Funkify

A social media platform for musicians

Final Degree Project

Author: Sara Bourjila
Director: Joan Pastor

Defense Date: 28/04/2022

Bachelor Degree in Informatics Engineering
Software Engineering

Polytechnic University of Catalonia
Barcelona School of Informatics

Index

Contents

1 Context

1.1 Introduction

This document consists of the follow-up report of the Bachelor Thesis *"Funkify, a social media platform for musicians"* and which belongs to the studies of Bachelor Degree in Informatics Engineering specialization in Software Engineering at Barcelona School of Informatics of the Polytechnic University of Catalonia. As a document submitted during the development process of the project itself, its main objectives are two.

First of all, to establish a documented main view of the project: that is, what is the state of the project, what work has been done so far, and what are the tasks that, according to the contextualization and approaches presented, it remains to be done. In this way, this document aims to reflect the reality of the project at the time of its presentation.

Secondly, this document serves as a guarantee for the development and use of the project. In accordance with the established deadlines, and the tasks performed and to perform, the details of the work performed and the relevant considerations must be a test of the viability of project presentation.

To encompass these objectives, contextualization and details of the environment in which we carry out the work, as well as related informationwith the management and planning in relation to everything established in GEP. In this sense, they close considerations such as the emergence of deviations and alternatives, and integration of the knowledge and theoretical aspects that have had to be considered for development of the project.

1.2 Problem to be solved

Music has tremendous social, economic and political value and it is a sector that has been adapting to new technologies over the years. In the last decade the number of users in social media has been increasing and it seems that it's a non-stop growing, it's estimated an increase of almost 4.41 billion in 2025[3].This has affected us in so many ways, but the most important one is the way how we interact with other people, which has become more accentuated in recent months due to the confinement of this year, although this change was already coming. The way we interact and socialize affects the business world in turn, although I will not go into much detail and focus more on the music sector.

The music industry has also been affected by the era of social media and has been adapting to it. New tools have been emerging (Music Streaming, Live Streaming) if this is added to the viral trends that have its positive and negative parts, we have a new concept of what it is to be a musician in these times. There has also been a strong increase in the use of social networks due to the confinement of the musicians, since it was impossible for them to work in conditions. All this has influenced so much that if you don't have many followers on social networks, nobody knows you and if nobody knows you sometimes it implies that people, in this case musicians, cannot fully dedicate themselves to it, because let's be honest doing organic networking would only reach a small portion of your area where you live and even smaller these days due to Covid-19.

1.3 Stackeholders

1.3.1 Professional and non-professional Musicians

Musicians who want to interact with other musicians and want their music to be heard or they are looking for a record label in their area. These would be the main stakeholders.

1.3.2 Record Label

Record companies especially independent record companies that want to make themselves known and are looking for new talents, this social network would allow them to listen to the work of various musicians depending on the genre.

1.3.3 Event/Festival Organizers

Organizers of festivals or events that want to make themselves known and that are looking for musicians to participate in their events. Both these and the record companies can organize castings in the application to facilitate their work when choosing musicians.

1.3.4 People interested in Music

People who are interested in music in general and who want to support musicians or if you have gone to an event and liked a band / musician, you can easily find it according to the event / festival.

1.3.5 Project Developer

The person in charge of developing the project, which in this case is me, although I also consider myself part of the first group of Stakeholders because I am also a singer. With this project I will be able to demonstrate all the skills obtained throughout the career to finally be able to obtain the final degree and I will also be able to apply computer science to music, which was one of my life goals.

1.3.6 Project Administrator

The content manager of the application will be in charge of maintaining a pleasant community without any kind of abuse or disrespect.

1.3.7 Project Tutor

The project tutor is responsible for guiding and advising during the development of the project and therefore verifying that it follows the established planning. In this project specifically it is Joan Antoni Pastor Collado, from the Department of Services Engineering and Information Systems who is in charge.

2 Justification

As I mentioned before, the rise of social networks is increasing and will continue to increase in the coming years, there have been many different types of social networks all with the same goal: connect more people and make themselves known if you own a business or your job requires a lot of networking which is the case of the musicians.

2.1 Existing Applications

After doing an exhaustive search for existing applications whose base was a social network oriented exclusively to musicians, these are the ones I found:

2.1.1 SoundCloud

Soundcloud is one of the best known websites where musicians can upload their music to the internet, what is called online music streaming has become a huge influence on how musicians grow their careers musicals at this time.

Strengths: It is one of the easiest music streaming platforms to load, in general the platform is easy to use and quite intuitive, which makes work much easier.
Weaknesses: Although it is possible to have fans, the platform is not very oriented to socialize with other musicians and fans.

2.1.2 Drooble

Drooble is a social media platform whose mission is to unite musicians and artists from around the world so that its users can gain more exposure, knowledge and experience.

Strengths: Includes your radio broadcast and promotion tools. They also have a system where you interact with other musicians to earn karma points. In turn, it is a currency that can be used to use promotional tools on the platform.

2.1.3 ReverbNation

ReverbNation is a platform that aims to help artists in the music industry. You can upload your songs, logos and album covers, etc.
Strengths: You can register as an artist or as a fan, so the two types of users are quite distinguished. They also have payment plans to promote your music.
Weaknesses: You can have quite a few fans, but maybe it doesn't allow you

to have a lot of interaction with them.

2.2 Conslusion

After doing this analysis I have been able to extract both their strengths and weaknesses of each one of these aplication mentioned previously and thus be able to create a new application inspired by some of the applications already mentioned, although with some new functionalities that to my point of view, at the moment, they are quite key.

So, in conclusion, the objective of this project is to develop a social media networked multiplatform (Android and IOS) for professional and non-professional musicians, geolocatable by person or place, which could be used to see and interact with other musicians in your area and to find rehearsal venues or music events near you, and the like. The main goal of this application is for musicians to interact with each other and promote their work, for example, collaborate and communicate with producers, record labels, and other similar tasks.

Application	**Geolocation**	**Types of user**	**Allows multimedia posts**	**Chat**	**User search by Proximity**
SoundCloud			X	X	
Drooble		X	X	X	
ReverbNation		X	X	X	
My application	X	X	X	X	X

Table 1: Comparison of the different applications

3 Scope

3.1 Objectives

3.1.1 Personal objectives

• Learn to develop a software application from scratch: since it is the first time that I have done a project of this size by myself and I am going to use technologies that I did not know previously, so it is a challenge for me.

• Obtain more knowledge and experience in project management as well as in the resolution of any problem or doubt that may arise during the course.

• Learn to document in Latex [1]: I think it is a good opportunity to learn to document in Latex, which I have always believed necessary but until now I had not had the opportunity to do it, so I have decided to make it a personal goal.

• Leave open the possibility of continuing with the development of the application once the TFG has been delivered. If the application goes as expected, the possible continuation of it is not ruled out since I think that it could be a very enriching experience and that it can be quite useful in these times.

3.1.2 Objectives of the application

• Get users to promote their music and make themselves known.

• Search for nearby users and events. (Allow a user, for example, to search for users who know how to play a specific instrument, who want to form a band, etc.).

• Encourage the socialization of users as well as their fans by attending events created by the users themselves. (For example, a band is going to play in a bar or a discography has organized an event, etc.)

• Allow the publication of multimedia content to share with other users such as your portfolio, add links to concerts recorded in other applications. (For example, add videos uploaded to YouTube, upload recorded songs).

• Share knowledge by publishing FAQs to help other users with the most frequent questions they may have when using the application. (For example, a user who does not know how to upload her recorded songs or who finds it difficult to create an event).

• Interact directly with other users through chat.(For example, a musician who wants to talk to his fans).

3.2 Sub-objectives

• Bring independent music to more people.

• Help musicians to socialize without having to leave their homes.

• Mitigate a little the damage that this sector has suffered caused by the pandemic.

3.3 Non-Functional Requirements

The non-functional requirements are the following:

• **Usability:** It should be an easy-to-use and fairly intuitive application.

• **Availability:** the application must always be available, so system crashes should be avoided, since the user must be able to use the application at any time of the day.

• **Scalability:** The application is a social network so the system must be prepared to receive a large number of users in the long term.

• **Adaptability:** The application must adapt to the different existing systems, that is, it must be a multi-platform application.

• **Security and privacy:** The system must ensure that the data accessed by the user is their own and limit / prevent access to the private data of other users.

• **Legality:** The system must comply with current legislation, especially the RGPD.

3.4 Risk and Obstacles

There are many different possible risks that could affect the development of the project, here are the main ones that I have considered and they are classified by the severity of the situation.

• **Covid-19:** It should be noted that the project is being developed during the coronavirus pandemic by a single person and it may be the case that the main developer can contract the virus, which could lead to the suspension of project development for a weather picture.

• **Loss / breakdown of the material:** Although the loss of all the material is quite unlikely since the project is uploaded to a repository in the cloud, it could be the case of loss or breakdown of the computer with which the project is being developed.

• **Inexperience in the technologies used:** The fact of using technologies that the developer has never used before increases the learning curve and therefore implies that a lot of time has to be spent on development.

• **Fixed delivery date:** In every project there are setbacks, unforeseen events in the development or testing phase, but having a fixed delivery date implies that delivery cannot be delayed and therefore the final product loses so much quality.

4 Methodology and rigour

4.1 Agile Methodology for one single person

Taking into account that the team is basically made up of one person, the methodology that has been chosen for this project is the agile methodology called SCRUM, because it is a software development model that makes it possible to establish, step by step, in a successful and organized way.
Later I will explain in more detail how to adapt it when the team is made up of only one person, but first we will start by making a summary of the different roles existing in this methodology.
Next, we will explain which are the phases of this methodology, and what each one of them consist of. And once the context has been entered, we go on to explain how it will be applied in our project.[2]

4.1.1 Roles

A SCRUM methodology is made up of a team of people with three different roles , the product owner, the Scrum master and the team of developers. As the only contributor to the project, these roles will be carried out by a single person, therefore they must be adapted to the new way of working as follows :

• **Product owner:** The product owner is the Source of product information, he is the one who has the final result in mind from the beginning, and therefore is in charge of sharing this vision with the rest of the team. As your own product owner, you must be able to clearly define the final product you want to obtain, to achieve this you must identify the functionalities and requirements of the project, always keeping in mind that the delivery of this project is the final objective.

• **Scrum master:** He is in charge of facilitating the iterations by guiding the team and increasing its performance. As a Scrum master herself, you have to be able to solve all the Roadblocks that are presented, this requires careful study of the project you are working with.

• **Development team:** Being the entire team of developers is the easiest role among those described, but it must be borne in mind that a team of Scrum developers is self-managed and self-motivated to complete the iterations, and it is important to approach the project with this attitude and face the difficulties that may be generated to improve one's personal process.

