

PROJECT PROPOSAL

MyPark app



Team Members

Flavia Zancanaro

Cliverson Thomas Obrzut

Advanced Studio 1

TABLE OF CONTENT

TABLE OF CONTENT	1
LIST OF FIGURES	3
LIST OF TABLES	4
1 INTRODUCTION	5
2 RESEARCH STUDY	6
2.1 TECHNOLOGY RESEARCH	6
2.1.1 Mobile Application	6
2.1.1.1 <i>What is a native application?</i>	6
2.1.1.2 <i>What is a hybrid application?</i>	7
2.1.1.3 <i>Native or Hybrid?</i>	8
2.1.2 Cross Platform Development	9
2.1.2.1 <i>Xamarin</i>	10
2.1.2.2 <i>Apache Cordova</i>	11
2.1.2.3 <i>Ionic Framework</i>	12
2.1.3 Database for Mobile Applications	13
2.1.3.1 <i>Firebase</i>	14
2.1.4 Web Services	14
2.1.5 Mobile Hardware	14
2.1.5.1 <i>GPS</i>	15
2.2 MARKETING RESEARCH	15
2.2.1 Official Parks Information	15
2.2.2 Existing Mobile Solutions	15
2.2.3 Survey	17
2.2.4 Mobile Devices Research	20
2.3 SOCIAL RESEARCH	21
2.4 RESEARCH CONCLUSION	21
3 SYSTEM DETAIL DESIGN	23
3.1 SYSTEM REQUIREMENTS	23
3.2 SYSTEM FEATURES	23

3.3	SYSTEM FLOW CHART	25
3.4	DATA DICTIONARY	26
3.5	LAYOUT	27
3.5.1	Colours	27
3.5.2	Wireframes	28
3.5.3	Mockups	30
4	SYSTEM ARCHITECTURE	32
5	TECHNOLOGY REQUIREMENTS	33
5.1	SOURCE CODE MANAGEMENT	33
5.2	MOBILE DEVELOPMENT	33
5.3	HARDWARE	34
6	PROJECT PLANNING	35
6.1	LEARNING CURVE	35
6.2	IMPLEMENTATION PLAN	35
6.2.1	Schedule	35
6.2.2	Gantt	37
6.3	COMMUNICATION PLAN	38
6.4	CONTINGENCY PLAN	38
6.5	TEST PLAN	40
7	CONCLUSION	42
8	REFERENCES	43
	APPENDIX A – SURVEY QUESTIONNAIRE	44

LIST OF FIGURES

FIGURE 1 – Native Application concept - Image courtesy by web-hitters.com.....	7
FIGURE 2 – Architecture of a Typical Hybrid Application - Image courtesy by Mobisoft Infotech.....	8
FIGURE 3 – Cross-platform native application developed with Xamarin - Image courtesy by xamarin.com	10
FIGURE 4 – Hybrid application Architecture with Cordova - Image courtesy by cordova.apache.org.	11
FIGURE 5 – Technologies related to Ionic - Image courtesy by Devmedia.	12
FIGURE 6 – Hybrid Application Architecture using Ionic - Image courtesy by forum.ionicframework.com.	13
FIGURE 7 – City of San Francisco Recreation and Park mobile app - Image courtesy by SFexaminer.....	16
FIGURE 8 – City of Chesterfield Parks, Recreation & Art mobile app - Image courtesy by Group US LLP.....	16
FIGURE 9 – Playground Finder mobile app - Image courtesy by playgroundfinder.com.....	17
FIGURE 10 – Survey Monkey result – How old are you? – April/2017	18
FIGURE 11 – Survey Monkey result – How often do you visit public parks in Sydney? – April/2017	18
FIGURE 12 – Survey Monkey result – Why to you go to the Parks? – April/2017	19
FIGURE 13 – Survey Monkey result – How do you find park information? – April/2017.....	19
FIGURE 14 – Comparative showing how many devices are owned by the interviewed – Courtesy by Deloitte...	20
FIGURE 15 – Comparative showing which brand is most owned by interviewed – Courtesy by Deloitte.....	20
FIGURE 16 – MyPark App Activity Flowchart	25
FIGURE 17 – MyPark App Layout Colour Scheme.....	27
FIGURE 18 – MyPark App System Architecture representation – Image courtesy by dotnettricks.com.....	32
FIGURE 19 – GitHub’s ATOM text editor view - Image courtesy by pplware.....	33
FIGURE 20 – MyPark App Implementation Phase Schedule	36
FIGURE 21 – MyPark App Project Gantt Chart	37

