iOS application for detailed weather prevision in Continental Portugal

Author: Tiago Duarte[1], Adviser: Arlindo Oliveira[1], Support: Rosa Trancoso[2], Carlos Palma[2]
[1]Instituto Superior Técnico, Av. Rovisco Pais 1, 1049 Lisboa, Portugal,
tiago.joao@ist.utl.pt, *arlindo.oliveira@cg.ist.utl.pt*
[2]METEO-IST, Departamento de Engenharia Mecânica,
meteo.ist@gmail.com

Keywords: Mobile applications, Apple Inc., iPhone, iOS, Software Development, Software Engineering, Weather Forecasting, Web Services, Design

Abstract: This work is the result of a need to move the meteorological forecast system developed by METEO-IST to an iOS application. METEO-IST is a weather computational server owned by IST that calculates with great accuracy the different weather conditions (rain, wind, humidity, etc.) anywhere within the Portuguese continental territory. The predictions are calculated frequently (at every 15 minute intervals) and exhibit a great precision, distinguishing it from other global meteorological systems that are currently available. Currently the system makes the forecasts available through the group's website. At the request of several users, the objective was to create a native iOS application for the iPhone that provides these forecasts by taking advantage of the device's capabilities.

1 INTRODUCTION

This project report presents the design and development of an iOS application called 'Weather IST'.

The app is primarily designed for people who usually get their weather forecasting information by accessing the METEO-IST website, and its development is divided in two parts. The client side consists of the iOS device whose programming is done through Xcode using objective-C (a variant of C and C++) and iOS SDK. The other part is the web services that provide the weather forecasting data to the mobile device.

Mobile communications are so integrated into our lives that many people already feel uncomfortable without a mobile phone. The time when mobile phones served only to make calls and send text messages is gone and the most important reason was the arrival of the smartphones.

What are these smartphones and how do they contribute to a higher user satisfaction compared to previous devices?

A smartphone is a designation given to a mobile device with greater computing power (and connectivity). Close to the end of the twentieth century the first smartphones combined the functionalities of a PDA with those of a mobile phone.

Despite the growing evolution in hardware, the operating systems running on the phones were, comparatively, lagging behind in evolution and did not allow much customization by the end users even if they were already demanding more software choice and greater freedom of customization. Companies started to launch more applications with their devices such as games and utility applications (i.e. calculator, alarm, etc.). Nokia, for example, was made famous for putting the Snake game on their first devices. These early devices changed people's views of mobile phones and together with a decrease in prices and an increase in battery autonomy and network coverage led to a rise in sales.

1.1 Reinvention of the smartphone

In January 2007, Steve Jobs (the co-founder and CEO of Apple at the time) unveiled what would be a major revolution in the mobile industry with a device called the iPhone. It revolutionized so many components (both in hardware, software and design) compared to previous phones that it sold 6.1 million units in the first 15 months after its release. In 2008, only one year after, ‘Apple’ was already one of the five largest mobile phone manufacturers by revenue.

The single major change introduced, when compared to older phones, was the fact that the iPhone did not have a physical keyboard. Instead, it provided the keyboard on the screen when it was needed (i.e. focus on a textbox) and its buttons were just the right size to use our fingers as the pointer. The second major difference was its 3.5 inch LCD touch screen, shown in Figure 1, which was able to render an image at 480 × 320 pixels with 163 PPI’s[1]. It also had a 2 Megapixel’s camera, and a GPS, among other hardware features.

Figure 1: iPhone’s 1st generation

The user interface was a touch-based control, which was one of a kind. It gave the user the freedom to use their fingers to interact with the device. It was developed in such a way that different touches or gestures on the screen were recognized and each gesture had a specific functionality, making it a multi gesture touch based device.

In March 2008, Apple introduced the iPhone SDK[2] for developers to build applications for the iPhone as well as other Apple devices like the iPod Touch[3]. It provided the developers with the same set of development tools and Application Programming Interfaces (APIs) that Apple uses for building native applications on the iPhone and iPod Touch. Within a week Apple had 100,000 downloads of the SDK [2]. The SDK is available through Apple’s ‘iOS Standard Developer Program’ which is €80/year or the "iOS Enterprise Developer Program" (iDEP). The iOS Enterprise Developer Program is for apps which are not meant to be released and are only for institutional use. These do not require a membership fee but need to be approved by Apple.

1.2 App Store

The Apple App Store is a digital application distribution platform for iOS, developed and maintained by Apple. The service allows users to browse and download applications that were developed with Apple's iOS SDK. The apps can be downloaded directly to an iOS device or onto a personal computer via iTunes[4]. Both the App Store and the iPhone SDK were released simultaneously as they are meant to complement each other. The App Store was made available to iOS first version through an update on iTunes and was made available natively on the second version of the operating system (released that same year) and all later versions.

The next section describes the related work with a general overview of why a mobile app was needed for METEO-IST and a survey of the most highly rated, and used, weather applications. In section 3 a general overview of the work done at METEO-IST is presented.

Section 4 will give a general overview on the development bundle of iOS applications.