4.1.2 SCRUM phases

The SCRUM process is organized through iterations, each iteration has a duration that is established according to the development effort (see Working Method section) and consists of a series of phases. In the case of adaptation for a person, these phases are the same and are repeated for each iteration:

1.Sprint Planning: In this phase, you must act as a Product Owner to define the product requirements and as a Scrum master and team of developers to plan the process and create goals to achieve at the end of the iteration in question.

2. Daily Scrum: Daily from the beginning of the iteration a priority "meeting" should be held where, as the only member, the work done during the previous day should be reviewed and any problems that may have occurred should be identified. Then you should act as a Scrum Master to find solutions to any identified problems. And finally, there will be a projection of the work that could be completed for the next daily scrum.

3. Sprint Review: At the end of the iteration, you should act as a team of developers to assess and consider both what has gone well during the iteration and the possible difficulties or problems. On the other hand, you must also act as a Product Owner to evaluate how the results are aligned with the established requirements.

4. Sprint Retrospective: In this last phase, as a member of the developer team you have the opportunity to inspect yourself and create a plan for improvements that are addressed during the next iteration.

4.2 Work method

After having carefully studied each of the roles and phases that make up this agile work methodology, we can now go into detail on how it will be applied to our project.

Initially, a list of user stories has been defined in order to satisfy all the objectives mentioned above that the mobile application will have (see Scope section). These user stories will be developed in the backend and the frontend simultaneously in order to guarantee that the whole set is working properly.

Next they have been classified into groups according to the functionalities to which they belong, this will allow to make a planning of all the iterations. Considering that each group contains several iterations, then each one of them is called without going into much detail in order to be able to make a rough

planning of the project:

1. Management of the different users, 2. Management of the geolocatable search engine and the chat. 3. Management of the different types of events, 4. Management of multimedia publications

4.3 Work tool

To get a good follow-up of the entire project, it has been decided to use a set of tools that facilitate this task. Regarding the monitoring of the generated code, the Github[5] collective development platform is used, in which projects can be hosted using the Git version control system. On the other hand, the Trello[4] tool (a project management software) is used that will allow to organize each one of the tasks associated with the project and classify them according to the state in which they are, until their completion.

And finally, it has been decided to use Clockify (a time control software) that together as an extension to Trello will allow us to have an estimate of the time dedicated to each of the tasks.

4.4 Validation Methods

To validate a product, it means that you have to ask yourself if you are building the correct product.To ensure that, a series of tests have should be carried out during the development of the product. On the one hand, the simplest tests, which means, the manuals, both in the back-end and in the front-end, have been tested that all the calls do what they should do, checking that for each one of them what is expected is obtained.

Once these types of simple tests have been passed successfully, and as usual members of agile methodologies, we proceed to do the so-called unit tests, which roughly consist of pieces of code designed to verify that the main code is working as it should be.

Finally, automated User Interface (UI) tests are also carried out, these include functional tests of the user interface controls, that is, they allow to check if the entire application, including the user interface, works correctly.

5 Description of tasks

Before describing the different types of tasks and estimate the time they will last, it is necessary to take into account other factors that influence, in my case, when it comes to being to have a more realistic estimation. The first factor is that this semester I am not only going to dedicate myself exclusively to this project, but at the same time I am taking another subject and working about 8 hours a day. The second factor to take into account is that the date of the presentation has not yet been decided, although I will try to assume that it is as soon as possible.
Without any further contemplation, let's procede to the description of the tasks.
Assuming that this project star date September 21, 2022 and the end date is April 28, 2022, also taking into account the factors described above, the project will have a total of 521 hours distrubuted as follow:

• 31 week with this schedulle:

Monday	Tuesday	Wensday	Thursday	Friday	Saturday	Sunday
1h	-	4.5h	4.5h	4.5h	8h	6h

Table 2: Working hours per week

After having assigned the time that will be dedicated to this project, we will proceed to the description of tasks. Before starting, remember that this project, as we had described in previous sections, follows an Agile Methodology, specifically Scrum.
We proceed to the description of the different tasks to carry out this project successfully. Each task described below are arranged chronologically. The hour has been taken as the unit of time, in order, to carry out the temporal planning of the entire development of the project because working with a methodology based on user stories, it is the unit make most sense to use it. Therefore, each of the tasks has a number of hours associated with it.

1. **Project Management:** The project manager will write all the documentation on :
   - Learning and familiarization with Latex and other documentation tools. -The scope and contextualization of the project.
   -Temporary planning of the project.
   -The economic management and sustainability of the project.
   -Throughout development of the project.

2. **Learning about Swift and iOS Development**:The developer has to

gather the necessary knowledge to be able to work with iOS Development.

3. **Environment Setup**
   - The developers install and configure all the tools necessary for the development of the project.
   -The developers will prepare and deploy the repositories both through the Backend and the Frontend of the project.
4. **User registration:** Application users will be able to create a new user account in the application.
5. **Login with email:** Users can log in to the application through their email.
6. **Login with google:** Users can log into the application through their Google account.
7. **Reset password:** Users will be able to reset their password to access the application (if the registration has been done with the mail).
8. **Delete account:** Users will be able to delete their account.
9. **View user profile:** Users will be able to see the information associated with a user.
10. **Follow user:** A user can follow other user, the followers will be called fans.
11. **Create a Post:** Users can post a publication on their wall.
12. **Add photo to Post :** Users can post a photo on their wall.
13. **Integrate with Youtube**The user will be able to connect their Youtube accounts,in order to be shown in their profile.
14. **Like user a Post:** Users can like for a specific Post of the system.
15. **View home page:** Users will be able to see the main page of the application.
16. **Create event:** Users will be able to publish a new event in the system.
17. **Event details:** Users will be able to view the information associated with a specific system event.
18. **Attend an event:** Users can add themselves as attendees of a specific event in the system.
19. **Search by users nearby:** Users can see others users who are nearby in a list ordered by proximity.

20. **Search by tags nearby:** Users can see others users/events who are nearby with a specific tag in a list ordered by proximity.

21. **Chat with a user and Inbox:** Users will be able to have conversations with other users of the system.

22. **Project defense preparation:**The manager will prepare the project to be able to present it to the audience and to the defense team on the day of the defense.

6 Resources

6.1 Human resources

The human resource included in the project is a single software developer, (both at the Backend and Frontend level) it is said that she works a total of almost 30 hours a week.

6.2 Material resources

Resource	Type of Resource	Goal
MacBookPro 2.7Gh Intel Core i7 4-core processor, 16GB memory, macOS Catalina (Version 10.15.7)	Development	To be able to run the project software
Smartphone(iPhone) with iOS version 15	Development	Test the application
Xcode Version 12.5 for Mac	Development	Develope the software and emulate the application
Github	Development	Manage repository where of the project
GitKraken	Development	Control the flow of the development process
Overleaf	Management	Tool to write the documentation of the project
Trello	Management	To manage tasks of the project
Teamgantt	Management	to be able to design the Gantt chart for project planning
Athena's courier service	Comunication	To be able to communicate with the project management teacher and view the documentation
FIB email service	Comunication	To be able to communicate with the director of the project

7 Estimates and the Gantt

Among the different tasks described before there is a set of precedence dependencies, wich means, all those tasks that must be performed first to be able to move on to other tasks that require the previous ones to be completed. To see clearly and well specified, both the logical sequence, duration in hours, as well as the precedents between the different tasks, a Gantt Chart (See on Appendix A) created with TeamGantt [6] and a planning table (See on Appendix B)have been carried out at the task level, these figures are included in the Annex, at the end of this document.

8 Risk Management

8.1 Obstacles

The following risk have been considered:

Risk	Impact	Probability
Covid-19	High	High
Loss/Breakdown of the material	Low	Low
Inexperience in the technologies used	High	High
Fixed delivery date	Medium	Medium

Table 3: Risk Management

These deviations mentioned would affect the development phase of the project, causing the increase of the hours of some tasks, and consequently, so would the total hours estimated in the total realization of the project.
As for resources, these would not be affected, in the worst cases there would be an increase in the energy consumption of the equipment.

8.2 Alternative Plans

In order to be prepared for the possible risks, some measures have been taken to solve them and / or soften their impact on the temporary planning of the project:

• Familiarization time with the technologies: A considerable amount of hours have been devoted to learning the project development technologies. With this measure, it is intended that, in the event of any deviation, its development is as fluid as possible.
• Overestimated working hours: A small overestimation has been made of the hours of both the weekly workday and the time assigned to tasks, in case unforeseen events arise that require us to use a little more time.
• Margin of days: As last measure, the temporary planning of the project has been proposed in such a way that ending on January 30, 2021, there is a margin of at least 15 days until the reading of the project to be able to manage any deviation .

9 Economic Management

9.1 Budget

Throughout the project all the resources that we have called in previous sections will be used (see temporal planning), these resources they carry a set of costs for the project. These are those specified below and have been divided into direct costs (human resources) and indirect costs (hardware, software and other associated costs).

9.1.1 Direct Costs

As this project is developed by a single person, this person during the project development process will play all the associated roles. These are the ones described in the following table, with their corresponding costs:

Human Resources Costs			
Role	**Hours**	**€/h**	**Cost**
Project Manager (PM)	25h	20 €/h	500 €
iOS Developer (AD)	496	16 €/h	7936 €
TOTAL	**521h**	-	**8436€**

Table 4: Human Resources Costs

The calculation of the cost per hour is computed using data from the Spanish territory for the year 2017. It has been calculated by dividing the average annual wages (gross) of each role in the country[8], by the average number of annual hours worked by the inhabitants in an active employment situation of the same.[9]
As can be seen, the number of hours to work for each role has been established according to the set of tasks to be performed specified in the Gantt chart (see Appendix A). To have a more exact and graphic idea of the cost for each of the planning tasks,please see the attached table with the breakdown of the costs of all the project tasks in the Appendix C. Each of the roles have been shortened to facilitate reading.

9.1.2 Indirect Cost

Hardware Cost
The following table indicates the budget in terms of hardware along with its corresponding amortizations. To perform its calculation, we have established that the years have 355 working days remember that we are counting weekends too). To simplify it, we have considered that the hours of use by the hardware are equivalent to the duration of the project, so we are considering a total of 477 hours as mentioned in the temporary planning section, and an average of 4 hours per day.

