



# One More Song

**Manex Garaio Mendizabal**

Máster en Diseño de UX y dispositivos móviles

**Alberto Zamarrón Pinilla y Carlos Caballero González**

20 de junio del 2016

© (Manex Garaio Mendizabal)

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

## FICHA DEL TRABAJO FINAL

<b>Título del trabajo:</b>	One More Song
<b>Nombre del autor:</b>	Manex Garaio Mendizabal
<b>Nombre del consultor:</b>	Alberto Zamarrón Pinilla and Carlos Caballero González
<b>Fecha de entrega (mm/aaaa):</b>	06/2016
<b>Área del Trabajo Final:</b>	
<b>Titulación:</b>	Máster en Diseño de UX y dispositivos móviles

### Resumen del Trabajo (máximo 250 palabras):

Este trabajo resume el proceso de creación de la aplicación OneMoreSong, aplicación de iOS que tiene como objetivo facilitar a las usuarias la búsqueda de concierto de los grupos que les gustan, además de poder ver los conciertos programados en su zona, independientemente del grupo que sea.

A lo largo del trabajo se hablará sobre el análisis previo, la investigación con usuarios, el diseño, la evaluación de la usabilidad y el desarrollo y las pruebas realizadas. Para finalizar se incluyen las conclusiones finales y las posibles mejoras que se le podrían hacer a la aplicación.

**Abstract (in English, 250 words or less):**

This report summarises the creation process of the app OneMoreSong, an iOS app that aims to facilitate the search of concerts of the bands users like, besides showing all the concerts scheduled for the area the user is located in, regardless of the band.

The report will start with the initial analysis, and will continue following the phases the project has had: investigation with users, design, evaluation of the user experience, development and testing, and the final conclusions and future work that may be done if the development of the app continues.

**Palabras clave (entre 4 y 8):**

iOS, API, concert search, OneMoreSong

# Index

## Aurkibidea

<b>1. Introduction</b>	<b>1</b>
1.1 Context and motivation of the Dissertation	1
1.2 Objectives of the Dissertation	1
1.3 Focus and method	1
1.4 Dissertation planning	2
1.5 Brief summary of the obtained products	4
1.6 Brief description of the objectives of the remaining chapters of the report	4
<b>2. Definition of the app</b>	<b>5</b>
2.1 Title	5
2.1 Operating system	5
2.1 Category	5
2.1 A priori end-users	5
2.1 Benchmarking	5
Bandsintown <sup>2</sup>	6
Songkick <sup>3</sup>	6
Gigbeat <sup>4</sup>	6
Thrillcall <sup>5</sup>	7
Timbre <sup>6</sup>	7
<b>3. Research with users</b>	<b>8</b>
3.1 Field work	8
Script	8
Sample	9
3.2 Analysis of the gathered information	10
Focus group	10
Interview	10
3.3 Main discoveries	10
3.4 Person modelling	11
3.5 Functionalities	11
<b>4. Design</b>	<b>13</b>
4.1 Initial design	13
Information architecture	13
Wireframes	14
Prototyping	16
4.2 Final design	19
Information architecture	20
Prototyping	20
User flows	22
<b>5. User experience evaluation</b>	<b>24</b>
5.1 Evaluation of the prototype	24
Heuristic evaluation	24
Evaluation with users	25
5.2 Evaluation of the app	26
Script	26

Sample .....	27
Main discoveries .....	27
<b>6. Development and Testing .....</b>	<b>28</b>
6.1 Songkick API <sub>9</sub> .....	28
6.2 Database diagram .....	29
6.3 Class diagram .....	29
User Interface layer .....	30
Business Logic layer .....	31
Data layer .....	32
6.4 Testing .....	33
Methodology .....	33
Test types .....	33
<b>7. Conclusions and future work .....</b>	<b>34</b>
7.1 Conclusions.....	34
7.2 Future work .....	34
<b>8. Bibliography .....</b>	<b>36</b>
<b>Appendix A: Scope redefinition .....</b>	<b>37</b>
A.1 Redefined scope.....	37

## List of figures

1. Figure: Gantt Diagram of the Dissertation .....	3
2. Figure: Information architecture of the initial design. ....	13
3. Figure: Wireframe of the Feed page.....	14
4. Figure: Wireframe of the Bands page.....	14
5. Figure: Wireframe of the VenueXYZ page.....	15
6. Figure: Wireframe of the BandXYZ page.....	15
7. Figure: Wireframe of the Plans page. ....	15
8. Figure: Wireframe of the Cities page. ....	15
9. Figure: Wireframe of the ConcertXYZ page. ....	16
10. Figure: Prototype of the Bands page.....	17
11. Figure: Prototype of the Feed page. ....	17
12. Figure: Prototype of the Cities page. ....	17
13. Figure: Prototype of the Venues page. ....	17
14. Figure: Prototype of the VenueXYZ page.....	18
15. Figure: Prototype of the BandXYZ page.....	18
16. Figure: Prototype of the Plans page. ....	18
17. Figure: Prototype of the CityXYZ page. ....	18
18. Figure: Prototype of the ConcertXYZ page.....	19
19. Figure: Information architecture of the final design.....	20
20. Figure: Prototype of the BandXYZ page.....	21
21. Figure: Prototype of the Plans page. ....	21
22. Figure: Prototype of the Bands page.....	21
23. Figure: Prototype of the Feed page.....	21
24. Figure: Prototype of the ConcertXYZ page.....	21
25. Figure: Database diagram of the app. ....	29
26. Figure: Class diagram of the UI layer. ....	30
27. Figure: Class diagram of the Business logic class. ....	31
28. Figure: Class diagram of the Data layer. ....	32

# 1. Introduction

## 1.1 Context and motivation of the Dissertation

Nowadays, listening to music is more widespread than ever before, because of the different means of listening to music (streaming services like *Deezer*, with our mobile phone/mp3, etc.) and the increasing variety of bands. Consequently, the amount of music fans has also increased. That fact also increases the likelihood for a person to attend one live concert in order to watch bands that will normally hear through her headphones.

However, a great amount of live shows is scheduled every weekend (and also during the week) so it is hard to filter them to the ones we are interested in. So, in a nutshell, that is the motivation of the Dissertation, provide a tool that will ease the process of looking for concerts and will increase the probability for people to attend live shows. Besides, many different APIs are available that facilitate the access of such data, so that the workload dramatically decreases to fetch it.

## 1.2 Objectives of the Dissertation

The objectives are fairly similar to the aforementioned motivation. The app intends to be a useful help for its users, so that they can get to know about the events they like without investing a lot of their time to do that research. The app ideally will have a one-time setup and after that will just run and provide the information users need.

Depending on the habits of each user the amount of information shown could be overwhelming so it is necessary that the information is formatted and shown in a way that is efficient and easy to understand.

## 1.3 Focus and method

The focus of the Dissertation will be to develop a product that will help to accomplish the aforementioned goals starting from the beginning, i.e. it will not just be an improvement of a currently developed product. There are two reasons for that. The first one is that the area the app is included in is not matured yet, so there is no standard that users expect to see in the app, therefore increasing the flexibility of design decisions and allowing to start fresh. However, it should be noted that starting a new design will also require to look to design decisions made by the developers of the already existing products. That research will be explained further in following chapters.

On the other hand, adapting a current product is not an option, as they are based on proprietary software, so it is not possible to use their source code and make adjustments and improvements starting from their code base.

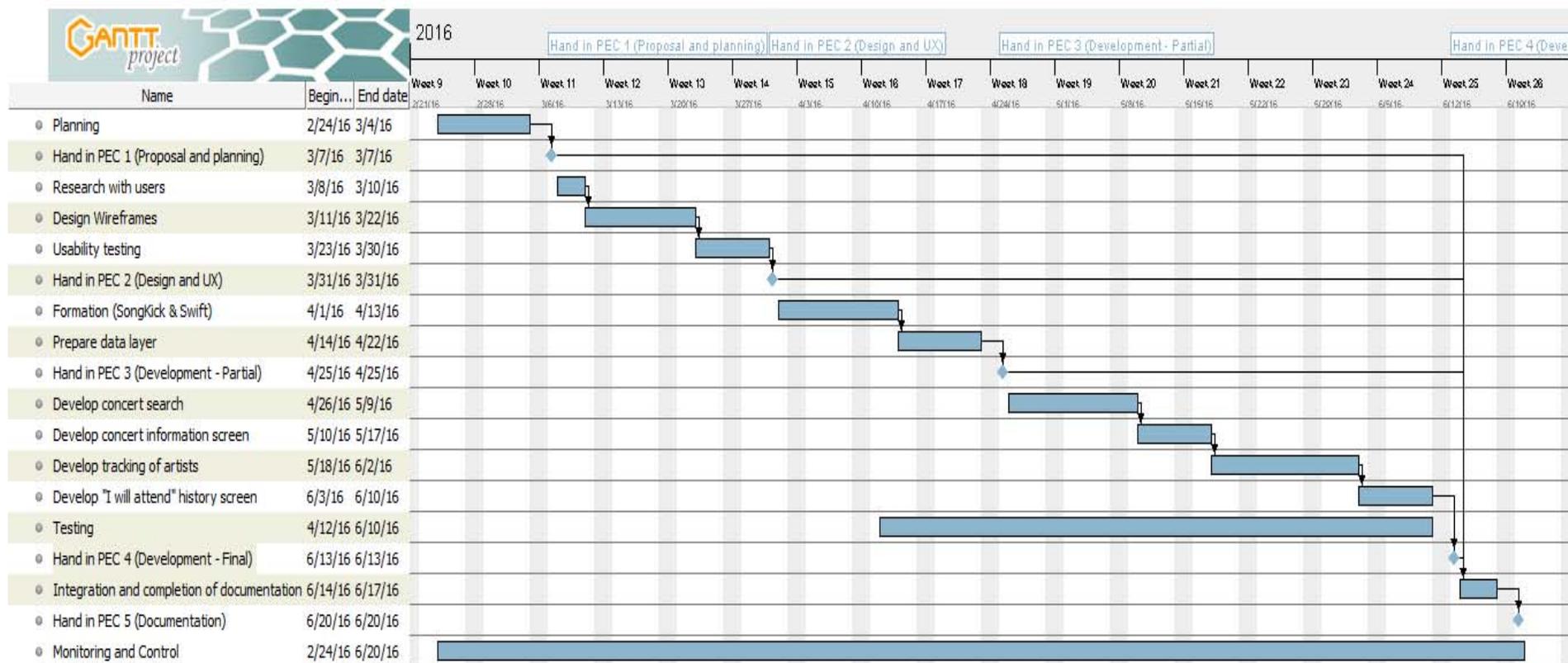
## 1.4 Dissertation planning

The planning of the Dissertation has been very influenced by the academic schedule of the master, as it required frequent hand-ins that determined the phase of the project that should be worked on each week. Therefore, the deliverables of the project have also been determined by the master's calendar, and are the following:

- Proposal and planning: It contains the initial idea of the app, and also information about the research of potential functionalities and the planning of the project.
  - Due on the 7<sup>th</sup> of March.
- UI design and UX: This document will specify the screens of the app.
  - Due on the 31<sup>st</sup> of March.
- Development and UX Evaluation (partial hand-in): Non complete implementation of the app.
  - Due on the 25<sup>th</sup> of April.
- Development and UX Evaluation (final): Final implementation of the app.
  - Due on the 13<sup>th</sup> of June.
- Documentation: Document that reflects all the work carried out throughout the Dissertation.
  - Due on the 20<sup>th</sup> of June.