Section 5 will describe the application development cycles. Section 6 will give some

[1] Pixels Per Inch

[2] Software Development Kit

[3] A pocket computer designed and marketed by Apple Inc. that can be used as a music and video player.

[4] A media player and media library application developed by Apple Inc.

insights on the application analytics and in section 7 the conclusions are presented.

2. RELATED WORK

A METEO-IST user gets its weather information by visiting the website. Although it provides accurate information about weather forecasting it is not as fast and easy to use (usability wise) as it could be.

The information is scattered throughout the website and it requires many clicks before a user can get all the information he wants.

The application 'Weather IST' was designed to provide weather information to users in a fast (efficient) and easy way. It also takes advantage of iPhone features like local storage, for persistent application settings, or GPS to determine the user's location and give the respective weather information for it.

2.1 Weather Applications

There are currently more than 400 weather applications on the App Store, roughly half of them paid and the other half free. In order to understand what makes users choose some applications over the others, I decided to study the features and design of the most downloaded and rated weather applications on the App Store.

The most highly rated paid application on the App Store is called **Weather Live**. It has almost 21,000 reviews with an average rating of 4.5 out of 5.

As you open the app for the first time you notice the thoughtful design and nice animations that help illustrate the weather conditions with the assistance of real images as background.

The application allows users to see the weather forecast by hour (up to 24 hours) and day (up to 7 days). The available metrics include temperature, rain, humidity, wind velocity and direction, atmospheric pressure and visibility, and the app allows the user to change the units of some of these (metric versus imperial system).

Weather Genie has almost 400 reviews and an average rating of 4 out of 5 stars making it the second most rated paid application. Just like Weather Live, the app tries to catch the user's attention by using HD images that illustrate the weather conditions with animations that represent for instance rain and snow. The application shows hourly predictions, up to 24 hours, and daily previsions, up to 7 days.

Weather HD is the third most highly rated paid application of the App Store with an average rating of 4 out of 5 stars. One of the key features of this app is that it contains animated radar maps with the weather data being supplied by the National Oceanic and Atmospheric Administration (NOAA) which makes it very accurate. Another key feature is the ability to share the weather previsions on Facebook and see other people's shares.

Jumping on to free applications, **Thermometer!** is the most highly rated application on the App Store. It has almost 300 reviews with an average rating of 4 stars out of 5. The application is based on a minimalist design and only shows predictions for the current day which include temperature, rain, atmospheric pressure and wind velocity and direction.

AccuWeather has approximately 83,000 reviews with an average rating of 3.5 out of 5 stars. Unlike the other applications, it contains daily forecasts up to 15 days and the usual 24 hours. One of the key features of this application is that it can be integrated with the iPhone calendar, which allows a user to check the weather predictions without having to open the app. In terms of the design, the app is well planned out although it revolves around a simplistic design more focused on the information it provides rather than its image, as shown in Figure 2.

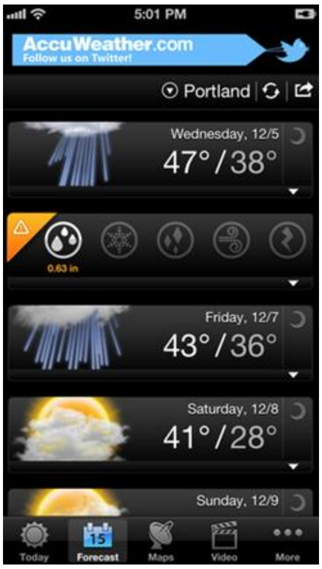


Figure 2: AccuWeather application

The Weather Channel has about 360,000 reviews and an average rating of 3.5 stars out of 5 making it the third most rated free weather application on the App Store. The application contains daily forecasts up to 10 days and 24 hours.

Like AccuWeather the application design is very minimalist and focuses mainly on the carefully placed display of the metrics. However, it contains interesting features like dynamic backgrounds that change with the weather conditions and it allows users to choose their own background from their personal images. The application also contains a notification system that warns about harsh rain forecasts, hurricanes or even the amount of pollen in the air.

2.2 Weather IST's target

Although existing applications already contain the most common metrics in weather forecasting (i.e. temperature, rain, etc.) they do not allow users to see them discriminated for the current and following days. This means users cannot have a perspective of the metric's evolution for the next few hours or days. One of the 'Weather IST' key features is that it will make the weather metrics independent which will allow the user to define and isolate the information that most matters to him. There are no optimal weather conditions for all users as each one has its own needs. For example, a firefighter on duty would search for a period of the day (or next days) with less wind velocity and more humidity, while a surfer would be interested in the sea wave's size and not really care about humidity or wind velocity.

Another advantage of our application is the precision and reliability of the forecasts that the METEO-IST servers produce for continental Portugal. The formulas behind the estimation of the forecasts are backed by years of scientific studies and are constantly being validated and improved. The next Chapter will focus on explaining the work done at METEO-IST as well as the developments made to support the app functionalities.

Although the app has many strengths, the goal is not to try and surpass the apps described above. The aim was to create an easy to use application with precise forecasts which the current METEO-IST website users can have as an alternative source of weather information. Section 5 will describe user satisfaction.

