base. Together, both these languages combined offer a modern, scalable approach to developing web applications, widely used in the industry nowadays.

3.3.2 Figma

Figma was the main tool used to bring user interface concepts to life before implementing them in the actual application. It is a design tool specifically built for user interfaces, and by using their local libraries, It was used to develop a visual prototype before developing the application's frontend. During the planification phases of the "Sprints", it was also used to add new elements to the original prototype, helping to grasp at what that new element would look like and how it fit in with the rest of the application's look.

Figure 3-10. Figma Logo

3.3.3 Loveable

Loveable is an Artificial Intelligence tool, which creates landing pages for web applications. This was used as a tool to gather inspiration as to how some of the main pages in the web application could look like. Since it generates these pages using a React-based frontend, inspiration not only for how the application could look but also to how the elements could be made reusable was a source of inspiration.

Figure 3-11. Lovable Logo

3.3.4 Notion

Notion is an organization tool, which was widely used throughout the development of this app. As explained in the "Workflow" section from Chapter 1, this project was organized in weekly sprints. *Notion* is where the backlog was kept and updated as well as where the "Kanban" board was constantly updated whenever a tasks status changed or a new one was added. Not only was it useful in the day to day, but it was also important during the planification phase. *Notion* is where all the applications use cases, actors and notes about possible technologies to be used was kept. To sum it up, it served as a hub for all information related to the development

process, where any doubts regarding the specifications or state of the development process could be resolved by consulting *Notion*.

Figure 3-12. Notion Logo

3.3.5 GitHub

Github is a collaborative platform for hosting Git repositories. This was where all the code relating to the project was contained. This helped in being able to access the code from any device authenticated to access the repository and providing extra security as opposed to only having the project saved locally on one machine which could cause the whole project to get lost. *Github Desktop* was used to easily commit and visualize changes easily when developing. For future work that can possibly be done on this application, *Github* also provides CI/CD tools for automatic deployments which are made easier through the dockerized structure the project contains already.

Figure 3-13. Github Logo

3.3.5 Postman

Postman is an API platform that offers the possibility to make HTTP requests. It was used throughout this project to test backend endpoints and be able to execute operations in a fluent manner without the need for a frontend implementation for those operations. This was extremely useful at the beginning of the development process to ensure the backend was correctly working before its corresponding frontend components were developed.

Figure 3-14. Postman Logo

Chapter 4 – Development

This chapter is dedicated to explaining the development process of this application and detailing how it is structured. The overall structure of the project will be

detailed along with a more detailed explanation of the most important elements which provide the most value for this project. This will be divided into two sections: backend and frontend.

For this project, a modern approach on the Model-View-Controller design pattern. The model manages the business logic and communicates with the database. The view represents the user interface and displays the data to the user while forwarding user commands to the controller. Lastly, the controller manages user input and acts as an intermediary between the model and the view.

4.1 Backend

The backend is organized to act as an API, providing the necessary endpoints to conduct any operations considered necessary to the functioning of the application. Examples of these necessary operations are creating universities, modifying user data, creating user-mentorship group relations, etc.

During the development of this project, Docker was used to deploy both the backend and the database separately by using two separate Docker compose files for each. To work on the backend the only necessary action was to deploy these containers and access the local ports associated with each one (3000 for backend and 5432 for PostgreSQL database).

4.1.1 Directory Structure

The directory structure is built according to Node.js best practices and uses the basic template that is applied whenever Node.js is installed into the directory. The following image contains the structure mentioned along with an explanation on every significant component that appears:

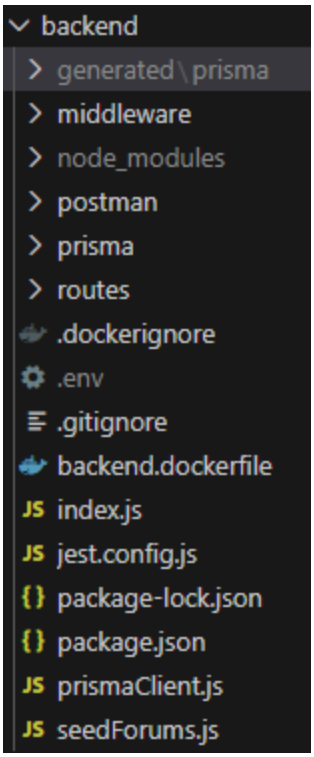

Figure 4-1. Backend File Structure

- **Generated/prisma:** contains all necessary files for Prisma to be able to operate as an ORM for this application
- **Middleware:** Contains the authorization middleware which is used for user log in and authorization.
- **Node_modules:** Contains all dependencies needed for this project.
- **Postman:** Contains a postman collection and environment to test all backend functionalities.
- **Prisma:** Contains the prisma database configuration file, in which the relational database is defined.
- **Routes:** Contains the JavaScript files for each route which contain the logic behind every endpoint.
- **Backend.dockerfile:** Configuration file for Docker indicating what actions it must perform when building.

4.1.2 Logic

For this section, as this backend intends to act as an API, all endpoints will be listed and a short self-descriptive title will be added along with the type of HTTP request it carries out. Each one of these sections are defined independently inside express routers, meaning the separations made on this list directly correlate to the express routers used in the backend of this project. A *Postman* collection with all the endpoints is included in the project as it serves as a great helping hand for accelerating certain use cases, especially useful during testing. This following section contains the router title and all its endpoints listed under it:

- **Authorization**
    - Get Profile (GET)
    - Login User (POST)
    - Register User (POST)
- **Users**
    - Get all users (GET)
    - Get User by ID (GET)
    - Get Users by University (GET)
    - Create User (POST)
    - Update User (PUT)
    - Delete User (DELETE)
- **Universities**
    - Get all universities (GET)
    - Search universities with query params (GET)
    - Get university by ID (GET)
    - Create university (POST)
    - Update university (PUT)

    - Delete university (DELETE)
- **Cities**
    - Get all cities (GET)
    - Search cities by query params (GET)
    - Get city by ID (GET)
    - List city's universities (GET)
    - Create city (POST)
    - Update city (PUT)
    - Delete city (DELETE)
- **Countries**
    - Get all countries (GET)
    - Search countries by query params (GET)
    - Get country by ID (GET)
    - List country's cities (GET)
    - Create country (POST)
    - Update country (PUT)
    - Delete country (DELETE)
- **Erasmus**
    - Get Erasmus' by user ID (GET)
    - Get Erasmus' by university ID (GET)
    - Get relevant contacts through query params (GET)
    - Create Erasmus assignment (POST)
    - Update Erasmus assignment (PUT)
    - Delete Erasmus assignment (DELETE)
- **Events**

    - Get all events (GET)
    - Get available events for user by ID (GET)
    - Get event by ID (GET)
    - Get event attendees (GET)
    - Create event (POST)
    - Attend event (POST)
    - Unattend event (POST)
- **Reviews**
    - Get reviews by university (GET)
    - Get reviews by user (GET)
    - Get University average ratings (GET)
    - Create review (POST)
    - Delete Review (DELETE)
- **Forums (Same for University, City, Country and Mentor Group)**
    - Get topics (GET)
    - Get a topic and its posts (GET)
    - Create topic (POST)
    - Create post for topic (POST)
- **Mentor Availability**
    - Get mentor's weekly availability (GET)
    - Create availability slot (POST)
    - Update availability slot (PUT)
    - Delete availability slot (DELETE)
- **Student-Mentor relationships**
    - Get student-mentor relationships using query params (GET)

- Create student-mentor relationship (POST)
- Delete student-mentor relationship (DELETE)

- **Mentorship Group Management**
  - Get all mentorship groups (GET)
  - Get mentorship group by ID (GET)
  - Get admin dashboard statistics (GET)
  - Create mentorship group (POST)
  - Add member to mentorship group (POST)
  - Update mentorship group information (PUT)
  - Remove member from mentorship group (DELETE)
  - Delete mentorship group (DELETE)

4.1.3 Database Structure

Having seen how the API is designed, the database was adapted as this backend was being developed, meaning it goes hand in hand with how the API is structured. The database provides everything necessary for this backend. In the appendixes of this document there is a link to the project's *Github,* where there is a *DbDiagram.io* file where a relational-entity diagram of the projects database is included.