Product	Unities	Price €	Useful Life	Amortization
MacBook Pro	1	2.699,00 €	7 years	385,6€
Smartphone with iOS 15	1	529 €	5 years	105,8€
TOTAL	-	**3.228,00 €**	-	**491€**

Table 5: Hardware Costs

Software Cost

The table below shows the budgets for the software used throughout the project along with their amortization. To calculate them, we have used the same criteria as in the case of hardware costs:

Product	Unities	Price €	Useful Life	Amortization
Android Studio 4.0.1 for Mac	1	0€	3 years	0€
Github	1	0€	3years	0€
GitKraken	1	0€ (Student Licenses)	3years	0€
Overleaf	1	0€	3years	0€
Trello	1	0€	3years	0€
Teamgantt	1	880,2€	3years	293,4 €
Athena's courier service	1	0€	3years	0 €
FIB email service	1	0€	3years	0 €
TOTAL	-	**880,00 €**	-	**293,4€**

Table 6: Software Costs

Other Inderect Costs
Finally, it has been decided to set aside a series of costs that do not fit into any of the previous categories and that are described in the following table:

Product	Unities	Price €	Estimated Price
AWS	6	0€	0€
Internet Access	5 months	40€/month	200€
Electricity	199,357 KW/h	0,1255€/kW	25,02€
TOTAL	-	-	**225,02€**

Table 7: Software Costs

9.1.3 Amortization

As already mentioned at the beginning of the indirect costs section, a set of parameters is being used to calculate the depreciation of each of the equipment necessary for the development of the project. The formula used to perform this calculation is detailed below.

$$\text{Amortization} = \frac{521 \text{ hours of use}}{\text{time life} \cdot 355 \text{ days} \cdot 4\text{hours/day}} \cdot \text{price of the product}$$

9.1.4 Contingency Plan

Every budget must have a contingency plan, to be warned in case of unforeseen circumstances, in this project, it has been decided to establish that 10% of the total budget (adding direct and indirect costs) is allocated for this purpose, with this amount is believed to be sufficient as the budget is done in sufficient detail. This means that if we have a total budget of 9279,6 € , the contingency plan for our project is 9220,64 €. approximately.

Type of Cost	Cost(€)	Contingency %	Final Cost (€)
Human Resources Cost	8436 €	10 %	9279,6€
Indirect Cost	608,4 €	10 %	669,24 €
Other Inderect Cost	225,02 €	10 %	247,5 €
Total	9269,42 €	10%	**10196,34 €**

Table 8: Contingency Plan

9.1.5 Unforeseen Costs

Every project may have unforeseen events and that is why it is necessary to take into account a certain amount of money destined for them and their probability of occurring. Two possible types of contingencies are foreseen:

1. Complications in the interface design. This could make it take more time for development (prob. 40%, cost 320 €, additional 20h of the mobile application developer).
2. Error in estimating the end date. It could be the case that the established completion date is reached but it will take a little more time to finish completing some tasks. (Prob. 15%, cost 240€ , additional 15h of the mobile application developer).
3. Loss/Breakdown of the material. As we mentioned in the risk an obstacles sections we should consider the cost of the MacBook Pro if lose it o break it. (prob 5%, additional cost 2.699,00 €).

Unforseen Cost	**Cost**	**Prob.%**	**Cost with Prob.%**
Complications in the interface design	320 €	40%	128 €
Error in estimating the end date	240€	15%	36 €
Loss/Breakdown of the material	2.699,00 €	5%	135€
Total	-	-	**299€**

Table 9: Unforseen Costs

9.2 Final Budget

Let's summarize in a table all the costs seen above to be able to see the true cost of the project:

Type of Cost	**Cost(€)**
Human Resources Cost	9279,6 €
Indirect Cost	669,24 €
Other Inderect Cost	247,5 €
Unforseen Cost	299 €
Total	**10495,34 €**

Table 10: Final Budget

9.3 Management Control

As for hardware resources, it is not expected that there will be any deviation from those initially budgeted, since all that will be needed for the full development of the project have been determined and this will not change, at most we could have a fault but this does not It would entail no extra expense, since it would be covered by the guarantee of the same product.

10 Sustainability Report

10.1 Self-Evaluation

On a personal level I do not have much knowledge about sustainability, only those that I have worked on throughout my degree in some subjects, but never as deeply as I am working on this section of the project. I knew the three dimensions of it, but I had never thought about them in a real case, and the fact of applying these three perspectives in your own project makes you realize the impact that any project has both economically and socially. It also makes us reflect on the fact that society we live in has little awareness of this issue, the number of projects that arise every day, and how little mentality society is about the harm that we are doing to the planet , since most companies only look out for their own interests, and it will not be until the moment they put that aside, when we are really helping to reduce this impact.

After carrying out the survey and reflecting on the different points discussed, a summary of those positive aspects that sustainability brings us can be made:

• Less impact: This approach can lead to an improvement in the environmental impact, the quality of life of people, as well as the economic flow and its transparency.

• Better future: We are currently experiencing a very delicate environmental situation, therefore sustainable solutions could guarantee that the natural resources available to the planet would not be exhausted and that there would be enough so that they could be renewed or at least , could last a good number of years (as would be the case with oil or natural gas).

And as negative aspects we have:

• Increase in costs: The economic increase in projects due to the increase in time spent in the search for sustainable solutions.

• Unemployment in some areas: Changes to preserve and care for ecosystems can cause many industries to have to reduce their activities in some sectors, this can lead to unemployment of many people who have dedicated their lives to work in this sector (an example of this would be the coal industry).

10.2 Economic Perspective

10.2.1 The project put into production

To carry out this project, a series of costs have been estimated, trying to be as low as possible (see section on Economic Management), since it is a final degree project and resources are more limited, apart from that as well the economic impact is minimized. For example, in terms of human resources, it has been decided to use a minimum number of roles with a fair salary assigned to each member but without it being disproportionate, as well as in terms of software, as far as possible, it has been tried that this out of free software in order to save a considerable part of the budget. And as for the hardware, the one that was already in possession before the start of the project has been used and therefore, this means that the costs of this are lower (since they are no longer products just released on the market).

10.2.2 Exploitation

Currently most existing tools equivalent to the project to be carried out are free to use, but we do not know what economic resources have been used, that is, we do not know the costs in human resources, software , hardware or other costs such as electricity used, renting a space, etc ... anyway for this project, the resources used are the minimum necessary as mentioned in the previous section, so economically speaking, it is intended that the economic impact of this application is as small as possible, improving it or at least equating it (due to ignorance) to the existing solutions.

10.2.3 Risk

Economically speaking, I do not think that scenarios could be produced that would harm the viability of the project, since as has been said it is a small-scale project and that few resources are needed, where a set of mostly free tools are used, and also a is, in the current existing market, other tools with the same purposes could be found that could be used in the place of the current ones if the case arose.

10.3 Environmental Perspective

10.3.1 The project put into production

To estimate the environmental impact that the project will have, it has been decided to calculate the carbon footprint caused by electricity consumption during its development[7]. Thus, as estimated in the indirect costs section, the

project will consume an electricity equivalent to 199,357 Kwh, taking into account the use of the laptop and the smartphone. Adding to it the nearby consumption of the web server (which is not specified but as it is a simple one, we calculate that it will spend approximately twice that of a Raspberry Pi 3, since it has similar hardware) of about 3W / day, that is 0.072Kwh, during the 5 months 10.8kwh, this makes a total of 210,157 Kwh. Using an online converter we have come to the conclusion that the project will have a footprint equivalent to 77.76 kg of CO2. Being able to reduce if the electricity used to write the memory were subtracted. Another factor to consider is that in order to minimize this impact to the maximum, it has been decided to reuse resources such as the work laptop and the mobile phone, since the same ones that are used in the personal field are used for day to day, thus achieving , not having to use (and consequently spend more electricity) extra tools for the development of the project.

10.3.2 Exploitation

Currently the other applications developed previously called must serve a larger number of users because they are already in post-production, and more complex and powerful web servers are needed to be able to supply all users, this consequently entails a greater ecological footprint, taking into account It also takes into account that a large part of society does not take this type of impact into account, this could mean a greater consumption of CO2. The project that is proposed in this report, if it were carried out once the development was completed, to the market, the capacity and power of the contracted server should also be increased, but doing so taking into account the environmental impact that it could cause, it could allow us look for the best option keeping in mind the objective of producing the least possible impact.

10.3.3 Risk

As has already been said in the previous section, the greatest risk of the project would be the fact of having to increase the characteristics of the web server contracted to be able to satisfy user demand, but in any case, this project is focused on the development of this application and not in the useful life that it could have afterwards, so despite being an important potential risk to consider in post-production, it does not affect us at the moment studied.

10.4 Social Perspective

10.4.1 The project put into production

Carrying out this project provides me with learning both at a technical level, that is, in new technologies, the realization of mobile applications from scratch,

in documenting in a more detailed way and at a social level regarding the world of music, I have gotten more involved In a world that was already very interesting for me, I have informed myself more deeply both about the current situation and the needs of the people within it, I have spoken with people with the same interests because they would give me their opinion and point of view to To be able to focus on my application, and all this has allowed me to continue learning and to channel the application to the one that people need.

10.4.2 Exploitation

The problem that my application is trying to solve has already been tried to be solved by other applications, but from my point of view all these need to go deeper into the direct interaction of their community, making it easier for people to be able to stay with musicians close to you, to be able to express yourself and / or teach other users through a space dedicated to just that. I think my application is mainly focused on helping all these people with a common bond to communicate in the best possible way allowing them to learn and help each other like no other application has done so far. Since this need is real, people seek to find other people who are in the same situation as them and can understand whatever is going on.

10.4.3 Risk

Like all applications that try to unite people with common interests, there is always the risk factor of encountering a person with malicious interests, the application cannot guarantee 100% that there are no users of this type (despite having the figure of an administrator) may be the case of someone who wants to take advantage of other people in one way or another. Not by any particular segment of the population, but if by a person who was more trusting or easy trick, it will depend a lot on the ability that person has to detect toxic people, as they do in their day-to-day lives. Apart from this possible risk, I can not find any other, since the application as explained in the previous section is focused on improving and facilitating coexistence with other musicians and fans.

11 Requirements Specification

11.1 Functional Requirements

11.1.1 User Stories

Users

Login with Google

Me as *new user or already existing user*
I want to*access with Google*
In order to *use and access the application*
Acceptation Criteria

- Have an account in Google

User Registration

Me as *new user*
I want to *register in the application*
In order to *use and access the application*
Acceptation Criteria

- Have a valid account
- The email should be used by another user
- The password should contain 8 characters minimum
- An email, username, password and location must be provided.

Reset Password

Me as *existing user*
I want to *reset my password*
In order to *recover the access to user account*
Acceptation Criteria

- Have a valid account
- Be user account owner and be able to prove it.

Logout

Me as *existing user*
I want to*logout from my user account*
In order to *login with another account or maybe for security reasons*
Acceptation Criteria

- Have a valid account
- The user should be already logged in order to logout

Login with Email

Me as *existing user*
I want to*login in my user account*
In order to *use and access the application*
Acceptation Criteria

- Have a valid account
- Enter a correct password.

