



# User-Centered Design of a weather forecast application for smartphones

Master's Thesis

**Master's Degree in Multimedia Applications**

Professional itinerary

**Author: Jordi Gómez Alberti**

Consultant: Sergio Schvarstein Liuboschetz

Professor: David García Solórzano

17 June 2016

## Copyright

This document is subject to the License Creative Commons:

Attribution – Non Commercial - Share Alike (BY-NC-SA)



A copy of the license can be obtained on:

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

NOTE: Product names, logos, brands, and other trademarks featured or referred to within this document are property of their respective trademark holders. These trademark holders are not affiliated with this document, project, or application. They do not sponsor or endorse our materials. This list is subject to change without notice.

## Thesis file

<b>Thesis title:</b>	<i>User-Centered Design of a weather forecast application for smartphones</i>
<b>Author's name:</b>	<i>Jordi Gomez Alberti</i>
<b>Consultant's name:</b>	<i>Sergio Schvarstein Liuboschetz</i>
<b>Professor's name:</b>	<i>David García Solórzano</i>
<b>Hand in date (mm/aaaa):</b>	<i>06/2016</i>
<b>Masters' program:</b>	<i>Multimedia applications</i>
<b>Master's Degree itinerary:</b>	<i>Professional</i>
<b>Language:</b>	<i>English</i>
<b>Keywords:</b>	<i>Design, Usability, User Experience</i>
<b>Resum del Treball (en català, màxim 250 paraules):</b>	
<p>La tesi realitzada per aquest màster és basa en dos fenòmens molt habituals en el Regne Unit, la rapidesa en que canvien les condicions meteorològiques, les quals afecten en gran mesura les activitats quotidianes, tal com inundacions. I el segon fenomen basat l'augment de dispositius connectats a Internet: smarhpones, tablets, etc...</p> <p>En l'actualitat per aconseguir la previsió meteorològica, hi ha moltes aplicacions, però la majoria d'elles no informen de les condicions meteorològiques greus, o no són totalment gratuïtes. A demes si el usuarios volen tenir les dades més actualitzades els usuarios han de fer ús de diversos mitjans, com poden ser la televisió, internet o premsa, o sinó recórrer a les aplicacions de pagament.</p> <p>Aquest projecte pretén adreçar aquest problema, oferint una aplicació gratuïta i fàcil d'utilitzar que permeti mantenir els usuarios al dia de les condicions meteorològiques.</p> <p>Per tal d'aconseguir aquest objectiu es seguirà la metodologia de disseny centrada en l'usuari, en la qual durant tot el procés de conceptualització, disseny i avaluació de l'aplicació els usuarios estan involucrats activament, per tal d'oferir una aplicació que s'adapti a les seves necessitats, fent ús d'una interfície senzilla d'utilitzar.</p> <p>Una vegada completats tots els processos, s'obtindrà una aplicació ben dissenya i provada, la qual permetrà als usuarios accedir a previsions meteorològiques precises i actualitzades;</p>	

i a més podran rebre notificacions per canvis en la previsió o per alertes greus a qualsevol lloc i temps sense cap cop per ells.

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

This thesis is based on two factors that are very common in the United Kingdom, the fast changing weather conditions, that greatly affect daily activities, such as severe floods and the increased usage of interactive devices connected to the Internet, like smartphones, tablets, etc.

Currently, in order to obtain the weather forecast, there are many applications, however most of them do not inform of severe conditions, or they are not free to use. And if users want to be updated, they have to check different sources, like TV forecast, newspapers, internet, or buy paid applications.

The problem that is being addressed is how to offer an easy and free to use application that can be used by everybody in order to keep up-to-date with all the forecast changes.

The approach used to tackle this issue is following the User-Centered Design methodology, in which during the whole process of conceptualization, design and testing of the application the users are involved in order to tailor the app to their needs while offering a great user interface.

Once all the phases of process have been completed, a well design and tested application will be obtained, that will allow the users to access precise and up-to-date weather forecast data and as well be able to receive notifications, for weather changes and for severe weather at any time for any location with no cost for them.

## Dedicatory

To my parents, grandparents and brother for all their love and support. I appreciate their sacrifices and wouldn't have been able to get to this stage without them.

To Sadia for her unending support, level-headedness and love. I wouldn't have gotten through this Masters' if it wasn't for her.

## Acknowledgments

I would like to express my greatest appreciation to my partner Sadia, for her tireless love, support, understanding and help in making this Thesis happen. To Sergio Schvarstein for his knowledge, guidance and contributions, without whom this Masters' Degree would not have happened.

I would also like to express my sincere gratitude to:

- To all those people who have participated unselfishly in any part of the Masters' Degree Thesis and especially to those, who performed the usability test for your patience and kindness to perform all the tests and tasks.
- All consultants, fellow students and teachers who have been by my side throughout the years of the Masters' Degree in Multimedia Applications.
- My "most excellent" friends who have supported me and kept me going on this journey.

## Notations and conventions

In order to complete this Masters' Degree Thesis, the font Arial, designed by Robin Nicholas and Patricia Saunders, has been used with a spacing of 1.5.

It has been used with different sizes, colours and styles as described:

**Heading One: Arial 16pt Bold**

**Heading Two: Arial 14pt Bold**

**Heading Three: Arial 12pt Bold**

Normal: Arial 11pt

**Heading (Non in table of content) Arial 16pt Bold**

**Captions: Arial 9pt Bold**

## Table of contents

Proposal of tittle.....	10
Keywords .....	10
Summary of proposal .....	10
Justification and motivation. ....	13
Market analysis .....	14
Competitors applications.....	14
Weather Data API.....	20
Objectives and scope .....	25
Primary objectives .....	25
Secondary objectives.....	26
Product scope.....	27
Work plan.....	28
Methodology.....	31
Requirement analysis .....	33
User requirements .....	35
Functional requirements .....	35
Non-functional requirements.....	36
Interface requirements .....	37
Usability requirements .....	37
Technological requirements.....	39
Maintenance requirements.....	40
UCD Design of the application.....	41
Task analysis.....	41
Use cases.....	43
Use Context.....	44



Sample Personas .....	45
Content tree and information flow .....	47
Prototyping .....	48
Sketches.....	48
Wireframes .....	49
Usability and accessibility evaluation.....	50
Heuristic evaluation .....	51
Planning the heuristic evaluation .....	53
Conducting the heuristic evaluation .....	54
Analysing the results and reporting .....	55
Cognitive walkthrough.....	55
User testing .....	56
Interactive Prototype .....	58
Calendar deviations.....	61
Future projection .....	62
Conclusions.....	64
References.....	66
Bibliography .....	68

## Index of tables

Table 1 Work plan .....	28
-------------------------	----

## Index of images

Figure 1: Weather applications .....	14
Figure 2: Accuweather.....	15

Figure 3: BBC Weather .....	16
Figure 4: Met Office Weather.....	17
Figure 5: The Weather Channel .....	18
Figure 6: Yahoo Weather .....	19
Figure 7: Work plan .....	30
Figure 8: Diagrams of User-Centered Design methodology.....	32
Figure 9: Preview of summary responses.....	34
Figure 10: Number of users targeted.....	39
Figure 11: Sample persona Mark Robinson.....	45
Figure 12: Sample persona Martha Woods .....	46
Figure 13: Sample persona Paul Johnson.....	46
Figure 14: Main screen tree.....	47
Figure 15: Sketches .....	48
Figure 16: Wireframes.....	49
Figure 17: Correlation between the number of errors found and the number of evaluators conducting the test. ....	53
Figure 18: Correlation of ratio of benefit and cost depending the evaluators employed. ....	53
Figure 19 : Now and detailed daily (day) views.....	59
Figure 20: Detailed daily (Night) and hourly views .....	60

## Proposal of title

User-Centered Design of a weather forecast application for smartphones

The proposed title summarises the topics and objectives planned to achieve in the Master's Degree Thesis, however as the title indicates I plan to program an application, although this may be not entirely possible, as it depends on all the tasks and milestones proposed on the work plan.

## Keywords

Android, Application, Design, Interface, JAVA, Usability, User Experience

## Summary of proposal

The present project is based on two very common situations in the United Kingdom, which are becoming more common and relevant.

The first is based on weather conditions, which have been recently altered in the UK [1,5], due to climate change, making it more difficult to predict temperature variations throughout the year, making it challenging to forecast snow, frost and rain [10], which has led to severe flooding throughout the country [2,11].

The second is the increase in population that makes regular use of interactive Internet-connected devices, such as smartphones, tablets, etc.

Reviewing the analysis and conclusions of recent reports on the usage habits of mobile devices, media and entertainment conducted by Deloitte and Ofcom in 2015 [3,6], it can be observed that every year the number of mobile phone users have greatly increased, the same has happened with the 3G(HSDPA) and 4G(LTE) Internet connectivity [7]. Moreover, the results show how users are changing their connection habits, meaning that users nowadays prefer to use smartphones or tablets as opposed to computers, the decreased use of these devices are causing them to fade into the background. [8,13].

Based on these two phenomena this project aims to analyze and document in detail all the stages that need to be carried out in order to create a multimedia and interactive application for interactive devices that utilize Android platform, this project will mainly focus its attention especially on smartphone devices. The ultimate goal is to develop a weather forecast application, i.e., an application that allows users to view weather forecasts, news related to the weather and receive weather alerts notifications of the most relevant situations.

The Master's Degree Dissertation is structured accordingly to the different phases of the User-Centered Design methodology [4], which is based on a set of phases that are executed iteratively, which engages the users in all phases of development, in order to obtain a quality product, which has been tested in usability, heuristics and user experience, to be able to guarantee that the application designed has been developed following the requirements and specifications identified by the users and includes all the functionalities they require.

Over the course of the project, all the different phases of the UCD methodology, will be documented in a precise and detailed format, which includes the following stages:

- Plan and research
- Evaluate and analyze
- Design and develop
- Program and implement
- Test and refine.

Initially in the first phase, the planning and research stage, will be used to define the requirements and context of use needed for the application, which will be obtained after observing and questioning the user with the recommended methods and techniques.

Once the required has been obtained, we can define the users' profiles, possible use cases and generic personas.

For the conceptualization of the application the existing normative and regulations in terms of usability and accessibility will be taken into consideration, aiming to provide an application usable for the majority of the users.

Once the theoretical foundations have been created, the first low-quality sketches will be designed, so the first set of tests can be done such as: user testing, app navigation, content sorting.

Based on these low-quality models of the interface, a second iteration of higher quality will be designed, these ones can be wireframes or high-quality mock-ups and once completed they will be used again for user testing and refinement.

In the last phase of development and prototyping an interactive prototype will be created, where another set of tests will be conducted, such as: heuristic evaluation, cognitive walk, etc. With these last tests performed by experts and users, the interface will be validated according to the needs and requirements that were set on the early stages of the project, guaranteeing an application tailored to the real needs, specifications and requirements.

During the development of the touch interface, special care must be taken to adapt the system to the real needs of users, whilst ensuring its accessibility and usability by the vast majority of the population, regardless of age, knowledge or skills.

Once these stages of the UCD methodology have been completed, conceptual and design guidelines will have been created, and will be possible to transform from a conceptual framework to a real application.

Finally, this would be the last phase or milestone, which would be the programming of an Android application in native JAVA code, it should be noted that the application offered, will be limited in functionalities and will be presented as the first alpha. On which future development will carry on until the final application is completed offering the same functionalities as the prototype.

## Justification and motivation.

During my Graduate studies in Multimedia I specialized in Interactive interfaces, and now with my Master's Degree I have expanded my knowledge in this area especially and other relevant areas related with the design of multimedia applications, however after all these years I feel like I have not achieved my final goal, to fully develop an app, which involves completing the whole life cycle of an application that includes: planning, designing, developing, programming and testing.

The main objective of this project is not only learn more about usability, heuristics, user experience, prototyping, designing interfaces and user testing, but also get to know a recent technology and market that is booming and greatly increasing in users and in revenue, which this project would allow me to experiment with it, learn a new discipline and a new field of the life cycle of an app, which could form a good basis to apply to my professional curriculum.

The most attractive aspect from my point of view is the technological challenge to program the application in native JAVA code, because I have basic knowledge obtained during my last years at university, however I have never used it to create a whole product.

Moreover, besides from gaining new knowledge and skills I will have achieved a fully functional weather forecast application on the completion of the dissertation.



This section will analyze some of the most popular ones and our main competitors, to see their characteristics:

### AccuWeather

- Between 50.000.000 - 100.000.000 applications installations.
- Free to use.
- Using Google Material design for Android.
- Updates frequently.
- Offers very accurate forecasts for: current, hourly and daily basis.
- Offers interactive maps and video updates.
- The interface follows the new design guidelines from Google.