4.2 Frontend

The frontend of this project is the main part where all the actions mentioned in the backend section come into play. All the components that comprise this section are included in the following chapter, so this section will cover how the elements and code are organized, while the actual elements and what they aim to achieve will be covered in the following chapter.

4.2.1 Directory Structure & Implementation

This project's frontend was developed with Next.js, using its project structure meaning the folder structure directly correlates to the different URLs that need to be used

to access every element. Below all these elements are portrayed by their URL terminations, as well as a list of the reusable components along with a brief description as to what every component provides the application with:

- **Reusable components**
    - **Forum Topic:** Everything inside a forum topic: topic title, comments, new comment box, etc.
    - **Forum Section:** Everything in a forum's main page: topics, create new topic button, etc.
    - **Erasmus Section:** Part of the user profile screen where user's Erasmus info and buttons to modify it are included.
    - **Event Calendar:** Calendar where all relevant events for a user are represented.
    - **Event Form:** Form for creating new events
    - **Navigation:** Navigation bar with quick links to the most relevant sections depending on the user's role.
    - **Uni Cards:** Cards that show brief information about a university.
    - **Uni Info:** Where basic information about a university including its ratings and mentors is shown.
    - **Uni Rating Form:** Form for star ratings to be applied to any certain university.
    - **Uni Search:** University search screen backbone.
- **/admin:** Admin Dashboard
    - **/universities/add:** Add new universities/cities/countries
- /**city/[id]:** City information screen
- **/country/[id]:** Country information screen
- **/erasmus:** Relevant contact search
- **/login:** Log in screen

- **/mentor:** Event calendar and information
    - **/event/[id]:** Specific event information
    - **/group:** Information about a student's mentorship group
    - **/profile/[id]:** Mentor's profile
- **/register:** Registration page
- **/university/[id]:** University information page
- **/user:** User profile
- **/:** Landing page for all logged in users

Chapter 5 – User Experience

Having detailed all the technical information in the earlier chapters, this section will explain what the end user sees and can achieve with this application. These experiences will be shown in order, varying by their degree of exclusivity: starting with what any user can see, whether logged in or not; followed by what registered students can access; then the additional features available to mentors; and finally, the screens reserved for administrators. In other words, every user will have access to their section plus everything in the preceding sections, but nothing from the ones that follow.

5.1 All Users

These following functionalities are those which are available to all users, whether logged in or not, with the caveat that they can neither rate universities nor participate in any forums although they are able to see the average ratings and read all forum entries. When accessing the webpage, they will be shown the university search screen, and the navigation bar will only show a link to the search page.

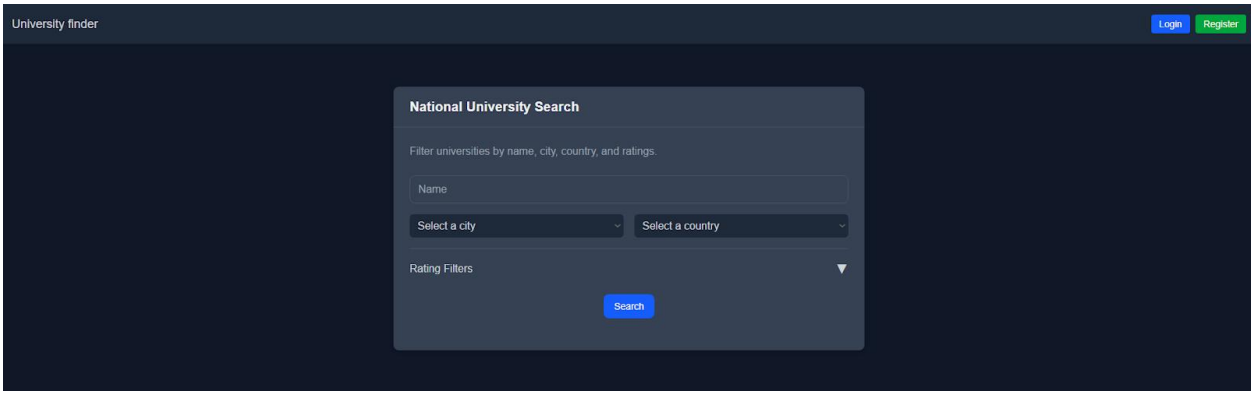

Figure 5-1. University Search Screen

5.1.1 Login

For ease of access, the log in screen is as simple as can be. Any registered user can log in with their email and password. This screen also includes a link to the registration page for any user who isn't yet registered and wishes to do so.

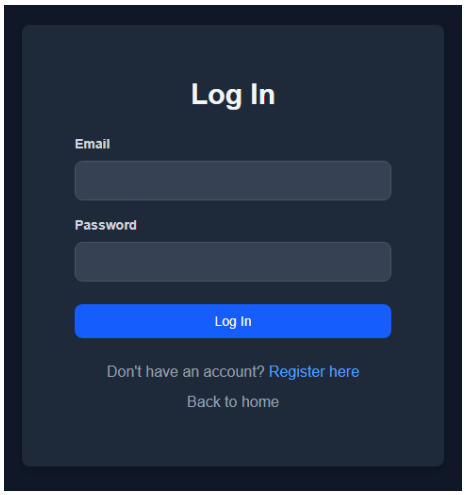

Figure 5-2. Log In Screen

5.1.2 Registration

The registration screen is for any user who wants to sign up for *UniversityBuddy*. This registration screen is a dynamic one, where more options pop up if a user is a student to select their university. If that student also has Erasmus experience, they can select the Erasmus checkbox and select the university where it took place and which academic year it was in. This registration screen also includes a privacy option which is directly related to the relevant contact search, where if a user doesn't want to be able to be found in searches, they can untick the box and add that extra layer of privacy.

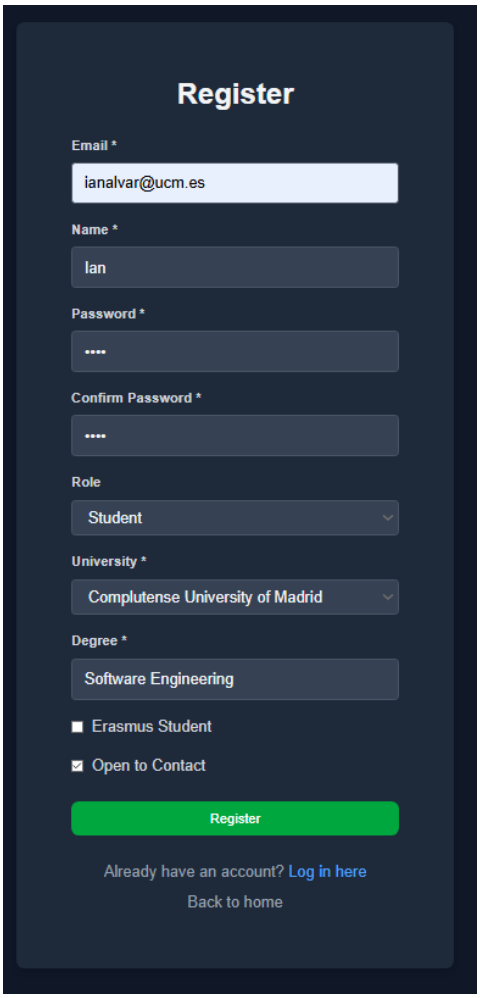

Figure 5-3. Registration Screen

5.1.3 University Search

The university search functionality provides any user with the possibility to search for universities, applying certain filters, which can be combined. The possible filters you can apply are by city, country or by maximum and minimum ratings you're looking for in a university. These ratings are classified in different types, which are also shown on a university's main page and, will be described in the following section.