LIST OF TABLES

TABLE 1: Native x Hybrid application comparative.....	9
TABLE 2: MyPark app Features - Functionalities	23
TABLE 3: MyPark app Features - Layout	24
TABLE 4: MyPark app – Data Dictionary	27
TABLE 5: MyPark app Wireframes - Registration, Login and Search.....	28
TABLE 6: MyPark app Wireframes – Search Result List and Park Details	29
TABLE 7: MyPark app Mockups - Registration, Login and Search.....	30
TABLE 8: MyPark app Mockups - Search Result List and Park Details.....	31
TABLE 9: Software Requirement for MyPark app implementation	34
TABLE 10: MyPark app Project Plan - Matrix of Communication	38
TABLE 11: MyPark app Project Plan – Risks x Contingency.....	39
TABLE 12: Metrics for Probability and Impact of Risks.....	40
TABLE 13: MyPark app Test Plan – Test Priorities.....	41
TABLE 14: MyPark app Test Plan – Types of Tests to be performed.....	41

1 INTRODUCTION

Our body ought to be thought of as an ordeal connected to insight, to the full advancement of the individual. Physical prosperity is fundamental for our development, particularly for youngsters, who are setting up body, mental, and engine records.

There are numerous potential outcomes for physical exercises, and they fortify coordination, adjust, agreement, fixation, quality, deftness, train and numerous different aptitudes. So, it makes a sound and dynamic way of life, which gives relaxation and prosperity.

In 2014–15, an expected 11.2 million grown-ups (63%) were overweight or corpulent—6.3 million (35%) were overweight and 4.9 million (28%) were stout. One of every 4 kids matured 5–17 (27%, or 1 million) were overweight or fat (ABS 2015).

Among kids and youngsters matured 5–17 years in 2011–12, 80% did not meet physical movement suggestions on every one of the 7 days of the week. Rates expanded from 64% for those matured 5–8 to 94% for those matured 15–17 (ABS 2013).

Thinking in solutions to convey welfare advantages to population, places where people can do physical exercises are the first to reach our mind.

Sydney, is one city with several parks, green areas and recreational places. But it did not present information about parks to population in a simple, quick and easy way.

In connection to these outcomes, it was chosen to build up an application not only to provide centralized information, but also to make it easier for people find what they like most.

MyPark will enable individuals to interface with nature, give a support of pursuit parks to have a great time. The clients can look for play areas for kids and places to have a grill or cookout and furthermore to do works out. At the point when individuals are in the parks, they are detached and interfacing with others, other than being exceptionally charming and unwinding.

Besides, the application will be produced with simple cooperation, in a straightforward and target way, giving maps of area and relevant data about the recreation centers.

2 RESEARCH STUDY

2.1 TECHNOLOGY RESEARCH

Before start developing *MyPark* application, some research was necessary to establish the best technology approach. *Which technology are most used? How the best choice considering project schedule, project size and learning curve?* To answer these questions the following research was conducted and provided support to previous decisions in terms of technological solution for the proposed system.

2.1.1 Mobile Application

Since the “*Smartphone Era*” started, business companies around the world started developing applications specifically designed for mobile operating systems (OS). At the beginning, the applications were native applications. However, with the complexity of managing two different source codes for the same application, keeping the consistency of the them for the final users and managing teams with different knowledge, new technologies were introduced: the cross-platform development and the hybrid application.

2.1.1.1 What is a native application?

Native application is exactly what comes to mind when talking about application. It is one that is programmed in a language unique to an operating system and is found in App Stores.

There are two types of dominant operating systems on smartphones: Android and iOS. The difference between them is not just aesthetic, what was developed for one does not work on the other.

The native application is programmed in the language of each system, like Java in Android and Objective-C in iOS, and can be written in other languages for each system. Each platform presents its own tools and interface elements.