As the master dictates the development rhythm, the date range for each of the tasks to be included in the planning is already taken care of. All tasks will be sequentially organised but two: *Testing* and *Monitoring and Control*. The latter expands over the whole timeframe of the Dissertation in order to keep track of the progress of the work and update it accordingly should any task be ahead or behind schedule. *Testing* will also be independent of the hand-ins as it spans throughout the entire development process, as it is necessary to make sure each functionality works properly before advancing to the next.

The Gantt diagram for the project can be seen on the next page.



1. Figure: Gantt Diagram of the Dissertation.

### 1.5 Brief summary of the obtained products

Besides the hand-ins of the Dissertation itself, which include the design prototypes and the partial implementation of the app, two major products will be obtained. The most important of them is the application, which will be the final result of all the work done during the Dissertation, both in design and implementation levels. The other product is a demonstration video that will show the design of the app and its functionalities.

### 1.6 Brief description of the objectives of the remaining chapters of the report

The remaining chapters of this report will explain the process the app has gone through from the first stages to the final ones. Those include the initial analysis, the design, the development and testing and the final conclusions.

## 2. Definition of the app

This chapter describes the outcome of the initial analysis process of the app, such as the type of application that will be developed, and an introductory research of the current state of the art and the potential users the app may have.

### 2.1 Title

#### **One more song**

The name originates from the chant that is so usual at live concerts, when attendees ask the performing band to play one more song, even if they have already left stage.

### 2.2 Operating system

The app is going to be developed for the iOS operating system. The main reason why iOS has been selected is that the developer has a personal interest in developing for that OS in order to enhance his skills with an unknown OS. Furthermore, iOS is one of the Operating Systems with more handheld devices (only after Android)<sup>1</sup> what means there will be a great amount of potential users if the app is released to the general public.

### 2.3 Category

The app could be classified as an app devoted to Music and/or Entertainment. However, looking at the apps classified as the latter, this app has a closer bond with those apps in the former category. Therefore, the app will be regarded as a Music app.

### 2.4 A priori end-users

The end-users for this app are those who have ever attended a live music show or those who intend to do so. Among that wide range of people there will be different degrees of use. Some of them will just want to add their favourite bands to the app and check it once in a while. Some others, however, will make a more active use of it, and perform searches for those upcoming shows according to different criteria, such as, location and/or date. That way, they can make plans for their trip weekend or for their holiday trip.

### 2.5 Benchmarking

In this section an analysis of similar applications will be performed. Contrary to initial belief, there is a wide array of different apps that have similar purposes. However, there is one inconvenient with some of the apps that have been analysed; they are based in the US. Thus, none or very little information of European concerts can be found in some of them.

On the other hand, all apps that have been analysed, regardless of their functionalities, have one feature in common; they all use APIs to fetch the data about concerts. Some of those apps use their own API (e.g. Bandsintown and Songkick), while others use the API of Bandsintown and the like.

### Bandsintown<sub>2</sub>

Bandsintown is one of the major apps in the field. The app is an auxiliary way of fetching the data available in their website. It allows users to track their favourite artists and receive notifications if one of them is playing nearby. In order to suit the preferences of the user it allows configuring the maximum radius, so that the app notifies a given concert only if it is located in that range. Besides, it also allows restricting the dates in which those concerts should be.

In order to ease the initial work of tracking each artist on an individual basis, Bandsintown can synchronise that list from the music library available in the device and also in popular online music streaming services (for example, Spotify and Last.fm). It can also be connected with social networks in order to find friends or invite them to use the app.

On the other hand, Bandsintown is capable of looking for concerts based on the artist, the city or the venue. Once the user chooses to see the information of a concert, it shows a brief description of the concert, and also the location of the venue. Users also have the chance to purchase tickets for such concert, and could also share the event with his/her friends through Facebook and Twitter. Furthermore, a user can see all the concerts for which she has purchased a ticket or has marked the show as a concert she is interested in.

### Songkick<sub>3</sub>

Songkick is another major app in the field of apps that track concerts. Its functionality is fairly similar, although is not as powerful. For example, it does not allow searching concerts based on a date, what makes it harder for travelling users to find concerts in their sightseeing/business destinations, for example. Furthermore, it is not possible to configure the maximum distance a user is willing to travel for a concert, which is not that flexible for those users willing to make longer journeys in order to watch a given concert.

Finally, the interface of the app is based on tabs instead of pages, and navigating across different pages can be difficult because of that design. As the main screen has five different tabs, it requires much interaction from the user to get from one end of the tab group to the other.

### Gigbeat<sub>4</sub>

Gigbeat is quite similar to the aforementioned apps, but as it is developed by an individual, it is not as powerful. However, it does have some advantages. For example, besides tracking artists, users can also track their favourite venues. This way, if they like a venue because of the bands they usually book they can be informed about all the concerts, and can decide whether they want to go or not, even if they may not be familiar with that band.

On a negative note, when viewing the information of a concert, a lot of information is shown (navigation options of the app, information about the venue, etc.) and it can be overwhelming to locate the relevant information at first.

#### Thrillcall<sub>5</sub>

Thrillcall is not comparable to the previous apps functionality-wise, but it contains some interesting features. On one hand, it offers last minute tickets at a cheaper rate, which could be interesting for those not really convinced about a given band or those with a tight budget. On the other hand, the app also publishes raffles to give away free tickets for those users that are interested in certain band.

#### Timbre<sub>6</sub>

Finally, Timbre plays a completely different role comparing to the aforementioned apps. It does not rely on the music tastes on the user. On the contrary, it just shows those concerts that are scheduled nearby. It will play a song for each of those bands. If the user likes it, she can purchase the tickets. If not, she can swipe left and it will play a song for another band playing in the area. It is a different app model, but it enhances the chances a user has of discovering new bands.

## 3. Research with users

In this section we will summarise the research that has been carried out. Such research has been done with people that meets the criteria specified for a priori end-users. In this case, those users share a common interest in music that leads them to attend live music shows regularly.

As a side note, it should be taken into account that the timespan to perform this research has not been ideal. Thus, the amount of participants that have been interviewed is smaller than the amount that projects of this kind should have.

### 3.1 Field work

Two different methods have been used for this research, the focus group and the interview. Even though both of them are qualitative methods, there is a slight difference between them. The former is based on doing a joint interview with a group of people: the interviewer asks questions (the extent of those questions will depend on whether is structured, semi-structured, etc.) and the interviewees answer them collectively, by complementing each other's ideas. On the other hand, the individual interview will also consist on the interviewer asking questions but, in this case, the answer will be given by the interviewee, without any other interaction.

#### Script

The script for both methods tries to address the general concepts that an app of such characteristics should take into account in terms of what a potential user would be looking for in the app, how she would expect to use it, etc. The script is included below:

With this exercise we intend to extract relevant information about the process each of you follows when looking for concerts to go to. We will ask you some general questions in order to understand your needs better.

To begin with, how do you look for concerts? *Examples: specialised websites, official websites, etc.*

Once you have found a concert that you are willing to go to, what is the information you consider essential about the concert?

Have you ever used an app or website specialised in providing information about concerts in different venues/cities, etc.? If so, what is your opinion about it? What improvements would you make?

The individual interview will have an additional section in which the user will be asked to complete some common tasks with one of the apps that have been benchmarked (in this case, Songkick), under the supervision of the interviewer. That way, possible improvements could be found by watching the user interact with the design or functionalities of the app, which will be valuable information in the design process.

Now, we will ask you to fulfil some tasks with an app that is currently available in the market, Songkick. Tasks:

- Track the artist called *Civet*.
- Find concerts for an artist you like and, then, set that you want to go to one of those concerts.
- Look for the concerts that are going to be held in Bilbao.

### Sample

Regarding to the sample used for the research, three people in total took part on it. The sample for both research methods is unique, i.e. the same person did not participate in both methods. Once again, it should be noted that on a proper research the sample size should be larger. However, at least the sample is representative of the a priori end-user group, which means that the information ought to be accurate.

The focus group was carried out with two people, one girl and one boy, age 30 and 25 years old, respectively. Both of them frequently attend live music concerts, even though their preferred music style varies. Furthermore, one of them is used to attend concerts outside his city and the other one is not.

On the other hand, the interview was done with one boy who is 24 years old. He is a regular in both summer music festivals and local music concerts. Having played on a music band for several years, he has a great knowledge on music and his music taste includes a wide range of music genres.

Finally, the recruitment process has been fairly informal, meaning that the participants were chosen among friends and relatives of the researcher. Because of that, contrary to standard practices, there was not a form shipment to those susceptible of forming part of the target profile in order to gather their sociological data (to make sure the sample is balanced according to gender, age, etc.) or their willingness to take part. Instead of that, social gatherings were used to approach suitable participants and, in case of agreement, make the exercise right away.

### 3.2 Analysis of the gathered information

In this section we will briefly summarise the most important remarks made by the participants in both research methods.

#### Focus group

- The main sources of information to look for concerts are the Twitter accounts of their favourite bands and also the website of a given venue (if looking for a plan for a specific date).
- The essential information is: venue and how to arrive to such venue, time the venue opens, concert start date/time, price (when purchasing before the date and also on the same day as the concert) and whether people below 18 years old can attend.
- Disadvantages of some of the current concert search services: some music styles are over-represented while other have very few concerts. As a consequence, many different websites have to be visited.
- In order to know if a concert is going to be crowded, useful to display the amount of probable attendees to the concert.