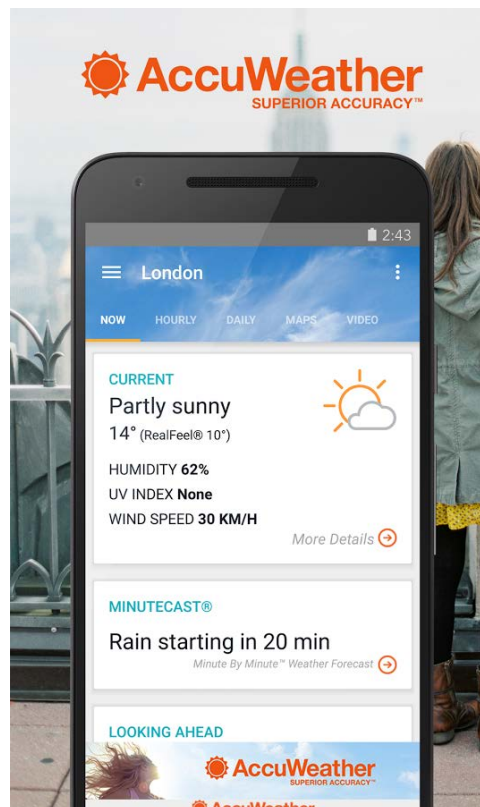


Figure 2: Accuweather

<https://play.google.com/store/apps/details?id=bbc.mobile.weather>



## BBC Weather

- Between 5.000.000 - 10.000.000 applications installations.
- Free to use.
- Updates frequently.
- Simple and clear design for easy access to the most detailed and up-to-date information.
- Offers very accurate forecasts for: current, hourly (UK only) and daily basis (international).
- Offers Sunrise/Sunset times, UV, with Pollen and Pollution information for the UK.
- See most likely high and low temperatures, and temperature ranges, for days 6-10



Figure 3: BBC Weather

<https://play.google.com/store/apps/details?id=bbc.mobile.weather>

## Met Office Weather

- Between 10.000 - 50.000 applications installations.
- Free to use.
- Updates frequently.
- New Interface following Material design
- Get the most up-to-date and accurate weather forecast information available for the next seven days, for many locations across the globe.
- UK national weather forecast video
- UK National Severe Weather Warnings
- UV Index, visibility, humidity, pressure.
- Official application of the UK Met Office

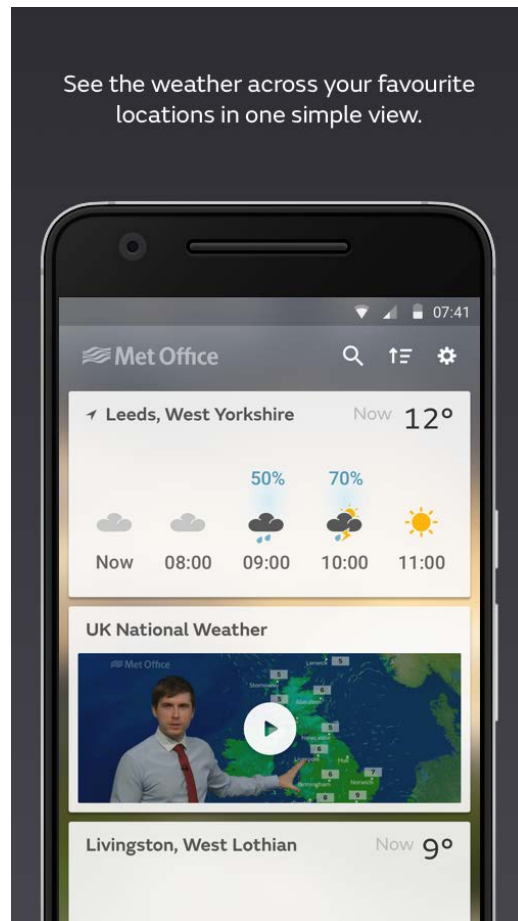


Figure 4: Met Office Weather

<https://play.google.com/store/apps/details?id=uk.gov.metoffice.weather.android>

## The Weather Channel

- Between 50.000.000 - 100.000.000 applications installations.
- Free to use.
- Updates frequently.
- Interface looks and feels outdated.
- Offers accurate forecasts for: current, hourly and daily basis.
- Get the most up-to-date and accurate weather forecast information available for the next fifteen days, for many locations across the globe.
- Offers Sunrise/Sunset times, UV Index, visibility, humidity, pressure.
- Weather forecast video depending on the region.
- Severe Weather Warnings on the region.

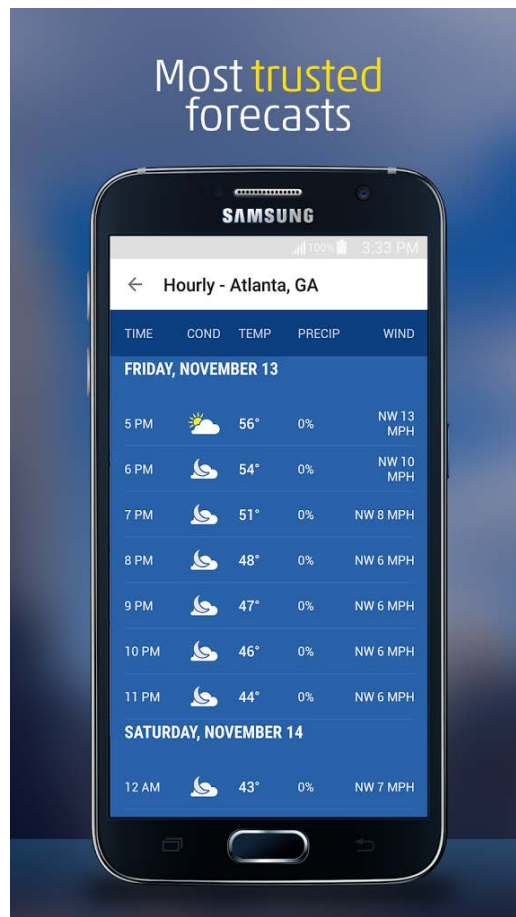


Figure 5: The Weather Channel

<https://play.google.com/store/apps/details?id=com.weather.Weather>

## Yahoo Weather

- Between 10.000.000 - 50.000.000 applications installations.
- Free to use.
- Updates frequently.
- New Interface with actual pictures of cities.
- Offers very accurate forecasts for: current, hourly and daily basis.
- Get the most up-to-date and accurate weather forecast information available for the next ten days, for many locations across the globe.
- Offers Sunrise/Sunset times, UV Index, visibility, humidity, pressure.



Figure 6: Yahoo Weather

<https://play.google.com/store/apps/details?id=com.yahoo.mobile.client.android.weather>

## Weather Data API

In final stages of the UCD methodology, especially during the implementation stage, in order to show real and actual data to the user, data needs to be previously obtained from a weather data provider.

There are many websites providing weather data using their API's, some of them are free, while others offer different pricing plans.

Once the data has been obtained in JSON or XML, it has to be processed to be able to manipulate it, and show it to the user. This is the most important factor after the design of the interface, providing accurate and reliable data is what makes the application important and relevant to the users.

This section will include an analysis and comparison of the different providers in order to find the best suitable data provider for the application.

### AccuWeather

- Offers weather forecast for more than 3 million locations worldwide
- The access is done via a RESTful web service
- Data is available in more than 100 languages and dialects
- Response to the API is returned in JSON
- Data calls can be secured using SSL protocol
- No free plan available, need to contact sales for a quote.

<http://apidev.accuweather.com/developers/>

As it is not offering a free plan this provider is discarded.

### Aeris Weather

- Can buy add-ons to tailor the API to your requirement.
- Response to the API is returned in JSON

- There are five API plans available depending on the functionalities, the price depends on the number of calls to the API per day:
  - Developer API: up to 750 calls per day.
  - Basic API: less than 5.000 calls per day. Pricing starts at 25\$/month
  - API Plus: less than 50.000 calls per day. Pricing starts at 150\$/month
  - API Premium: less than 50.000 calls per day. Pricing starts at 360\$/month
  - API Enterprise: more than 1.000.000 calls per day. Need to contact sales for a quote.

<http://www.aerisweather.com/signup/#api>

This provider can be considered as it is offering an API free of charge with a good set of functionalities.

## Open Weather

- Access current weather data for any location including over 200,000 cities
- Current weather is frequently updated based on global models and data from more than 40,000 weather stations
- Response to the API is returned in JSON, XML or HTML
- Two API available:
  - Free API: up to 60 calls per minute.
  - Premium API: there are different tiers of pricing, depending on the number of the calls per minute or functionalities of the API:
    - Startup: less than 600 calls per minute. Pricing starts at 40\$/month
    - Developer: less than 3.000 calls per minute. Pricing starts at 180\$/month
    - Professional: less than 30.000 calls per minute. Pricing starts at 470\$/month
    - Enterprise: less than 200.000 calls per minute. Pricing starts at 2.000\$/month

<http://openweathermap.org/api>

<http://openweathermap.org/price>

This provider can be considered as it is offering an API free of charge with a good set of functionalities.

### The Dark Sky Forecast API

- The Forecast API lets you query for most locations on the globe, and returns:
  - Current conditions
  - Minute-by-minute forecasts out to 1 hour (where available)
  - Hour-by-hour forecasts out to 48 hours
  - Day-by-day forecasts out to 7 days
- Free to use up to 1.000 calls per day, after that costs \$0.0001 each.
- Response to the API is returned in JSON
- Need to Credit us with a “Powered by Forecast” badge that links to <http://forecast.io/> wherever you display data from the API.

<https://developer.forecast.io/>

This provider can be considered as it is offering an API free of charge with a good set of functionalities.

### Weather Bug

- They have the largest networks of real-time weather and lightning sensors in the world.
- The API provides real-time weather and lightning information, forecasts utilizing forecast models with the lowest margin of error, and other top-notch weather data.
- The access is done via a RESTful web service.
- API undergoing maintenance only available the legacy API.
- No free plan available, three paying plans available:
  - Basic starting at 20\$/month.
  - Plus starting at 150\$/month.
  - Pro starting at 300\$/month.

<http://legacy.weather.weatherbug.com/pulseapi.html>

<http://legacy.weather.weatherbug.com/pulseapi/compare.html>

Cannot register to use the actual API or the legacy one, as of this the provider is discarded.

### **Weather Underground**

- Reliable data, accurate forecast, & global coverage in 80 languages
- The access is done via a RESTful web service.
- Response to the API are returned in JSON or XML
- There are three tiers available, depending on the service and functionalities they offer:
  - Stratus plan (basic package)
  - Cumulus plan (intermediate package)
  - Anvil plan (complete package)
- There are different tiers of pricing, depending on the number of the calls per minute or per day to the API:
  - Developer: less than 500 calls a day or 10 per minute. Free to use
  - Drizzle: less than 5000 calls a day or 100 per minute. Pricing starts at 20\$/month
  - Shower: less than 100.000 calls a day or 100 per minute. Pricing starts at 200\$/month
  - Downpour: more than 100.000 calls a day. Need to contact sales for pricing

This provider can be considered as it is offering an API free of charge with a good set of functionalities.

<https://www.wunderground.com/weather/api>

<https://www.wunderground.com/weather/api/d/pricing.html>

### **World Weather Online**

- Provides current weather conditions and hourly weather forecast for up to 15 days across worldwide locations. Local weather API also provides access to monthly climate average data to deliver aggregate weather across the world.
- Free to use up to 250 calls per day.
- Two API available:
  - Free API: up to 250 calls per day.



- Premium API: more than 250 calls a day. Pricing starts at 10\$/month
- The Premium API can be tailored to your needs with the functionalities you require.

<http://developer.worldweatheronline.com/api/>

<http://developer.worldweatheronline.com/api/compare.aspx>

This provider can be considered as it is offering an API free of charge with a good set of functionalities.

### **Yahoo Weather**

- Get up-to-date weather information for any location, including 5-day forecast, wind, atmosphere, astronomy conditions, and more. You can lookup weather by id, city name or lat/long.
- Response to the API are returned in JSON or XML
- Free to use up to 2.000 calls per day.
- Need to Include a "powered by Yahoo" logo along with any data from a Yahoo API.

<https://developer.yahoo.com/weather/>

This provider can be considered as it is offering an API free of charge with a good set of functionalities.

After this first research the weather providers that can be used for the application are:

- Aeris Weather
- Open Weather
- The Dark Sky Forecast API
- Weather Underground
- World Weather Online
- Yahoo Weather

Later on when all the requirements of the application have been found an API will be chosen, based on the functionalities that the users require.

## Objectives and scope

The present dissertation is based on the professional itinerary of the Master's Degree and as a consequence it is aimed on developing a product or a service.

In this particular case, the dissertation is mainly focused on the UCD methodology, where all the different stages are going to be completed to achieve a final product to answer a certain problem, whilst complying with a set of requirements defined by the users.

Therefore, the final objective is the creation of a mobile application for Android, that is based on all the studies realized during the analysis and design cycles.

### Primary objectives

Create a mobile phone application that allow the users to receive the weather forecast for the cities they have searched and added to favourites, and as well receive notifications of severe weather conditions that may affect them.

- Be able to obtain weather data from a certain location
  - Create scripts and code to be able to download remote data and process it.
- Allow to search places by name or user location
  - Use of different API's, for this use case Geo-localization
- The application has to include all the functionalities stated by the users and has to be able to execute them in a logical manner.
  - Use the UCD methodology to obtain all the requirements and needs of the users, and complete different sets of tests to verify the design and implementation of the tasks and data structure.

## Secondary objectives

- Define the profile of the users that are going to use the application.
- State their needs and requirements and all the functionalities the program has to offer and create hypothetical use cases.
- Create sketches, wireframes and interactive prototypes of the application.
- Analyze and evaluate the mockups and prototypes to assess the accessibility, usability and user experience.
- Verify that the application complies with the 10 Usability Heuristics for User Design Interfaces defined by Jakob Nielsen [9].
- Develop all the analysis and designing of the application with the knowledge acquired in the Master's Degree subjects' such as: Interactive Interface Development, Advanced Management of IT projects.
- Document all processes and stages executed in a detailed and formal manner for the dissertation.
- Obtain in-depth knowledge of usability, accessibility and interface design by using specific software.
- Expand the knowledge acquired throughout the Master's Degree by researching and studying new techniques for development interactive applications.
- Learn and use a specific technology: Android-SDK, development, configuration and API's usage.