Because they are programmed exclusively for the operating system, the native application is faster and more reliable than the others. They present a better user experience by being able to use all the features offered by smartphones such as camera, GPS and push notifications.

This own programming for the operating system makes the native application performance very good. They also have a longer usage time than others because they can work without an internet connection.

When programming a native application, developers respect a design pattern. Guides are offered for each operating system, such as the Android design guide and the iOS design guide, which contains best practices to provide a good user experience.



FIGURE 1 – Native Application concept - Image courtesy by web-hitters.com.

2.1.1.2 What is a hybrid application?

The hybrid application, as the name itself suggests, is a blend of a native app and a web app.

The hybrid application is built in HTML5, CSS and JavaScript, as well as the mobile site. This code is placed inside a container, integrating the features that your device offers, offering a better user experience than the web Apps.

So, only part of the native code should be written for those Apps. This allows only part of the code to be rewritten if you want to offer the application to another platform.

They will be available for download in the App Stores, offering a traffic and download channel. Like the native app, the hybrid has a maintenance cost in App Stores.

It is simpler and therefore faster to develop. You do not need to download whenever an update is made to your app. But its use will still depend on internet connection and speed of the same, not working as fast as a native application.

The Figure 2 shows the typical architecture of a hybrid application

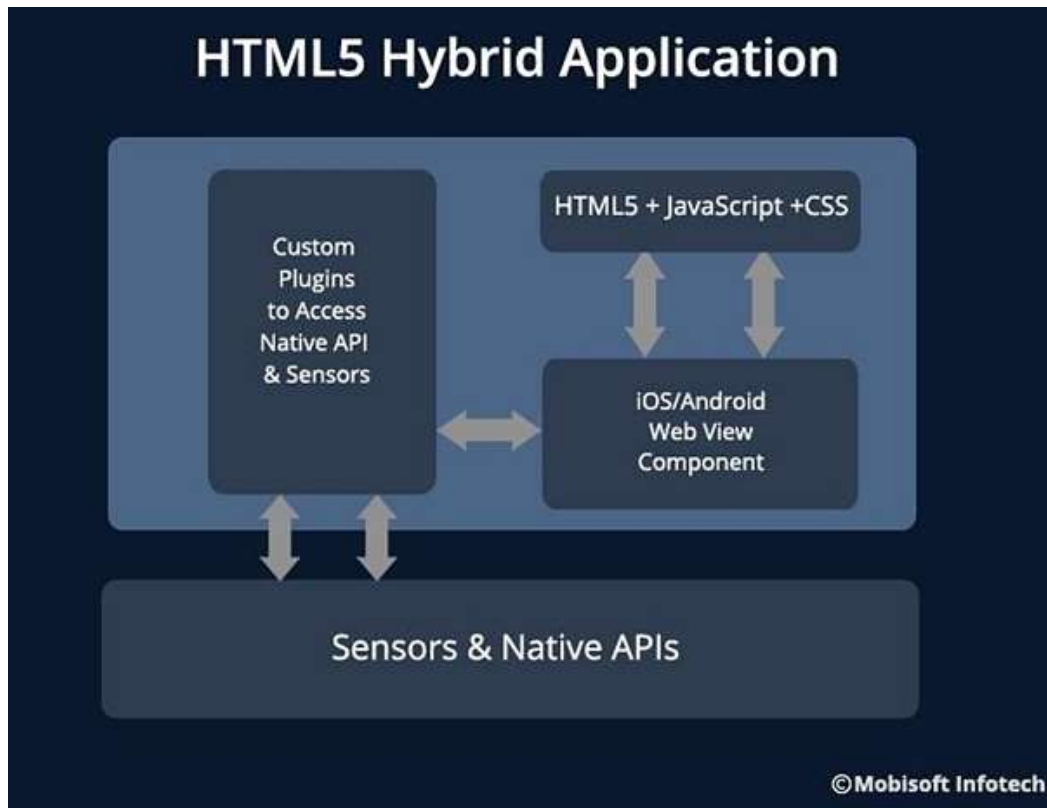


FIGURE 2 – Architecture of a Typical Hybrid Application - Image courtesy by Mobisoft Infotech.

2.1.1.3 Native or Hybrid?

The following image displays a comparative between Native and Hybrid applications and which are the advantages and disadvantages in terms of development skills, costs, performance, available features and a suggestion of when to use one or another.