3. METEO-IST

METEO-IST is a numerical weather prediction group founded at IST. It is part of the mechanical engineering department in the energy and environment section. Although this thesis does not involve forecasts calculations itself, but only the results, I describe in this section the evolution of numerical weather prediction in general and at IST.

3.1 Numerical Weather Prediction

Numerical weather prediction (or NWP) uses mathematical models of the atmosphere and oceans to predict the weather based on current weather conditions.

The first attempt at this system was made in the 1920's, but due to the lack of computer computational power the first realistic results were only produced in the 1950's.

These mathematical models can be used to generate either short-term weather forecasts or longer-term climate predictions with the latter being widely applied towards understanding and projecting climate change.

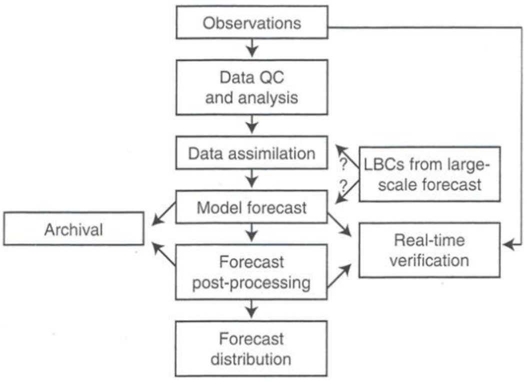


Figure 3: The various components of a NWP system

The NWP used at IST is based on two models. The MM5 (short for Fifth-Generation Penn State/NCAR Mesoscale Model) and the Weather Research and Forecasting model (WRF) are offered free of charge and are supported by the United States National Center for Environmental Prediction (NCAR). Both models are suitable for a wide range of applications with a scale ranging from a few meters up to thousands of kilometers. Initial and boundary conditions to both models are provided by the Global Forecast System (GFS), a global spectral weather prediction model running at the National Centers for Environmental Prediction (NCEP).

The NWP systems were first brought to IST by Prof. Delgado Domingos in 1995 using a cluster of 4 Intel PC's. He also developed the early scripts to run the required software for a viable and reliable operational model.

After exhaustive testing and configuration the model was first brought to public in 2001 where users could access the weather forecasts by visiting their website[5].

The website allowed users to pick any location on continental Portugal by providing a pair of coordinates (latitude and longitude) and it was the first system of its kind in Portugal to allow such a free use of a weather prevision system to get real time information of the most important metrics like temperature, rain, etc.

Initially, the model produced 72-hour forecasts with domains of 81, 27 and 9 kilometers, Figure 4, updated four times in the day (00Z, 06Z, 12Z and 18Z).

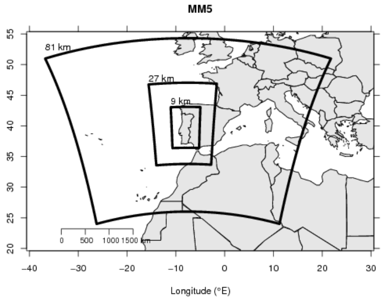


Figure 4: Weather forecast domains of the first MM5 model implemented in IST

In 2007, with the upgrade to GFS and the introduction of a new NWP model, the WRF, both models began using the new GFS, as shown in Figure 4. The outer coarse domain of MM5 was dropped and the WRF model, now sufficiently mature to be operationalized, was added. WRF was implemented with nested grids of 9 km for Portugal, and a 3 km in a square domain with approximately 200x200 km in extent centered in the city of Lisbon. Today, each model produces 72-hour forecasts updated four times in the day (00Z, 06Z, 12Z and 18Z) and 180-hours forecasts updated each day at 00Z.

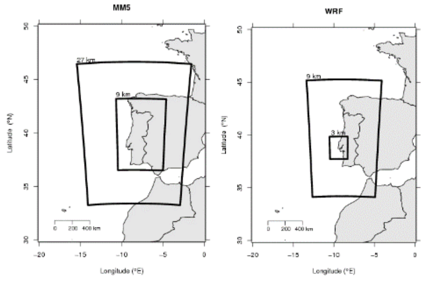


Figure 5: Domains in IST 3-day and 7-day forecast system for Portugal, since 2007

4. TECHNICAL DETAILS

This chapter provides a broad overview of the tools and technologies that are currently available for the development of applications in iOS.

4.1 iPhone Development Bundle

The first iPhone was released in June 2007 and since then it has evolved every year, both in software and hardware, with the release of new versions of its operating system and models.

Having an OS that works exclusively in a restricted number of devices makes it easier to assure performance and stability, something that Android, for example, is not able to do.

iOS is derived from OS X. The operating system used in Apple's desktop and laptop devices like the iMac and Mac Pro, respectively, and is similar to the UNIX operating system.

The home screen displays application icons and a dock at the bottom of the screen where users can pin their most frequently used apps (up to four). The home screen appears whenever the user unlocks the device or presses the "Home" button (a physical button on the bottom part of the device) while using another app, and it is the doorway to open any application in iOS.

Although iOS is derived from OS X, one does not need to know how to program for OS X in order to develop iOS applications. I would classify iOS applications in three different categories.