User Settings

Me as *existing user*
I want to*my user account*
In order to *modify the settings or add information to my user profile*
Acceptation Criteria

- Have a valid account
- User should be already logged in

Watch user profile

Me as *existing user*
I want to *my user account*
In order to *to watch other user's profiles*
Acceptation Criteria

- Have a valid account
- User should be already logged in

Modify user profile

Me as *existing user*
I want to *access to my user account*
In order to *to modify my profile or add new content*
Acceptation Criteria

- Have a valid account
- User should be already logged in

Follow a user

Me as *existing user*
I want to *access to my user account*
In order to *to follow a user so I can stay tuned to the latest content updates*
Acceptation Criteria

- Have a valid account
- User should be already logged in

Posts

Add video from my YouTube's channel

Me as *existing user*
I want to *access to my user account*
In order to *to add new video from my YouTube's channel to my profile*
Acceptation Criteria

- Have a valid account
- User should be already logged in

Add Post

Me as *existing user*
I want to *access yo my user account*
In order to *add new post*
Acceptation Criteria

- Have a valid account
- User should be already logged in

Add comment to a post

Me as *existing user*
I want to *access yo my user account*
In order to *to make a comment in a specific Post*
Acceptation Criteria

- Have a valid account
- User should be already logged in

Like a Post

Me as *existing user*
I want to *access yo my user account*
In order to *to make a like in a specific post*
Acceptation Criteria

- Have a valid account
- User should be already logged in

Delete a post

Me as *existing user*
I want to*access to my user account*
In order to *delete a Post*
Acceptation Criteria

- Have a valid account
- User should be already logged in
- The Post must be created by the user if it is a simple user account
- The user should have administrator permission if it is Band or Record Label user type.

List of all posts

Me as *existing user*
I want to *access to my user account*
In order to *to see all posts of the use's that I follow*
Acceptation Criteria

- Have a valid account
- User should be already logged in

Search

Search users by proximity

Me as *existing user*
I want to*access to my user account*
In order to *to find other user's nearby*
Acceptation Criteria

- Have a valid account
- User should be already logged in

Search users by tags

Me as *existing user*
I want to *access to my user account*
In order to *to find other user's by tags nearby*
Acceptation Criteria

- Have a valid account
- User should be already logged in

Event

Add Event

Me as *existing user*
I want to *access to my user account*
In order to *add new public Event*
Acceptation Criteria

- Have a valid account
- User should be already logged in

Attendance to an event

Me as *existing user*
I want to *access to my user account*
In order to *notify my attendance to an event*
Acceptation Criteria

- Have a valid account
- User should be already logged in

Cancel attendance to an Event

Me as *existing user*
I want to*access with Google*
In order to *cancel my attendance to an event*
Acceptation Criteria

- Have a valid account
- User should be already logged in

Cancel Event

Me as *existing user*
I want to *access to my user account*
In order to *to cancel an Event*
Acceptation Criteria

- Have a valid account
- User should be already logged in
- Be the Event owner

List of Events nearby

Me as *existing user*
I want to *access to my user account*
In order to *to see all the Events nearby*
Acceptation Criteria

- Have a valid account
- User should be already logged in

Watch Event details

Me as *existing user*
I want to *access to my user account*
In order to *watch the Events detail*
Acceptation Criteria

- Have a valid account
- User should be already logged in

Chat

Chat with a user

Me as *existing user*
I want to *access to my user account*
In order to *chat with a user*
Acceptation Criteria

- Have a valid account
- User should be already logged in

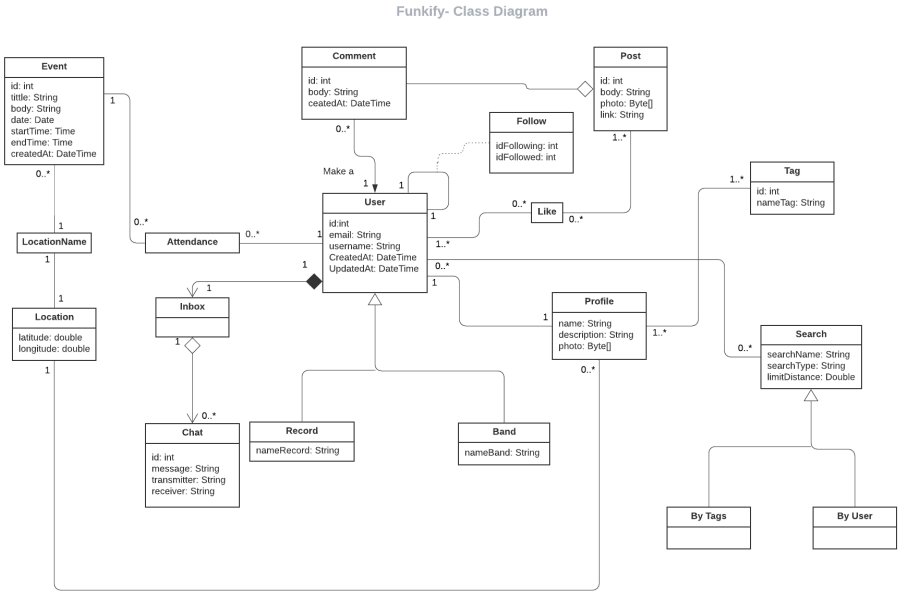
See inbox

Me as *existing user*
I want to *access to my user account*
In order to *see my inbox*
Acceptation Criteria

- Have a valid account
- User should be already logged in

11.1.2 Class Diagram

Based on the specification of the system requirements, it has been decided to create a UML class diagram, in order to have an overview of the structure of the system showing its classes, their attributes, and relationships between the different objects that comprise it. . In this way we can have a general idea of the structure of the project.



Textual Constraints:

1- Foreign keys: (User: id), (Comment: id), (Post: id), (Event: id), (Chat: id), (User-searched: idUser, searchedName), (Follow: idFollowing, idFollowed)

2- The event date (date) must be after the creation date (createdAt)

3- The resulting distance of User-searched (distance) must be less than or equal to limitDistance.

6- A user can Like the same publication only once.

7- A user can notify attendance at the same event only once.

8- To know the distance a user is, the location where it is located is taken into account find the user who created this pet.

9- The date of birth of a user (age) cannot be later than the current date.

10- A user cannot follow himself.

11.2 Non-Functional Requirements

Following Volere's classification, the non-functional requirements found by this system are the following:

Look and feel
• The appearance of the application will be pleasant and will give a sense of the order.

Performance
• All system functionality must respond to the user in less than 5 seconds. Speed and latency.

Usability and humanity
• The application can be used by users without much knowledge. Ease of use.
• The application will provide informational and targeted error messages.
• The application will have well-formed graphical interfaces.

Understanding
• The application will allow the user to select between three languages. Personalization and internationalization.

Operation
• The application will be updated every few months Requirements for new versions.

Maintenance and support
• The application for the moment will work on Android devices but it could be easily scalable to other operating systems, due to the technologies used. Adaptability.
• Maintenance design patterns will be implemented. Maintenance.
• Good programming practices will be used and code will be provided. to understand. Maintenance.

Security
• The application will prevent the user from entering incorrect data. Integrity.

Legal
• The user's personal information will comply with the regulations of the LOPD and the GDPR. Compliance.

12 System Architecture

12.1 Overview

The system architecture is a high level representation of the structure of the application and the interactions within components. It also defines how these components can interact with each other, defining their characteristics in an abstract manner. The architectural design patterns and best practices used to organise iOS applications into logical components have evolved over the last years. Below, the patterns that have been used in the application will be described along with their explanation.

12.2 Why do we need patterns?

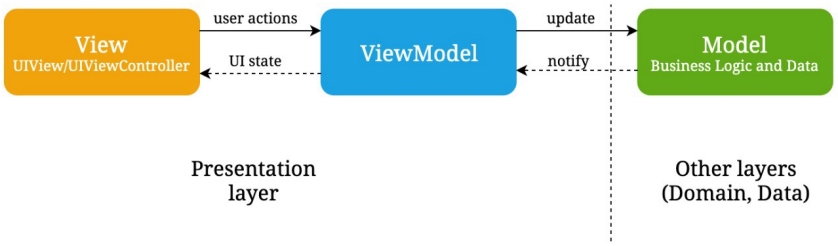
Patterns are being used as a common way to solve specific problems. The problems faced when applications are developed are, at the core level, very similar to other existing problems already solved.
Some time ago, some smart people put together a set of patterns, which in general try to model a problem and give a general solution on how to solve it. Instead of going by myself and crafting a new solution, I have decided to use the patterns that have been proved to work for certain kinds of problems.

12.3 Design Patterns

12.3.1 Model-View-ViewModel

Model-View-ViewModel (MVVM) is a UI architectural design pattern which decouples UI code from the business and presentation logic of an application. Decoupling has been around computer science since the very beginning. The idea behind decoupling two components is that one component can change without affecting the other. Also, it facilitates testing and the evolution of the application itself.
The concept of decoupling has absolute importance in object oriented programming as most of the design patterns are based on it due to other benefits such as reusability, maintainability, testability, which are always desirable properties for systems. As it comes from the name, MVVM divides an application into three compo- nents to help separate the code: the model, the view, and the view model.
View defines the layout, appearance and structure of how things would look like in the UI. The view informs the ViewModel about user interactions and observes state changes exposed by the ViewModel. ViewModel is responsible for wrapping the model and providing state to the UI components. It also defines actions that can be used by the view to send events to the model. However, it shouldn't have access to the view. Model defines core types and implements application business logic. It is the first thing defined when thinking

about the problem, and is completely independent of the view and view-model and reusable in many places across the application.
As mentioned above, the use of this pattern facilitates the maintainability, reusability, and readability of the application, as we are separating concerns between the model and the views. At the end this results in a cleaner and more organized code that might facilitate some designers working on the View and other developers working on the Model or ModelView separately. Furthermore, using the ViewModel as an adapter would be a good solution in the case we have a very tight model that cannot be changed.



• **View** defines the layout, appearance and structure of the UI. The view informs the ViewModel about user interactions and observables state changes exposed by the viewModel.

• **ViewModel** is responsible for wrapping the model and providing state to the UI components. It also defines actions, that can be used by the view to pass events to the model. However, it shouldn't have access to the view.

• **Model** defines core types and implements application business logic. It is completely independent of the view and view-model and reusable in many across the application.

12.3.2 Decorator Pattern

Decoration comes in hand as an alternative to inheritance when giving responsibilities or behavior to an object. The main difference is that decoration happens dynamically so that classes could be decoupled, which as explained above brings a lot of benefits. The problem with inheritance is that it is static, which means that a specific behavior cannot be changed in run time. Another disadvantage is that a class can only inherit from a parent, that means that we could only have one behavior. The Decorator pattern basically wraps an object with another object with specific behavior, which could have been already wrapped inside another one, so it might end up with

multiple behaviors set dynamically without the need of extending a specific class. The Decorator pattern is based on the following OO principle: "Classes should be open for extension but closed for modification". This principle mainly suggests to avoid extending things/code that already work as the possibilities to introduce a bug into the existing functionality is high. Instead, extending the component is not going to hurt any old code. That gives confidence to the developers and helps them to work faster as they know that their new code is not going to affect the existing.
In the end all these benefits give us flexibility in the design.