	Native	Hybrid
Skills needed to reach Android and iOS	Objective-C, iOS SDK, Java, Android SDK	HTML, CSS, JavaScript, Mobile Development Framework
Distribution	App Store/Market	App Store/Market
Development speed	High	Moderate
Development cost	High	Moderate
Maintenance cost	High	Moderate
Graphical performance	High	Moderate
App performance	Fast	Moderate
Device Access and features		
Camera	Yes	Yes

Push Notifications	Yes	Yes
Contacts	Yes	Yes
Offline access	Yes	Yes
Geolocation	Yes	Yes
File upload	Yes	Yes
Gyroscope	Yes	Yes
Accelerometer	Yes	Yes
Swipe Navigation	Yes	Yes
Microphone	Yes	Yes
Best Used For	Games or consumer-focused apps where performance, graphics, and overall user experience are necessary	Apps that do not have high performance requirements, but need full device access

TABLE 1: Native x Hybrid application comparative

The comparative shows advantages for native solutions in terms of development speed if you are targeting only one operating system, and application performance, especially when involve graphics and overall user-experience. On the other hand, hybrid applications have advantages in terms of skills for development, costs and distribution time, especially for simpler systems that not request high performance.

The answer in choosing native or hybrid cannot be based on these characteristics only. Each project has its own particularities, such as deadline, team expertise, budget and the type of product they want to build.

2.1.2 Cross Platform Development

The definition of cross platform mobile development is “*the creation of apps that are compatible with multiple mobile operating systems*” (TechTarget, 2015). This not mean the application will be hybrid though. It also can be native.

Native applications normally were built using the mobile operating system’s Software Development Kit (SDK) that uses its own preferred programming language, for example, the Android Studio SDK for Android OS that uses Java as the developing language and the XCode SDK for IOs OS that uses Objective-C or Swift as the developing language. However, the SDK’s provide Application Programming Interfaces (API), which allows to develop using other programming languages not supported by the OS vendor (Ritesh Patil 2016).

Hybrid applications are considered essentially Graphical User Interfaces (GUI) with connection to web services to access the large part of their business logic (Ritesh Patil 2016).

In both cases it is perceptible the opportunity to have part of the codebase as one and deliver for different OS. The advantages of that goes from reuse of code and benefits for maintenance to save money and the use of specialists in one development language only (Ritesh Patil 2016).

The most popular cross platform framework for native applications is Xamarin and for hybrid application is Cordova (Ritesh Patil 2016).

2.1.2.1 *Xamarin*

Xamarin enables the development of native applications from the .NET Framework using the C # language. So, the compiler will generate native code, which will have the advantages of optimum performance, ideal UI, and direct use of device features, from the same APIs used in each OS SDK.

However, Xamarin has one limitation: code is not completely reused (shared) on target platforms. Sharing is limited to about 75% (depending on the application), since there is a need to create specific code, even in C #, for each platform. This code includes user interfaces, access to device features, and so on. In other words, there is sharing of the core of the application, but another large part of the code needs to be built separately, with C #. This makes Xamarin's learning curve not as low as expected, as we also need to learn concepts related to iOS and Android apps (such as configuring layout files, features, and other details of those platforms). This can be seen clearly in Figure 3.



FIGURE 3 – Cross-platform native application developed with Xamarin - Image courtesy by xamarin.com .

The great advantage of Xamarin lies in the final product, which will be a native application, and its ease of maintenance. With just a team of C # developers, using tools like Microsoft Visual Studio, considered one of the most complete IDE's in the market, it is possible to build native applications for the major mobile platforms.

2.1.2.2 Apache Cordova

Apache Cordova is “an open-source mobile development framework. It allows you to use standard web technologies - HTML5, CSS3, and JavaScript for cross-platform development. Applications execute within wrappers targeted to each platform, and rely on standards-compliant API bindings to access each device's capabilities such as sensors, data, network status, etc.” (Cordova Apache Documentation 2017).