## Product scope

Apart from the primary and secondary objectives, these items or tasks are the ones that need to be completed in order to successfully achieve the goal of this particular dissertation.

- Analysis and design of the user needs and requirements, creating user personas and use cases.
- Create the navigation tree and flow of information.
- Conceptualize and design mockups and interactive prototypes, to test and verify them.
- Create an initial but fully working application, with the main functionalities included.

## Work plan

The dissertation work plan has been planned taking into consideration the hand-in dates of the different PAC, which have been marked as partial milestones, being PAC5 the last one and being considered as final milestone.

Some of the tasks may overlap, but this doesn't cause any issue, as one of the main key point of the UCD methodology is being iterative.

Table 1 Work plan

Task	Duration	Start Date	End Date
<b>PAC1</b>	15 days	29/02/2016	14/03/2016
Proposal of title	1 day	29/02/2016	29/02/2016
Keywords	1 day	29/02/2016	29/02/2016
Proposal	7 days	01/03/2016	07/03/2016
Justification and motivation	4 days	08/03/2016	11/03/2016
<b>PAC2</b>	14 days	15/03/2016	28/03/2016
Market analysis	5 days	15/03/2016	19/03/2016
Objectives	2 days	20/03/2016	21/03/2016
Scope	2 days	20/03/2016	21/03/2016
Developing the work plan	4 days	22/03/2016	25/03/2016
<b>PAC3</b>	28 days	29/03/2016	25/04/2016
Methodology	2 days	29/03/2016	30/03/2016
User survey	2 days	30/03/2016	31/03/2016
Survey distribution	4 days	01/04/2016	04/04/2016
Survey analysis	1 day	05/04/2016	05/04/2016
User requirements	3 days	05/04/2016	07/04/2016
Functional and non-functional requirements	3 days	05/04/2016	07/04/2016

Interface requirements	3 days	05/04/2016	07/04/2016
Usability requirements	3 days	05/04/2016	07/04/2016
Use cases	2 days	08/04/2016	09/04/2016
Task analysis	2 days	08/04/2016	09/04/2016
Navigation tree	5 days	10/04/2016	14/04/2016
Information flow	5 days	10/04/2016	14/04/2016
Sketches (Lo-Fi)	3 days	15/04/2016	17/04/2016
Wireframes (Hi-Fi)	4 days	18/04/2016	21/04/2016
<b>PAC4</b>	28 days	26/04/2016	23/05/2016
Interactive prototype	10 days	26/04/2016	05/05/2016
Heuristic evaluation	4 days	06/05/2016	09/05/2016
Cognitive walk	4 days	06/05/2016	09/05/2016
User testing	4 days	06/05/2016	09/05/2016
Results and modifications	3 days	10/05/2016	12/05/2016
Coding	24 days	26/04/2016	19/05/2016
<b>PAC5</b>	21 days	24/05/2016	13/06/2016
Completion of remaining tasks or objectives (Safeguard time)	12 days	24/05/2016	04/06/2016
Video demonstrating application	4 days	05/06/2016	08/06/2016
Presentation	4 days	09/06/2016	12/06/2016
Monograph completion and revision	20 days	24/05/2016	12/06/2016

After declaring the different tasks that need to be completed, the next Gantt chart is obtained:

It can be seen in full detail in the file [workplanTFM.mpp](#)



## Methodology

To complete this project, the User-Centered Design (UCD) methodology is going to be used to customize the features and adapt the interface to the users' needs

UCD methodology outlines the phases throughout a design and development life-cycle all while focusing on gaining a deep understanding of who will be using the product.

Known the philosophy of UCD, this process will be applied in the project, thus dividing it in three phases (analysis, development and evaluation). However, it's important to remark that UCD is an iterative process, meaning that this is not a closed process or unidirectional. It is dynamic process that evaluates and re-evaluates itself constantly in order to offer the best product tailored to users' needs.

In the first phase: the analysis and research phase, there will be an ongoing dialogue with the potential users, our target, is to be able to define properly the scope of the project, its requirements, their real needs, etc.

Different techniques will be used to obtain qualitative and quantitative data. Once processed, the data and all the basic input for the development of the application will have been detailed.

This stage of the project will be used to define the structure and scope of the project, that will be implemented during the development phase.

On this stage there will also be a benchmarking with similar applications and later on all application architecture will be developed including task analysis, use cases, contents tree and flow diagrams.

The last phases: development and evaluation will be conducted simultaneously to provide a tested and tailored application to the users' real needs.

Once all the basics of the application are defined, such as requirements, scope, etc. low quality mock-ups will be designed (sketches and wireframes), lastly a high quality interactive prototype will be created, to verify once again the interaction and user experience.

At the same moment a constant evaluation will be carried to assess and verify the usability and accessibility of the prototypes created, using techniques such as: heuristic evaluation,



cognitive walkthroughs, tests with users, etc. To improve or correct problems that have been overlooked during the previous stages.

Finally, after the prototypes have been verified a preliminary version of the application will be coded, in which later on all future upgrades will be built on to.

Lastly a short video will be recorded, to show the performance and some of the functionalities available to the users and at the same time, it will be used as an introduction tutorial on how to use and interact with the application.

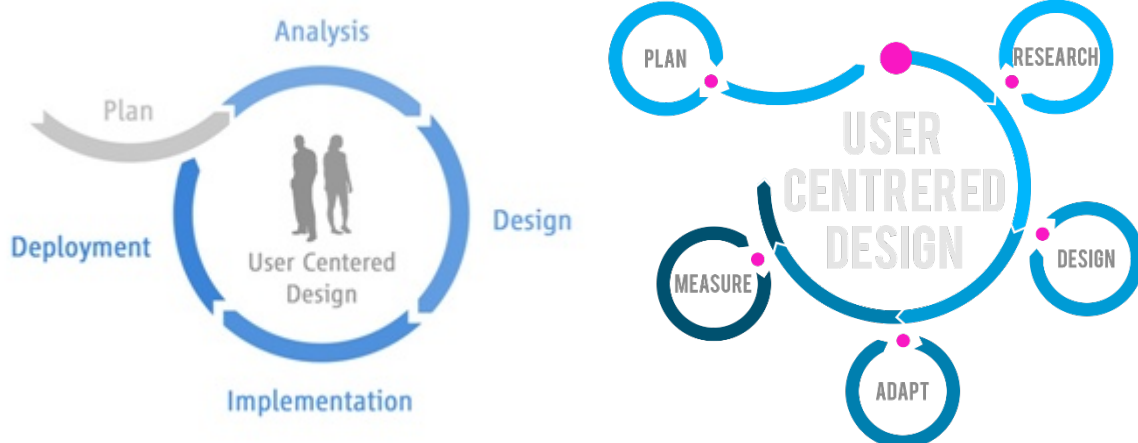


Figure 8: Diagrams of User-Centered Design methodology

## Requirement analysis

This section plans to define and establish all the requirements needed to successfully complete the design and implement the weather application.

This stage of analysis during the phase of design is essential, as the requirements observed in this phase are going to be used during the whole project design.

All the requirements detailed in the next sections, haven been obtained by an online survey in which 117 people took part.

Survey link: Data collection for creation of weather application.

<http://goo.gl/forms/kbA67kXVDw>

In this survey the participants where asked a set of questions in order to find out their needs and what would they expect from a weather application.

The questions were:

- Gender
- How old are you?
- How often do you check for the weather forecast?
- Would you be interested in cloud maps? (Radar imaging)
- Would you be interested in videos related to weather?
- Would you be interested in flood warnings?
- Would you be interested in UV values?
- Would you be interested in Wind & Pressure data?
- Would you be interested in Pollen data?
- Would you be interested in Sunrise and Sunset data?
- Would you be interested in a notification with current weather data, providing access to a detailed view?
- What forecast data would you be interested in?
- What kind of application would you be interested in?
- If you had to pay to use the application, how much would you be willing to pay?

Link to responses in a data table:

[https://docs.google.com/spreadsheets/d/1SzQ2XmIGspJCs\\_6FLZekoQSQrcH3FPEvAoCZ8i4nc-Q/edit#gid=244062220](https://docs.google.com/spreadsheets/d/1SzQ2XmIGspJCs_6FLZekoQSQrcH3FPEvAoCZ8i4nc-Q/edit#gid=244062220)

Link to response in a graphic view:

<https://docs.google.com/forms/d/11eFDyrkJxcgq0iUTTLRTpUoZfQUXUmFN6XfV7s6st4Q/viewanalytics>

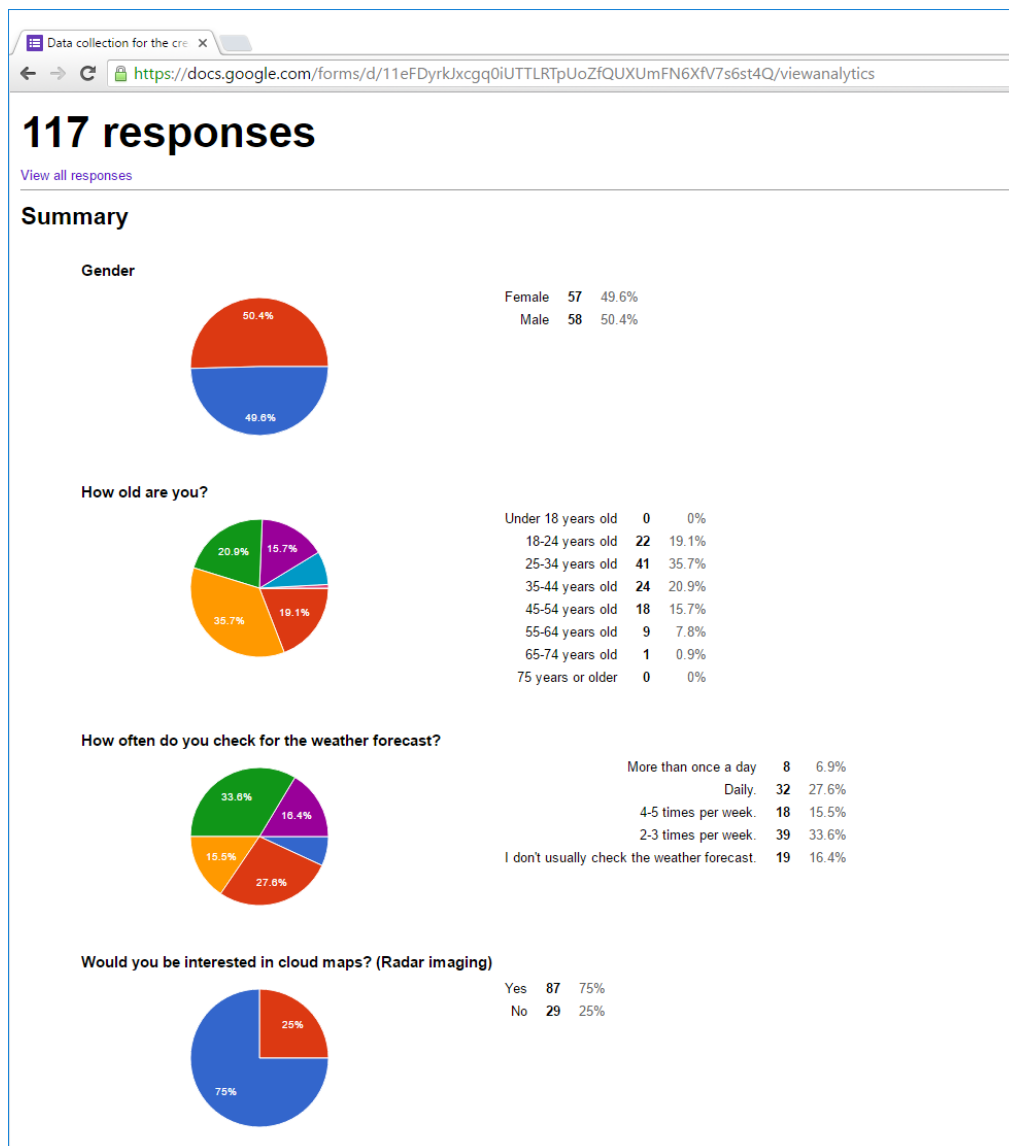


Figure 9: Preview of summary responses

The different sections are going to evaluate different kinds of requirements, such as: user requirements, technological requirement, functional and non-functional requirements, interface and usability requirements, that need to be achieved during the consequent phases of the weather application development.

## User requirements

All the functions available to the users:

- Users can access weather forecast data and receive notifications with weather updates.
- The weather data has to be for: current, hourly, daily, and up to 10 days' forecasts.
- Users can access radar imaging data.
- Users can receive severe weather conditions notification; such as flood warnings.
- Users can access UV data values.
- Users can access Wind and Pressure data values.
- Users can access Pollen data values.
- Users can access Sunrise and Sunset data values.
- The application can be used by a large number of users simultaneously. This means the API has to allowed a large number of calls per minute.
- The performance of the application cannot decrease while adding new functionalities

## Functional requirements

These are the requirements that define the internal behaviour of the application. They are based on the Use Cases and guarantee a fully functional application.