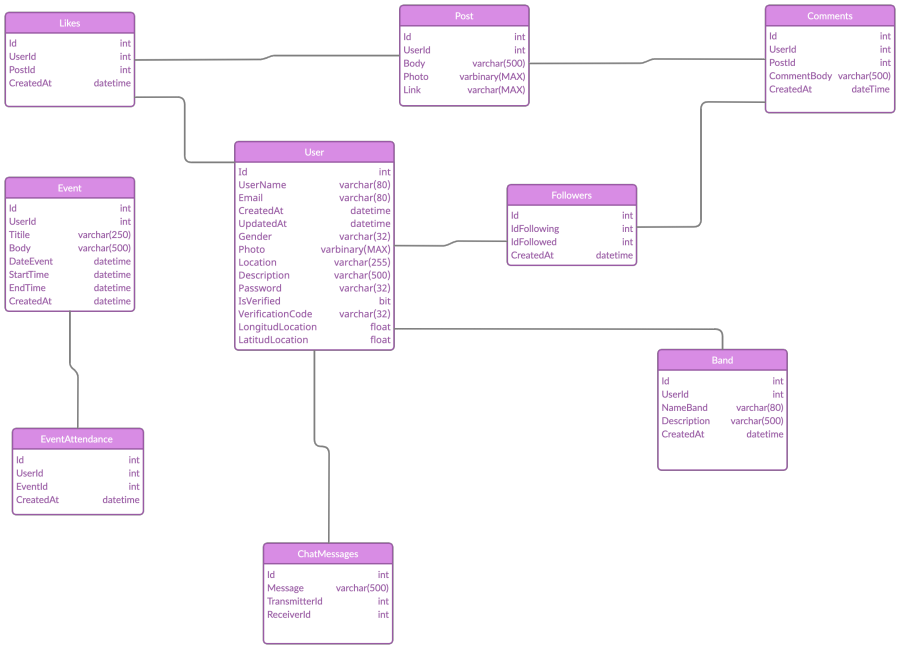
12.3.3 Dependency Injection

Before explaining Dependency Injection I'll explain the concept of dependency. In Software Engineering, dependency is often understood as, for example, an object A relying on something from object B. Therefore, we could say that object A depends on object B. Dependency injection is not a pattern itself but more of a concept which is based on one object supplying the dependencies another object needs. For example, let's say we are building a house from scratch, the structure of the building needs to be designed and there is an urgent need for a new door. Instead of creating this door, we could use one that has been created by others and just 'put it' in the house. This reflects the meaning of injecting a dependency. The door is a dependency, instead of building it by ourselves another door already built is used. We have three principle types of DI: - Constructor injection: When building the object dependent another external object is used at the moment of creation. - Setter injection: The object that has the dependency is already built and the dependency is set using the setter methods. - Interface injection: Is a type of a setter injection but the setter method lives in an interface class, which at the end let defer the injection to the concrete class. Dependency Injection is a very important concept as most of the patterns make use of this. For example, the pattern mentioned before, the decorator pattern, makes use of dependency injection to decorate the object with another behavior.

12.3.4 Singleton

Singleton pattern is a creational design pattern that assures that a class only has one instance and gives a single access point for accessing it, preventing being able to instantiate it more than one. There are particular situations when we need to use this pattern, for example to maintain database connections or creation of threads as we'd run into problems otherwise. In this application, this pattern has been used as I'm sure there should only live one instance when the application runs, so that memory problems might be avoided in the case many instances are created and the garbage collector doesn't work properly.

12.4 Database Diagram



• **Users:** Represents users registered in the application, this table stores all the data related to them, from login information to user profile information. As you can see, the primary key in this table is the Id, but in order to prevent a user from registering with the same email twice, it is not allowed to repeat.

• **ChatMessages:** Represents all messages that are sent between two users, so the identifiers of the two users who are maintaining the chat are stored, differentiating who sends and who receives each message, as well as the message itself.

• **Followers:** This table stores all the pairs of users that are "followed" by their user identifiers, just as one user can only follow another user once the primary key is formed by the two identifiers. users. From this table can be obtained for each user, both the users who follow him and the users he "follows".

• **Likes:** Save all the votes that users make in multimedia posts, for each of them the pair of identifiers is stored, on the one hand what corresponds to the

user who made the vote and on the other hand. to the post you voted for. As you can see, this pair of IDs is the primary key, as a user cannot vote for the same photo twice.

• **Post:** Represents media posts made by users through photographs and accompanying comments, such as how a post is made by a user, this table also stores the identifier of the user who created this post.

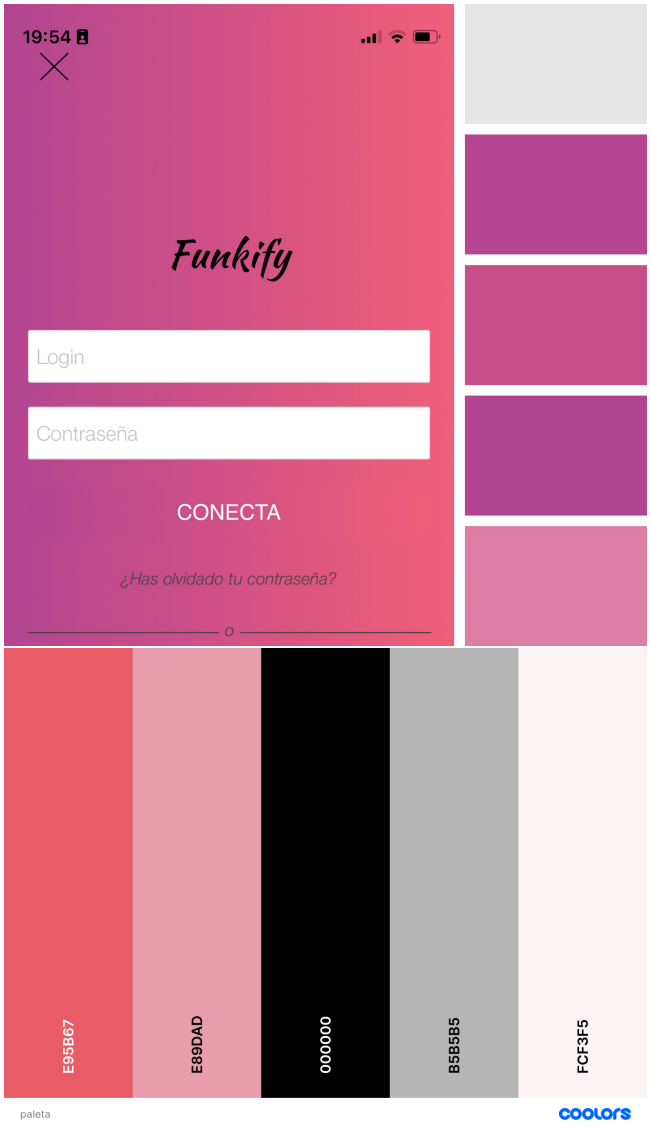
• **Comments:** It stores all the comments that users post in the different threads of the forum, the way to relate these comments to the commented thread is through the identifier of the user who comments and the identifier of the thread in which he does so. In this case, the primary key is only the comment identifier, as a user is allowed to create all the comments they want in a single thread.

•**Event:** Saves information for all events created by users of the application. When creating an event, users must enter all the information in the system, including the date and day of the event, an optional photo, a description of it, as well as all the information related to its location.

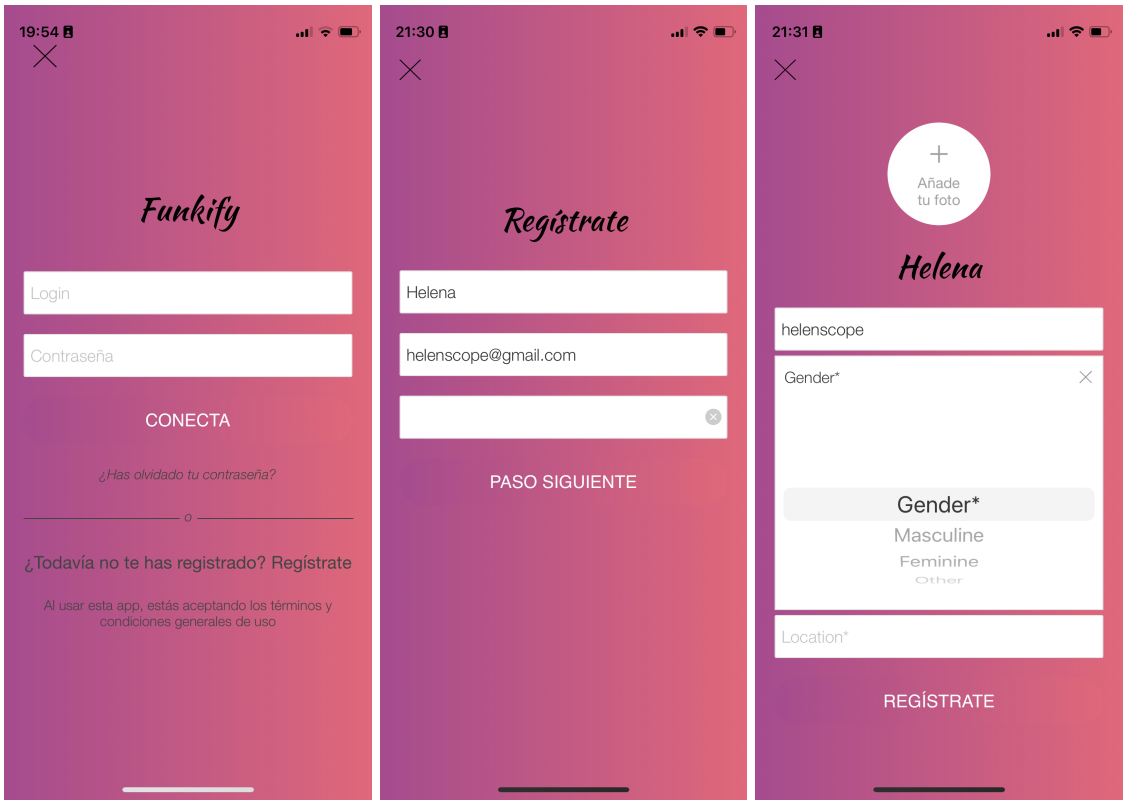
•**EventAttendance:** This table is responsible for saving all attendees to an event created by a user, to do so, the table stores the identifiers of the user who wants to attend and the event you want to attend. . Since a user can only attend the same event once, this pair of identifiers has been set as the primary key of the class.