#### Interview

- The main sources of information to look for concerts are the entertainment guides in the press and/or specialized forums.
- The essential information about a concert is the venue, time/date and the price.
- One of the main disadvantages of Songkick is that it seems that only *mainstream* bands appear, something that does not suit every user.

### 3.3 Main discoveries

One of the initial discoveries of the conducted research is that, even if several apps and websites are available to look for concerts, they are not so widely spread yet. Some of the users know they exist but have never actually used them. Therefore, as its use is not general is not that necessary to follow all the practices in that kind of apps with the pretext of not confusing the users.

Another interesting discovery is that the main players in the market still have not reached maturity in terms of the amount of concert data they provide, as to this date the information they provide does not cater for the needs of music fans of different backgrounds. In other words, the information they provide leaves behind those music styles or bands that are not as popular. Of course, this problem should be fixed at the API level and not in the UI but, anyhow, it is advisable to have it in mind.

### 3.4 Person modelling

- Oihana: Oihana is 23 years old and works as an industrial designer. Due to the nature of her work she is constantly listening to music at work. Because of that, her music tastes are expanding greatly and she discovers new bands that she likes as those tastes evolve. She normally looks for music concerts one by one in her favourite band's social network feed or official website. However, due to the vast amount of bands she listens to, it is impossible to keep track of all of them. She is interested in using the app so that the app takes care of looking for concerts and lets her know when one concert she may like is happening in her area.
- Jokin: Jokin is 35 years old and works as a teacher. Even though he does not do it that often any more, he has been a sound technician for many years. As a consequence, he has a vivid interest in live music as he can observe how different bands sound depending on the configuration and the venue. His leisure time is planned around music, he regularly attends concerts in his area and also travels to festivals in order to watch those bands he does not get the chance to see in his town. His motivation to use the app is that he will be able to look for concerts in the towns he is planning to visit and will also keep a record of those concerts that he has found and is interested in going to.

### 3.5 Functionalities<sup>1</sup>

Regarding to the functionalities the app will have, here is a summary:

- The app will have a general feed that will show the concerts that have been scheduled in the area the user is located in.
- The app will have a page that shows the artists the user is tracking and it will also have the chance to add more artists to that list. If the user taps on a given artist the page for that artist will open.
- The app will have a page that contains the cities the user is tracking and it will also have the option of adding new cities to that list. If the user taps on a city, the page for that city will launch. A list of concerts that will take place in such city will be shown in that page.
- The app will have a page that contains the venues the user is tracking and it will also have the option of adding new venues to that list. If the user taps on a venue, the page for that venue will launch. A list of concerts that will take place in such venue will be shown in that page, along with its location.
- The app will have a page that will show the concerts the user is going to in chronological order.

---

<sup>1</sup> The scope of the Dissertation was adapted later on to adapt to the time available for development. Therefore, not all of this functionalities will be developed. This change is explained in appendix A.

- The app will have a page that shows the information of a given concert that will contain the information about date, time, venue and line-up of such event. It will also offer the chance to get directions to the concert based on the location of the user.
- The app will have a page with the configurable settings, like the maximum distance the user is willing to travel for a concert, the information that the feed will contain (all concerts in the user's area, only concerts from the bands the user is tracking, only concerts scheduled in the venues and cities the user is following, etc.), whether the app should remind the user when a concert she has marked as "I will go" is near...

That set of functionalities have been defined after taking the information gathered at the research with users into account. Nevertheless, not every aspect has been introduced to the design. For example, one of the features that was considered helpful by the participants was to provide information about the price of an event and also the time in which the venue gates would open. However, the APIs that are available in the market do not provide such information. Thus, the app will also lack that information and will not be included in the design.

If compared to the apps that were analysed in the benchmarking, there are some differences in terms of the way the functionalities are devised. For example, the feed located in the home page tries to expand the amount of information it provides in order to be a hub of all the information the user might be interested in, amount that will be configurable through the settings to make sure the user is comfortable with it. On the contrary, other apps just show a set of events they consider that might be of interest but without much flexibility for choosing the information that should appear.

On the other hand, the app will enable the user to find concerts easier, compared to the other apps. The reason is that there will be different criteria to look for those events, such as, the artist, the city and the venue, which is more straightforward for the user that already knows what she wants to search for. Those who do not, however, will also take advantage of such categorised search as well as the feed located in the home page.

## 4. Design

Once the information about the requirements of the app has been gathered, this chapter will describe the design process, starting from the low-fidelity wireframes to a higher-fidelity prototype, which will be the base for the implementation.

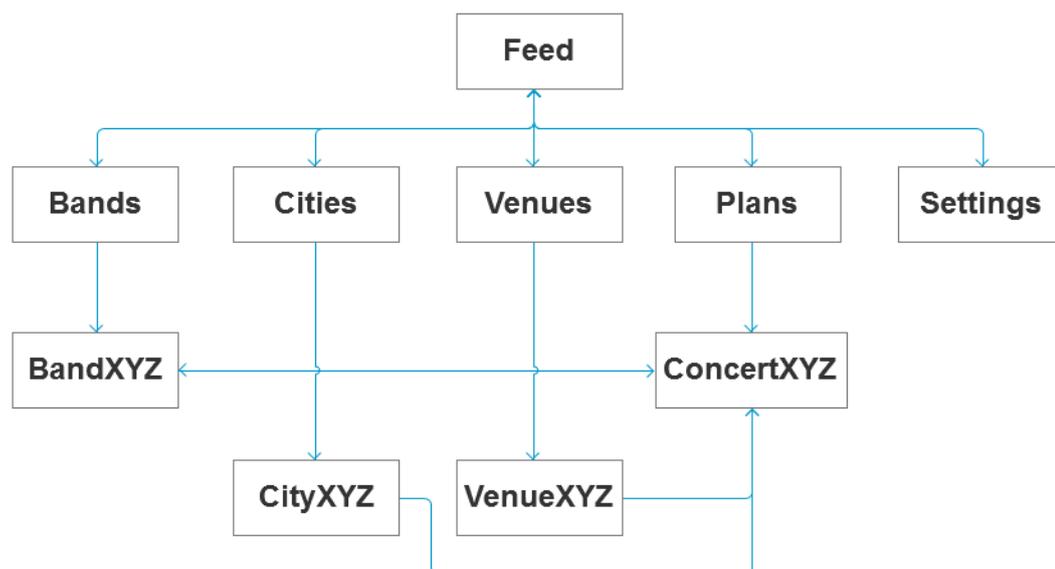
As mentioned in the previous chapter the scope of the project was altered in the implementation phase, leaving some functionalities out of this version of the app. However, by the time the scope was altered the design for those functionalities was already done, so it will be described anyway. After that, the current design will be described.

### 4.1 Initial design

This section contains the first design of the app, i.e. the design that ought to be implemented had not changed the scope.

#### Information architecture

The information architecture of the app could be classified in two different groups, main pages and secondaries. On one hand, the main pages will link between each other and will be the first thing a user sees when launching the application. On the other hand, those pages ending with *XYZ* are secondary pages and will be used to give more information about the elements shown in the main pages. However, it should be noted that the term secondaries could be misleading, as they are the pages that will provide some of the most important features such as tracking a band. These secondary pages will have the option to return to the main page that precedes them and in *BandXYZ* and *ConcertXYZ*'s case, they will also be able to navigate between them.

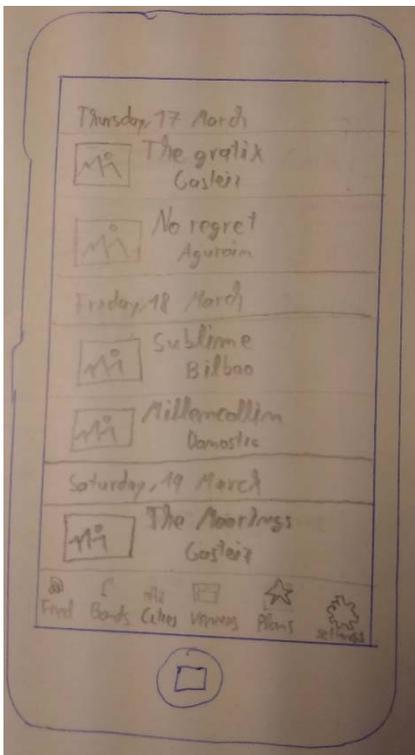


2. Figure: Information architecture of the initial design.

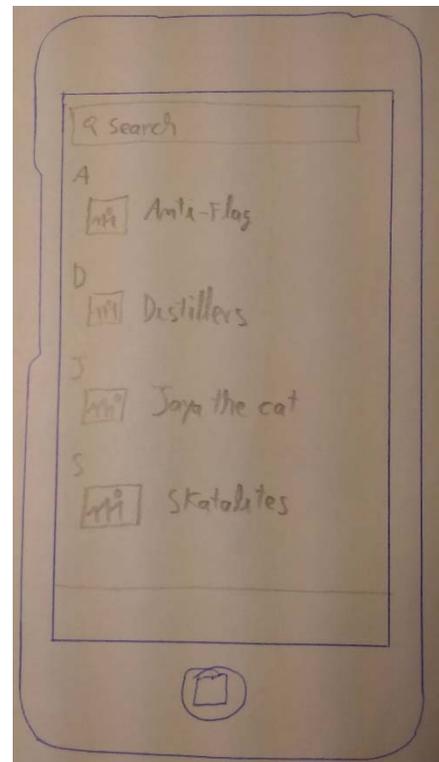
## Wireframes

In this subsection the Wireframes for the app will be shown. All the main pages have been sketched, with some exceptions. For example, the design for the pages *Venue* and *Cities* on one hand, and *VenueXYZ* and *CityXYZ* on the other are very similar. That is why, each of those pairs has only been sketched once, in order to avoid duplicated work. Later on, those screens will be properly defined in the prototype.