[5] http://meteo.ist.utl.pt

System Applications

System applications are apps made by Apple that can talk to the underlying hardware directly. For example, switching the Wi-Fi on and off or installing a new app on your device is done by this type of applications. “Settings” and “App Store” applications are examples of these.

Native Applications

Native applications are apps also made by Apple but which are developed using the same layers of architecture (API’s) as the ones the developers have access to. “Calculator” and “Notes” are examples of such applications.

Third Party Applications

Third party applications are apps done by any developer, other than Apple, and are limited in their use of some architecture layers. ‘Weather IST’ is a third party application.

4.1.1 iOS Architecture

iOS acts as an intermediary between the underlying hardware and the apps created for the iPhone. As stated above, apps do not talk to the underlying hardware directly. Instead, they communicate with it through an available set of system interfaces. These system interfaces are divided in four different layers, shown in Figure 6, that provide fundamental services and technologies in the lower levels and more sophisticated services in the higher levels.

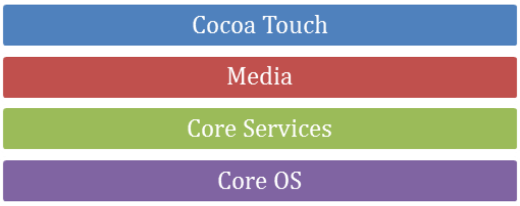


Figure 6: Layers of iOS [5]

Core OS

The Core OS layer contains the low-level features upon which most of the other technologies are built and is subject to change in every OS.

Core Services

The Core Services layer contains fundamental services for apps and it contains, for example, the Foundation framework which is used by every navigation controller in iOS applications.

Media

This layer contains the graphics, audio and video technologies developers use to implement multimedia in their apps.

Cocoa Touch

The Cocoa Touch contains frameworks that define the appearance of the app and also provides the basic infrastructure and support for technologies such as multitasking, touch-based input (i.e. tapping or zooming), push notifications, etc.

4.1.2 Development Tools

Xcode

Xcode is an IDE that contains software development tools (all developed by Apple) that support the development of applications for OS X and iOS.

Both the iOS SDK and Xcode come in a single package. Apple provides rich API documentation as a development guide and, for each API, they also provide code samples for reference. Most of the code in an iOS application is written in Objective-C.

Xcode handles most of the project details and takes the project from inception to deployment containing documentation for every API it provides. The IDE contains an interface builder which is a tool that allows developers to create their own interface by dragging elements to the view and customizing them. It supports C, C++, Objective C, and Objective C++, among others. Xcode provides tools to manage the entire development workflow, from creating the application, testing, optimizing, and submitting it to the App Store.

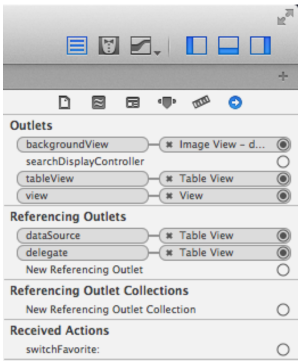
Interface Builder

The Interface Builder provides collections of user interface objects to the developer. These user interface objects contain items like text fields, data tables, sliders and buttons and can be found in the “Object Library” of Xcode.

An interface file is saved as a package that contains the interface objects and relationships between them (events, inherency, etc.). All of the objects used in the interface are archived (process also known as serialization) as an XML file or a property list file with a .nib extension.

Connections Panel

In order to associate Objective-C code with an interface element a developer needs to somehow connect both elements. The connections panel is a context-specific window that shows up only when there are outlets or actions associated with a specific object. To associate a code declared property with an interface object, the developer must declare the property as an "IBOutlet" which will let him then connect that property to the object like shown in Figure 7.



Outlets

backgroundView — ✖ Image View – d... ●

searchDisplayController ○

tableView — ✖ Table View ●

view — ✖ View ●

Referencing Outlets

dataSource — ✖ Table View ●

delegate — ✖ Table View ●

New Referencing Outlet ○

Referencing Outlet Collections

New Referencing Outlet Collection ○

Received Actions

switchFavorite: ○

Figure 7: The Interface Builder connections panel

iOS Simulator Application

The SDK provides an iOS simulator that can emulate the iPhone on the Macintosh. When a developer wants to deploy an application he has to choose the target where he wants to deploy it to. If there is an iOS device plugged into the computer, and it is properly provisioned Xcode automatically adds it to the target options list and lets you deploy it directly onto the device with debugging ability. iOS Simulator allows a developer to simulate several iOS devices and versions of its operating system. Each simulated software version is considered its own simulation environment, independent of others, with its own settings and files.

Performance Tools

As the device has limited memory and computing power, it is important to test the application's performance so that it does not consume the entire power of the device thus leaving all other applications pendent.

In a previous Xcode update, an application called 'Instruments' was introduced. It can run multiple performance testing tools simultaneously and view the results as a timeline-based graphical representation. It monitors CPU usage, memory leaks, garbage collection, disk reads and writes, thread activity, etc.

5. DEVELOPMENT

This section provides a general overview on the various phases of the software development lifecycle.

5.1 Version 1.0

The idea of a mobile application for METEO-IST had been first suggested, by users of the website that wanted a portable version of the forecast service.