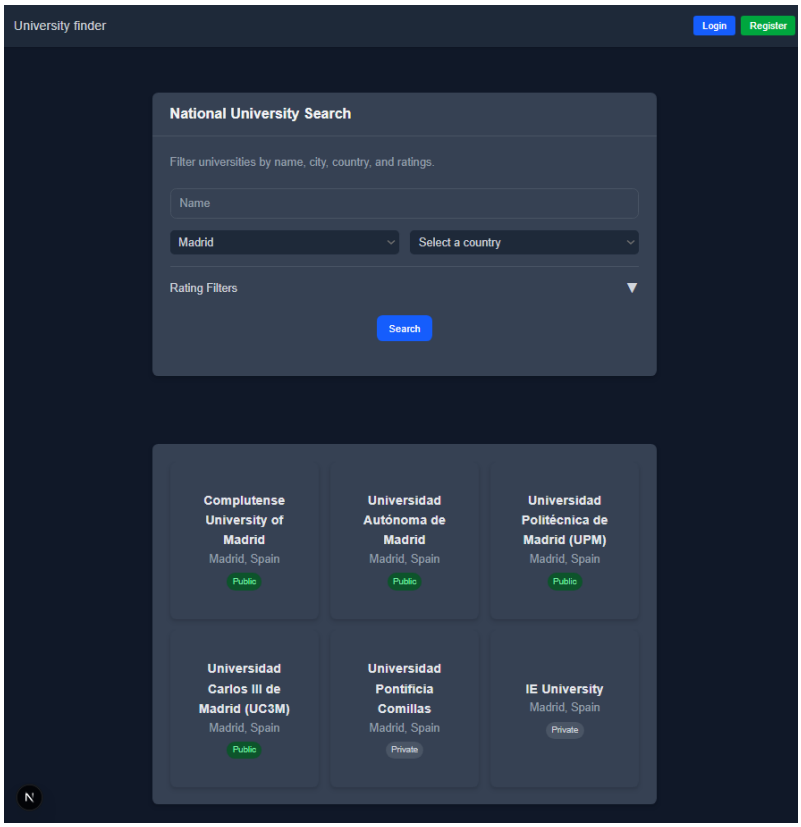

Figure 5-4. University Search in action

5.1.4 University Information Screen

After searching for a university and clicking on it, it will redirect users to the main university page. This page contains basic information of the city and country as well as links to the corresponding information screens which are described below. If you are a student of said university you have the option to rate the university, grading it based on these 6 metrics: Overall rating, Facilities, University Life, Accommodation, Academic Level and Activities. The average ratings from all students are also displayed on this screen.

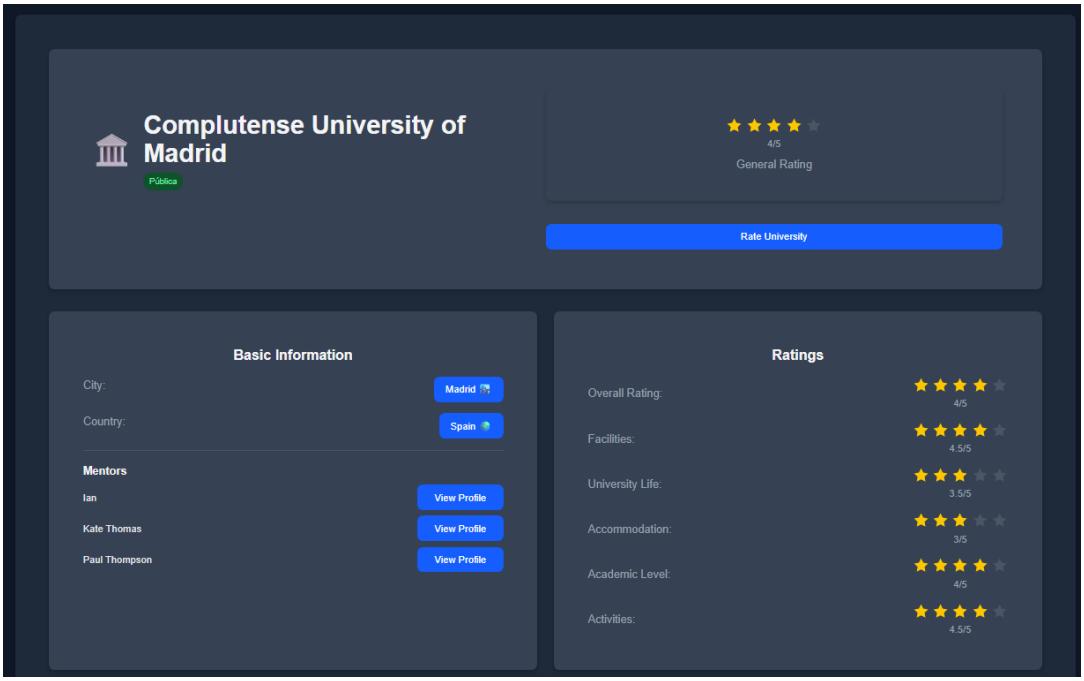

Figure 5-5. University Information Screen

At the bottom of these sections is a forum section, where users can ask about relevant topics that can be organized into different types of questions depending on the nature of the information needed. These forums are open to any student and can be created by any student as well.