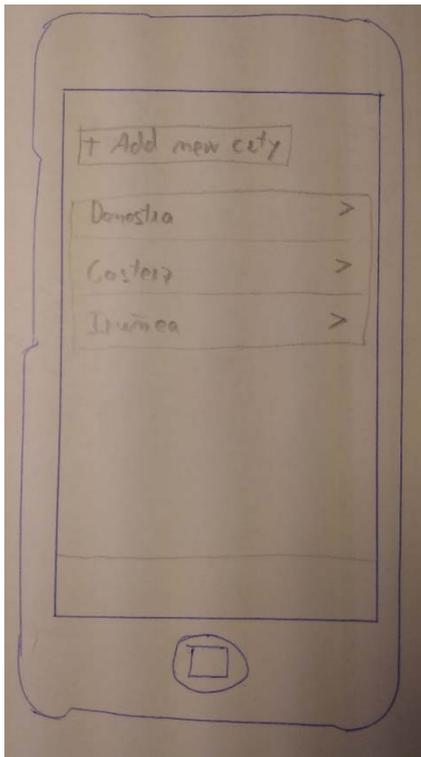
Furthermore, the main pages should have a navigation bar in the bottom. For simplicity's sake, that bar has been omitted from some pages. Those pages are *Bands*, *Cities*, *Venues* and *Plans*.



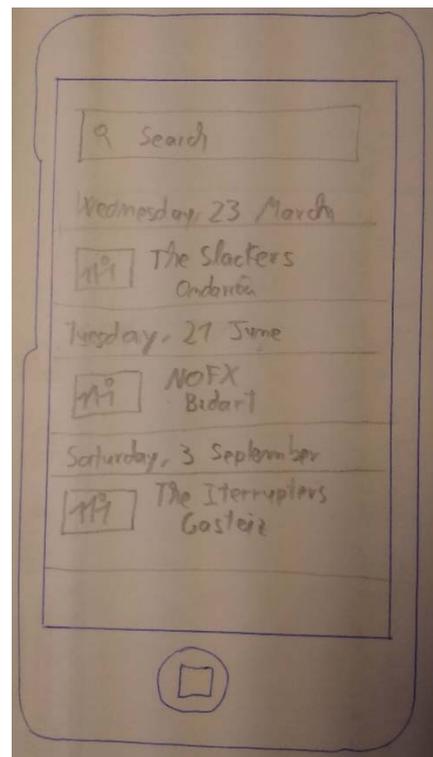
3. Figure: Wireframe of the Feed page.



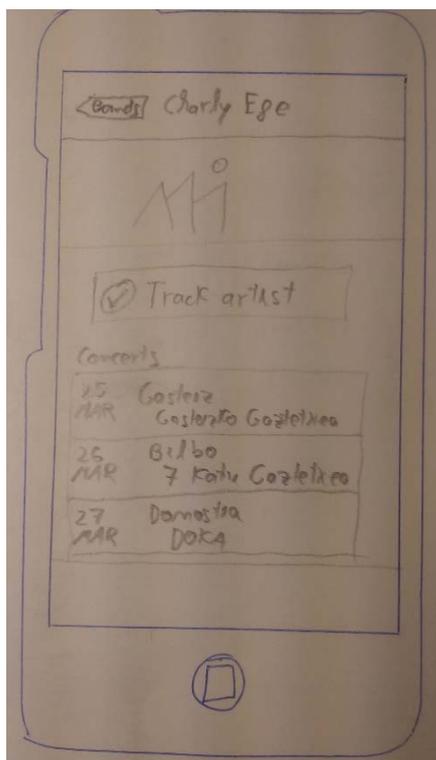
4. Figure: Wireframe of the Bands page.



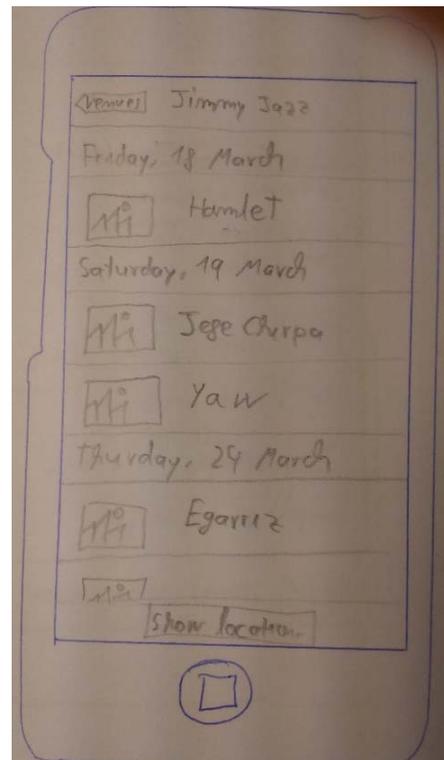
8. Figure: Wireframe of the Cities page.



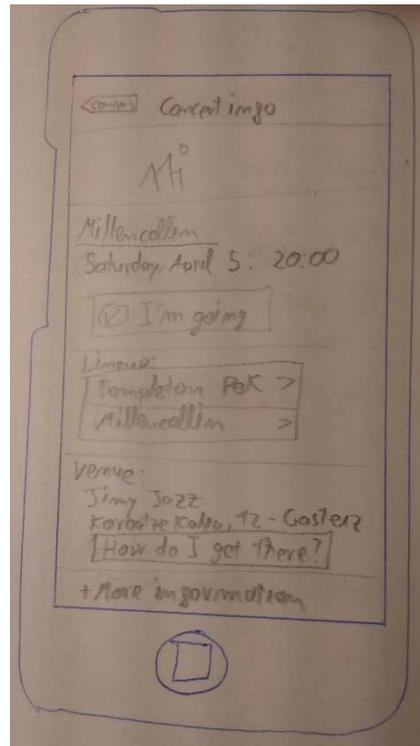
7. Figure: Wireframe of the Plans page.



6. Figure: Wireframe of the BandXYZ page.



5. Figure: Wireframe of the VenueXYZ page.



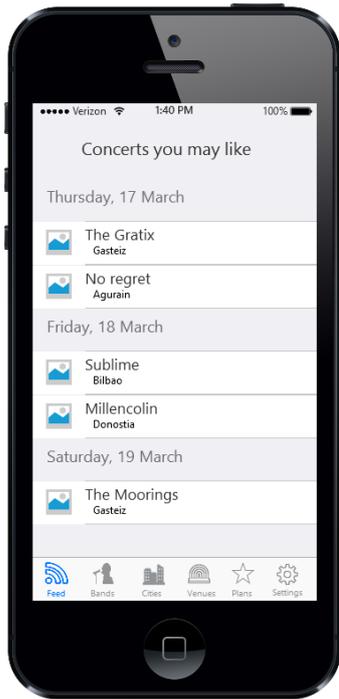
**9. Figure: Wireframe of the ConcertXYZ page.**

### Prototyping

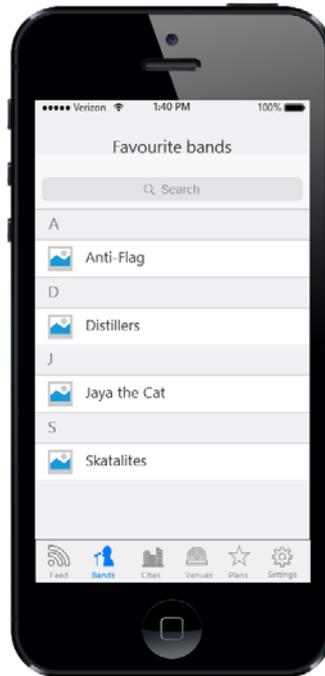
The prototype has been designed according to the aforementioned wireframes. However, some changes have been applied to the initial design, in order to fix possible design flaws. For example, in the wireframes the main pages did not have a header. The header was added in order to facilitate the user to locate herself in the hierarchy of the app.

On the other hand, the prototype has been created with the iOS Operating System in mind. Thus, the design tries to be as similar as possible to the design guidelines of such OS. Nevertheless, because of the lack of experience of the designer some features may have to be altered in the development process in order to match the standard procedures and looks. Related to that, the font and some User Interface elements that have been used are not the same as in iOS, so there will also be changes in that aspect.

Finally, some external icons have been used for this prototype, which have been taken from a library<sup>7</sup> freely available in the Internet.



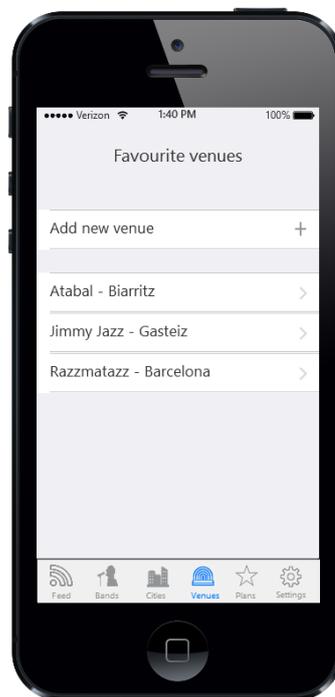
11. Figure: Prototype of the Feed page.



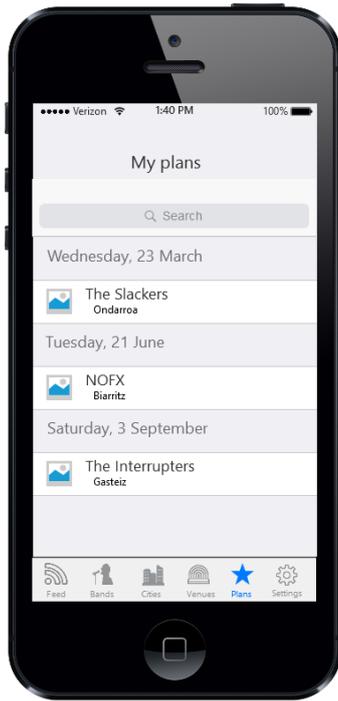
10. Figure: Prototype of the Bands page.



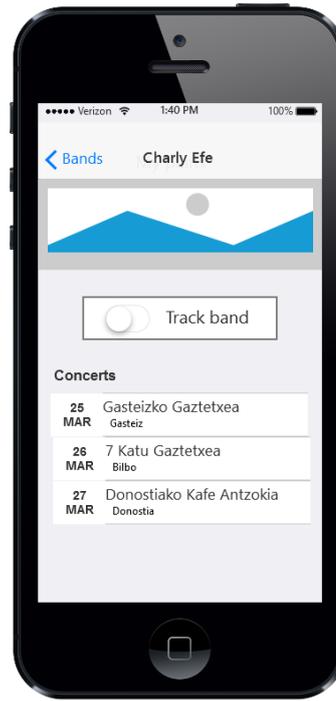
12. Figure: Prototype of the Cities page.