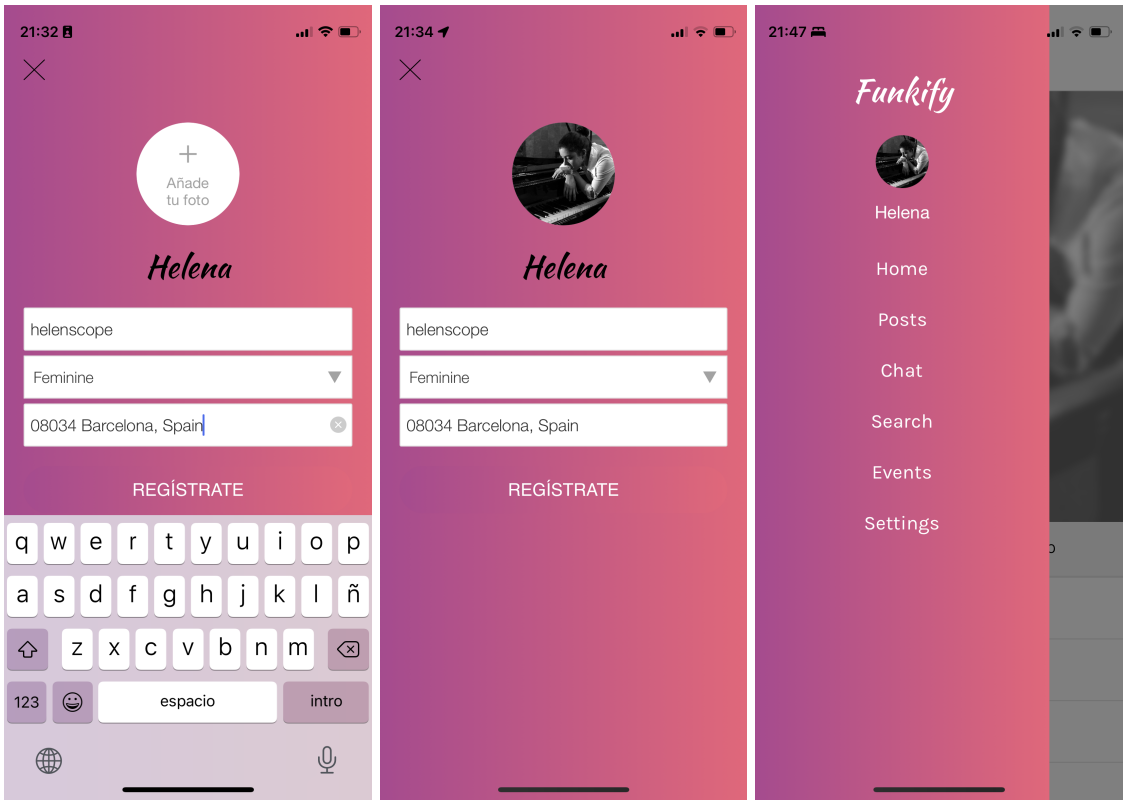
12.5 Graphic Interface Design

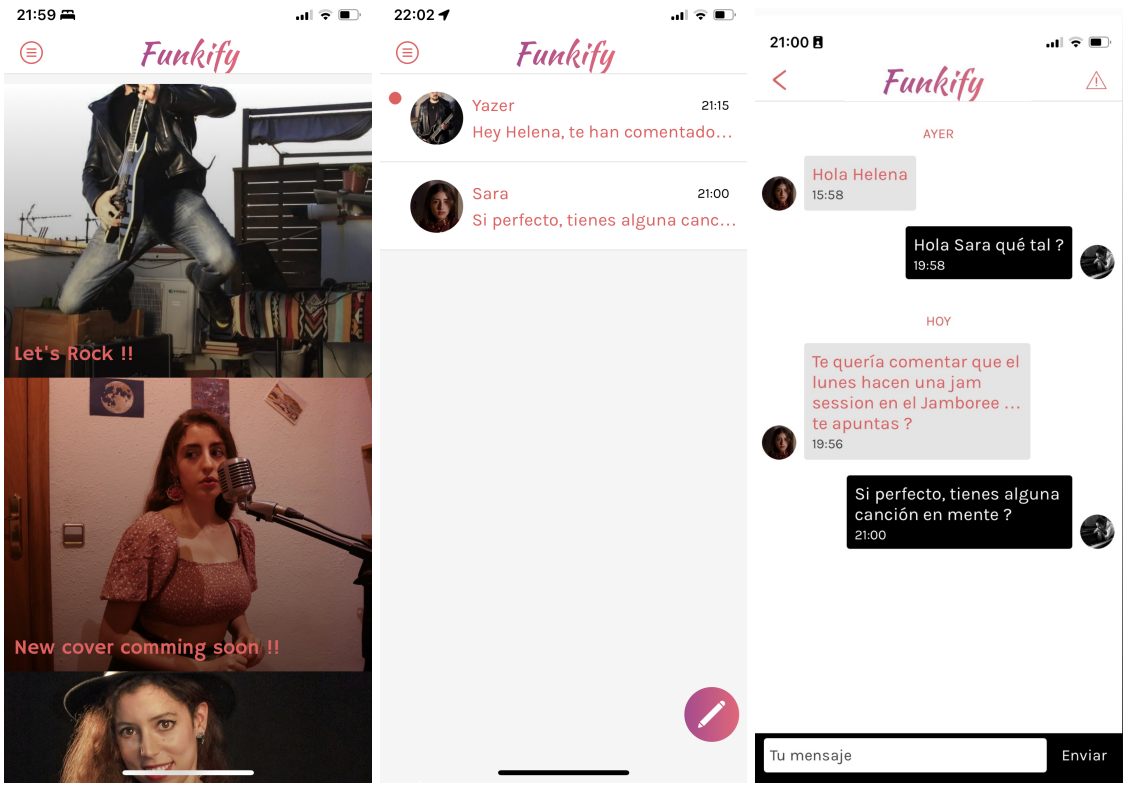
12.5.1 Colors Palette

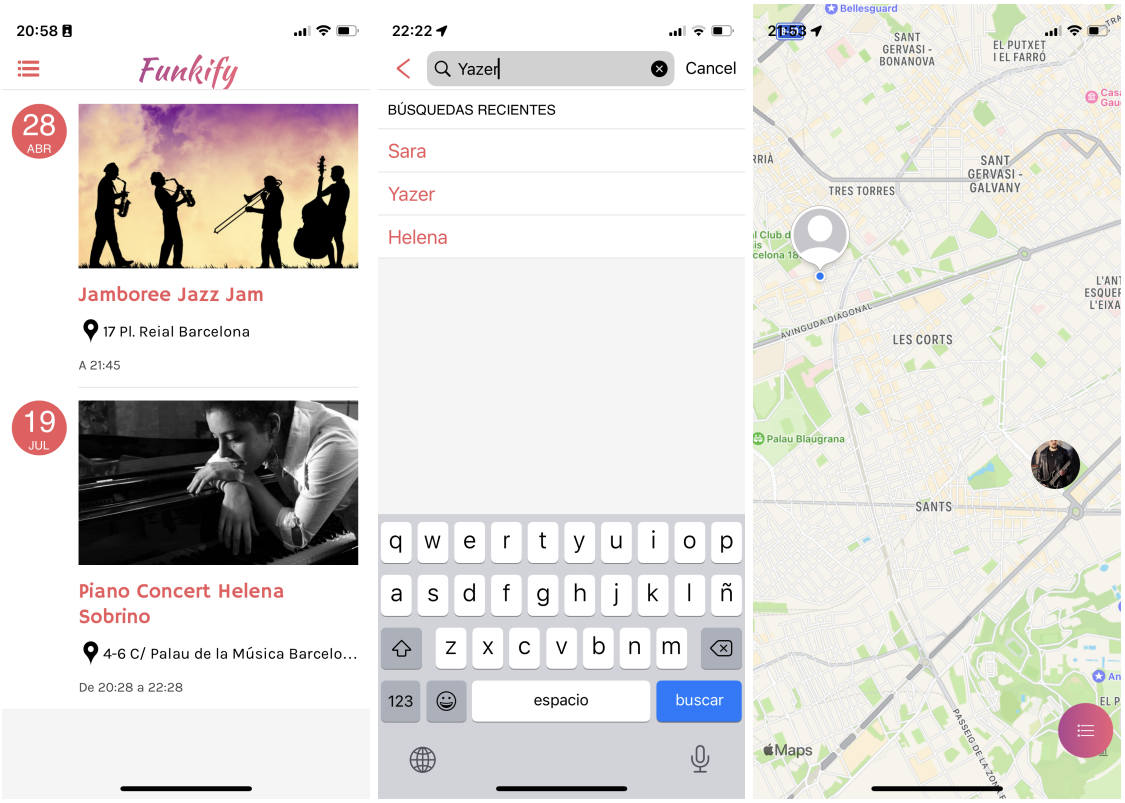


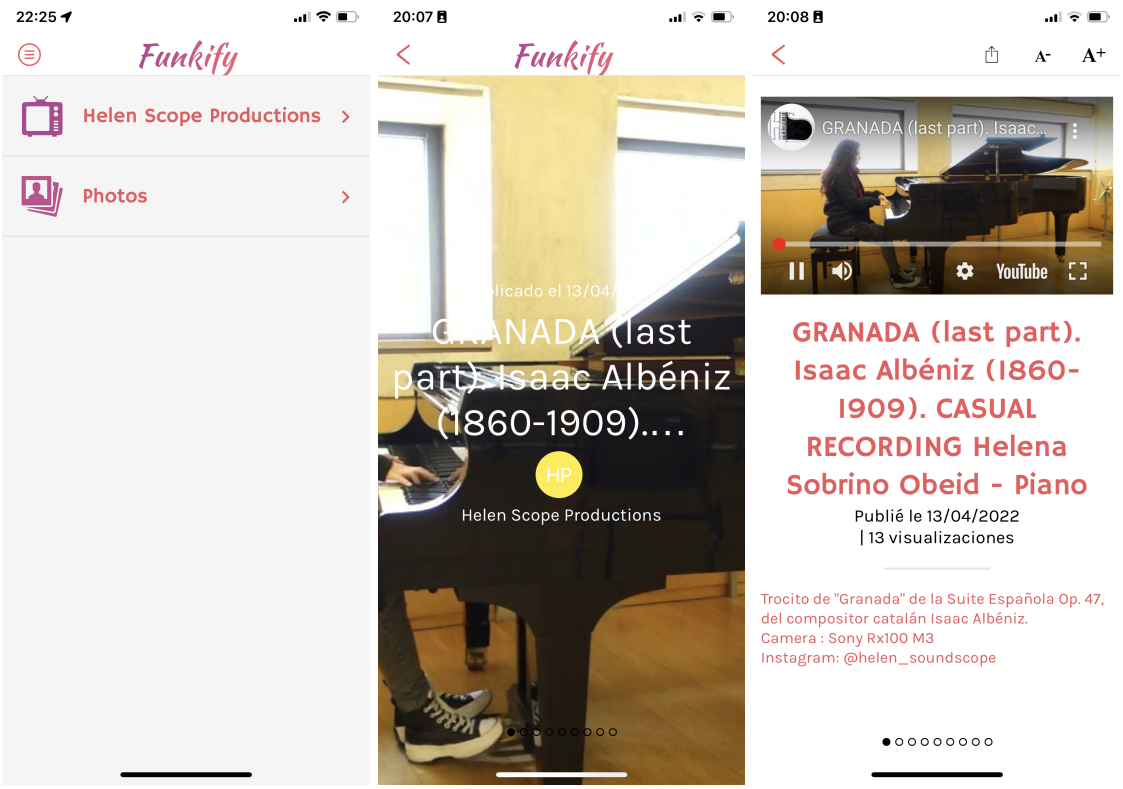
12.6 ScreenShoots of the Application

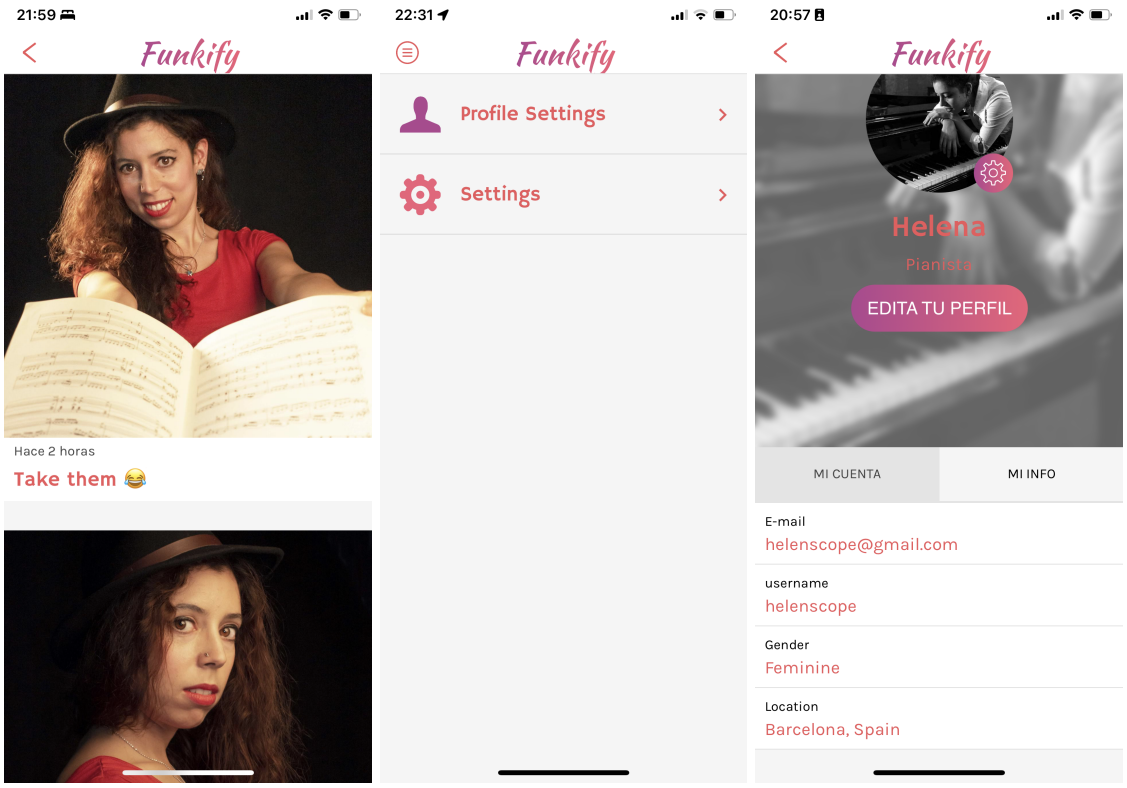


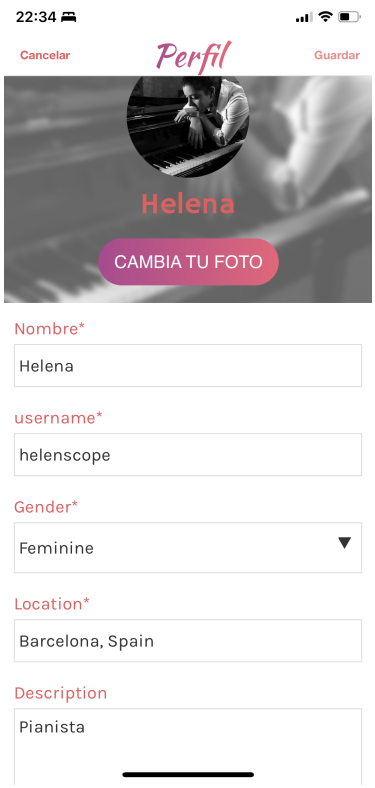

21:32
Añade
tu foto
Helena
helenscope
Feminine
08034 Barcelona, Spain
REGÍSTRATE
q w e r t y u i o p
a s d f g h j k l ñ
z x c v b n m
123
espacio
intro
21:34
Helena
helenscope
Feminine
08034 Barcelona, Spain
REGÍSTRATE
21:47
Funkify
Helena
Home
Posts
Chat
Search
Events
Settings


21:59
Funkify
Let's Rock !!
New cover comming soon !!
22:02
Funkify
Yazer
21:15
Hey Helena, te han comentado...
Sara
21:00
Si perfecto, tienes alguna canc...
21:00
Funkify
AYER
Hola Helena
15:58
Hola Sara qué tal ?
19:58
HOY
Te quería comentar que el lunes hacen una jam session en el Jamboree ... te apuntas ?
19:56
Si perfecto, tienes alguna canción en mente ?
21:00
Tu mensaje
Enviar


20:58
Funkify
28
ABR
Jamboree Jazz Jam
17 Pl. Reial Barcelona
A 21:45
19
JUL
Piano Concert Helena Sobrino
4-6 C/ Palau de la Música Barcelo...
De 20:28 a 22:28
22:22
Yazer
Cancel
BÚSQUEDAS RECIENTES
Sara
Yazer
Helena
q w e r t y u i o p
a s d f g h j k l ñ
z x c v b n m
123
espacio
buscar
Bellesguard
SANT GERVASI - BONANOVA
EL PUTXET I EL FARRÓ
SANT GERVASI - GALVANY
TRES TORRES
AVINGUDA DIAGONAL
LES CORTS
Palau Blaugrana
SANTS
Maps


22:25
Funkify
Helen Scope Productions
Photos
20:07
Funkify
GRANADA (last part). Isaac Albéniz (1860-1909)....
HP
Helen Scope Productions
20:08
GRANADA (last part). Isaac...
YouTube
GRANADA (last part). Isaac Albéniz (1860-1909). CASUAL RECORDING Helena Sobrino Obeid - Piano
Publié le 13/04/2022
| 13 visualizaciones
Trocito de "Granada" de la Suite Española Op. 47, del compositor catalán Isaac Albéniz.
Camera : Sony Rx100 M3
Instagram: @helen_soundscope


21:59
Funkify
Hace 2 horas
Take them
22:31
Funkify
Profile Settings
Settings
20:57
Funkify
Helena
Pianista
EDITA TU PERFIL
MI CUENTA
MI INFO
E-mail
helenscope@gmail.com
username
helenscope
Gender
Feminine
Location
Barcelona, Spain


22:34
Cancelar
Perfil
Guardar
Helena
CAMBIA TU FOTO
Nombre*
Helena
username*
helenscope
Gender*
Feminine
Location*
Barcelona, Spain
Description
Pianista

13 System Development

13.1 Implementation of User Stories

This section explains in detail the development of some of the most complex user stories in the project.

Geolocation

One of the strong points that gives value to the application is the fact that it incorporates geolocation. This application allows the user to obtain a list of users/events by proximity by assigning them a specific location to both user and event.

To achieve that, it is essential to use the MapKit and CoreLocation, iOS APIs. CoreLocation is responsible for handling location data while MapKit has to do with visual operations, such as rendering maps among many other functionalities.

Thus, once we have obtained a location, it contains coordinates, that is, a latitude and a longitude. These two fields are the ones that we will save for all those models that need to use the location, that is, the Users and the Events.

User Registration and Login