A hybrid application developed with Cordova has many components as the architecture diagram shows below:

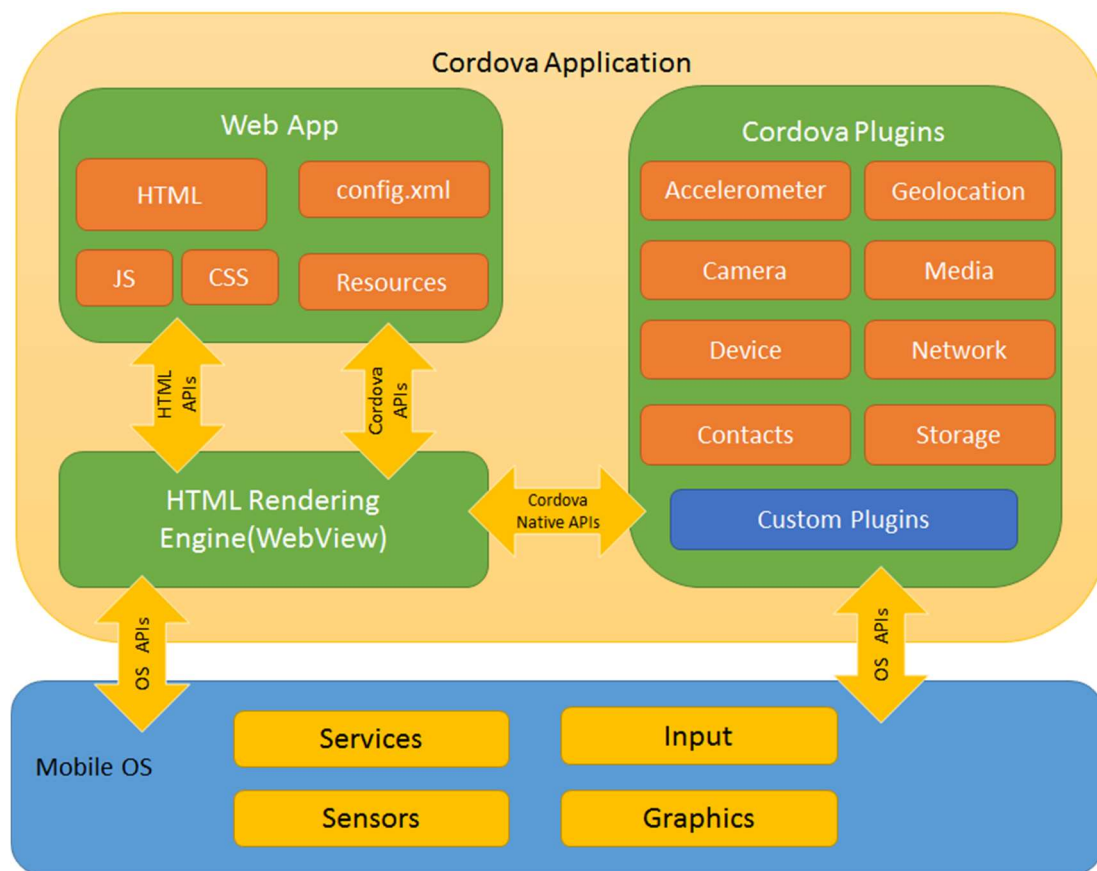


FIGURE 4 – Hybrid application Architecture with Cordova - Image courtesy by cordova.apache.org.

WebView is the name given to a special type of browser that starts to run as soon as the hybrid app is opened by the user. It is within that browser that the app is executed. The user does not realize that it is inside a browser because this WebView does not contain any characteristic components such as the address bar and favourites bar, for example. The WebView contains just what is needed for HTML, CSS and JavaScript to work. It behaves like the render engine of the app (Cordova Apache Documentation 2017).

Web App is where the application code is stored. It may be seen like a Single Page Applications (SPA), but it will still be like a website, using HTML, CSS, JS, and so on, with all normal web limitations (Cordova Apache Documentation 2017).

Plugins provide an interface for Cordova to communicate with devices standard API

components, such as Camera, Network, Storage, GPS, etc (Cordova Apache Documentation 2017).

Apache Cordova has many benefits such as: simple to learn, compatible on multiple platforms, provides cost effective development and extensive plugin libraries, which helps to achieve the full potential of the phone device using the native features (Vensi, Inc. 2017).