5.1.1 Design choices

The METEO-IST's website interface is not very user friendly when used on a mobile device mainly, because it requires a lot of zooming and dragging just to set the right variables (i.e. location of the forecasts).

To start a requirement analysis, the first step was to access the site and navigate through it as a regular user, and after the survey, some of the initial features to be included in the application were:

- **Search Location:** A user would be able to search any location by typing the street or city name and get smart results that match it.
- **User's Location:** Taking advantage of the iPhone capabilities, with GPS capacity, a user would be able to get weather forecasts for his current location.
- **Favorite Places:** In order to make it faster for a user to check the forecasts for a usual place (i.e. his home or work location) a user would be able to save his favorite locations.
- **Chart Graph:** Keeping an existing feature in METEO-IST's website, a user would be able to see the evolution of a metric in a line chart graph.
- **Settings View:** A user would be able to change the temperature and wind speed units, and store these preferences on the iPhone.

5.1.2 Use Case Analysis

After the desired features of the software were defined, the Use Cases were wrote to better understand how the application should behave. It is designed in such a way that it is efficient and does not use a lot of processing or computing power of the mobile device, as all the complex operations are made on the METEO-IST's servers.

5.1.3 Application Architecture

The iOS application will behave as a client in the proposed architecture and will use METEO-IST's servers as the weather forecast provider. It will be necessary to define a new service interface that will support the application requirements, and also that conforms to the mobile paradigm (i.e. fast and small responses).

The new services will be available on the Internet, although the weather forecast data file, which is generated by the METEO-IST's servers, will only be available on the IST intranet meaning that the client application ('Weather IST') will only be accessing the interface that is available on a public machine inside the IST network. Because this machine is inside the network, it will then have connectivity and permission to read the data file and return the results. Figure 8 illustrates the architecture.

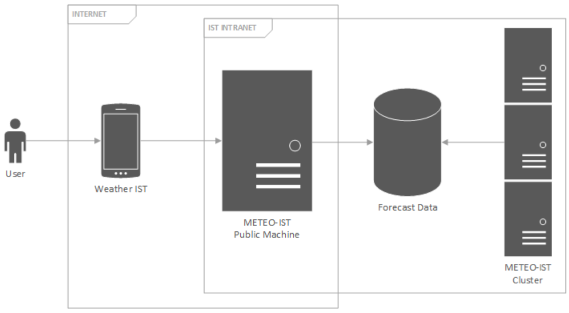


Figure 8: Top level architecture for 'Weather IST'

5.1.4 METEO-IST's Web Services

Web services were developed in METEO-IST in order to support the application requirements.

The services were developed in PHP which is a server-side scripting language designed for web development but is also used as a general-purpose programming language. The service's responses are all written in JSON, which is derived from the JavaScript scripting language and is used for representing simple data structures and associative arrays.

Before the requests are sent, the app gets the device current date and time (using native iOS API) and converts it to UTC[6]. This conversion ensures synchronization and precision in the forecasts. The app also gets the user's coordinates either by doing a search, or by getting his location using the GPS. The coordinates are then rounded to floating point's numbers with a precision of 4 decimal places and put in the request. These attributes are mandatory since as all the forecasts are time and location specific.

Another attribute required in some services is the metric, and, in order to identify it, between the two endpoints, we had to define an enumerator that would be the same in the app and in the web services.

Even though this protocol is required it does not limit the extensibility of the app. The code is made in such a way that if a new metric was implemented in the response, the application would not crash but simply ignore it as it would not recognize it. By just updating the enumerator on the app it could start parsing and displaying it.

5.1.5 Other Web Services

In order to support the location search in the app the Google Maps API was used, which is free with a usage limit of 100,000 requests every 24 hours.

Google Places provides a service called 'autocomplete' that given a string of text, returns location results that match the search.

The idea is to call the service every time the user writes something in the search text field so it gives the idea of autocomplete and refreshes the results for the new string.

This service is very interesting because it allows the search of other places besides streets or cities (i.e. a person could search for 'Oceanário Vasco da Gama' and have weather forecasts for that location without even knowing where it is).

[6] Coordinated Universal Time

However, this service does not return the coordinates for each place, so when a user taps a result, another service has to be used called ‘Place Details’ which receives a place id and returns more information about the place including the coordinates.

Figure 9 shows the implementation of this feature depicted using a sequence diagram.

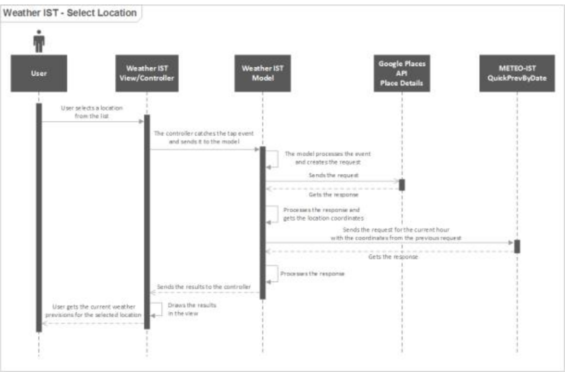

Weather IST - Select Location
User
Weather IST View/Controller
Weather IST Model
Google Places API Place Details
METEO-IST QuickPrevByDate
User selects a location from the list
The controller catches the tap event and sends it to the model
The model processes the event and creates the request
Sends the request
Gets the response
Processes the response and gets the location coordinates
Sends the request for the current hour with the coordinates from the previous request
Gets the response
Processes the response
Sends the results to the controller
User gets the current weather previsions for the selected location
Draws the results in the view