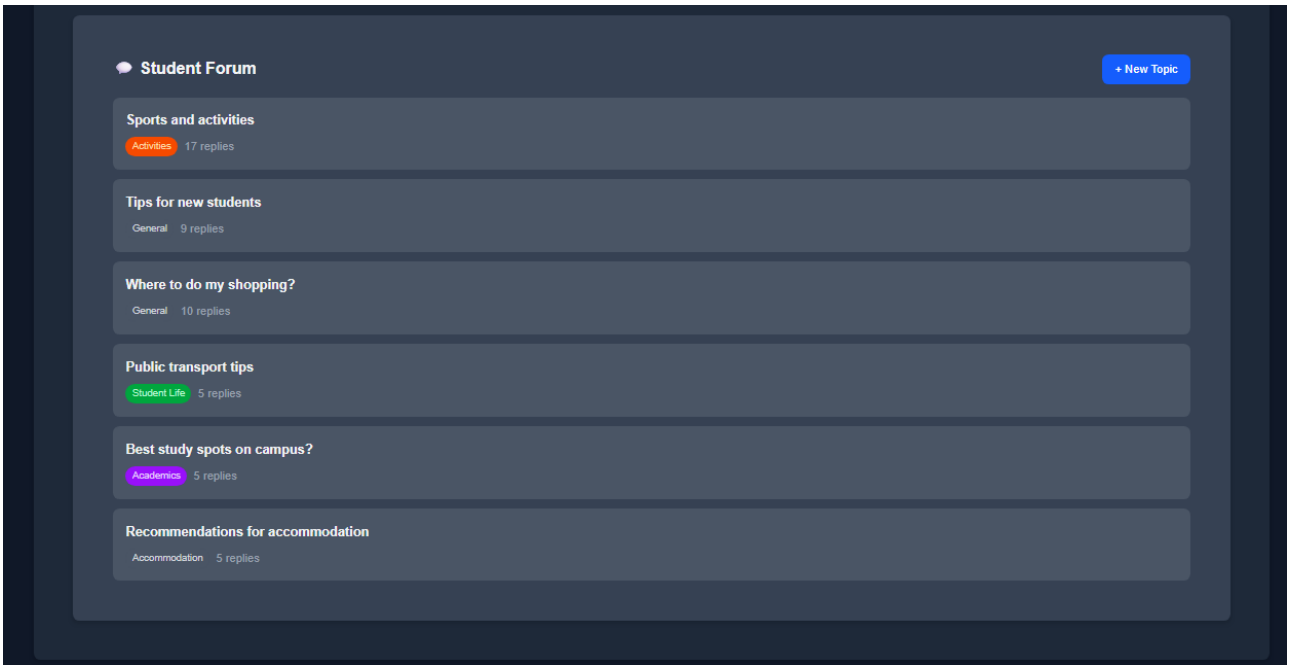

Figure 5-6. Student Forum

5.1.5 City Information Screen

As described above, the university screen has direct links to show you the city information screen. This screen contains a map so users can situate where it is they plan on studying, and a forum for questions and tips on life in the city (which serve a broader audience as they can also be useful for any student in that specific city, not just Erasmus). It also includes a list of all universities in that city, so a user interested in studying in a specific city has the information about all universities in that city at hand.

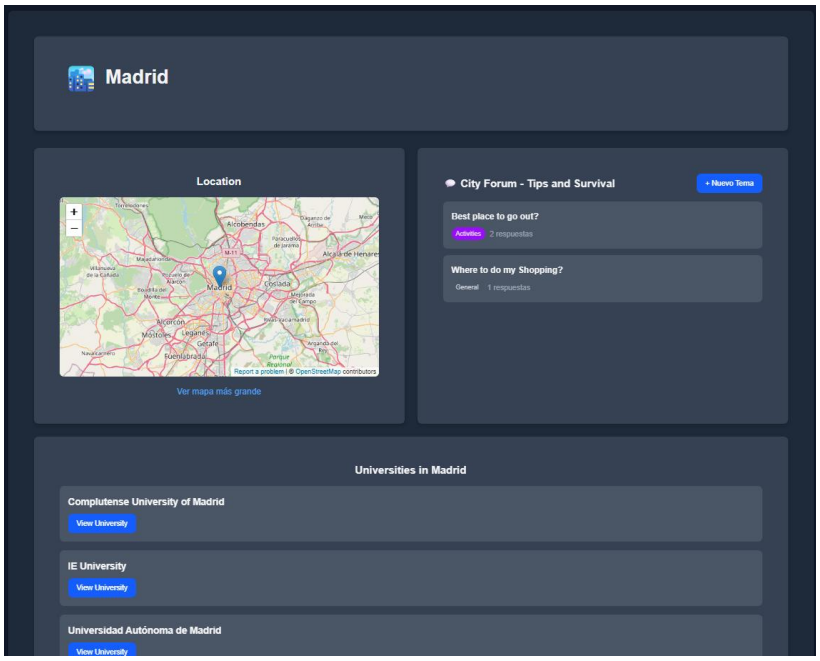

Figure 5-7. City Information Screen

5.1.6 Country Information Screen

Similarly to the city information screen, the country information screen gives a small insight into life in that country through the country forum. Any user who is new or interested in studying there can ask questions about it and receive answers to help guide their decision. It also includes a list of cities, where users can narrow down their decisions to these cities having the information about them at hand though links to their corresponding information screens.

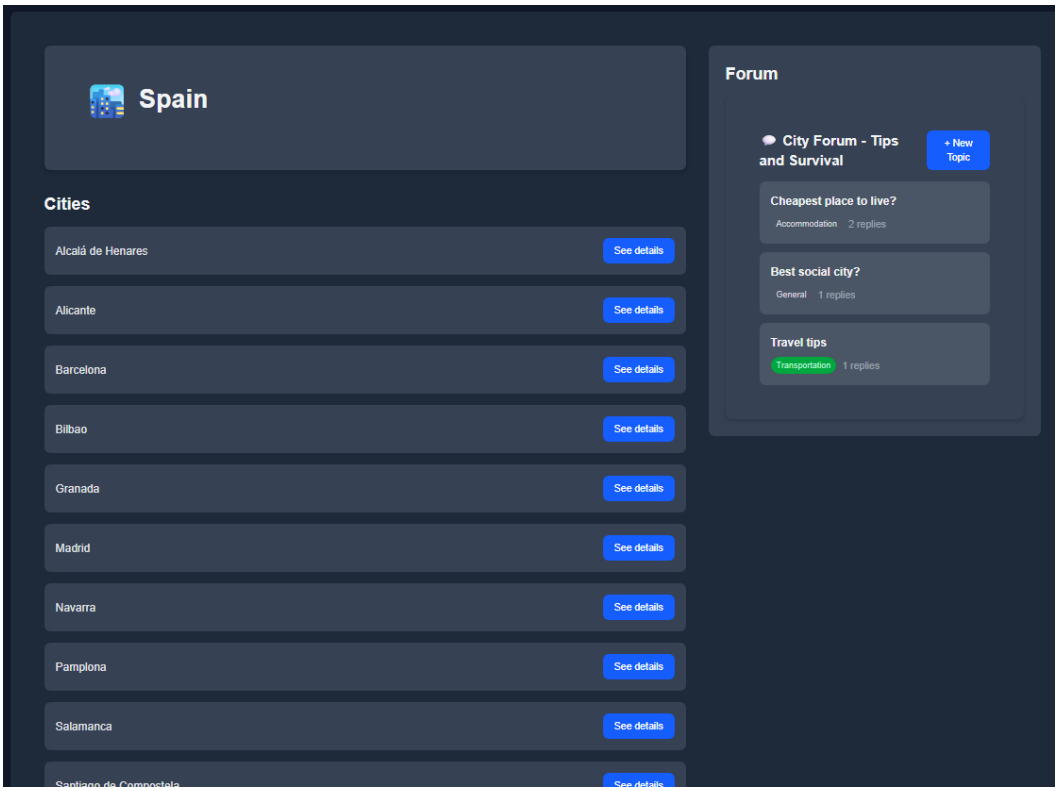

Figure 5-8. Country Information Screen

5.2 Students

Once a user has logged in, they now have access to more functions, unblocking these options in the navigation bar as giving them access to the main landing page, where there are links to the three main sections in the web application: University Search, Mentoring and Relevant Contact search(Erasmus Help)