- Users can add and remove locations to receive their weather data.
- Locations can be added by user coordinates (Geo Position) or location search.
- Application shows notifications on weather update on selected locations
- Application offers data on the location saved by the user, in ranges of:

- Current
- Hourly
- Up to 10 days.
- Notifications allow to access a detailed information.
- On severe weather conditions users gets notified.
- Users can access different types of data:
  - Radar imaging
  - UV data
  - Wind Pressure
  - Pollen
  - Sunrise and Sunset
- Data shown is obtained and aggregated from different sources.
- The application must support different languages, through the use of Locales in Android, so the language selection is transparent to the user, as it will load the text in the language defined in their device.

## Non-functional requirements

This section includes and analyses all the requirements of the system that are not contemplated in the Use Case Scenarios, but they are essential for the smooth running of the application. These are needed as they define basic aspects to successfully launch the application and must be met before being able to offer a final product that meets all other requirements set.

- Application has to be modular, in order to add new functionalities easily.
- Application has to follow Material Design Guidelines set by Google, and maintain a unique interface all around the application.
- The application content has to be available 24/7. Use of multiple weather API in order to aggregate data in order to function even with some providers down.
- Low response time, all actions should be processed instantly and data should update promptly.

- Application should work without errors in order to be fast and provide a good user experience. However, in case errors happen they should get saved in a log file including ID of the error, reason, data and hour, so they can be analysed and fixed.

## Interface requirements

- Interface design must be flexible and scalable to be used in different devices with different screen resolutions and sizes.
- Interface should be easy to use, rather than intimidating and requiring user learning.
- Interface should be intuitive and provide easy to understand items such as: buttons, headings, messages or errors.
- The screen layout and colours should be attractive.
- Interface must be consistent: elements should be organized similarly across the interface, allowing users to predict what will happen when an action is taken.
- Content should be displayed in a logical and natural order.
- Interface should display all the necessary information at all times, so users do not need to remember information.
- Users have to know at any time, in which part of the application are located.
- Users can navigate to the home page from any part of the application.
- Interface response has to be fast, to avoid waiting time for users or making them think that the application does not respond, the ideal response time should be less than 0.2 seconds.
- Progress bar or a timer should be shown in case of slower response times.

## Usability requirements

- Users do not need to have any specific skills for normal use of the application.
- Images and icons used must be representative.

- Text must be large enough to be readable without difficulty, using scale-independent pixels (SP), in case the standard defined size does not offer enough readability users can change the text size.
- Efficiency of use: goals (tasks) must be easy to perform and should not lead to confusion.
- Application should be easy to learn.
- All possible actions and elements of the interface must react accordingly and consistently.
- In the event of error messages, they should explain how to fix the error.
- Undo option should be offered in the majority of the actions, especially the permanent ones.
- Offer feedback to the users after completing an action.
- Help must be sensitive to the context; it should explain how to perform tasks.
- When a new functionality is added, provide relevant information and guided steps on how to use it.

## Technological requirements

This list includes all the requirements for the execution of the project and for the usage of the application

### Mobile Device: smartphones.

Requirements for a recommended experience of the application.

- Operating System Android 4.0.3, API 15, in this way we can target 97.3% of the active users on Google Play Store.

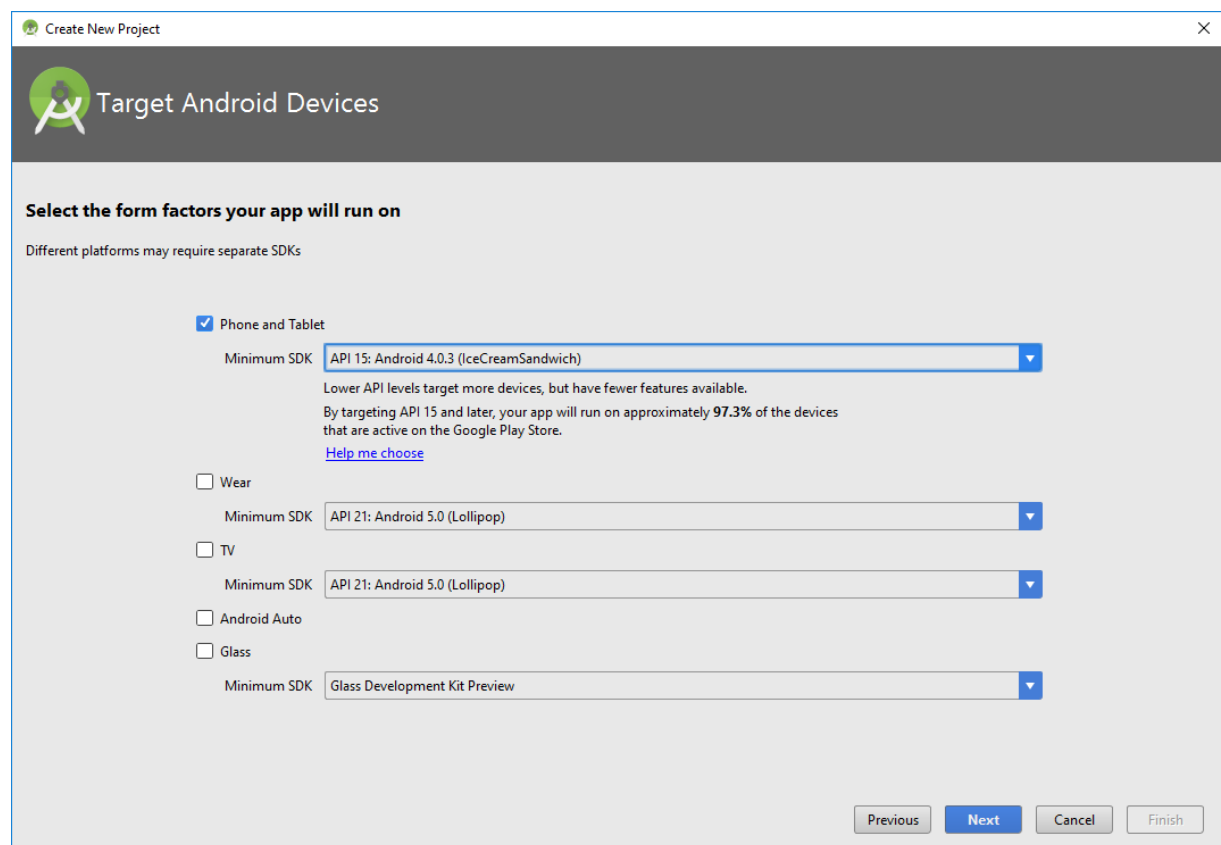


Figure 10: Number of users targeted

- Low usage of CPU, to prevent high use of battery.
- Low use of data connection on mobile data, as some data plans are capped.
- Low usage of RAM, in order to not slow down the device.



## Software

Software used for the development of the project

- **Microsoft Word, Excel, Project, PowerPoint** used to write all documents related to the project.
- **Xtensio** [14] used to create the personas.
- **Balsamiq** used to create the sketches and Wireframes
- **Justinmind, Axure RP** used to create the Interactive prototype.
- **Adobe Fireworks, Photoshop, Illustrator CC** used to capture, design and process images.
- **Adobe Premiere, Audacity** used for capturing audio and creating audio tracks.
- **Acrobat Professional** used for the final editing of the Dissertation
- **Google Maps** used for user positioning, weather maps...
- **Android Studio** to code, compile and pack the application.

## Maintenance requirements

After following all the stages of UCD methodology detailed in this project, a well-designed and tested application will have been coded, however this will only be the first release of the application, in which all future modifications and upgrades will be built.

Once the launch and distribution of the application is completed through Google Play Store, the first phase of lifecycle of the application will be finished and the maintenance and upgrade period will begin. In this period different tasks will be required, to be able to continue and guarantee the optimum performance and up to date functions, such as:

- **Functional and usability upgrades:** update the application with new functions obtained through user feedback and new usability guidelines.
- **Interface and graphic upgrades:** to keep the UI up-to-date and follow any modifications on the design guidelines set by Google.
- **Technical upgrades:** to upgrade the code to new coding standards and to make sure the application is compatible with new releases of the O.S. or new devices.

- **Routine maintenance:** to guarantee that the application keeps working, offering a good service and user experience, verifying that the integration of the different API's is well executed.

## UCD Design of the application

UCD is defined by involving the users from the beginning until the end of the designing of the project. Thus, using this approach, the conceptual model of the application will be developed following the mental model of user, such as: characteristics, needs and goals.

As opposed to other methodologies, which start with planning, designing the application and later on the users evaluate the prototypes; UCD methodology starts thinking and researching about the users that are going to use the application, to find exactly what they really need. One of the goals of UCD is to adapt the User Interface (UI), to the different needs of the users that will be using the application, instead of forcing the users to change their way of working and interacting with the application.

The phases of design, prototyping and evaluation are cyclical and should be performed continuously to eliminate different errors, such as: usability, navigation and accessibility; until the application fulfils the needs and expectations of users.

Applying this methodology, a high quality application will be achieved that will be useful to users.

## Task analysis

The application has to allow the users to complete different sets of tasks in an efficient and simple manner.

On the first release of the application the users will be able to access the following tasks, such as:

### Location:

- **Add location:** users can add a new location in order to get updated weather data, this task can be completed by adding the location by a search or by using the Geo-position.
- **Remove location:** users can remove a location from which they do not want to receive more weather updates.
- **Sort locations:** users can sort the locations that they have previously saved, in order to see the data in the order they want.

### Settings:

- **Customize units for temperature:** users can choose between Fahrenheit or Celsius.
- **Customize units for wind speed:** users can choose between Miles per hour or Kilometers per hour.
- **Notifications:** users can enable or disable weather updates notifications.
  - **General notifications:** providing up to date forecast data.
  - **Severe weather:** notifies the users from important weather updates.

### Weather Data:

- **Forecast data:** users can access weather and other data for their saved locations:
  - **Time interval:** the data than can be accessed is separated into sections:
    - **Current:** provide the user the actual forecast data for a location.
    - **Hourly:** provide the user hourly forecast data up to 24hours.
    - **Up to 10 days:** provide the user daily forecast data up to 10 days.
  - **Other data:**
    - **Radar imaging:** provide the user cloud forecast data.
    - **UV data:** provide the user data relative to the Sun intensity values.
    - **Wind Pressure:** provide the user data related the wind strength and direction.
    - **Pollen:** provide the user data related to pollen, to prevent allergies
    - **Sunrise and Sunset:** provide the user data about the sunrise and sunrise.

### In-App purchases:

- **Ad-free:** allows the user to remove the ads of the application, with a single purchase.

- **Extra features:** allows the users to obtain premium version of the application with extra functionalities.

## Use cases

These scenarios describe some of the tasks that users' needs to complete in order to achieve a goal:

### Scenario 1: Add a new location

- **Persona:** Peter B, business man 34 years old, used to travel a lot.
- **Context:** He is going to be travelling to many different cities in the next month and he wants to know the weather forecast in advance.
- **Actions:** Launch the application, access the location menu, search for the different locations and add them to favorites.

### Scenario 2: Check weather forecast for a specific date.

- **Persona:** Peter B, same person as the last scenario.
- **Context:** In 3 days he is going to fly to Colombia, Bogota and he needs to know what kind of clothing to pack.
- **Actions:** Launch the application, access the location menu, select the location where he wants to obtain the data. Access the daily forecast section, and check for the date he is interested in.

### Scenario 3: Change the measure units

- **Persona:** Charlene W., student 23 years old, on Erasmus to study on Oxford University.
- **Context:** She just flew from US to study in the UK, she wants to know what weather is going to be like for tomorrow, but all the data is expressed in a unit she doesn't comprehend-
- **Actions:** Launch the application, access the settings menu and change the temperature (Fahrenheit /Celsius) and speed unit (Miles per hour/ Kilometers per hour).

#### Scenario 4: Sort the locations

- **Persona:** Charlene W., same person as the last scenario
- **Context:** She is going to be living in Oxford for at least a couple of years, so she does not want to have her hometown Atlanta as the main location
- **Actions:** Launch the application, access the location menu, click on edit locations and in that menu the locations can be sorted in any order.

#### Scenario 5: Checking pollen values

- **Persona:** Charles G, veterinary student 20 years old, he is allergic to pollen.
- **Context:** He is going on a field trip with the university to a farm on the countryside, he needs to know how are the pollen levels in order to know if he needs to take extra medication.
- **Actions:** Launch the application, access the location menu, search for the location he is interested in, select the day and check for pollen values.

#### Scenario 6: Disable automatic updates and notifications

- **Persona:** Marlene T, high school teacher 38 years old, on half term holidays.
- **Context:** She is going on holidays abroad and her plan doesn't allow data roaming so she needs to disable automatic updates, and just allow wireless updates.
- **Actions:** Launch the application, access the settings, change the connectivity options to wireless, disable the automatic updates and disable the notifications.

## Use Context

This topic describes the conditions in which the product will be used, in this case there are factors that will influence its use and satisfaction of specific users.

These variables or factors which influence the use of the application might come from the environment (place, time, noise...) or technical issues (connectivity, performance...).

In terms of environment:

- As it is a ubiquitous application, the environment can be completely different for each user, they can use the application anywhere, such as: their home, at work, commuting

by public transport, socialising in a bar or cafe, walking down the street, sunbathing on the beach...) provided they meet the requirements of connectivity.