Figure 9: ‘Weather IST’ location selection – Sequence Diagram

5.1.6 Problems encountered

Coding in Objective-C

The learning curve for Objective-C was steeper than expected. Objective-C is an object-oriented programming language but on top of that, it adds Smalltalk-style messaging to the C programming language. Smalltalk is also a dynamically typed, object-oriented language. It is a reflective programming language that was created in 1970 for educational use with the aim to underpin the “new world” of computing. In summary, Objective-C is easy to read but hard to write. It is easy to learn how to create a method in Objective-C, but because of all the semantics, it takes a while for a developer to fully understand it.

METEO-IST’s web services

Another problem we faced while developing the app was that the METEO-IST’s services were taking too long to respond. This was because every time a service was called, it would access the forecast data file which had more than 1 Gigabyte, and get the information it required. Because that file contained more data then what the app needed (i.e. it had temperature in various layers of the atmosphere), a separate file was created with only the information the app required. The file became smaller than 300 Megabytes and the services response got faster.

5.1.7 Testing

In order to transfer the app to users, for testing, a physical connection with the device was required. When the app was finished, a meeting was scheduled with some prospective users in order to connect their devices to the Mac and install the app.

5.1.8 Release & User Satisfaction

The app was submitted to the App Store on April 10, 2013. It took 7 days for Apple to approve the application and release it on the App Store. It usually takes Apple about 2 or 3 hours to review an application (depending on the app features) so the remaining time was queuing for someone to review it, because of the hundreds of applications that get submitted every day. The app was released to the public on April 17.

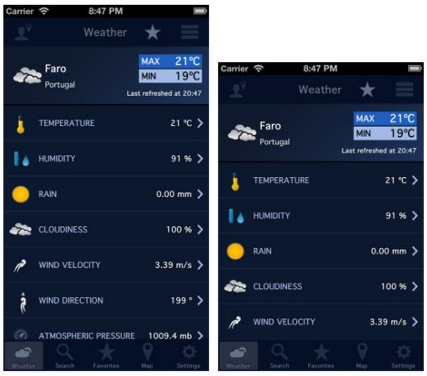
The application took about 5 months to develop, test and integrate with METEO-IST. We were eager to get results and see how the application was doing on the market. The first results were good: the app had about 500 downloads in the first two weeks and was able to maintain its rank position in the top 40 (in a total of 200 weather apps).

5.2 Version 1.0.1

Some months before the first version of ‘Weather IST’ was launched, Apple released its new iPhone 5. Throughout the years Apple had launch the device with the same screen size (3.5 inches), but with the iPhone 5 they decided to go bigger and increased the screen size to 4 inches. The screen only grew in height, not in width. This changed the resolution from 960 x 640 pixels to 1136 x 640 pixels. Developers now had 176 more pixels to draw on their app. This change came with the release of the iOS 6 and consequently brought a new SDK for developers with Xcode 4.

For apps that were not ready to take advantage of the iPhone 5 screens, iOS would put black bars on top and bottom of the screen to fill the remaining space displaying the app with the same height as it was initially developed. Our app was one of those cases as it was still compiled in the previous SDK and did not reflect these changes. I then recompiled the application and made some adjustments to the interface. It required some tweaking but overall it was fairly simple to recompile the app and take advantage of the 4-inch screen.

Figure 10 shows the height difference in ‘Weather IST’ on an iPhone 5 and an iPhone 4S.



Faro
Portugal
MAX 21°C
MIN 19°C
Last refreshed at 20:47

TEMPERATURE	21 °C
HUMIDITY	91 %
RAIN	0.00 mm
CLOUDINESS	100 %
WIND VELOCITY	3.39 m/s
WIND DIRECTION	199 °
ATMOSPHERIC PRESSURE	1009.4 mb

Faro
Portugal
MAX 21°C
MIN 19°C
Last refreshed at 20:47
TEMPERATURE 21 °C
HUMIDITY 91 %
RAIN 0.00 mm
CLOUDINESS 100 %
WIND VELOCITY 3.39 m/s

Figure 10: ‘Weather IST’ version 1.0 in iPhone 5 (left) and iPhone 4S (right)

5.3 Version 2.0.0

After the launch of the first version the final result was not satisfactory. While using it, the application felt complicated and the way the information was shown was not taking advantage of the powerful system supporting it (METEO-IST).

5.3.1 Requirement Analysis

The following requirements were created after reviewing the first version of the application.