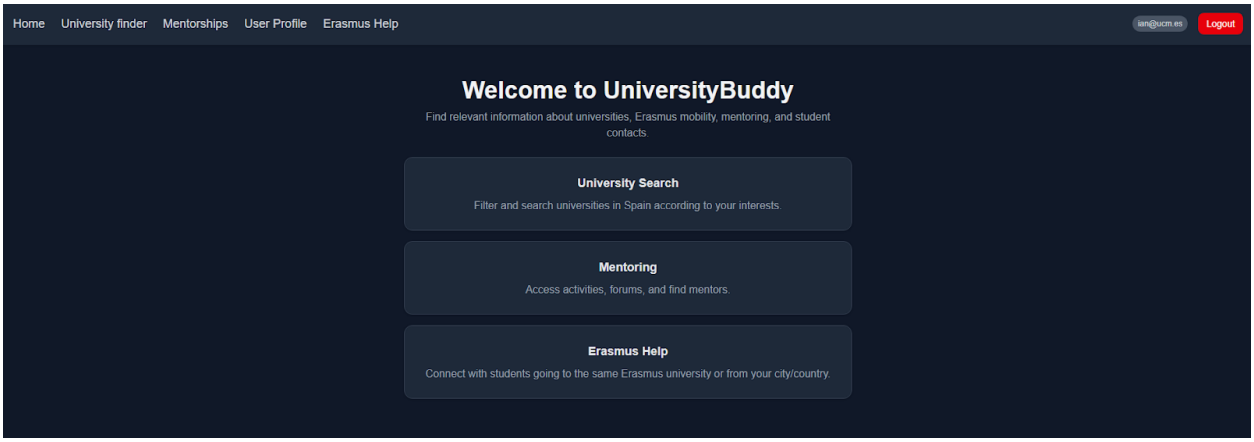

Figure 5-9. Student Landing Page

5.2.1 Mentorship Information

In the mentorship information screen, students are shown a calendar with their upcoming mentorship related events. These events are clickable both through the event list and the calendar directly. The events are separated into three categories, past, available and enrolled. The past events can only be seen in the calendar view, as the event list only shows those events in which the current user can be enrolled.

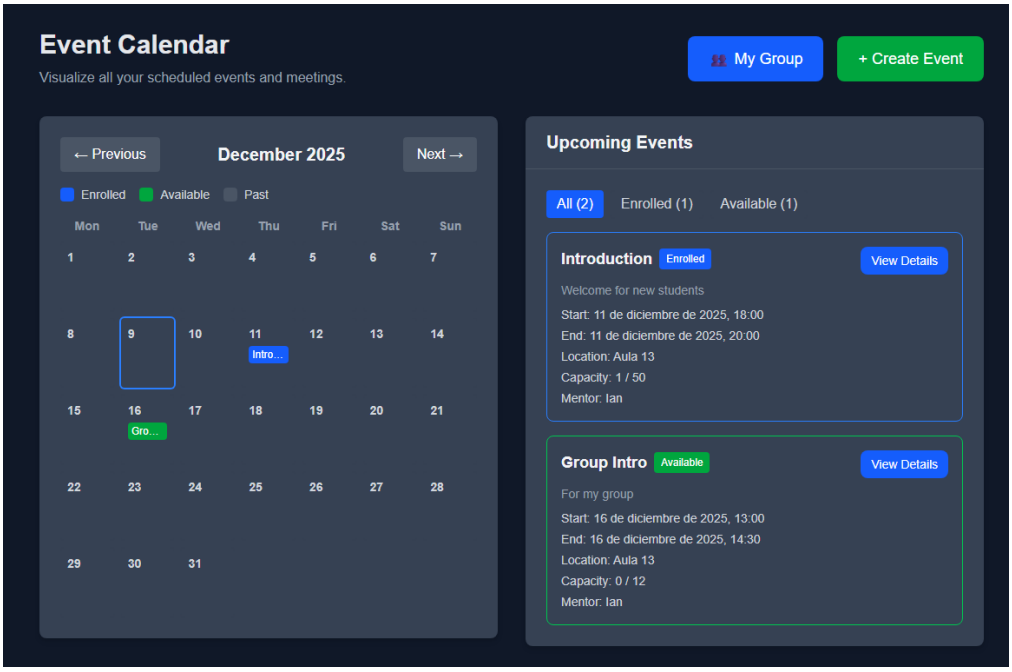

Figure 5-10. Mentorship Information Screen

5.2.2 Event Information

When an event is clicked users are redirected to the event information screen, where more information is displayed about the event. This information includes title, description, start and end time, location, mentor in charge and the current capacity, as well as which users have already signed up. To sign up to an event, a user must simply click the inscription button and will automatically be signed up and will receive a confirmation message on that same screen. When a user is already signed up, this inscription button is replaced by a deregistration button, where a user can easily deregister from an event.

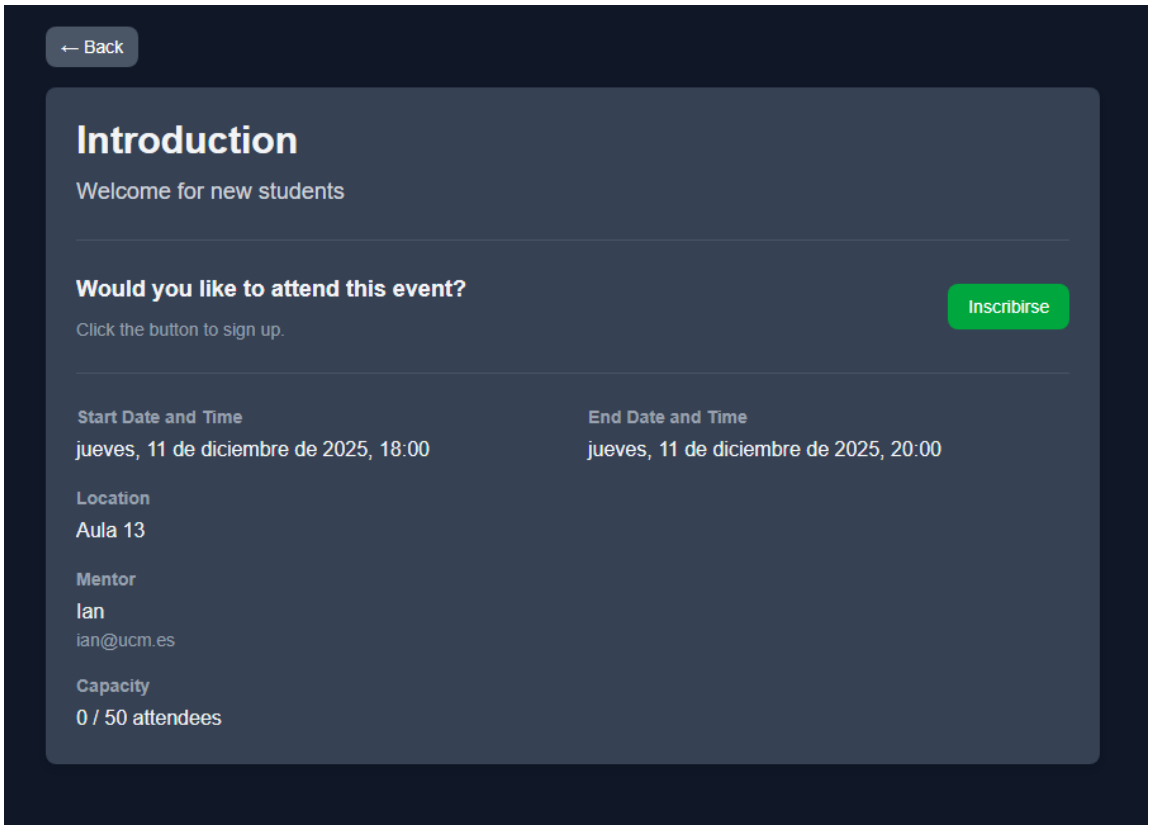

Figure 5-11. Event Information Screen

5.2.3 User Profile

When a user logs in they are redirected to this screen. Anything relevant to a student already studying can be found here. It is a summary of all relevant screens, with an event list and calendar such as the one in the mentoring screen, as well as quick links to all relevant screens for a current student, those being the event screen, university search(extremely useful for those students that are seeking an international experience) as well as link to their university screen to read forums and interact with fellow students.