Before being able to use the application, the user must register in the system. To do this, within the main screen of the application you will select the option to register.

Entering user information

The user registration consists of two parts, initially the information that will be used to access the application and to identify this new user must be entered, that is, a valid email, a password and a nickname. Both the password and the username must meet special conditions to guarantee minimum security requirements.
These conditions are as follows:

• Username: does not have a minimum length but a maximum length of 80 characters, it can only contain uppercase and lowercase letters, numbers, hyphens and underscores.

• Password: must have a minimum length of 8 and a maximum of 30 characters, the use of upper and lower case letters and symbols are allowed, but not spaces.
The password must be typed twice to avoid typo errors, since it is more likely

that user made mistakes typing the password and don't realize it because it is masked.
Once the access and identification data have been created within the application, you will be asked to create a user profile, this profile includes entering a set of information. Of these fields to enter, most are optional, others are mandatory:
• Photo: Optional field, if the user does not enter any image, the system puts one by default.

• Name: Required field, it is only necessary to write a first name.

• Gender: Required field, since one of the three options provided must be selected, "man", "woman" or "I prefer not to say". It is a required field but instead allows you to select the level of privacy that the user prefers.

• Location: Required field, since the application is strongly linked to offering users information on the proximity of other users, to ease their interaction. In any case, the privacy of each of the users is also very important, and to maintain it, in this field each user can choose the level of privacy they prefer, writing, for example, from the name of the street where they live , to the zip code of the neighborhood in which you live. But it must be taken into account that all distances with other users will be calculated according to the location provided. The process that allows us to register this location and the way in which the application does it, has been explained in the previous section.

• Description: Optional field, if a description is not entered, this field remains empty.

User verification Firebase Authentication is used for user verification, which will greatly simplify the user registration and login process. It allows the management of users from one or more access providers.
At the implementation level, it begins by importing the authentication library (FirebaseAuth) to the project code, having previously registered the project database in Firebase. Once logged in, the state of the application is saved in the same Firebase.

Search

Search by user When a user wants to search for a user in the system, the search will return a list of all users that meet the criteria described by the user, in order from least to greatest distance.
Search by tags The search allows the user to perform a search by applying a set of tags, the search will obtain a list of all the users that contain the tag or tags that the user has written in, in order from least to greatest distance.