- One view only for all the weather information. No more navigating through views, or opening graphs. The user would have all the information in just one view;

- Tab bar removed. To leave all the information in one screen we had to remove elements that only cause noise. Elements have to breathe in order for the design to work;

- The map view brings no value to the app, so it should be removed;

- A view just for the favorites is also an overhead. With this new design I wanted the user to access the favorites without having to push another view;

- The app has to become lighter. The first version of ‘Weather IST’ was too dark and plain. Users care much about design and they can easily switch to another application even if its forecasts are not as good. I chose to implement dynamic backgrounds that reflect the current weather;

- Weather icons are useful because they aggregate the time of day (with a sun or a moon), cloudiness and rain. In order to get a cleaner design the application will use these more often.

5.3.2 Use Case Analysis

Forecasts for the next hours

In the previous version a user could only check the forecast for the next few hours for the present day. For example, if he was checking the temperature forecasts at 22:00 he would only see the values for the current hour and for 23:00.

Removing a favorite place

The favorites view was removed from this version. So the way a user deletes a favorite is the same way he adds it, which is by clicking the top right corner star button.

5.3.3 METEO-IST’s Web Services

To support the new Use Case, described above (forecasts for the next hours), we developed a new service that given a date (with time), latitude, longitude and a metric returns the next 24 hours of forecast for that metric beginning at the given time.

5.3.4 Testing & Release

I performed the only testing work on this version as I did not distribute it to anyone including my supervisor. The reason for that is that I wanted to have this version finished in time to present it on this report as well as collect the results for user satisfaction. Skipping this step allowed me to jump directly to release. The app was submitted to the App Store in September 3, and was approved and released to the public on September 11.

6. APP ANALYTICS

The initial results for the first version of 'Weather IST' were good. The app had about 500 downloads in the first two weeks and was able to maintain its rank position in the top 40 (in a total of 200 weather apps). Figure 11 and Figure 12 show the evolution of downloads and rank (by week) of the app in the App Store in its lifetime.

The app also received some good reviews and was able to score a 4 star rating out of 5 from users.

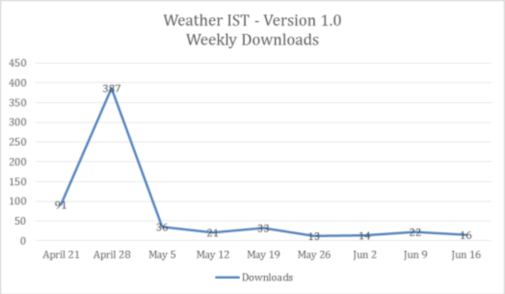


Figure 11: 'Weather IST' - Version 1.0 downloads

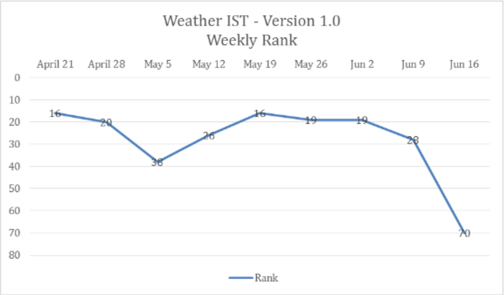


Figure 12: 'Weather IST' – Version 1.0 rank

Although the initial analytic results were good, Figure 11 clearly shows a stagnation in the number of weekly downloads with an average of 3 downloads per day. The results were coherent with my own analysis of the application. The final result of version 1.0 was not satisfactory and it felt complicated. Also the way it was displaying the information was not taking advantage of the powerful system supporting it (METEO-IST).

After the release of version 2.0.0 the number of downloads started to rise and the application rank got better. Figure 13 shows the number of downloads and updates per day for the first 3 weeks of version 2.0.0 lifetime.

The graph shows that version 2.0.0 had more than 300 updates in the first 3 weeks, which is 50 more than version 1.0.x[7] for the same period after release.

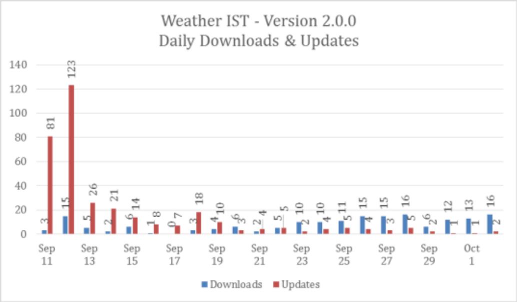


Figure 13: 'Weather IST' – Version 2.0.0 downloads/updates per day

On the first 2 days alone there were more than 200 updates, which shows that users were eager to upgrade the app. Also, the same chart shows an increase in downloads beginning on September 23, which coincided with the beginning of autumn and the first rainy days in Portugal. The third week of version 2.0.0 had more downloads than several weeks of version 1.0.x put together.

Also, the first major version of 'Weather IST' was only gaining 2 new users every day, whereas 2.0.0 version is gaining 8 new users every day. It is a 400% increase compared to its predecessor and the next chart (Figure 14) shows the impact of this change in months.

The ranking of version 2.0.0 has also been better than previous versions. Figure 15 shows stable values beginning on September 23 which coincide with the period of increased number of downloads shown in the chart in Figure 13.

[7] Merge of version 1.0.0 and 1.0.1 for analytic purpose.