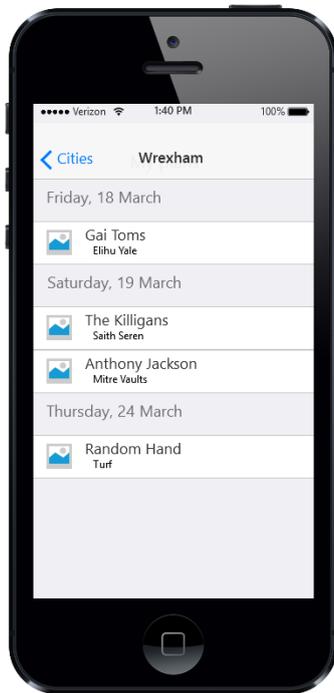
13. Figure: Prototype of the Venues page.



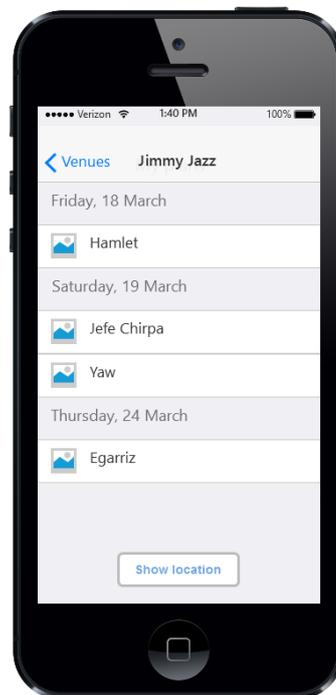
16. Figure: Prototype of the Plans page.



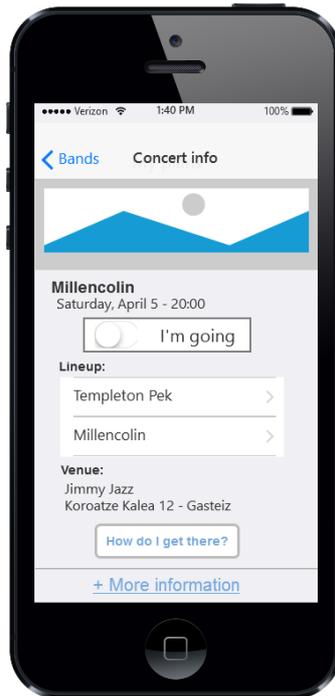
15. Figure: Prototype of the BandXYZ page.



17. Figure: Prototype of the CityXYZ page.



14. Figure: Prototype of the VenueXYZ page.



**18. Figure: Prototype of the ConcertXYZ page.**

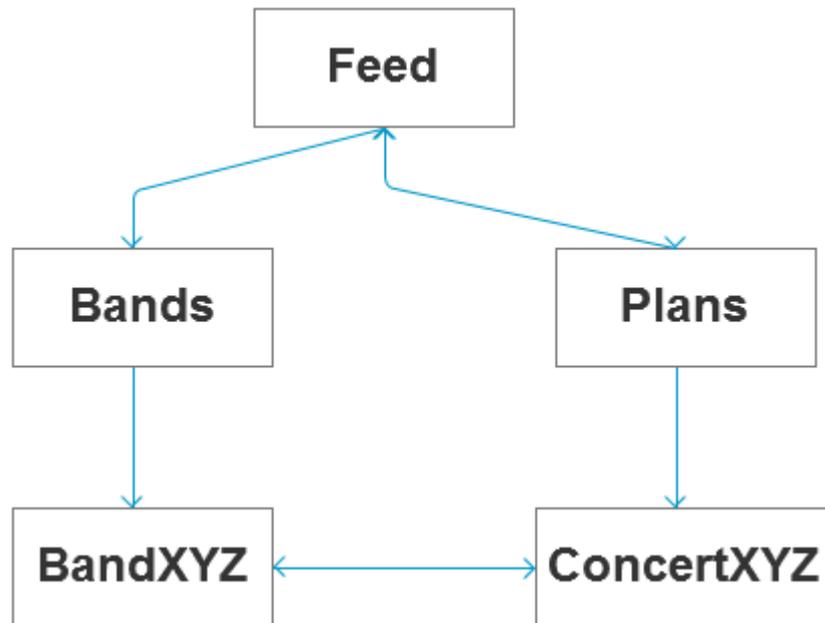
#### 4.2 Final design

Once the development had started the prototype had to be redesigned for two reasons. The first one is that the scope had been redefined and therefore some features were no longer needed. The second reason is related to the limitations of the API of Songkick. In the first iteration of the design many pages included images that were foreseen to be related to the band, such as an image of their logo or another similar photo, as that way the interface would look more appealing. However, the API has no option for that so it could not be accomplished. Furthermore, the price of the tickets for a given price could not be retrieved from the API neither, so it has been omitted from the final design too.

The reason for misconception of the capabilities of the API is that by the time the initial design was done, there was no definitive decision about which API to use. Some of the candidate APIs had the option to retrieve ticket prices and photos, but Songkick did not. When development started the design had to be changed, because the alternative of looking for another API to retrieve the images was not an option due to time limitations.

## Information architecture

Due to the changes, the information architecture of the app is quite simple now. As the previous design, it includes three main pages and two secondaries.



19. Figure: Information architecture of the final design.

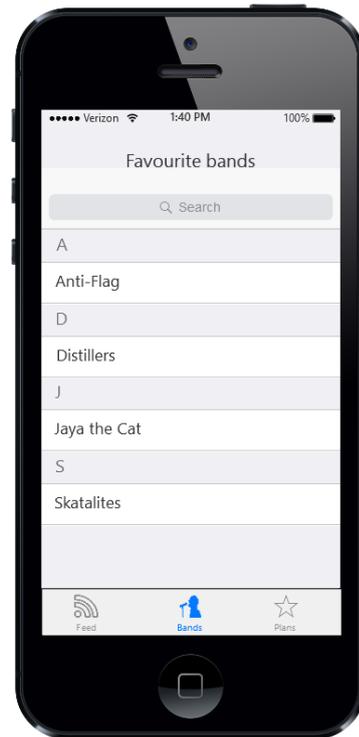
## Prototyping

The new iteration of the prototype has evolved from the previous prototype so there are no low-fidelity wireframes in this case. These are some of the most significant changes:

- The tab bar of the app has been updated and the items related to the discarded features have been removed.
- The tab bar has been included in all the pages (main and secondaries) in order to facilitate the navigation in the app.
- Page *ConcertXYZ* has been modified and instead of having a button offering to give directions to the venue, a map has been added with an additional in-map button that will show the directions.
- The link in the bottom of the *ConcertXYZ* has been removed because it was redundant, as all the main information is already displayed in the page.



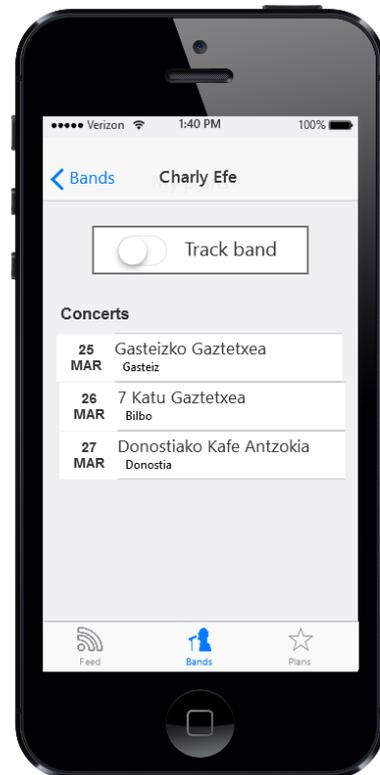
23. Figure: Prototype of the Feed page.



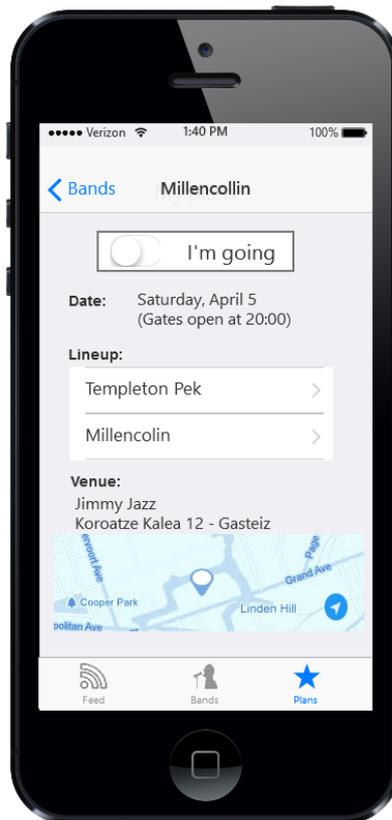
22. Figure: Prototype of the Bands page.



21. Figure: Prototype of the Plans page.



20. Figure: Prototype of the BandXYZ page.



24. Figure: Prototype of the ConcertXYZ page.

### User flows

The different user flows that correspond to the features of the prototype are briefly explained below, in order to have a better understanding of the process:

- Look for the list of events that might interest the user:
  - The user will access the home page and will see the feed, which will consist on the concerts that have been scheduled in the area the user is located in. The list will display the shows chronologically, starting from those whose date is nearest.
- Track a new artist:
  - The user will go into the page that contains the bands being tracked at the moment. The user will search for the band to be tracked and once it has been loaded, she will select the option to track such artist in its page.
  - It can also be accomplished when a user opens the artist's page from the *ConcertXYZ* page.

- Look for concerts of a given artist:
  - The user will go to the *Bands* page. If the band is already tracked, the user will select it from the list and the list of concerts of such band will be shown. If it has not been tracked yet, the user will have to search for it. Once she has found and opened it, the list of concerts will be displayed.
- Set a concert as *I'm going*:
  - Once the user has found a concert of her interest (it does not matter the way she has accessed it) she will select the option called *I'm going* and the concert will be added to the list of her plans.
- Look for concerts a user has chosen to go:
  - The user will go to the *Plans* page. A list of concerts she has marked as *I'm going* will be shown, ordered from those happening sooner to those happening later.

## 5. User experience evaluation

In this chapter the outcome of the user experience evaluation will be summarised. Two different evaluations were carried out throughout the project. The first one was performed after the first prototype had been finished and its aim was to verify that the prototype was easy to use so that no major usability error was introduced by design into the app, as it would be much harder to solve once it was implemented. The second evaluation was carried out after the implementation of the app, to make sure the app was useful before its release (in this case the release will be done to the academic staff, not the general public as it ought to be done in normal circumstances).