- As far as the time of use:
  - Smartphone: the application can be used at any time of day.
- Environmental noise will not affect the use of the application; however, a silent environment is recommended for the comfort of the user.


Technically there is only one possible option:

- Using a smartphone with Android O.S.
- To use the application and get updated weather data, the devices have to have an active Internet connection, otherwise the application will launch but the data will not update.
- The performance of the application will be based on the device, depending on the CPU and available RAM of each smartphone.

## Sample Personas

This section shows generic personas that can use the weather application.

### Mark Robinson



"Cyclist during the day, musician at night."

Age: 28  
Work: Bus driver  
Family: Single, living with family  
Location: London, United Kingdom  
Character: Extrovert, easy-going

Personality

Introvert	Extrovert
Analytical	Creative
Conservative	Liberal
Passive	Active

Enthusiastic   Friendly   Fun-loving

Leader   Sociable

Goals

- Wants to become a renowned pro-amateur cyclist.
- Wants to return to college to finish his career in arts and drama.
- Travel around the world with his bike.
- Release a CD music with his band.

Frustrations

- With his work, since he must work long hours and it is rota based.
- The family problems that have occurred, many of which have truncated his dreams and goals.
- The difficulty of interfaces that make using applications quite complicated, as he is not very skilled with technology.
- Having to use different weather applications in order to obtain a fully detailed information of weather data.

Bio

Mark is currently working as a bus driver, because he had to leave university to be able to help finance his family.

At the same time, he is attempting to save money to go back to university and once he graduates, he wants to travel around the world with his bike.

He uses his free time playing music in pubs and other events, riding his bike around London and competing in the amateur cyclists leagues around UK, he is also part of the cycling team of the university, which is the captain.

Currently he thinks that the available applications are not sufficient or they do not provide accurate forecast data. He aims to find an application that can provide him with all the data he needs, being accurate. He would prefer a free app, but wouldn't mind ads or paying a little money for a premium service.


Use Cases


- Be able to visualize weather data detailed by hour for the actual day.
- Be able to visualize forecast data for rain, wind, UV and sunrise and sunset to be able to properly plan his cyclist training.
- Keep updated on the weather while cycling, receiving notifications on his smartwatch.

Motivations

Family	▬
Achievements	▬
Growth	▬
Power	▬
Social	▬

Brands





SHIMANO

Figure 11: Sample persona Mark Robinson

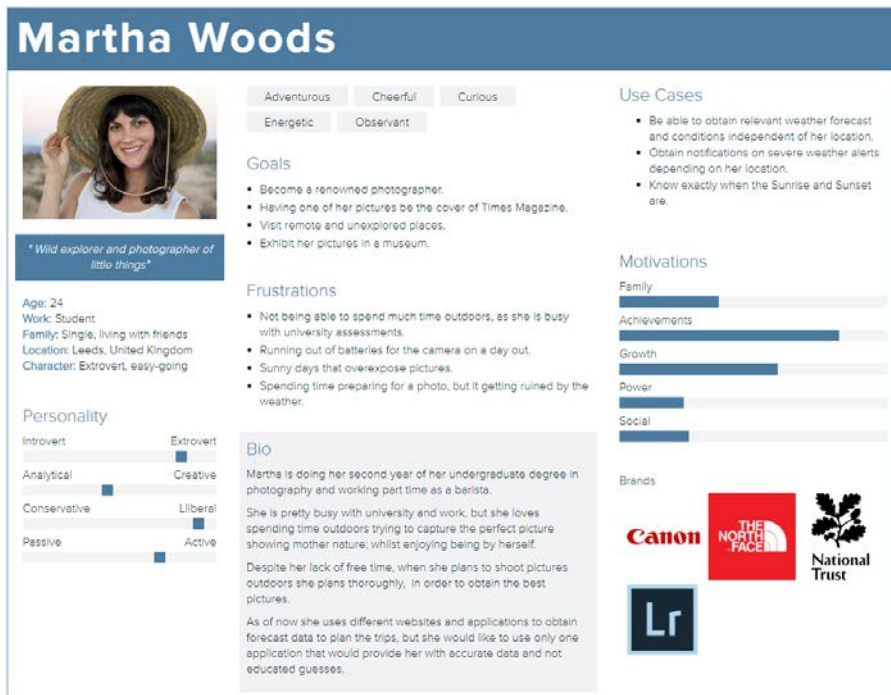


Figure 12: Sample persona Martha Woods



Figure 13: Sample persona Paul Johnson

## Content tree and information flow

This sections details the navigation system and all the content available in the application.

For all the details check file [WeatherApp.pdf](#)

### Main screen

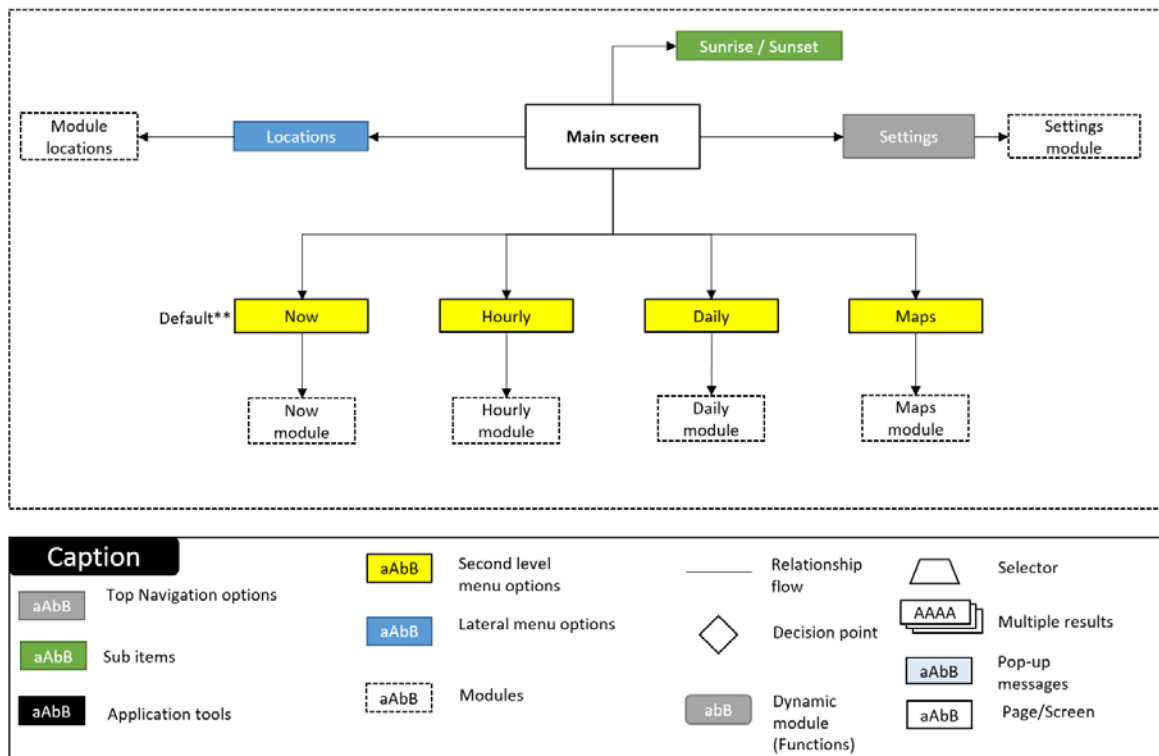


Figure 14: Main screen tree



# Prototyping

## Sketches

This shows sketches of the application

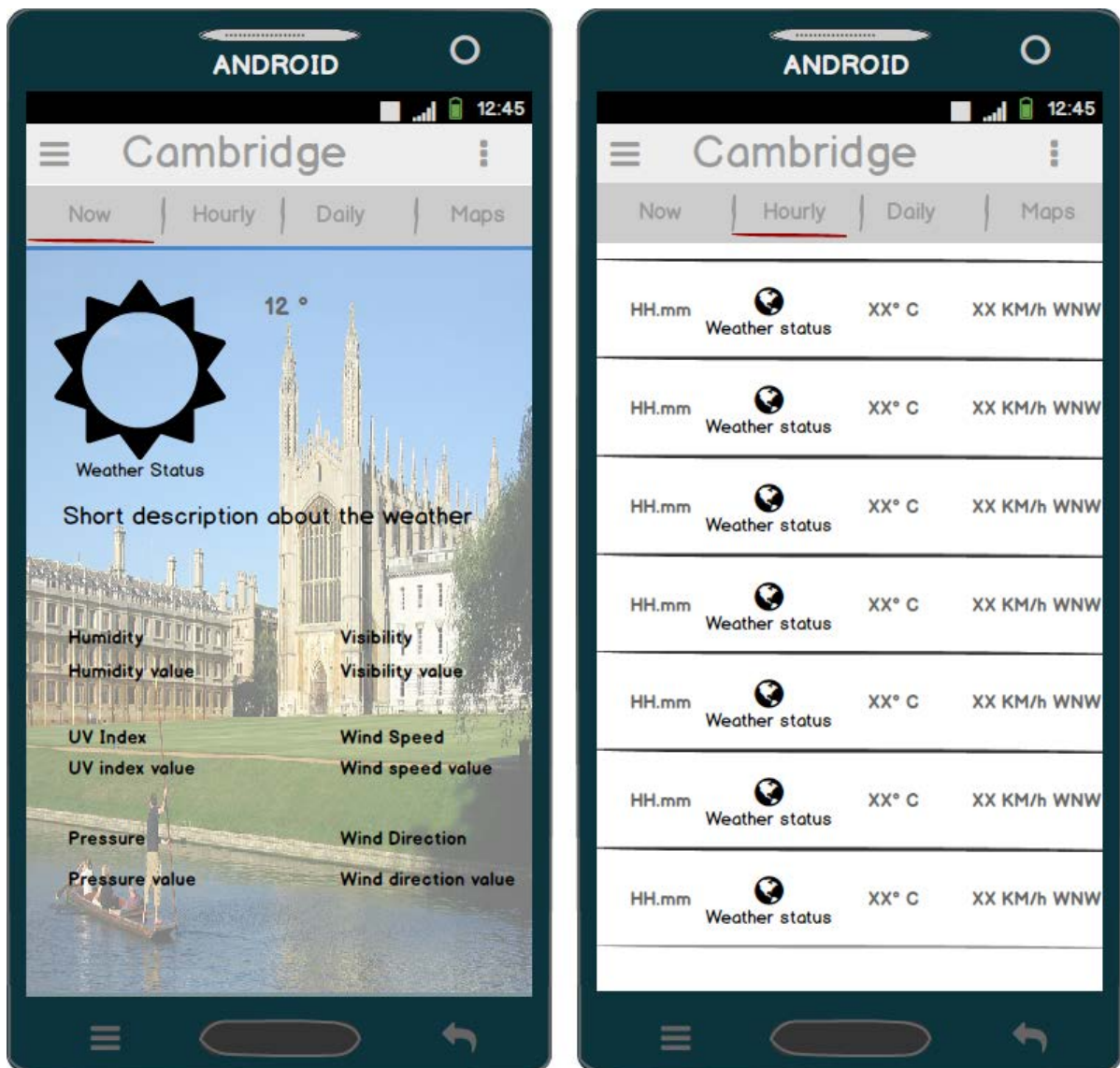


Figure 15: Sketches

For all the remaining Sketches check file [WeatherApp Sketchs.png](#)

## Wireframes

This shows wireframes of the application



Figure 16: Wireframes

For all the remaining Wireframes check file [WeatherApp Wireframes.png](#)

## Usability and accessibility evaluation

Once the sketches, wireframes and prototype of the application have been developed, several assessments are needed to be carried out to verify if the current design meets the requirements that were initially defined. In order to achieve the verification of the application, the following tests have been selected: Heuristic Evaluation, Cognitive Walkthrough and User Testing.

At this stage we are still in the early stages of the iterative process of designing the application, however the high quality mock-ups of the application and the wireframes are available.

The heuristic evaluation will be the first of the test to be conducted and it will be conducted by an expert, which will be held on the wireframes. This first evaluation will permit the detection and help avoid possible errors and will allow progress with the development without them.

Once the heuristic evaluation has been completed, the interactive prototype will be modified accordingly to the results found and the remaining tests will be conducted, this interactive prototype looks and behaves as the final application.

The second test that will be conducted will be the Cognitive Walkthrough, however this test will be conducted on the interactive prototype and it will be fulfilled by users and an evaluator, so it will be possible to obtain more relevant information as the test will be completed with users similar to the sample personas described previously, the target users.

The main goal of this evaluation is to verify the main functionalities and tasks are being performed rightly and the user experience is satisfactory. Explicitly verify that the different processes can be completed without any error and to ensure that all the processes and content distribution is following the users' mental model.

The last assessment, is only conducted by users, where they are faced with different hypothetical contexts, on which they have to complete a set of tasks, that they could happen in real life.

Once the users have finish the test, they will be presented with a questionnaire, aiming to get extra information about the usability, user experience and ease of use of the application.

## Heuristic evaluation

This is one of the most used techniques in usability evaluation, it was developed by Jakob Nielsen and Rolf Molich on 1990.

The main characteristic of this evaluation, is that it needs to be performed by experts' evaluators, assessing a previously defined set of heuristic principles.

Being a technique that evaluates the usability of an application, it's main objective is to measure the quality of the user interface for the users taking into consideration the ease of use, and the learning curve for a set of specific users in a concrete situation or context.