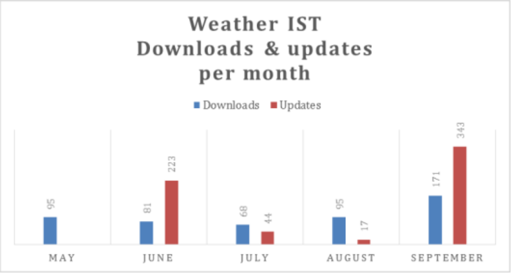


Figure 14: ‘Weather IST’ – Overall downloads and updates per month

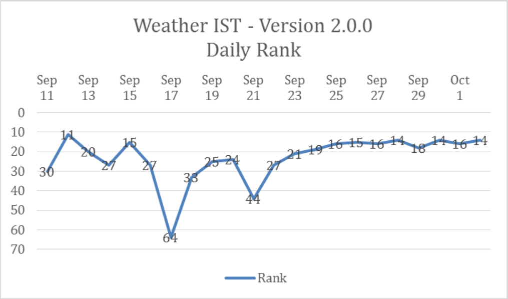


Figure 15: ‘Weather IST’ – Version 2.0.0 daily rank

The application has now more than 1000 downloads. It surpassed that number on September 1 with most of downloads being made in the Portuguese market although there were some in other countries as well. The following chart, in Figure 16, shows the top 5 countries overall.

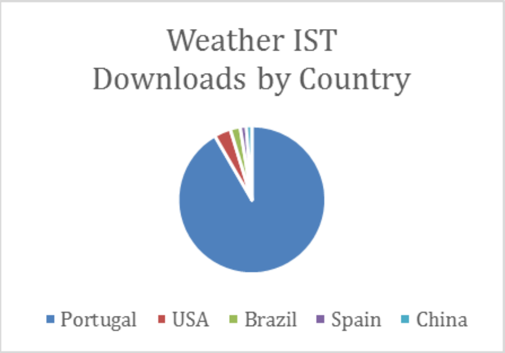


Figure 16: ‘Weather IST’ – Overall downloads by country

7. CONCLUSIONS

The final result of ‘Weather IST’ is, in my opinion, a really good application. The application got a more professional look and feel, and it is also very easy to use compared to the previous version.

Figure 17 shows the app new design.

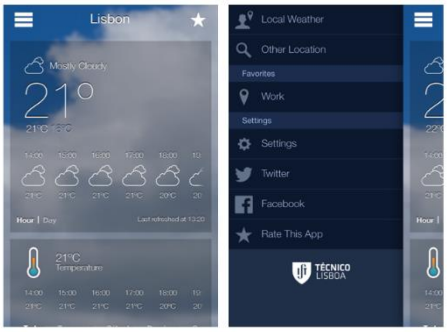


Figure 17: ‘Weather IST’ – Version 2.0.0

Mobile app development is a continuous process with many iterations. There is no such thing as a perfect design and the app has to dynamically be able to change in order to follow the trends, because the App Store is a trend market. What is good today, may not be good tomorrow. Developers have to always keep this in mind if they want their app to succeed in this market, and although testing is a very important step of this process, so the app is not published with bugs, the best test they can make to it is to release it on the App Store. It will reach millions of users throughout the world, all with different needs and tastes and only then will you have the results that allow you to develop the app in one or more directions.

7.1 Things Learned

Design

I thought that by having great and precise forecasts the design would just be a way of showing those forecasts. It is not. The design has to reflect the quality of those forecasts because the user has no idea of what the services can or cannot do. He only knows what he sees and interacts with.

Objective-C

Objective C was a completely new experience for me. All other object-oriented programming is relatively easy since the syntax is simple. All the brackets and the pointers with a different syntax for function call, memory management etc. offers a level of resistance at the beginning. I overcame this difficulty using some materials available on the web.

App Store market

App Store is like any market out there. It is not just about developing a great application, it is also about good marketing. One thing I learned is that, if you are about to release an app that will only have forecasts for a specific country, do not release it in the summertime. The tendency to open a weather application during the summer is too low compared to other periods in the year. The probability of having 2 or 3 months of sun and clear sky is very high in the summer and because of that, users will not be searching for these.

7.2 Future Work and Improvements

Forecasts for the entire world

Although the application only serves forecasts for Continental Portugal, it is available to download on the App Store in the entire world. Even if users in other countries like the application they cannot really use it in their places. What should be done is to open the search location to any country and if the chosen location is outside of Portugal, use an external service to get the weather forecasts while displaying them in the same way as they do now. There are a number of weather API available in the market like weather.com or wunderground.com.

REFERENCES

Allan, A. (2012). *Learning iOS Programming, Second Edition.* Sebastopol: O'Reilly.

Apple. (2009, January 21). *Apple Reports First Quarter Results.* Retrieved September 21, 2013, from Apple: http://www.apple.com/pr/library/2009/01/21Apple-Reports-First-Quarter-Results.html

Apple Inc. (2013, September 18). *About the iOS Technologies*. Retrieved from iOS Developer Library: https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html

Lynch, P. (2007, March). The origins of computer weather prediction and climate modeling. *Journal of Computational Physics 227*, 3431-3444.

Trancoso, A. R. (2012, April). Operational Modelling as a Tool in Wind Power Forecasts and Meteorological Warnings.