### 5.1 Evaluation of the prototype

This evaluation was carried out immediately after finishing the first iteration of the prototype, so the outcome is not entirely useful, as some of the features that were evaluated at the time were not included in the final prototype. However, some things still apply.

Two different methods were used in this evaluation, the heuristic evaluation and the evaluation with users. The first one will provide information on whether the prototype conforms to the interface design *best practices*, and the second will provide information from the point of view of the user.

#### Heuristic evaluation

The heuristic evaluation is an evaluation method in which no user takes part in the *traditional* way. Instead of being given tasks to accomplish with the prototype, the evaluator (it could be a user, an expert or a member of the development team) will assess the prototype according to a set of design guidelines (heuristics) that ought to be respected in every application.

In this case, the designer of the application itself has been the evaluator of the aforementioned heuristics. The subset of heuristics that has been chosen for this evaluation is the one suggested by Jakob Nielsen<sup>8</sup>. These are the main findings of such evaluation:

- When listing concerts, the separator used to specify the date of each group of concerts takes too much space. The date occupies more space in the interface than a concert, something that is not necessary, as it disallows to show meaningful content to users. Besides, the same result could be obtained by standing out the date by other design parameters, e.g. the colour, the font type, etc.
- The consistency is not total across different pages of the app when listing a concert, as not all of them show the same amount of information. For example, in the feed a concert is represented with an image, the name of the band and the city the show will take place in. On the contrary, in *VenueXYZ* only the image and the name of the band appear, and in *CityXYZ* the name of the venue appears instead of the name of the city.

Nevertheless, it is not considered an issue as that *inconsistency* is justified so that content is not duplicated. In the case of *VenueXYZ*, the user already knows where the concert will be, so explicitly repeating it in each row will not provide any additional information. On the other hand, in *CityXYZ* the user already knows the city the concert will take place in, so additional information can be shown instead, like the venue.

In conclusion, according to the heuristics it could be claimed that the interface is well constructed, as it follows the heuristics in a considerable proportion. However, it should be noted that an interface that respects the heuristics is not necessarily a usable interface from the users' point of view. That is why the evaluation with users will be carried out in the following section.

### Evaluation with users

The evaluation with users is one of the most important sources of information in terms of whether an interface is understood and easy to use for them. In an evaluation with users the user will be given a scenario and a set of tasks to fulfil and she will be watched while accomplishing them in order to look for any aspect of the interface that might not feel natural to her.

### Script

This is the script used to conduct the evaluation with the user.

- 1<sup>st</sup> scenario: You just discovered a new band called *Distillers* and it has become one of your all-time favourite bands.
  - a. Track the band called *Distillers*.
  - b. Look for the concerts this band has scheduled and choose that you want to attend the concert in *7 Katu Gaztetxea* on the 26 of March.
  - c. You are not sure on the location of the venue, find the address of the venue.
- 2<sup>nd</sup> scenario: You are going to Madrid next month to visit some friends. You want to look for concerts that you may like in Madrid as you are on your own in the evenings.
  - a. Look for the scheduled concerts in Madrid for the third weekend of April (April 15-17).
  - b. A friend has told you that a venue called *La penúltima* usually schedules concerts that suit your taste. Look for the concerts in that venue for the same weekend.

## Sample

The sample for this evaluation process has consisted on two users. The participants are a girl and a boy, aged 35 and 43 respectively. Only one of them can be considered to be part of the a priori end-user group. However, it does not represent a problem as in this case what is being evaluated is the interface and not the needs or functionalities the app should have. That happens because this app does not require a great amount of expertise in order to understand its functionality.

## Main discoveries

- When visualizing the *ConcertXYZ* page, the user thought that the venue information was clickable and would lead him to the appropriate *VenueXYZ* page.
- In the Cities and Venues pages is not very practical to look for concerts on a given date as if that date is some weeks ahead in the future the user will have to scroll the content until arriving to the date she is interested in.

## 5.2 Evaluation of the app

This evaluation is not as thorough as the previous one because, on the one hand, the time was very limited and tasks with higher priority were due, and, on the other hand, because the design of the app respected the original design, even though it was a limited version. Therefore, only the evaluation with users was performed as it is a method that provides a good diagnosis of the degree of usability of the app.

## Script

This is the script used to conduct the evaluation with the user.

- 1<sup>st</sup> scenario: You are looking for a plan for next weekend and you fancy going to a concert
  - a. Look for the concerts scheduled for next Saturday and select any of them.
  - b. You like the concert, add it to your agenda.
  - c. You are not sure where the venue is, how do you get there?
  - d. Look the list of concerts you have added so far.
- 2<sup>nd</sup> scenario: You recently discovered your new favourite band, *Hot Water Music*.
  - a. Add *Hot Water Music* to your favourite bands.
  - b. Check the list of the bands you liked so far.

- 3<sup>rd</sup> scenario: A friend told you that *The Interrupters* are playing in your town on the 30<sup>th</sup> of June:
  - a. Open the information of the concert.
  - b. What time does it start?
  - c. Which are the bands that will play in that event?

### Sample

Because of time limitations the sample for this evaluation consisted on one user: a girl aged 26 years old that is part of the a priori end-user group.

### Main discoveries

- When the user tried to get the directions to the venue in the first scenario, she tapped the map, thinking that was the way to do it. Because of the location the map was showing, the directions button was not as easy to spot, as the colour contrast was low. The button ought to be changed to a colour that will stand out with every background. Besides, the map should be opened on a standalone screen after a user taps it, so that she can see the location more easily and get the directions if she wishes to.
- The search results in the *Bands* page do not hide straightaway when the user taps on cancel or when the user goes to another page. The search results should only be displayed when the search is being active, and the list of favourite bands should be displayed otherwise.
- The API is sometimes slow when retrieving the results of a given command which leaves the app waiting for the response. An activity indicator is shown in the notification bar while waiting for that response but it is not very visible so the user is left wondering whether she did something wrong or whether the app crashed. Therefore, more explicit feedback has to be given to the user.

## 6. Development and Testing

This chapter describes the main aspects related to the development of the app, such as the structure of both the code and the database, as well as describing the API that retrieves the data for bands and concerts. Finally, it will summarise the testing process of the app.

### 6.1 Songkick APIs

The use of the Songkick API requires almost no setup in order to start retrieving data. The only requirement of the API is to apply for a key, in an application in which a brief description of the app has to be included. Once the request is granted, the API can be consumed in xml and json format, the latter being the format that has been chosen for the app, due to the simplicity with which it can be retrieved. These are the API functions used throughout the code:

- *Artist search*<sub>10</sub>: This function will receive a text and it will search for a band that contains said text. It will return a list of all the bands that match the criteria.
  - *Use in the app*: The search bar in the *Bands* page will get the search results using these API function. In order to control the amount of data that is received each time and speed-up the response, each API call will request 20 results.
- *Artist's upcoming events*<sub>11</sub>: This function receives an ID of a band and it will return all the events that have been scheduled for that band, ordered by date (starting from those events that will start soonest). Each event will also include information about the date, the location of the concert and the line-up of such event, among other useful information.
  - *Use in the app*: The list of concerts in page *BandXYZ* is retrieved with this API function. In order to control the amount of data that is received each time and speed-up the response, each API call will request 20 results.
- *Event details*<sub>12</sub>: This function receives an event ID and it will return all the information available about the event and the venue is scheduled in.
  - *Use in the app*: Page *ConcertXYZ* relies in this function to display the information to the current concert.
- *Location search*<sub>13</sub>: Songkick's location search is based on geographical areas regarded as *metro areas*. Each *metro area* corresponds to a town or city, and each of them will also include all those places that have not been included in a *metro area* of their own. This function receives a set of coordinates and it will return a list of *metro areas* located near it, ordered from those that are nearest to those who are furthest.
  - *Use in the app*: This function will be used as a helper function that will retrieve the *metro areas* that are near to the location of the user.

- *Metro area's upcoming events*<sup>14</sup>: This API function receives a *metro area* ID and will retrieve all the events that are scheduled there.
  - *Use in this app*: This function is used to retrieve the feed of concerts in the home page of the app. The *metro areas* located near to the location of the user have to be retrieved prior to the execution of this function. In order to control the amount of data that is received each time and speed-up the response, each API call will request 20 results.

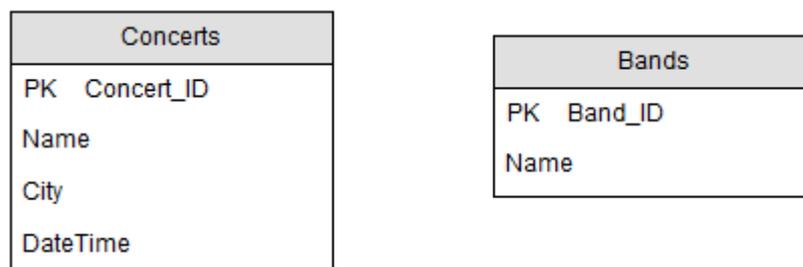
## 6.2 Database diagram

The structure of the database is very simple as it only contains two unrelated tables. The aim of the database is to store the basic information of the bands and concerts the user likes. That information is only used in pages *Bands* and *Plans*, and all the remaining information is retrieved from the API, as this way the information shown on screen will be more likely to be up to date.

Table *Bands* just contains the name and ID of the band. The name is stored so that it can be displayed in the list and the ID is stored so that the API can be called using that reference.

On the other hand, table *Concerts* contains the name, the city and the date so that concerts can be chronologically ordered and show both the name and the city the concert is scheduled in. The ID is stored so that the API can be called using that reference.

Finally, the database is stored locally in the device itself for simplicity's sake.