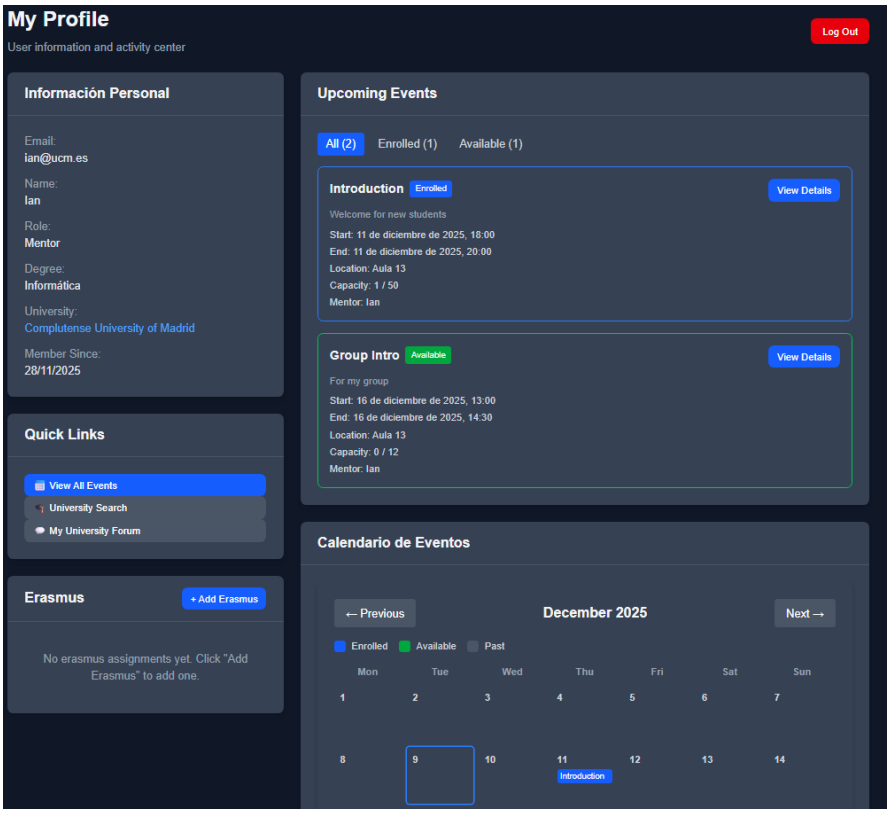

Figure 5-12. User Profile

Within a user's profile is a section where it shows their Erasmus universities in the case that they participated in an exchange program. In this section a user has the option to add an Erasmus experience whenever it begins, indicating the location and academic year it will take place. This helps for the next section, where if the user elected to be a part of the user search functionalities, they will now be able to be seen by those who might be participating and having the same destination university.

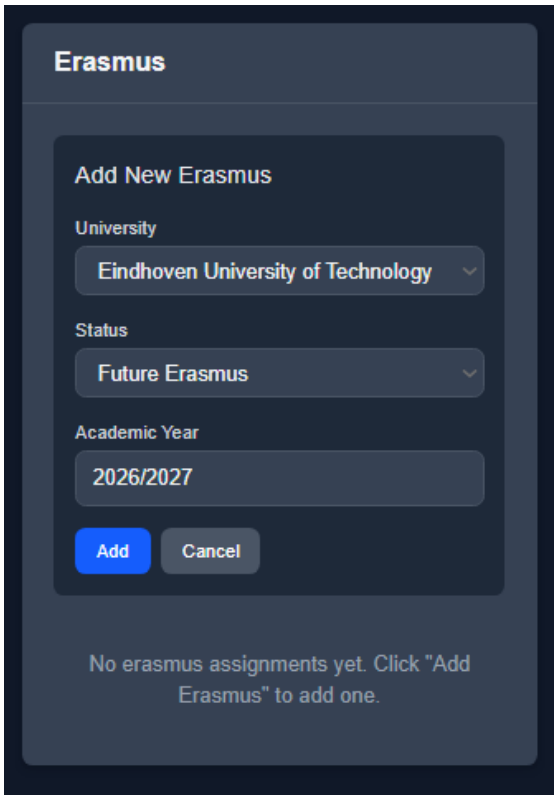

Figure 5-13. Add Erasmus Modal

5.2.4 Relevant Contact Search

This section is directly aimed at those students who are starting at a new university on an exchange program, although it can also be useful to those looking for a new university. Here, users will be able to search for users who study, studied or will study at their destination university and who come from their home city and/or country, as well as by what academic year their exchange took place. This search will provide a list of users, indicating their name, email address, home university and information about their Erasmus including the degree they study and what academic year they studied in.

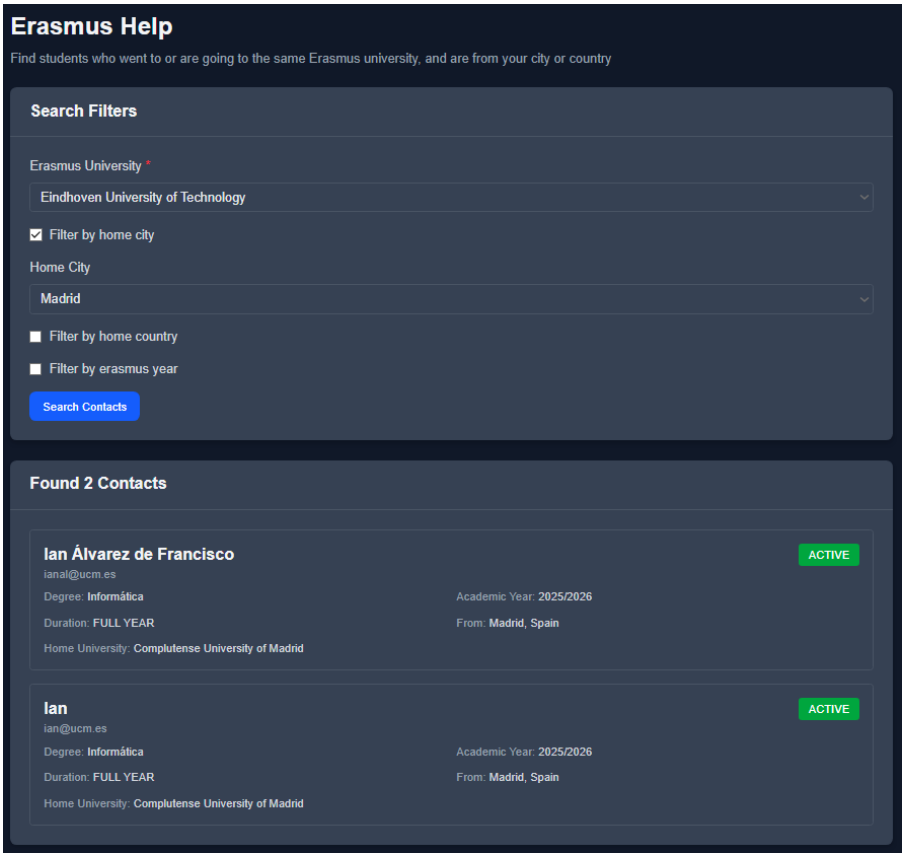

Figure 5-14. Relevant Contact Search

5.3 Mentors

Mentors are very similar to students in the sense that, apart from being students themselves, there is not much added functionality, except for that of managing their mentorship group. This is why they have an added functionality inside the event screen which is the option to create events.