This method compared to others offers the following characteristics:

- Usually each assessment of the interface requires about one or two hours.
- It is one of the cheaper methods of assessments, as it requires fewer resources and planning time as it can be performed in earlier stages of development.
- It is recommended to carry out the test by a group of usability evaluators, in order to obtain detailed and verified information.
- All heuristic evaluations must be performed under the same conditions and environment, to ensure that the results cannot be altered by external factors.
- It is mandatory to perform the test in the most objective manner possible, to avoid spoiled results by the evaluators, their knowledge or ideals.
- Once the evaluation has been concluded, all the errors need to be identified, sorted and prioritised and afterwards all the required modifications for the interface have to be defined.

The heuristics principles are broad rules for support making while designing an application it is also used to criticise a previously design application in order to improve it and make it more usable.

Nowadays most of the heuristics are still based on the definition by Nielsen and Molich, however, quite often it is necessary to modify and extend this guidelines or principles in order to be able to evaluate the application that is being assessed, to ease the detection errors.

The heuristics principles defined by Nielsen and Rolf are:

- 1. Visibility of the system status:** the application should always keep the users informed about what is happening, providing them with appropriate feedback within reasonable time.
- 2. Match between system and the real world:** the application should speak the users' language, such as: familiar concepts and common language; rather than system commands. Information should appear in a natural and logical order.
- 3. User control and freedom:** sometimes users choose system functions by mistake, so a clear "emergency exit" should be offered to avoid an unwanted situation. The application should allow undo and redo.
- 4. Consistency and standards:** users should not have to be concerned about different words or situation meaning the same thing. The application has to follow the conventions of the platform.
- 5. Error prevention:** It is recommended to avoid error messages by designing an application that prevents conditions that could lead to error situation by providing confirmation of the action about to be committed.
- 6. Recognition rather than recall:** users should not have to remember information and instructions of use of the application should be easily accessible. All objects, actions and options should be visible.
- 7. Flexibility and efficiency of use:** the application has to be adaptable to the user experience, providing accelerators for the experts' users in order to cater all kind of users.
- 8. Aesthetic and minimalist design:** Extra information on dialogues is not recommended as it diminishes the visibility of useful information.
- 9. Help users recognize, diagnose and recover form errors:** In case an error message is needed, they should be expressed in plain language (no code), suggesting a clear solution for the users.
- 10. Help and documentation:** the application should be usable without any kind of documentation, however, if required it should be easily accessible and concrete to the user's task.
- 11. Skills:** the application should improve, support or enhance the user's skills, knowledge or expertise and not replace them.

**12. Pleasurable and respectful interaction with the user:** user's interactions with the system should enhance their quality of life. Users should be treated with respect. The aesthetic design should be pleasing and functional.

**13. Privacy:** the application should help the user to protect personal or private information.

## Planning the heuristic evaluation

In order to perform a useful test, the following steps should be considered:

- The number of reviewers, should be between 3 and 5, as it has been proven that this ratio offers the cost/benefit results, being able to detect almost 70% of the errors. Adding more evaluators increase the number of errors found, however the ratio of cost/benefit gets greatly reduced.

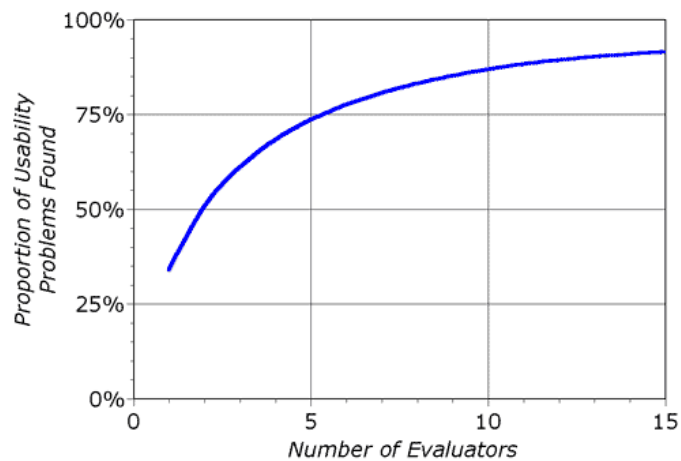


Figure 17: Correlation between the number of errors found and the number of evaluators conducting the test.

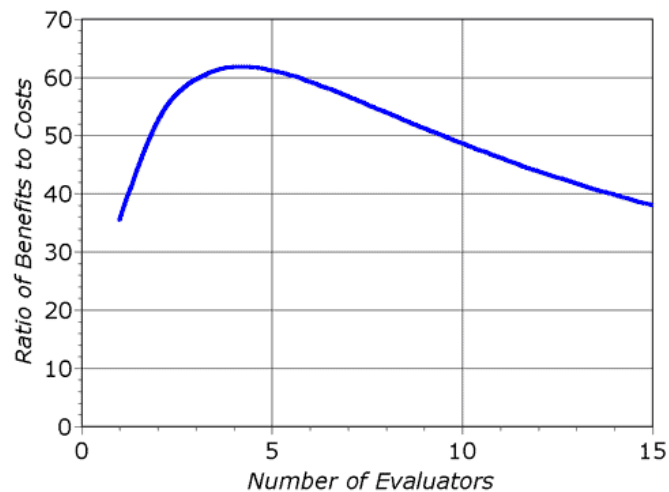


Figure 18: Correlation of ratio of benefit and cost depending on the evaluators employed.

- For the evaluation of this application, only one evaluator could complete the test, there was no possibility to make other reviewers take part.
- Selected heuristic principles: it is crucial to select heuristic principles suited to the analysis of the application.
- Selected evaluation questions: to be able to analyse the interface with a high level of detail.
- Use a template: provide a black space for the users to be able to note their comments related to the heuristic principles and questions.

### **Conducting the heuristic evaluation**

It is recommended that the evaluators perform the evaluation individually to avoid influencing each other.

Before starting the test, it is recommended that the evaluators examine the interface previously so they are familiarized with the interface.

Whenever they encounter a problem, evaluator must define the errors and sort them according to severity. In order to smooth this task, Jakob Nielsen formulated three questions:

1. The frequency of the error, is it common or rare?
2. When the errors occur, is it easy or difficult to overcome for the users?
3. The persistence of the problem, does it appear repeatedly bothering the users, or is it a one-time problem that they can overcome once known?

Similarly, he also developed a scale to rank the severity of the problem:

0. I don't agree that this is a usability problem at all.
1. Cosmetic problem only: need not be fixed unless extra time is available on project.
2. Minor usability problem: fixing this should be given low priority.
3. Major usability problem: important to fix, so should be given high priority.
4. Usability catastrophe: imperative to fix this before product can be released

## Analysing the results and reporting

Once all the evaluations have been conducted, each of the evaluators shall prepare an individual report, describing the problems found, how often it happens, the gravity of them and possible corrections or solutions to fix the errors.

When all reports are completed, the results shall be compared and aggregated to create a final report with the prioritised errors and solutions, after being discussed and agreed on.

The final report should comply with the following guidelines:

- It should be clear and direct.
- Each problem must be treated individually and must be explained as a heuristic principle.
- Every problem must have a possible solution.
- Each problem should be graded according to the frequency of occurrence, it's severity and ease of correction.

The Heuristic Evaluation conducted can be found on the file [Heuristic\\_Evaluation.pdf](#)

The results of the Heuristic Evaluation can be consulted on the file:

[Heuristic\\_Evaluation\\_Results.pdf](#)

## Cognitive walkthrough

This evaluation method is used to evaluate the design of the application, especially the navigation system, it's ease of use, so the test is performed with several users that have similar profiles to end users.

The test can be performed simultaneously with multiple users, however in our situation the evaluation will be performed individually, so the users cannot be influenced by others.



All the tasks of the test are performed by the users under the supervision of the evaluator and the users must record all the steps taken to perform the task. After the evaluation has been completed users and evaluators discuss the tasks.

This method compared to others offers the following characteristics:

1. This evaluation is considered a low-cost method, as it can be implemented on low quality mock-ups, as it only evaluates the navigation system of the application, not the user interface.
2. This technique demonstrates how easy it is to use the application without having to read the user's manual or use the help reference.
3. The tasks should be explained in a clear and simple language.
4. As it is focused on specific tasks, it can easily notice many problems, inconsistencies and improvements.
5. It can also be performed by evaluators, but its highly recommended to be carried out by real users, as the evaluators might be associated with the development of then application and might not have and objective opinion.

## User testing

This evaluation method is founded on observing and analysing the actions that a group of users performs in order to complete previously defined series of tasks.

It is performed by real users, in order to verify the ease of use of the application and the efficiency of it. The test validates the tasks defined in the requirements can be completed by the app, in a suitable and satisfactory manner. The main goal of this test is to find the remaining errors, if any remaining and discover the user experience with the application.

As the evaluation is being done, if any problem arises they have to be noted down to be corrected later on.

This method compared to others offers the following characteristics:

1. It can be performed in any stage of the development cycle, however it is advised to perform it after the heuristic evaluation as most of the usability errors will have been already detected and corrected.

2. It is recommended a minimum number of five with similar profiles to end users to complete the test.
3. The test must be performed individually in a comfortable environment where the user cannot be distracted by any external interference or noise.
4. The evaluator in this technique is a mere observer, it cannot help the user.
5. If an error occurs during the test, the evaluator must inform the user that it is a design problem and not theirs.
6. The evaluator or observer should be really aware of all the comments of the users and even more aware of their expressions and gestures as they can capture relevant information of the non-textual communication.

The Authorization Letter, Screening form, Cognitive Walkthrough, Tasks and Scenarios and Post-Questionnaire can be obtained from: [User Test User Copy.pdf](#)

The same version of the previous file used by the evaluator can be found on: [User Test Facilitator.pdf](#)

The answers from the users who realized the test; their responses, results and conclusions of the test can be seen on: [User Test Answers.pdf](#)

## Interactive Prototype

The interactive prototype offers all the functionalities that the final app will offer:

- All tabs are functional
  - Current weather
  - Hourly
    - Can check the details for 19:00, 20:00 and 21:00 (swipe between them)
  - Daily (scrolling up & down to see all the information)
    - Can check the details for 27/04/28/04/29/04 (swipe between them)
    - Day
    - Night
  - Maps
    - Can play the layer rain, other layers are not interactive.
- All options in options menu are functional change the view.
  - Notifications on lock screen enabled or disabled (swipe right for cleaning them/ tap to access them)
  - Units
  - Update interval can be interacted, however doesn't modify nothing.
  - Terms & conditions
  - Help & About
- Lock screen can be swiped to go back to main menu
- Can swap between cities
- Can see notifications
- Can simulate adding locations
- Can remove locations (Barcelona only)
- Can receive severe weather alerts.
- For changing tabs, swipe right and left is functional as well.
- The icon of the app allows to access main screen from any point.

The interactive prototype is available online at: [prototype](#)

The interactive prototype has been created using Justinmind Prototyper 7.2.2 Pro.

The final file of the prototype can be found at: [weatherAPP.vp](#)