2.1.2.3 Ionic Framework

Ionic is an open source framework for multiplatform mobile application development. It enables the implementation of the app using technologies commonly employed in building Front-end web solutions: HTML, CSS and JavaScript. However, as opposed to the framework that is based on Apache Cordova, it brings features that simplify development even further and give the app a more professional look.

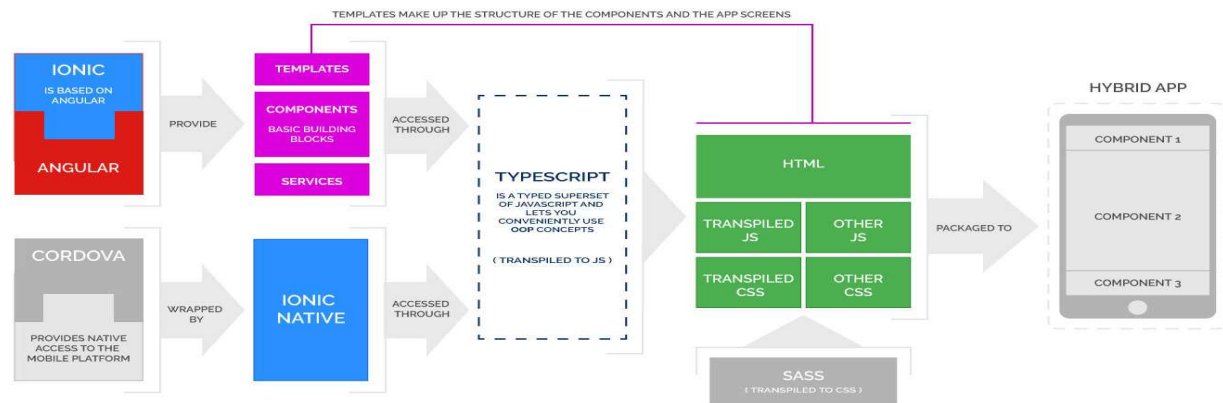
These differentials are related to the set of visual components that we can use to construct the Frontend of the solution, as well as the fact that Ionic brings with it another language and framework to provide a higher-level solution in terms of code and, consequently, design. These solutions are TypeScript and AngularJS (Figure 5).



FIGURE 5 – Technologies related to Ionic - Image courtesy by Devmedia.

Next, it is possible to visualize the architecture diagram for a hybrid application built using Ionic Framework.

A BIRD'S EYE VIEW OF IONIC AND RELATED TECHNOLOGIES



Created By: KishuPro - <https://forum.ionicframework.com/users/KishuPro/>

FIGURE 6 – Hybrid Application Architecture using Ionic - Image courtesy by forum.ionicframework.com.

There are many advantages in using Ionic, such as the cross-platform compatibility, using Cordova as the wrapper to run as a native app; the clean UI and the bundle of built-in CSS and JS components that cover most of the things needed in a mobile application, and the use of AngularJS to build creative mobile apps reducing the development time (TVS Next 2017).

2.1.3 Database for Mobile Applications

Database is “a collection of information that is organized so that it can be easily accessed, managed and updated” (TechTarget, 2017).

Since users are highly reliant on their mobile applications, and the information that comes to their apps depends on internet connection, there were several reasons to consider a suitable database to avoid bad experiences for the user.

Recently providers of database and cloud services have added synchronization and offline capabilities allowing mobile applications work both online and offline (InfoWorld 2017).

Some of the most known used solutions are: CouchBase Mobile (Couchbase), Azure Mobile Services (Microsoft), Cognito (Amazon) and Firebase (Google) (InfoWorld 2017).

With so many options, the following six keys criteria should be considered when evaluating mobile solutions: security, conflict resolution, topology support, sync optimization, modeling flexibility and platform support (InfoWorld 2017).

2.1.3.1 *Firebase*

Firebase is not only a Database. It is a complete solution that permits the development of web applications with no server-side programming. With just a few configurations it is possible to store data, verify users, define access rules and counts with security management (Priyanka Varma 2016).

Firebase supports web, IOs and Android mobile clients with the capability of store and synchronize data in real time. Through the client libraries or the access using REST API the Database and the Application have total integration (Priyanka Varma 2016).