**25. Figure: Database diagram of the app.**

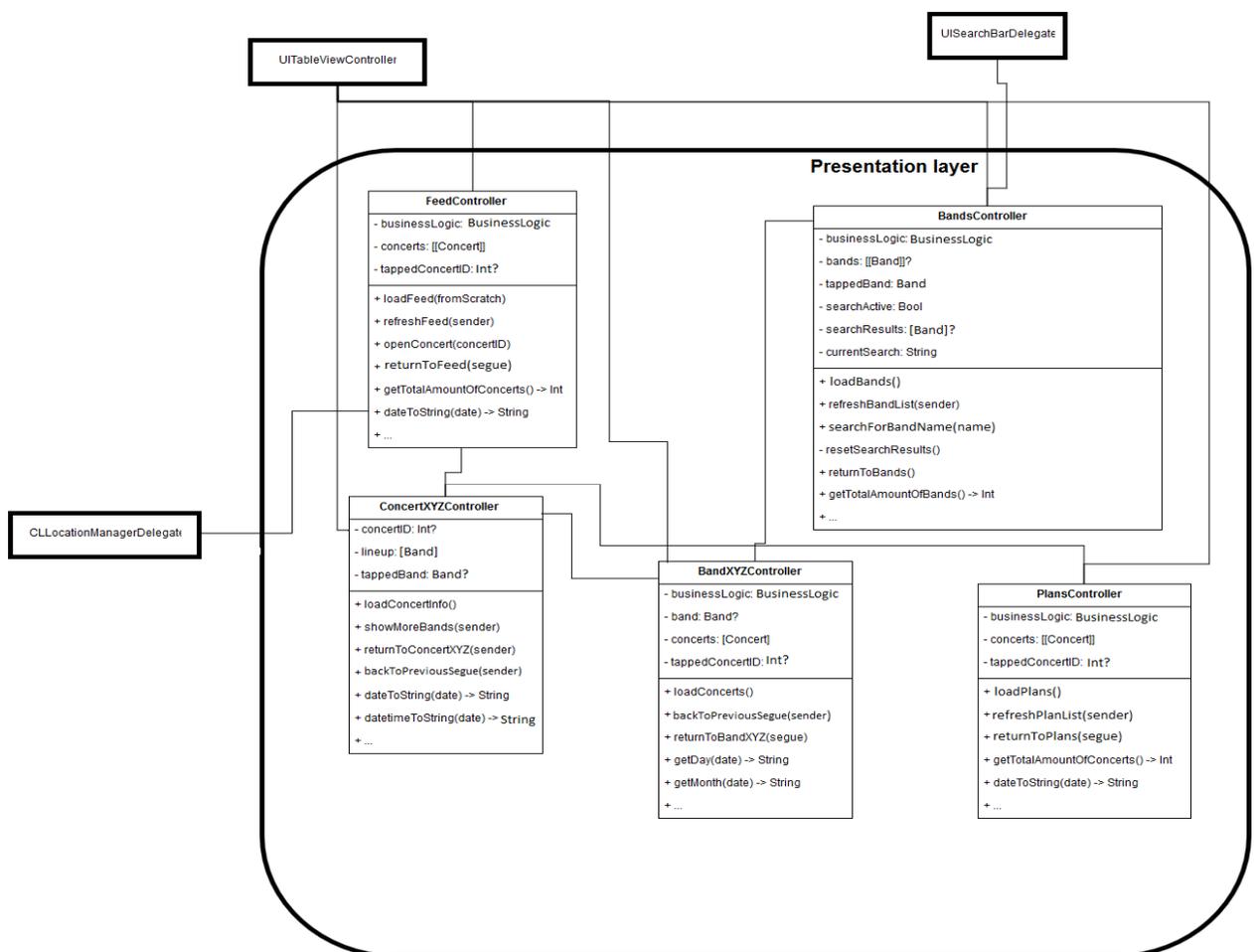
## 6.3 Class diagram

The application has been structured following the three-tier architecture, in which the interface, the database and the business logic are logically separated in order to make a code that is easier to understand and maintain. The class diagram has been divided in three smaller diagrams following the same model as it was too big to show at once, each of them representing one of the tiers.

## User Interface layer

The User Interface layer is the tier the user will directly interact with. Each of the classes of this layer represents a different page. All of them are of type *UITableViewController*. Besides, class *BandsController* will also be a *UISearchBarDelegate* in order to handle the actions of the search bar, and *FeedController* will be a *CLLocationManagerDelegate* in order to be able to receive the location coordinates of the user.

Some of the functions implemented by each class have been omitted and only those specifically implemented for the class (i.e. not those implemented because of requirements of the *UITableViewController*, for example) have been included. Apart from that, even if that relation does not appear all of the classes in this layer will interact with the Business Logic layer.

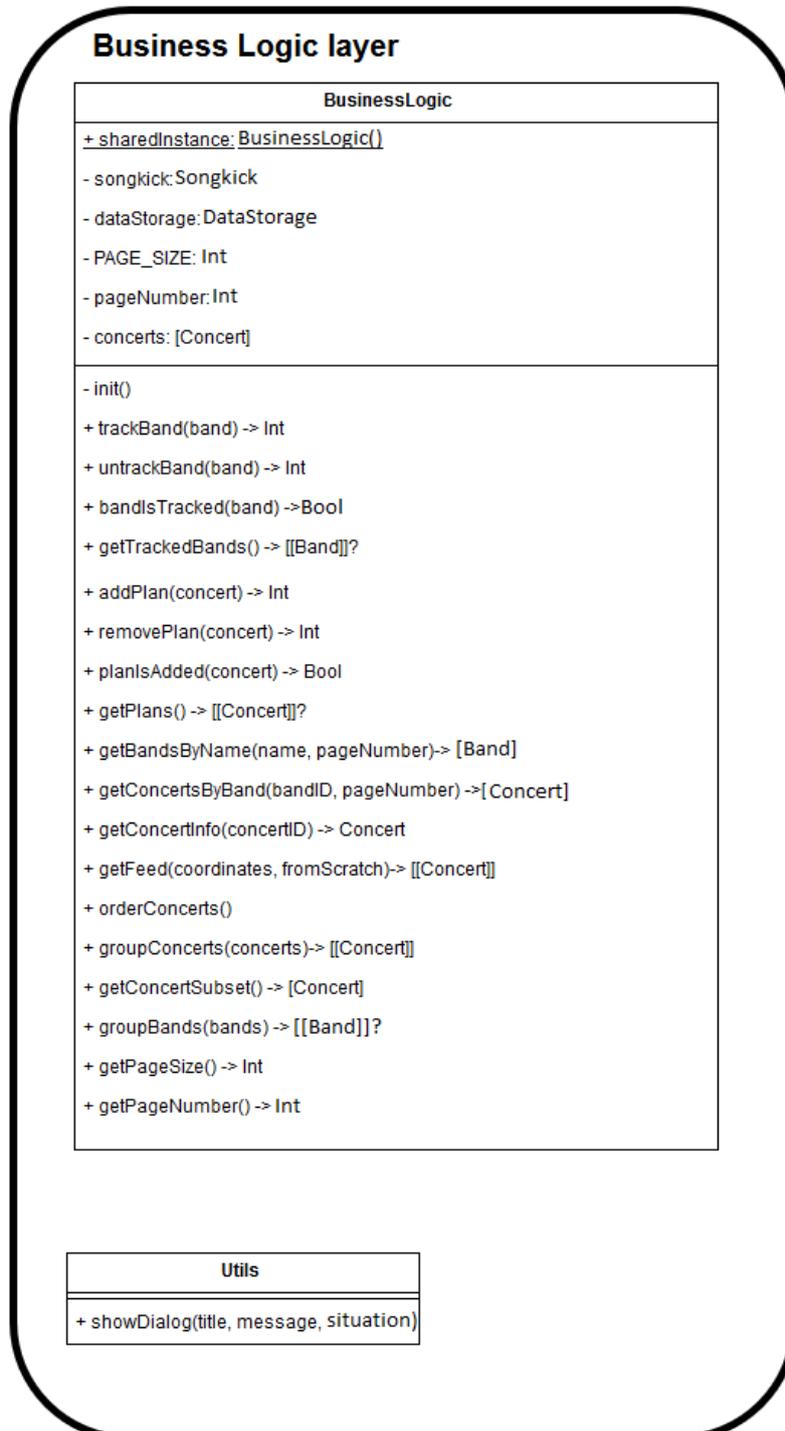


26. Figure: Class diagram of the UI layer.

## Business Logic layer

This layer will mainly act as an intermediary between the UI layer and the Data layer. Most of the function are proxies that will just make the call to a Data layer function and return the result. Some others, however, will also process the result, such as functions *orderConcerts()* and *groupBands()*, to name a few.

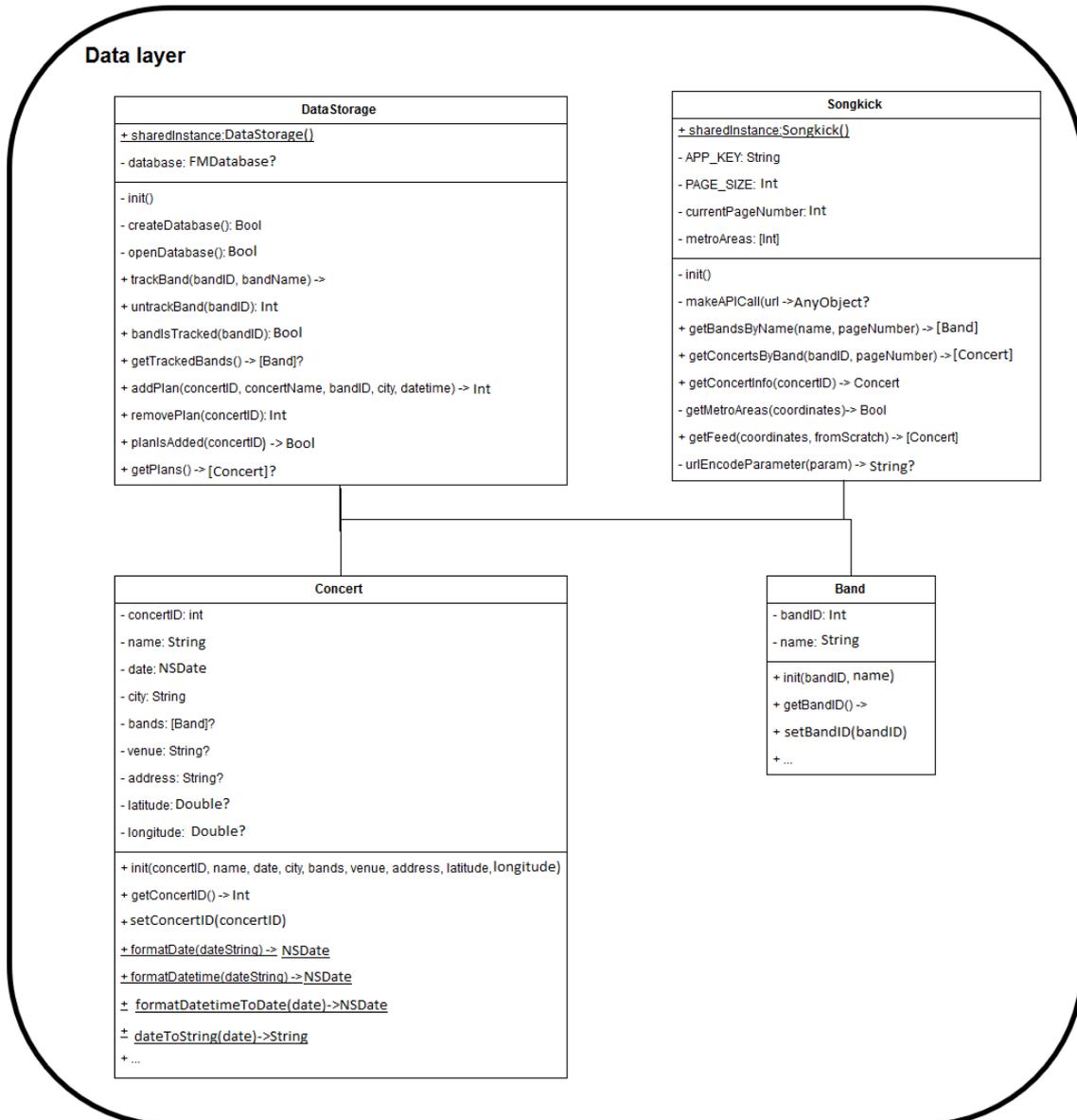
However, class *Utils* will not interact with the Data layer.