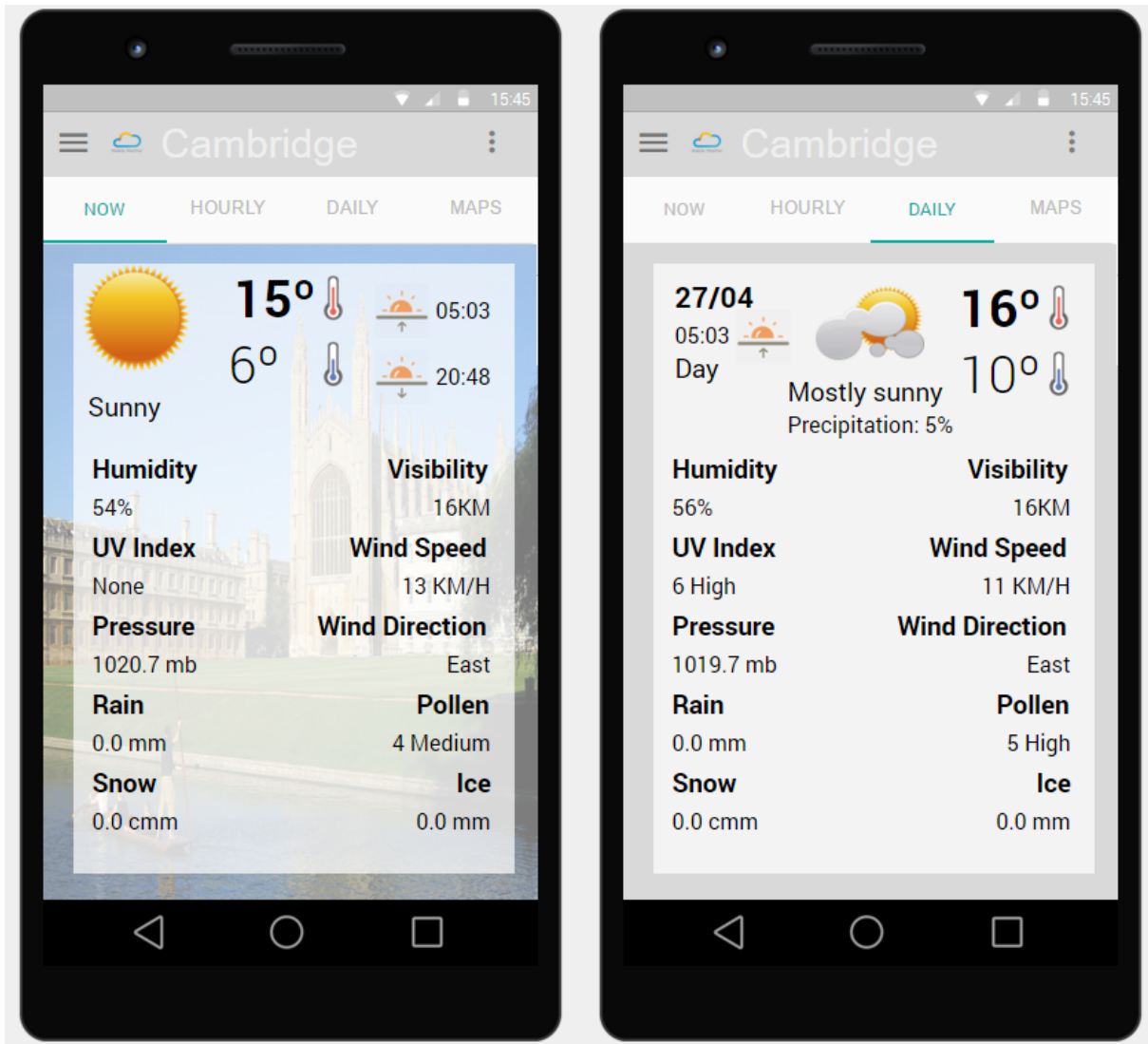


Figure 19 : Now and detailed daily (day) views

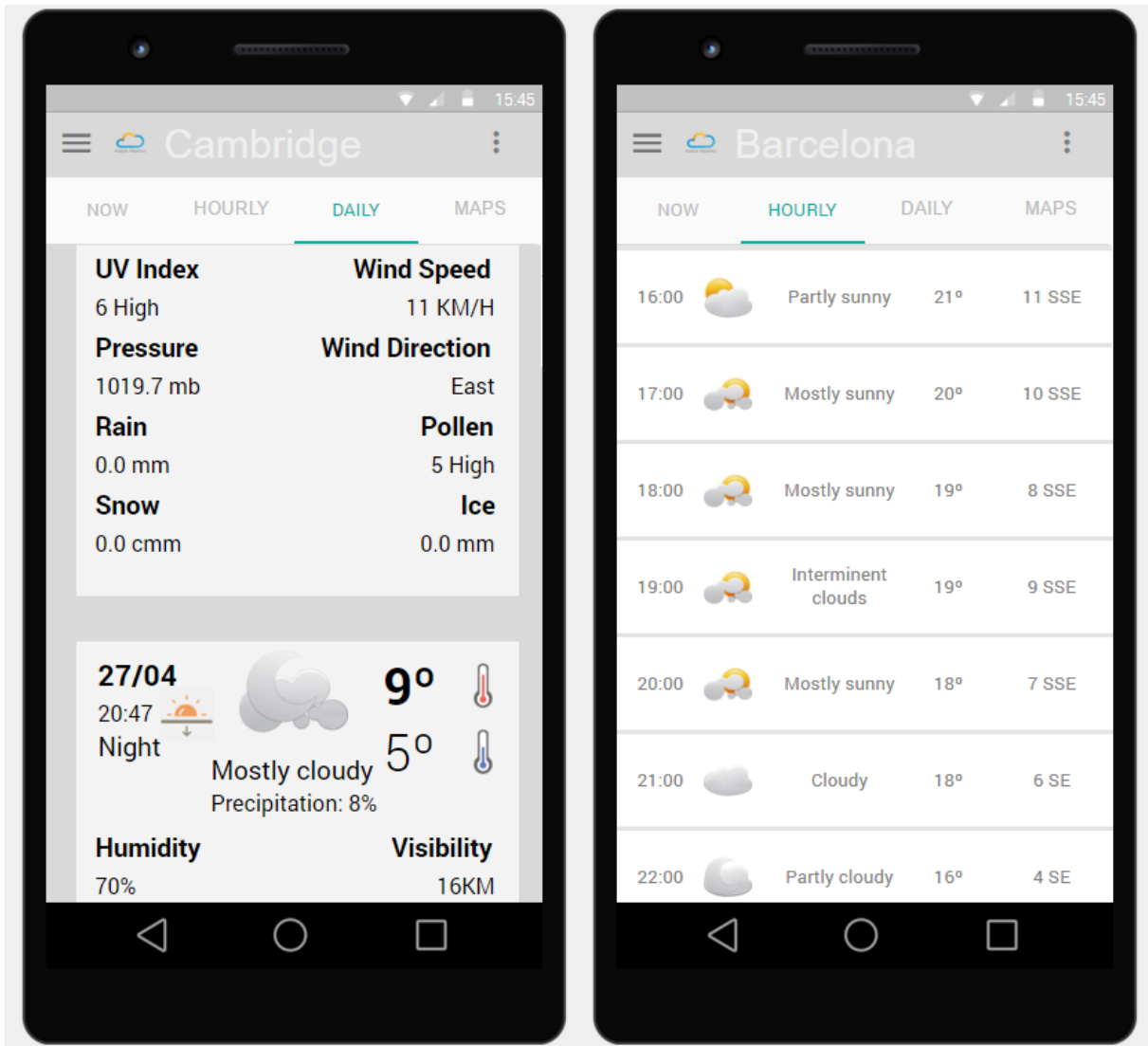


Figure 20: Detailed daily (Night) and hourly views

## Calendar deviations

During the majority of the project all the tasks were executed as previously planned, however, in the third partial hand-in was the most problematic stage, as I added and removed tasks in order to achieve the following stages of development, as I realised that important difficulties could appear with the coding of the application.

### **Added tasks:**

All these tasks were added to define better the application being build and define sample personas, which are our potential target.

- Technological requirements
- Maintenance needs
- Sample personas

### **Non finalized tasks:**

This task was being executed as planned, however it wasn't really finalised until the 4<sup>th</sup> partial hand-in, in order to have extra time to gain more knowledge for the coding stage.

- Sketches

### **Removed tasks:**

During the development stage this task was removed as I started to encounter major difficulties during the initial learning and coding using Android Studio, and I realised that I did not plan accordingly the time need to learnt and code, so I used the time designed for this task to gain more knowledge.

- Wireframes

Even with all these alterations the project followed its course, however on the 6<sup>th</sup> of June I started on a new job, and the final ending date of the project got postponed 4 extra days, so I could finish all the tasks as I had less time to work on the thesis.

Despite all these issues, for the final hand-in all the tasks that had been initially planned have been all completed.

The final calendar of the project can be seen on the file: [Final\\_workplanTFM.mpp](#)

## Future projection

Once the project has been completed, a very important milestone of the development of the application has been accomplished; obtaining a fully and detailed conceptual and design framework and as well a first demo of the application, however there is still a lot to do and achieve.

In order to proceed with the development of the application the following steps are needed:

- Finish all the application programming, conforming the current standards of accessibility and usability, while following the model reflected on the interactive prototype and wireframes obtained through the realization of this project.
- Enable an aggregation of multiple weather forecast data providers, in order to obtain data from different sources to get the average of them and provide the users with more precise forecast.
- By using multiple API weather forecast providers, the up-time of the application will increase, as even if any of them is offline, all the others can be used to obtain the forecast.
- Allow the users to report their current weather, to provide more accurate data and enable collaborative weather, permitting more precise weather as per post codes.
- Monetize the application, creating a free version supported by adds, and a full version purchased with a one-time payment.

Once the previous steps have been completed, a new stage of maintenance and upgrading of the application begins in order to keep offering a useful application and easy to use to the users. To achieve this objective, new functionalities must be added to the application, which comply with the requests formulated by the users and correcting bugs or errors that may appear during its use as more users start using the service provided by the app.

Beside all this, an iterative process of maintenance of the application will have to be implemented, in order to keep the application up-to-date and following the new standards and API implementations.



## Conclusions

After surveying, more than one hundred potential users it was found that there was a real interest of using a weather forecast application that would offer a lot of functionalities in an easy to use interface.

This dissertation aimed to analyse the feasibility of developing a weather forecast application, that enables the users to obtain precise and accurate weather data and notifications on weather change or severe weather conditions.

Observing the results attained, it can be stated that the initial objectives have been achieved and a fully detailed conceptualization and design of the weather application has been modelled.

As well, a complete interactive prototype and the first alpha release of the application has been created, establishing all the required guidelines for carrying on with the development.

After finishing the thesis, it has been demonstrated; as stated in the UCD methodology, that special attention should be taken when designing any kind of application, as it must be adapted to the real needs of the users, since they are the ones that will decide if the application has been a success or a failure. Which is why during the design of any project users must be taken into consideration; in order to improve the product quality and to reduce the future maintenance.

Throughout the dissertation, ease of use, efficiency, simplicity and clarity were prioritised in order to obtain an application that could be used by the majority of users, regardless of their knowledge or skills with new technologies.

In brief, what I wanted to achieve is an interface easily usable and as accessible as possible, in order to be able to offer the application to a large number of users.

Another relevant key factor for any project that should be noted is the planning of the scope. Even if some of the initial tasks require extra time to be completed, then all the planning could get delayed, affecting the overall progress.

In my case, during execution of the initial stage I found that I needed to add extra tasks in order to fully develop the application, otherwise some of its requirements and functionalities would not have been properly defined and tested.

During the 3<sup>rd</sup> partial hand-in one of the tasks that was planned in the original calendar wasn't completed, in order to offer an alpha of the application for the 4<sup>th</sup> partial hand-in, so I reassigned the time dedicated to the wireframes to the android section. However, this deviation was rectified for the following hand-in.

To summarise, it's important to create a defined and accurate planning for all the tasks required for the complete execution of the project, as the success of the project greatly depends on completing the tasks on time and following the calendar in the most accurate manner.

Personally, after completing the dissertation I can say that I have applied general knowledge acquired during my Undergraduate degree in Multimedia and more specific knowledge learned in the Master's, all of which is based on Interface Design, Usability and accessibility, Graphic Design, User Behaviour and user testing.

It has also allowed me to deepen my knowledge in areas related to design and prototyping. Permitting me to understand all the difficulties that may appear while building an application from the ground up.

Despite having set a quite difficult personal goal for myself, which was the coding of the application; as it was a new field to me and I had a rather limited time available, I managed to achieve a working alpha version of the application, that allows the users to check the weather data; it is still far away from all the functionalities available on the interactive prototype, however, it can be expanded as it is programmed in a modular manner, so new functionalities and modules can be implemented.

All the learning, research and implementation process during the extent of the Master's Thesis has been quite stressful, possibly due to the ambitious goal I established, especially considering the amount of time available and my initial skills, however after completing the thesis I can say that it has been satisfying, enriching and rewarding. As many of the concepts were new to me and I have broadened my knowledge.

It would be interesting to carry on developing the application, in order to have a full application published on the Google Play Market and have it available for the users, but to accomplish that I still need to carry on leaning on how to code Java for Android.

## References

- [1]. Connor, S. (2016). *Why severe weather events are the new normal*. [online] The Independent. Available at: <http://www.independent.co.uk/environment/uk-weather-why-the-recent-devastating-floods-will-become-the-new-normal-a6793291.html> [Accessed 7 Mar. 2016].
- [2]. Cookson, C. (2016). Climate change strongly linked to UK flooding - FT.com. [online] Financial Times. Available at: <http://www.ft.com/cms/s/0/831d04d4-b5ee-11e5-b147-e5e5bba42e51.html#axzz42odcKUJB> [Accessed 7 Mar. 2016].
- [3]. Deloitte United Kingdom. (2016). Deloitte Mobile UK 2015. [online] Available at: <http://www.deloitte.co.uk/mobileuk/> [Accessed 8 Mar. 2016].
- [4]. Garreta Domingo, M. and Mor Pera, E. (2011). Disseny centrat en l'usuari. 1st ed. [eBook] Barcelona: Universitat Oberta de Catalunya, pp.24-50. Available at: [http://openaccess.uoc.edu/webapps/o2/bitstream/10609/9743/4/PID\\_00158922-3.pdf](http://openaccess.uoc.edu/webapps/o2/bitstream/10609/9743/4/PID_00158922-3.pdf) [Accessed 9 Mar. 2016].
- [5]. Gov.uk. (2014). Climate change explained - Detailed guidance - GOV.UK. [online] Available at: <https://www.gov.uk/guidance/climate-change-explained> [Accessed 7 Mar. 2016].
- [6]. Media.ofcom.org.uk. (2016). Facts & figures. [online] Available at: <http://media.ofcom.org.uk/facts/> [Accessed 8 Mar. 2016].
- [7]. Media.ofcom.org.uk. (2015). The UK is now a smartphone society. [online] Available at: <http://media.ofcom.org.uk/news/2015/cmr-uk-2015/> [Accessed 8 Mar. 2016].
- [8]. Mobile Consumer 2015: The UK cut Game of phones. (2016). 1st ed. [eBook] Deloitte United Kingdom. Available at: <http://www.deloitte.co.uk/mobileuk/assets/pdf/Deloitte-Mobile-Consumer-2015.pdf> [Accessed 8 Mar. 2016].
- [9]. Nielsen, J. (1995). *10 Heuristics for User Interface Design: Article by Jakob Nielsen*. [online] Nngroup.com. Available at: <https://www.nngroup.com/articles/ten-usability-heuristics/> [Accessed 19 Mar. 2016].
- [10]. Parkinson, J. (2016). Five ways climate change could affect the UK - BBC News. [online] BBC News. Available at: <http://www.bbc.co.uk/news/magazine-35037983> [Accessed 7 Mar. 2016].