Some advantages of Firebase are: it uses a non-SQL database which enhance the scalability of the app; it cut down the time used to develop the app; stores data in JSON format what make it easy to read; it uses 2048-bit SSL encryption for data; and have good integration with frameworks like AngularJS (Priyanka Varma 2016).

2.1.4 *Web Services*

The definition of Web Services, according to the W3C, is a software system responsible for providing the interaction between two machines through a network. In another word, it provides a way for different applications, even in different platforms, to interact with each other. These interactions occur through Application Programming Interfaces (API's), that it is nothing more than methods that an application makes available to be called for other applications through the internet.

Web services are increasingly enabled by the use of standardizing data formats, such as Extensible Markup Language (XML) and JavaScript Object Notation(JSON) for exchange of data (TechTarget 2017).

Most mobile applications use webservices to access data through internet, such as read and write data in remote databases, sometimes using third party API services to bring some extra functionalities to their own applications. Firebase is a good example of a system that provides API's to use data storage, data security, scalability and other features, that can be integrated to a mobile application avoiding the needs of server development and management.

2.1.5 *Mobile Hardware*

Smartphones are basically computers. They run on processors. However, along with the processors they also have many other built-in technologies that enables them to do several different tasks providing ways for applications to use them, increasing the user experience.

The most common hardware features available and used in a smartphone are: the touchscreen, the camera, the network, the GPS, the accelerometer and others complex functions that runs in chips mobile phone services (SIM), to connect to internet (WIFI, Bluetooth), etc.

2.1.5.1 GPS

GPS, or Global Positioning System, is a system composed of 30 satellites placed into orbit with the purpose to provide information about location and time anywhere on Earth. The GPS receiver is capable of calculate the device's geographical position with the information received from the satellites.

The great majority of smartphones has a GPS receiver embedded, and many applications can use from this resource to offer a variety of services for users.

2.2 MARKETING RESEARCH

The marketing research has the purpose to define the project viability. The research for similar solutions and the survey with the population can define if the observed problems were problems to many people. Also, it can clarify the project scope boundaries, define the profile of the target audience and give the hints about which platform would be suitable for the proposed system.

2.2.1 Official Parks Information

The official channels providing information about parks and recreational places in Sydney, are the Sydney Official Tourism website from NSW Government, and all the Sydney Councils websites. None of them provides a mobile version of their website for the population.

2.2.2 Existing Mobile Solutions

The first similar occurrence for a mobile app that has many features like the ones MyPark would have was “City of San Francisco Recreation and Parks Official Mobile App”. It belongs to the city of San Francisco, California - USA.



FIGURE 7 – City of San Francisco Recreation and Park mobile app - Image courtesy by SF Examiner.

This app counts with a search system that integrates more than one option, returning information about places that matches the input. It also offers a list of parks in the city, and a map to see location and distances.

The second solution found was “City of Chesterfield Parks, Recreation & Arts”. It belongs to the City of Chesterfield, Arizona – USA.



FIGURE 8 – City of Chesterfield Parks, Recreation & Art mobile app - Image courtesy by Group US LLP.

This system counts with some direct filter options for recreation and concentrate information about parks and events happening in the city.

The third app found was “Playground Finder” mobile application.