5.3.1 Create Events

The event creation screen is where mentors can define all the characteristics that can be relevant to the event they are organizing. This form needs the following fields: Title, description, visibility (Public for all students and private for only their mentorship group), start and end date and time, location and capacity. If the event is created correctly, the user will be redirected to the event screen, where they will now see the new event on the calendar.

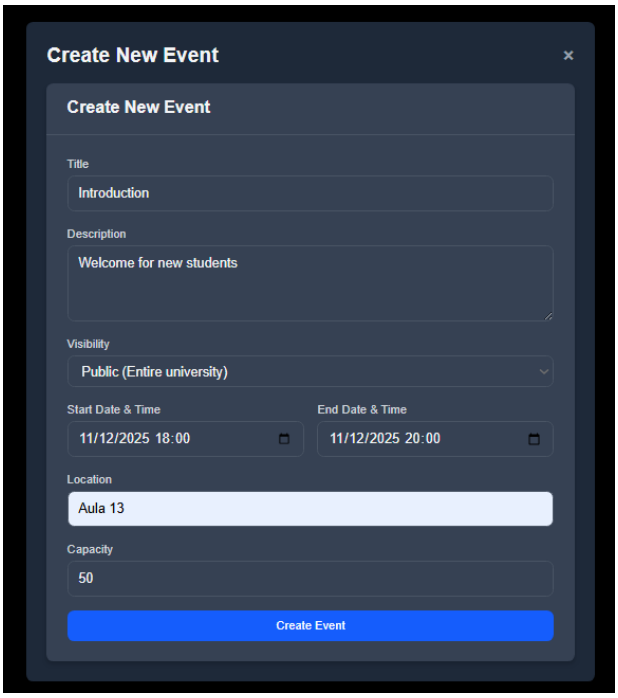

Figure 5-15. Create New Event Screen

5.4 Administrators

Administrators are capable of accessing anything users can access, but are granted with their own dashboard, which should include anything relevant to them as all other functionalities defined above are only useful for administrators to check that everything is working fine and to monitor general situations.

5.4.1 Admin Dashboard

The admin dashboard is where every administrator has all the relevant information. This information is related to mentorship groups, where all groups are listed and a summary of the overall situation is given at the top of the dashboard. This summary includes total number of students, number of mentorship groups, students currently in groups and number of unassigned students without a mentorship group. Mentors have the capacity to see who is in these groups and add and remove members on the spot.

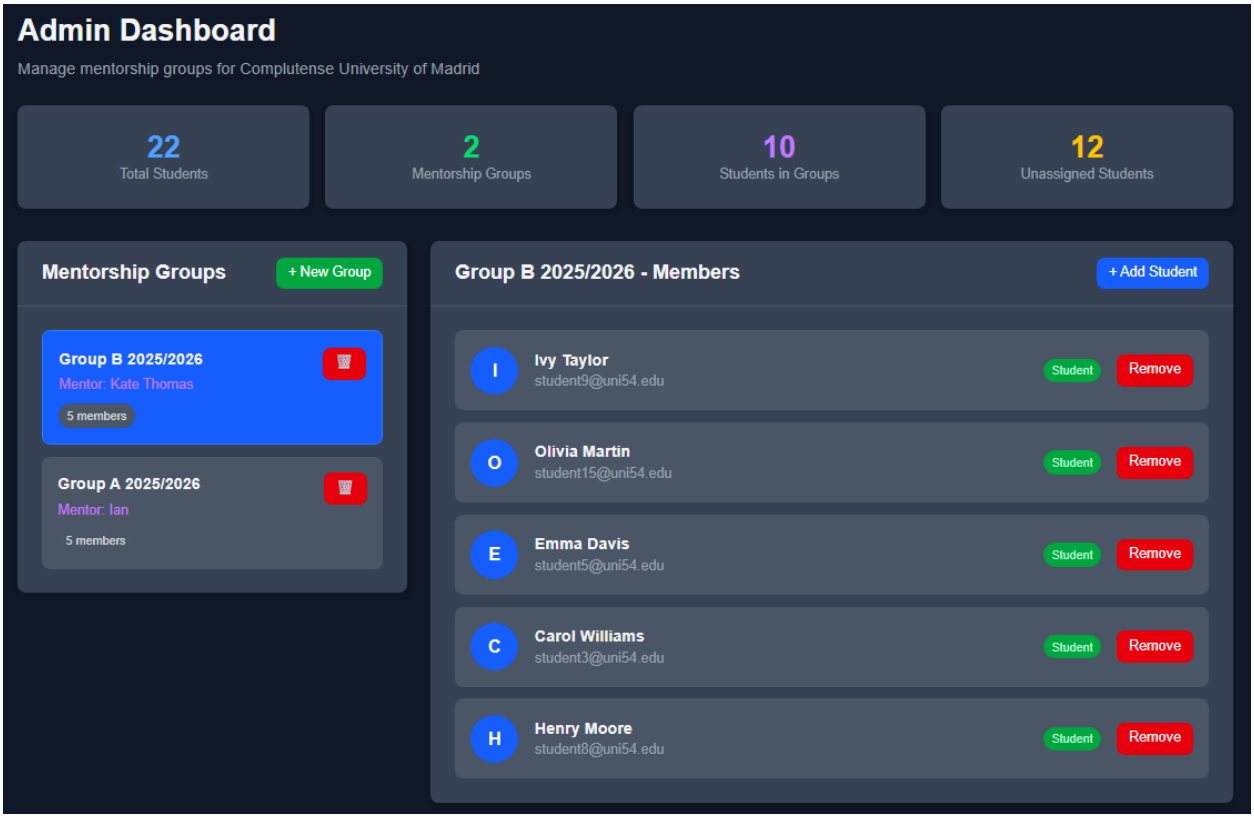

Figure 5-16. Admin Dashboard

They can also create and remove mentorship groups, where the only requisite when creating a mentorship group is assigning a mentor to it, meaning a mentorship group can't be created without a mentor.