- [11]. Rutkin, A. and Ravilious, K. (2016). *Understanding climate change's role in the UK's recent floods*. [online] New Scientist. Available at:  
<https://www.newscientist.com/article/mg22930554-700-understanding-climate-changes-role-in-the-uks-recent-floods/> [Accessed 7 Mar. 2016].
- [12]. Statista. (2016). *Number of Google Play Store apps 2015 | Statistic*. [online] Available at: <http://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/> [Accessed 15 Mar. 2016].
- [13]. The Communications Market Report. (2015). 1st ed. [eBook] Ofcom. Available at:  
[http://stakeholders.ofcom.org.uk/binaries/research/cmr/cmr15/CMR\\_UK\\_2015.pdf](http://stakeholders.ofcom.org.uk/binaries/research/cmr/cmr15/CMR_UK_2015.pdf)  
[Accessed 8 Mar. 2016].
- [14]. Xtensio. (2016). *Xtensio | A Toolbox For Your Startup*. [online] Available at:  
<https://xtensio.com/> [Accessed 8 Apr. 2016].

## Bibliography

- [1]. Aeriweather.com. (2016). Sign Up - AeriWeather. [online] Available at: <http://www.aeriweather.com/signup/#api> [Accessed 17 Mar. 2016].
- [2]. Annyas.com. (2011). *Shimano logo*. [online] Available at: <http://annyas.com/htdocs/wp-content/uploads/2011/07/shimano-logo-1973.jpg> [Accessed 9 Apr. 2016].
- [3]. Anon, (2016). *Xerox Heuristic Checklist*. [online] Available at: <http://www1.zie.pg.gda.pl/~msik/materialy/xerox-he-chklst2a.pdf> [Accessed 6 May 2016].
- [4]. Apidev.accuweather.com. (2016). AccuWeather API - AccuWeather API. [online] Available at: <http://apidev.accuweather.com/developers/> [Accessed 17 Mar. 2016].
- [5]. Brandsoftheworld.com. (2016). *Garmin logo*. [online] Available at: <http://www.brandsoftheworld.com/sites/default/files/styles/logo-thumbnail/public/0019/0473/brand.gif?itok=eQ8QJ9es> [Accessed 9 Apr. 2016].
- [6]. Brandsoftheworld.com. (2016). *National Trust UK logo*. [online] Available at: <http://www.brandsoftheworld.com/sites/default/files/styles/logo-thumbnail/public/022013/nationaltrust2.png?itok=hbDr76vM> [Accessed 25 Apr. 2016].
- [7]. Canon.co.uk. (2016). *Canon logo*. [online] Available at: [http://www.canon.co.uk/Images/Canon\\_Logo\\_350\\_tcm14-959888.jpg](http://www.canon.co.uk/Images/Canon_Logo_350_tcm14-959888.jpg) [Accessed 8 Apr. 2016].
- [8]. Carreras, O. (2008). *Las 75 directrices de accesibilidad de Jakob Nielsen*. [online] Olgacarreras.blogspot.com.es. Available at: <http://olgacarreras.blogspot.com.es/2007/02/las-75-directrices-de-accesibilidad-de.html> [Accessed 10 May 2016].
- [9]. Cdn.coresites.factorymedia.com. (2016). *Nort Face logo*. [online] Available at: [http://cdn.coresites.factorymedia.com/whitelines\\_new/wp-content/uploads/2015/11/1024px-The\\_North\\_Face\\_logo.svg\\_.png](http://cdn.coresites.factorymedia.com/whitelines_new/wp-content/uploads/2015/11/1024px-The_North_Face_logo.svg_.png) [Accessed 9 Apr. 2016].
- [10]. Creative Commons. (2016). Home - Creative Commons. [online] Available at: <https://creativecommons.org> [Accessed 6 Mar. 2016].
- [11]. Creative Commons, (2016). [image] Available at: <http://mirrors.creativecommons.org/presskit/buttons/88x31/png/by-nc-sa.png> [Accessed 6 Mar. 2016].

- [12]. D1d985mza66mv7.cloudfront.net. (2016). *Adobe Lightroom logo*. [online] Available at: <https://d1d985mza66mv7.cloudfront.net/assets/lr-og-ec768f6dee9255b37fc8999e2e7ae4a7.png> [Accessed 25 Apr. 2016].
- [13]. Danielikim.com. (2013). *What is User Centered Design? | Daniel I Kim | UX Designer | Interactive | Visual Design | Developer*. [online] Available at: <http://danielikim.com/what-is-user-centered-design/> [Accessed 29 Mar. 2016].
- [14]. Developer.forecast.io. (2016). The Dark Sky Forecast API. [online] Available at: <https://developer.forecast.io/> [Accessed 17 Mar. 2016].
- [15]. Developer.worldweatheronline.com. (2016). Weather API | Local Weather API | World Weather Online. [online] Available at: <http://developer.worldweatheronline.com/api/> [Accessed 19 Mar. 2016].
- [16]. Developer.yahoo.com. (2016). Weather - Yahoo Developer Network. [online] Available at: <https://developer.yahoo.com/weather/> [Accessed 18 Mar. 2016].
- [17]. Dogtiev, A. (2015). [online] *Businessofapps.com*. Available at: <http://www.businessofapps.com/app-usage-statistics-2015/> [Accessed 15 Mar. 2016].
- [18]. FindThatLogo.com. (2011). *Camel Logo*. [online] Available at: <http://www.findthatlogo.com/camel-logo/> [Accessed 25 Apr. 2016].
- [19]. GO-Globe. (2015). Mobile Apps Usage – Statistics And Trends. [online] Available at: <http://www.go-globe.com/blog/mobile-apps-usage/> [Accessed 15 Mar. 2016].
- [20]. Hassan Montero, Y. (2002). Introducción a la Usabilidad. *No Solo Usabilidad*, [online] (1). Available at: [http://www.nosolousabilidad.com/articulos/introduccion\\_usabilidad.htm](http://www.nosolousabilidad.com/articulos/introduccion_usabilidad.htm) [Accessed 11 May 2016].
- [21]. Hassan Montero, Y. and Martín Fernández, F. (2003). Diseño Inclusivo: Marco Metodológico para el Desarrollo de Sitios Web Accesibles. *No Solo Usabilidad*, [online] (2). Available at: [http://www.nosolousabilidad.com/articulos/disenio\\_inclusivo.htm](http://www.nosolousabilidad.com/articulos/disenio_inclusivo.htm) [Accessed 11 May 2016].
- [22]. Heaton, A. (2014). Mobile App Statistics. [online] *Smart Insights*. Available at: <http://www.smartinsights.com/mobile-marketing/app-marketing/mobile-app-statistics/> [Accessed 15 Mar. 2016].
- [23]. Legacy.weather.weatherbug.com. (2016). Weather API for Programmers and Developers - WeatherBug.com. [online] Available at: <http://legacy.weather.weatherbug.com/pulseapi.html> [Accessed 18 Mar. 2016].

- [24]. Met Office. (2016). Weather. [online] Available at:  
<http://www.metoffice.gov.uk/public/weather> [Accessed 7 Mar. 2016].
- [25]. Nielsen, J. (1995). *10 Heuristics for User Interface Design: Article by Jakob Nielsen*. [online] Nngroup.com. Available at: <http://www.nngroup.com/articles/ten-usability-heuristics/> [Accessed 12 May 2016].
- [26]. Nielsen, J. (1995). *Heuristic Evaluation: How-To: Article by Jakob Nielsen*. [online] Nngroup.com. Available at: <http://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/> [Accessed 12 May 2016].
- [27]. Nielsen, J. (2000). *Why You Only Need to Test with 5 Users*. [online] Nngroup.com. Available at: <http://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/> [Accessed 13 May 2016].
- [28]. Openweathermap.org. (2016). Weather API. [online] Available at:  
<http://openweathermap.org/api> [Accessed 17 Mar. 2016].
- [29]. Pinterest. (2013). *Stuff we like*. [online] Available at:  
[https://www.pinterest.com/pin/494903446521132309/?from\\_navigate=true](https://www.pinterest.com/pin/494903446521132309/?from_navigate=true) [Accessed 30 Mar. 2016].
- [30]. Play.google.com. (2016). Accuweather. [online] Available at:  
<https://play.google.com/store/apps/details?id=com.accuweather.android> [Accessed 15 Mar. 2016].
- [31]. Play.google.com. (2016). BBC Weather. [online] Available at:  
<https://play.google.com/store/apps/details?id=bbc.mobile.weather> [Accessed 15 Mar. 2016].
- [32]. Play.google.com. (2016). Met Office Weather. [online] Available at:  
<https://play.google.com/store/apps/details?id=uk.gov.metoffice.weather.android> [Accessed 15 Mar. 2016].
- [33]. Play.google.com. (2016). The Weather Channel. [online] Available at:  
<https://play.google.com/store/apps/details?id=com.weather.Weather> [Accessed 15 Mar. 2016].
- [34]. Play.google.com. (2016). Yahoo Weather. [online] Available at:  
<https://play.google.com/store/apps/details?id=com.yahoo.mobile.client.android.weather> [Accessed 15 Mar. 2016].
- [35]. Reebok News Stream. (2014). *Reebok logo*. [online] Available at:  
<http://news.reebok.com/GLOBAL/reebok-signals-change-with-launch-of-new-brand-mark/s/ff399034-0aac-4263-99ed-6104ef4eda20> [Accessed 9 Apr. 2016].

- [36]. Sánchez, J. (2011). En busca del Diseño Centrado en el Usuario (DCU): definiciones, técnicas y una propuesta. *No Solo Usabilidad*, [online] (10). Available at: <http://www.nosolousabilidad.com/articulos/dcu.htm> [Accessed 3 May 2016].
- [37]. SeekLogo. (2016). *Specialized Logo*. [online] Available at: <http://seeklogo.com/specialized-logo-130169.html> [Accessed 8 Apr. 2016].
- [38]. Teoh, C. (2006). *User-centred design (UCD) - 6 methods - Webcredible UX blog*. [online] Webcredible.com. Available at: <http://www.webcredible.com/blog/user-centered-design-ucd-6-methods/> [Accessed 30 Mar. 2016].
- [39]. Travis, J. (2016). *Carla Diptych*. [online] Flickr - Photo Sharing!. Available at: <https://www.flickr.com/photos/jasontravis/15330876992/in/album-72157603258446753/> [Accessed 8 Apr. 2016].
- [40]. Travis, J. (2016). *Clark Howard Diptych*. [online] Flickr - Photo Sharing!. Available at: <https://www.flickr.com/photos/jasontravis/14651883172/in/album-72157603258446753/> [Accessed 8 Apr. 2016].
- [41]. Travis, J. (2016). *Clark Howard Diptych*. [online] Flickr - Photo Sharing!. Available at: <https://www.flickr.com/photos/jasontravis/14651883172/in/album-72157603258446753/> [Accessed 8 Apr. 2016].
- [42]. Upload.wikimedia.org. (2016). *Apple logo*. [online] Available at: [https://upload.wikimedia.org/wikipedia/commons/thumb/f/fa/Apple\\_logo\\_black.svg/240px-Apple\\_logo\\_black.svg.png](https://upload.wikimedia.org/wikipedia/commons/thumb/f/fa/Apple_logo_black.svg/240px-Apple_logo_black.svg.png) [Accessed 8 Apr. 2016].
- [43]. Usability.gov. (2016). *User-Centered Design Basics*. [online] Available at: <http://www.usability.gov/what-and-why/user-centered-design.html> [Accessed 29 Mar. 2016].
- [44]. Usabilitynet.org. (2006). *UsabilityNet: Overview of user centred design*. [online] Available at: [http://www.usabilitynet.org/management/b\\_overview.htm](http://www.usabilitynet.org/management/b_overview.htm) [Accessed 29 Mar. 2016].
- [45]. Wagner, J. (2014). Top 10 Weather APIs. [online] ProgrammableWeb. Available at: <http://www.programmableweb.com/news/top-10-weather-apis/analysis/2014/11/13> [Accessed 16 Mar. 2016].
- [46]. Wikimedia.org. (2016). *Giant logo*. [online] Available at: [https://upload.wikimedia.org/wikipedia/en/thumb/3/3c/Giant\\_logo.svg/1280px-Giant\\_logo.svg.png](https://upload.wikimedia.org/wikipedia/en/thumb/3/3c/Giant_logo.svg/1280px-Giant_logo.svg.png) [Accessed 8 Apr. 2016].
- [47]. Wikipedia. (2016). *User-centered design*. [online] Available at: [https://en.wikipedia.org/wiki/User-centered\\_design](https://en.wikipedia.org/wiki/User-centered_design) [Accessed 30 Mar. 2016].



- [48]. Wunderground.com. (2016). API | Weather Underground. [online] Available at: <https://www.wunderground.com/weather/api> [Accessed 19 Mar. 2016].
- [49]. Yáñez Gómez, R., Cascado Caballero, D. and Sevillano, J. (2014). Heuristic Evaluation on Mobile Interfaces: A New Checklist. *The Scientific World Journal*, [online] 2014, pp.1-19. Available at: <http://www.hindawi.com/journals/tswj/2014/434326/sup/> [Accessed 6 May 2016].