Once the search has been narrowed down and the user has clicked on the search button, the notification is sent to the backend. The sending of notifications as well as the calculation of coordinates in terms of proximity towards the user doing the search are carried out using Apple's Core Location framework which is jointly integrated with MapKit to show nearby users on the map according to preferences.
In this functionality, the most complex part of querying in the backend is the fact that the API call only returns those users that are within the distance range selected by the user. To calculate these distances, the location of the user who owns each of the pets that meet all the other filters is used, compared to the location of the user who made the query, that is, the user who is logged in to the application. This location, as seen in previous sections, is stored in the tables using the latitude and longitude, these two fields are the ones used through the Haversine formula to calculate these distances and determine if they are inside or outside the search done.
Once the search has been done with its corresponding filters applied, the API call returns a list with all the users that meet the conditions, ordered from the least to the greatest distance from the user.

13.2 Used Resources

13.2.1 Programming Languages/Technologies

Swift

Swift is a multiplatform, compiled, general-purpose programming language built using a modern approach to safety, performance, and software design patterns developed by Apple Inc. Swift was created as a replacement for Objective-C language with similar features, but in a "safer" way, making it easier to catch software bugs.
It's the main programming language used to develop the application.

iOS

Is the operating system for Apple's mobile(iPhone) and iPod Touch. Written in c, c++, Objective-C, Swift, and assembly language. We are using the latest version which at this moment iOS 15.4.1 for the development of the application.

Latex

Latex is a markup language created in 1980 by Leslie Leslie Lamport It is written in Tex macro language and uses the same Tex typesetting for formatting the output text. This memory is written in LaTex through Overleaf web that is described below in the tools section.

13.2.2 Tools

Xcode

Xcode is an IDE for developing software for macOS, iOS, watchOS, and tvOS. It is the official Apple developing IDE that works together with Interface Builder (a grahical tool for creating user interfaces)for creating and publishing apps. Includes also a collection of compilers from the GNU Project (GCC) and can compile C, C++, Objective-C, Objective-C++, AppleScript, and Java code using a wide range of programing models.

Firebase

Firebase is a platform for the development web and mobile applications. For this project the following set of tools have been used:

RealtimeDatabase NoSQL database organized in the form of a JSON tree hosted in the cloud that allows you to store and synchronize data between your users in real time.

Firebase Auth User authentication system using only client-side code. Includes authentication using login providers such as Facebook, GitHub, Twitter, Google, Yahoo, and Microsoft; as well as the classic email and password login methods. Additionally, it includes a user management system whereby developers can enable user authentication with email and password to be stored in Firebase. It also provides other features such as account recovery and verification, both by email and by SMS, among many others.

Cacoo

Cacoo is an online service that allows you to create, share, and publish diagrams. It is a user-friendly online drawing tool that allows you to create a variety of diagrams such as site maps, wireframes, UML diagrams, software and network diagrams, etc ...

Overleaf
Overleaf is a collaborative web-based writing and publishing service that facilitates the writing of academic texts. It contains a LaTeX interpreter, which is a highly customizable document preparation system that allows the inclusion of packages for specific purposes.

Coolors
Coolors is a color palettes app and website where you can generate your color schemes, pick colors from your photos or simply choose existing palettes on the web or app and export them as images or PDFs.

14 References

References

[1] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Addison-Wesley, Reading, Massachusetts, 1993.

[2] K.Schwaber and J.Sutherland. La guía definitiva de scrum: Las reglas del juego. `https://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Spanish.pdf` Juliol 2016. (page 13).

[3] Statista. Number of social network users worldwide from 2017 to 2025 `https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users`

[4] Trello. Trello. url `https://trello.com` Accedit el 24-02-2019. (page 15).

[5] Github.`https://github.com/` Accedit el 24-02-2019. (page 15).

[6] TeamGantt documentation: `https://support.teamgantt.com/`

[7] CO2 footprint calculation `https://www.ceroco2.org/calculadoras/`

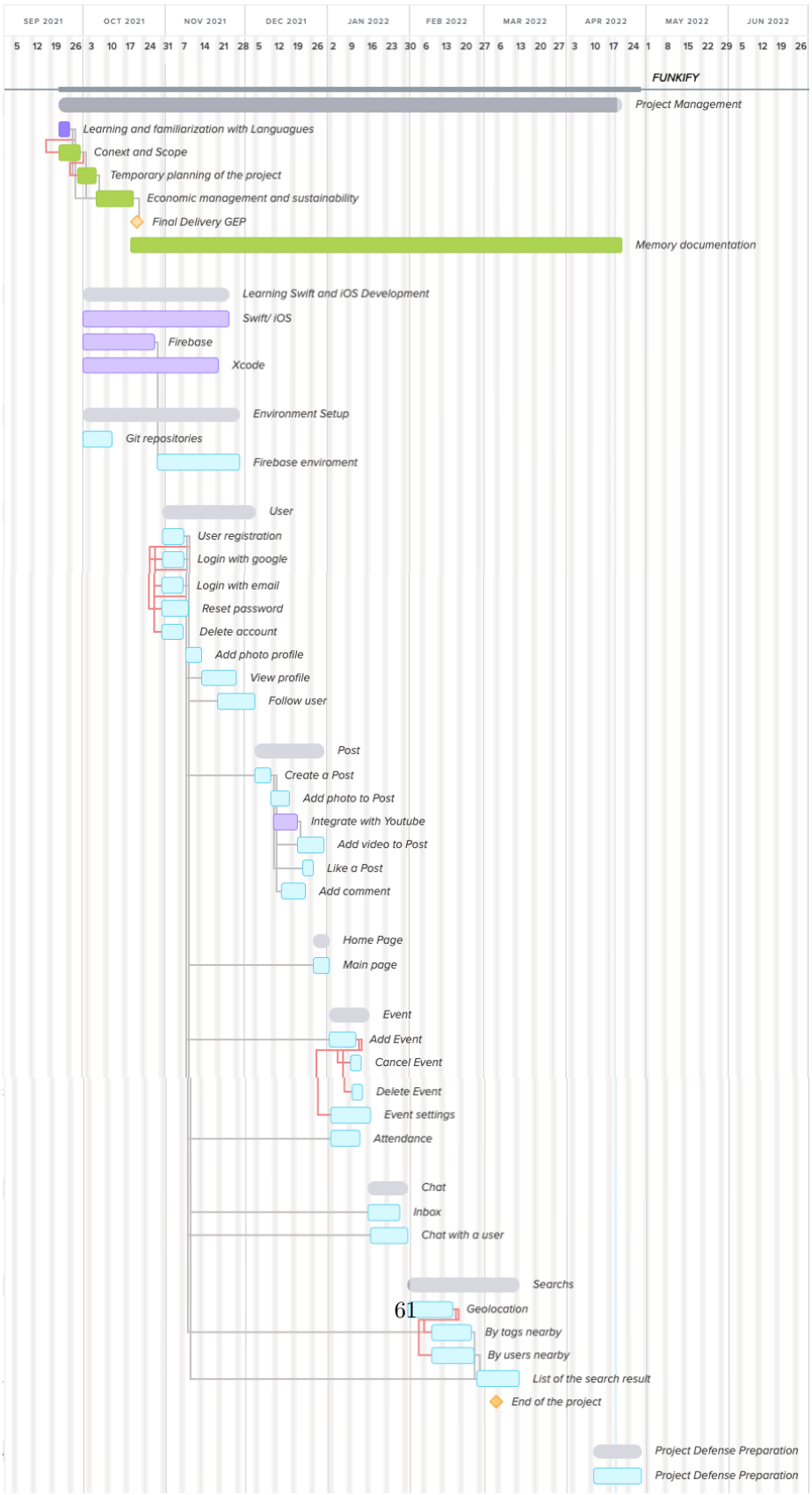
[8] PayScale: Salary data career research center(spain), 2020. URL `https://www.payscale.com/research/ES/Location=Barcelona/Salary`

[9] OECD. Average anual hours actually worked. `https://www.oecd-ilibrary.org/content/data/data-00303-en`

[10] SWIFT. About Swift language. `https://www.swift.org/about/`

15 Appendices

15.1 Appendix A : Gantt Diagram



15.2 Appendix B: Planing Table

Nº Task	**Name of the Task**	**Hours**	**Dependencies**
1	Project Management	10h	
2	Learning Swift and iOS Development	70h	
3	Enviroment Setup	30h	
4	User Registration	12	2
5	Login with email	10h	4
6	Login with Google	10h	4
7	Reset password	8h	4
8	Delete account	8h	4
9	View user profile	20h	4
10	Follow user	20h	4
11	Create a Post	20h	
12	Add photo to Post	20h	
13	Integrate with Youtube	40h	
14	Add voideo to Post	20h	11-12
15	Like a Post o	8h	10-11
16	Add comment	20h	
17	Home page	6h	
18	Add Event	35h	
19	Cancel Event	8h	16
20	Delete Event	8h	16
21	Event settings	20h	16
21	Attend an event	8h	16
23	Inbox	25h	4
24	Chat with a user	25h	4
25	Geolocation	15h	
26	Search by tags nearby	15h	
27	Search by users nearby	15h	
28	Project defense preparation	15h	
TOTAL	-	**521h**	

15.3 Appendix C: Planning Table with Costs

Nº Task	**Name of the Task**	**Role**	**Hours**	**Price/Hour**	**Cost**
1	Project Management	PM	10h	20€	200€
2	Learning Swift and iOS Development	iOSD	70h	16€	1120€
3	Enviroment Setup	iOSD	30h	16€	480€
4	User Registration	iOSD	12h	16€	192€
5	Login with email	iOSD	10h	16€	160€
6	Login with Google	iOSD	10h	16€	160€
7	Reset password	iOSD	8h	16€	128€
8	Delete account	iOSD	8h	16€	128€
9	View user profile	iOSD	20h	16€	320€
10	Follow user	iOSD	20h	16€	320€
11	Create a Post	iOSD	20h	16€	320€
12	Add photo to Post	iOSD	20h	16€	320€
13	Integrate with Youtube	iOSD	40h	16€	640€
14	Add video to Post	iOSD	20h	16€	320€
15	Like a Post	iOSD	8h	16€	128€
16	Add comment	iOSD	20h	16€	320€
17	Home page	iOSD	6h	16€	96€
18	Add Event	iOSD	35h	16€	560€
19	Cancel Event	iOSD	8h	16€	128€
20	Delete Event	iOSD	8h	16€	128€
21	Event settings	iOSD	20h	16€	320€
22	Attend an Event	iOSD	8h	16€	128€
23	Inbox	iOSD	25h	16€	400€
24	Chat with user	iOSD	25h	16€	400€
25	Geolocation	iOSD	15h	16€	240€
26	Search by tags nearby	iOSD	15h	16€	240€
27	Search by users nearby	iOSD	15h	16€	240€
28	Project defense preparation	PM	15h	20€	300€
TOTAL	-	-	**521h**	-	**8436€**