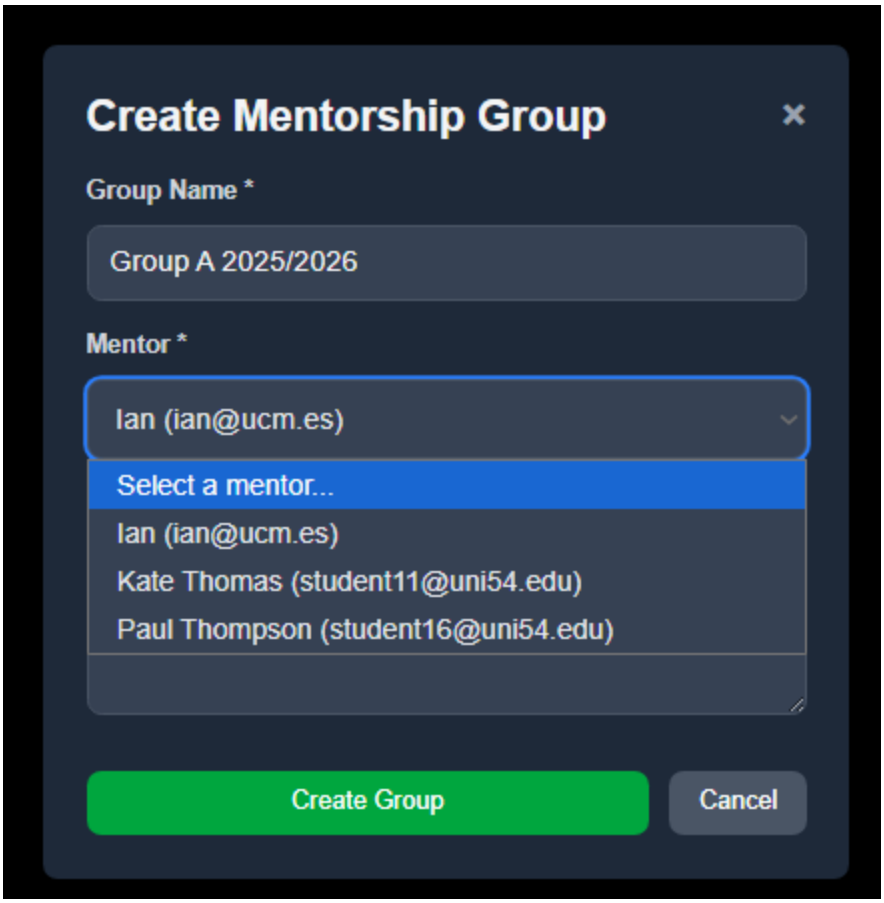

Figure 5-17. Create Mentorship Group

5.4.2 Create University

When a new university must be added to the application, administrators have a simple user interface where they can add the university as well as the country and/or city if it doesn't exist on the application yet as well. After creating this new university, administrators will be able to see their new admin dashboard with all relevant information as well as creating screens related to the university, activating forums and giving the information about that university.

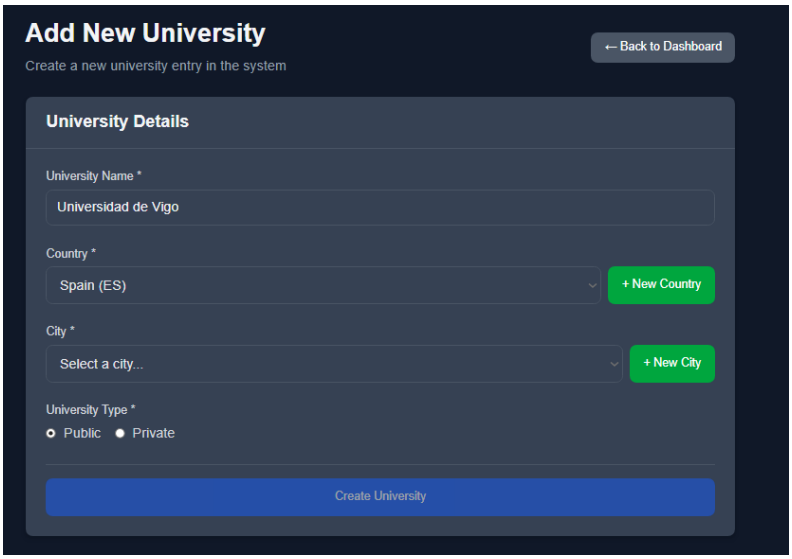

Figure 5-18. Create University Screen

When adding a country its name and code must be provided. When creating a city, the country it is in must be selected and its name, latitude and longitude must be defined.

Chapter 6 – Conclusions & Future Work

6.1 Conclusions

The development of this application has been a huge personal lesson, for technical and non-technical reasons. Technically, I've dove deeper than ever into web applications which have always been a huge personal interest. I've created an application that I'm happy with and which has been designed correctly in its development status and correctly built to support it in a possible future production state.

Not only have I learnt very valuable lessons from this project but I strongly believe this application could be a big help to many university students who, like myself, find new university experiences challenging and don't know where to find a helping hand to have a place they can look to for that help. That was the main objective from the start and I feel satisfied to have delivered a final product that can achieve that help I was looking for.

In terms of the technical aspects I have learnt to use a modern programming stack and build a project from the ground up. I have advanced my knowledge in PostgreSQL as well as learned how to use Prisma as an ORM. I have also advanced my knowledge in terms of backend applications and API design through the use of Node.js, Express and deploying these in a development state locally through the use of Docker, a feature I was very eager to incorporate as I had a keen interest in learning more about containers and related technologies and applying them to a real project was a huge boost in my understanding. In terms of frontend I learned to develop modern user-friendly user interfaces using Typescript with the help of previous HTML and CSS knowledge applied through the use of Next.js and the Tailwind CSS framework.

To conclude, this project resembles an accomplishment of the goal set before a single line of code was written or a design was sketched. The goal was to provide something that could help students, guide them in their quest to achieve stability and success in an environment they are assured is the correct one based on the feedback this application can provide them, knowing that if they wish to help others or use the help

for and academic change or international study experience, this application will be available to them and give them the most relevant information.

6.2 Future Work

UniversityBuddy in its current state offers students a helping hand, offering forums to participate in and a university rating system. Knowing that, it is built looking for scalability, and there are many improvements that can be made on the application to better its user experience and offer a better overall product. The following is a list with some possible improvements which could be made on the application:

- **Mobile Application:** Although the application is built around PC use as it is meant as a researching tool to calmly look at, having a mobile adaptation would be useful for users who may want to quickly check ratings or certain conversations.
- **Bookmarking feature:** A possible implementation to be added, especially if the application user base grows is the possibility to mark certain forum topics for easy access in the future and avoid large searches to find a topic.
- **Notifications:** In relation to the bookmarking feature, it could also be useful for the user to be notified whenever something new pops up related to a forum topic they are interested in, which would link nicely with a mobile application as well.
- **Deployment features:** With the current application structure, it could be improved to adapt it to be hosted on cloud servers, made easier with the current dockerized state of the application. In relation to better deployment features it could implement a CICD system for automatic deployment and testing, for all environments, creating a more reliable system structure.

To sum up, the application in its current state offers a helping hand to students looking for clarity. This application more than exceeds the expectations set at the beginning of this project and has given me a very interesting project that not only serves for this thesis but which I would love to keep developing further if possible. I feel that after this project I am not only more experienced but have learned several valuable lessons which I can apply to many different fields.

Bibliography

[1] RateMyProfessors, "RateMyProfessors," [Online]. Available: https://www.ratemyprofessors.com.

[2] Educaweb, "Educaweb," [Online]. Available: https://www.educaweb.com.

[3] Uniranks, "Uniranks," [Online]. Available: https://www.uniranks.com/ranking/europe.

[4] Universidad Complutense, "Experiencias Erasmus," [Online]. Available: https://www.ucm.es/experiencias-estudiantes.

[5] Prisma, "Github," [Online]. Available: https://github.com/prisma/prisma.

[6] MDN WEB DOCS, "JavaScript," [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/JavaScript. [Accessed December 2025].

[7] TypeScript, "The TypeScript Handbook," [Online]. Available: https://www.typescriptlang.org/docs/handbook/intro.html. [Accessed December 2025].

[8] MDN WEB DOCS, "HTML: HyperText Markup Language," [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTML. [Accessed December 2025].

[9] MDN WEB DOCS, "CSS: Cascading Style Sheets," [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/CSS. [Accessed December 2025].

Appendix A

User Guide

How to install the application

1. Download and install **Docker** and **npm** to your PC.
2. Clone the git repository.
3. Start the database and backend with docker compose: `docker compose up -d`
4. Install frontend dependencies: `cd frontend; npm install`
5. Run the frontend development server: `npm run dev`
6. If you wish to populate the database execute the following files: backend\seedUCMUsersAndForum.js, insertScript.py
7. You can now access the application's frontend at http://localhost:3000 and the backend at http://localhost:4000.
8. For testing purposes, a new user can be of any user type, including administrators.

Database diagram

Since the database diagram is complex and doesn't fit in an image, the following DBML code insert is valid to insert into https://dbdiagram.io and generate an interactive model of the database. This file is included in the project at the following location: https://github.com/Ian354/UniversityBuddy/blob/main/dbDiagram.dbml.