27. Figure: Class diagram of the Business logic class.

## Data layer

This layer is the responsible of retrieving data from either the API or the database. This layer has two groups of classes. The first one, which includes *DataStorage* and *Songkick*, will handle the retrieval of data from the database and the API, respectively. The second group, formed by *Band* and *Concert*, defines the objects used throughout the app to pass the data of bands and concerts, with the objective of simplifying that transmission.



28. Figure: Class diagram of the Data layer.

## 6.4 Testing

During the implementation phase tests were continuously performed. This section tries to describe the process that was carried out to perform those tests.

### Methodology

Even though the tests were done on a constant basis, the testing efforts of the app gradually increased as the implementation of the app went forward. At first the tests only tested that a given function worked as intended and slowly evolved into functionality-level tests. By then, each time a new functionality was finished and before moving on to the next one, that functionality and those developed before were tested in order to make sure that everything worked correctly and no unexpected behaviour occurred.

### Test types

Two different test types were used during the testing process, automated and manual tests.

Automated tests were developed for both the Data and Business Logic layers and test each of the functions developed in those classes. Those tests were divided in different classes so that each of them only tested one class, without any interference. The great advantage of this tests is that once they have been defined they could be repeated as much as necessary without having to do it one by one. This way it was made sure that no change has been done to the codebase that will alter the final result each function returned.

Unfortunately, time limitations prevented that automated test could be generated for the UI layer. These tests simulate the actions a real user would make in order to see that the interface responds to those events as expected. Consequently, the interface had to be tested manually after every change to the UI layer. Obviously, this is not ideal as it implies that the developer has to repeat those actions multiple times in order to make sure that every functionality developed to that time is working properly.

Finally, the tests were done using the iOS standard simulator (provided by xCode) and an iPhone 4S with iOS 7.1 operating system. The current iOS version is 9.3.2 so there is a big gap between both. That gap is an anomaly in iOS devices as they tend to be updated to the last version available. However, the physical device could not be updated due to performance issues.

Due to the aforementioned software limitation, most of the manual tests were performed with the simulator in different screen sizes and with the last available iOS version. The physical device was used to verify that the appearance of the UI was correct when being displayed in a real device.

## 7. Conclusions and future work

In this last chapter, a review of the accomplished work will be done and a brief brainstorming of the work ahead will be done.

### 7.1 Conclusions

These are the main conclusions that have been derived from the accomplished work:

- Overall, even if the project was oversized at first, the planning was adapted in time so that the planned tasks could be finished. For the next time, the preferences extracted from the investigation with users have to be taken into account but it is important to prioritise that input and delimit the project to a workload that can be assumed.
- If we look at the execution of the project and whether it respects the original design, it can be claimed that the design is respected and that the app is functional too, as intended. However, the final functionalities were completed in a hurry, so it would have been better to leave a bigger margin with the final delivery.
- In a personal note, the main objective of learning this master has been accomplished with this project, learning iOS. The first three (mainly the first year) semesters of the master were based on very theoretical concepts that made studying very dull. However, this semester has been enjoyable because of its practical contents, even if the academic workload of the master was not properly balanced between subjects and Dissertation, and the final month and a half of the master was a complete stress.

### 7.2 Future work

The time available to develop these product was very limited. Thus, many functionalities have been left behind and there are many others too that show the path to follow, should the development continue:

- First of all, the app should have those functionalities that were omitted in the scope redefinition (a page for the tracked venues and cities, notifications when a new concert has been scheduled, etc.). The remarks made during the evaluation of the UX should be included into the app too. Among those remarks, it is critical to provide a more visual feedback to the user when the request to the API is being made.
- In order to grant that a user can access her data from many different devices, user-generated data should be stored on the Internet and be retrieved via API. Before starting to use the app the user should create an account or log in. This way a *social* aspect could be added to the app and encourage interaction between users.
- Tracking all favourite artists can be a long task, the app should be able to import them automatically from the device's music library or from music streaming services such as Deezer or Spotify.

- Songkick includes a function in its API that returns similar artists to a given band<sup>15</sup>. The app should use that information to make accurate recommendations to the user when a concert for an artist similar to her favourites has been added.
- The app should offer users the chance to purchase tickets for a concert.
- This improvement is not directly linked to the functionalities of the app but automated UI tests ought to be implemented in order to guarantee the robustness of the app.
- Continuing with the robustness, the app should handle those events that are related to the context the app is used in and that have an impact on performance, such as screen orientation change, inability to capture a valid GPS signal, etc.

## 8. Bibliography

1. IDC: Smartphone OS Market Share [Internet]. www.idc.com. 2016 [cited 7 March 2016]. Available from: <https://www.idc.com/prodserv/smartphone-os-market-share.jsp>
2. Bandsintown [Internet]. Google Play. 2016 [cited 7 March 2016]. Available from: <https://play.google.com/store/apps/details?id=com.bandsintown>
3. Songkick [Internet]. Play.google.com. 2016 [cited 7 March 2016]. Available from: <https://play.google.com/store/apps/details?id=com.songkick>
4. Novak M. Gigbeat Concerts [Internet]. Play.google.com. 2016 [cited 7 March 2016]. Available from: <https://play.google.com/store/apps/details?id=fm.gigbeat.android>
5. Thrillcall Live Music [Internet]. Play.google.com. 2016 [cited 7 March 2016]. Available from: <https://play.google.com/store/apps/details?id=com.thrillcall.thrillcall>
6. Timbre Makes Finding a Concert as Easy as Pressing 'Play' | Evolver.fm [Internet]. Evolver.fm. 2016 [cited 7 March 2016]. Available from: <http://evolver.fm/2012/09/10/timbre/>
7. Version M. Icons8 - 28,700 icons + Icon Tools [Internet]. Icons8. 2016 [cited 18 June 2016]. Available from: <https://icons8.com/web-app/for/ios7/>
8. Nielsen J. Enhancing the explanatory power of usability heuristics. CHI 94 ACM Conference on Human Factors in Computer Systems. Boston; 1994. p. 152-158.
9. Concerts and Festivals API | Songkick [Internet]. Songkick.com. 2016 [cited 7 March 2016]. Available from: <http://www.songkick.com/developer>
10. Artist search | Songkick [Internet]. Songkick.com. 2016 [cited 19 June 2016]. Available from: <https://www.songkick.com/developer/artist-search>
11. Upcoming events for artist | Songkick [Internet]. Songkick.com. 2016 [cited 19 June 2016]. Available from: <https://www.songkick.com/developer/upcoming-events-for-artist>
12. Event details | Songkick [Internet]. Songkick.com. 2016 [cited 19 June 2016]. Available from: <https://www.songkick.com/developer/events-details>
13. Location search | Songkick [Internet]. Songkick.com. 2016 [cited 19 June 2016]. Available from: <https://www.songkick.com/developer/location-search>
14. Upcoming events for metro area search | Songkick [Internet]. Songkick.com. 2016 [cited 19 June 2016]. Available from: <https://www.songkick.com/developer/upcoming-events-for-metro-area>
15. Similar artists | Songkick [Internet]. Songkick.com. 2016 [cited 19 June 2016]. Available from: <https://www.songkick.com/developer/similar-artists>

## Appendix A: Scope redefinition

The scope of the product was redefined at the partial hand-in of the development, as the amount of work ahead was larger than the available time to perform those tasks. There are two factors that had a large influence in that decision. The first of them was that the app had to be developed while working on two different subjects of the master that demanded a workload far larger than expected in the beginning. Therefore, the available time for working on the app was vastly reduced.

On the other hand, the inexperience with iOS lead to a slower work pace (specially at the beginning) and even though it was expected to a certain extent the impact it had was greater, and combined with the workload of the aforementioned subjects it was too much a delay to assume.

### A.1 Redefined scope

In the initial scope of the app, users were going to be able to look for concerts according to a band, a city or a venue, apart from the general feed that showed concerts in their area. Besides, the app would also have configurable settings such as the maximum distance from the user's location to the concert, the amount of information shown in the feed (show only concerts of the bands the user likes, show all concerts, etc.), and the capability of configuring notifications.

After redefining the scope, the app was not going to search for concerts by city or venue, and would not have configurable settings neither. Besides, the app would have no notifications at all, not for recently announced concerts, not for reminders of concerts the user had liked.

Those functionalities were discarded because, even though they are helpful and were extracted from the information gathered at the research with users, they were not considered to be the most important ones. The ability to look for concerts based on a band is the most basic type of search, and also the most straightforward, so it is compulsory to have it in the app. Besides, the feed could act as a replacement for the venue and city search up to a certain extent, because it will show the information of the concerts nearby, no matter the band, what gives more flexibility on the results that are shown, in a similar way those two search types would do. Settings were also discarded because it was not considered a strategic feature, albeit a good complement. Finally, notifications were not kept in the new scope because of the difficulty of testing them in the simulator and the amount of time it would ask for. Notifications were the first feature to be included in the scope, should the redefined scope be fulfilled and more time was available for development but there was no chance for that in the end.