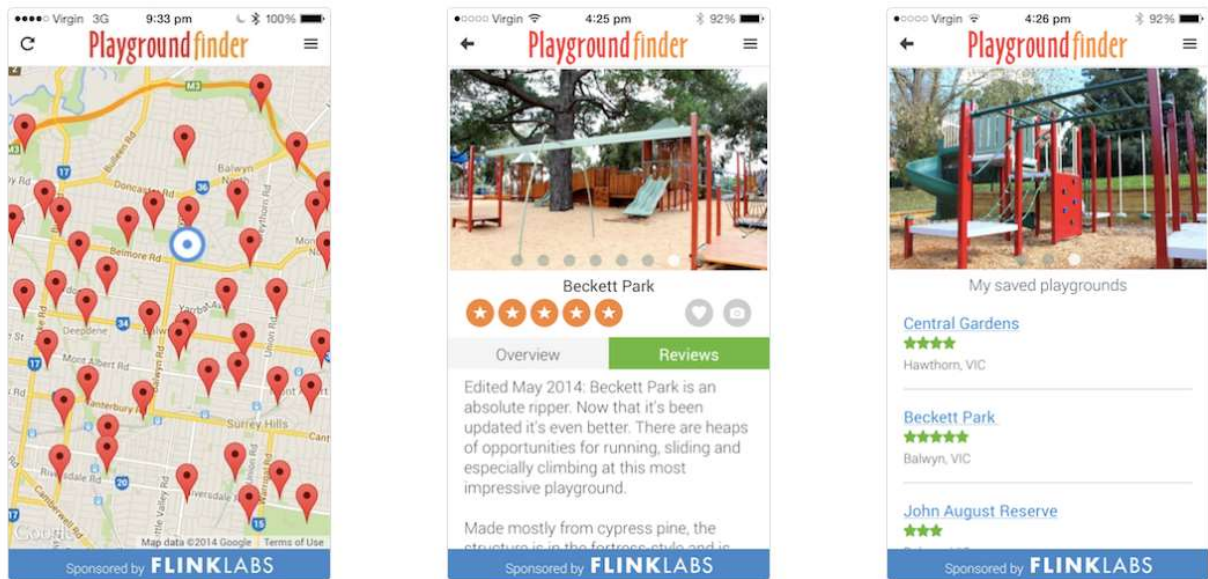


FIGURE 9 – Playground Finder mobile app - Image courtesy by playgroundfinder.com.

This application was built to provide information about Playground places in all cities of Australia. It has good functionalities as maps, reviews and ratings, but it does not extend to other kind of facilities parks usually have.

2.2.3 Survey

A survey was created using the tool Survey Monkey and it was conducted to discover the opinion of the population about visiting parks and understand their preferences.

The survey was composed of 10 questions (see Appendix A). Among them, there was questions about age, how often they go to a park, what they most like to do in parks, and specially how they research for information about parks.

Around 50 people answered the survey where some conclusions could be made.

95% of interviewed people with ages between 26 and 45 years old goes to park very often or extremely often. It corresponds to approximately 60% of people that visit parks frequently.

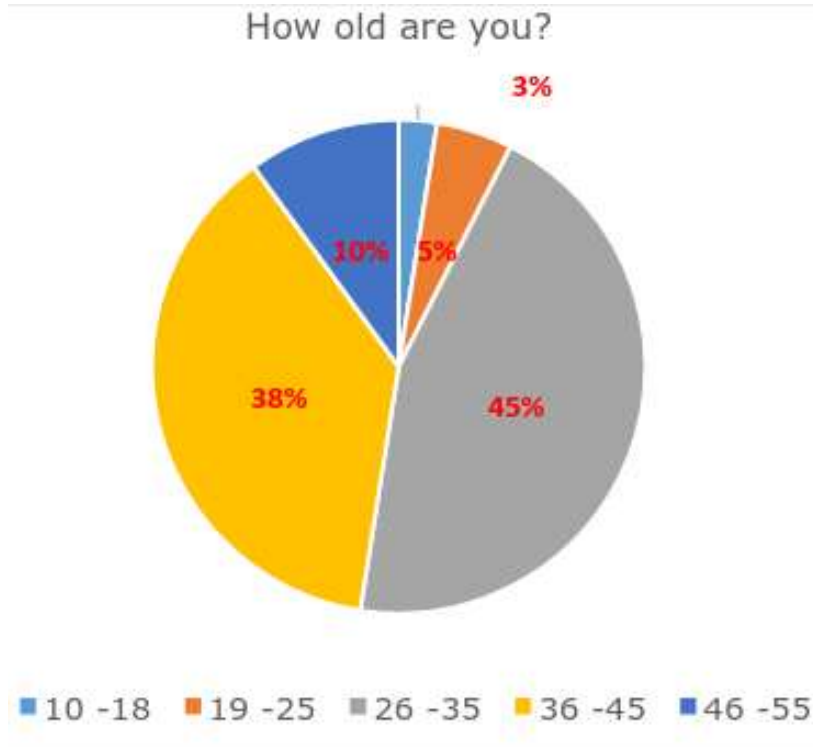


FIGURE 10 – Survey Monkey result – How old are you? – April/2017

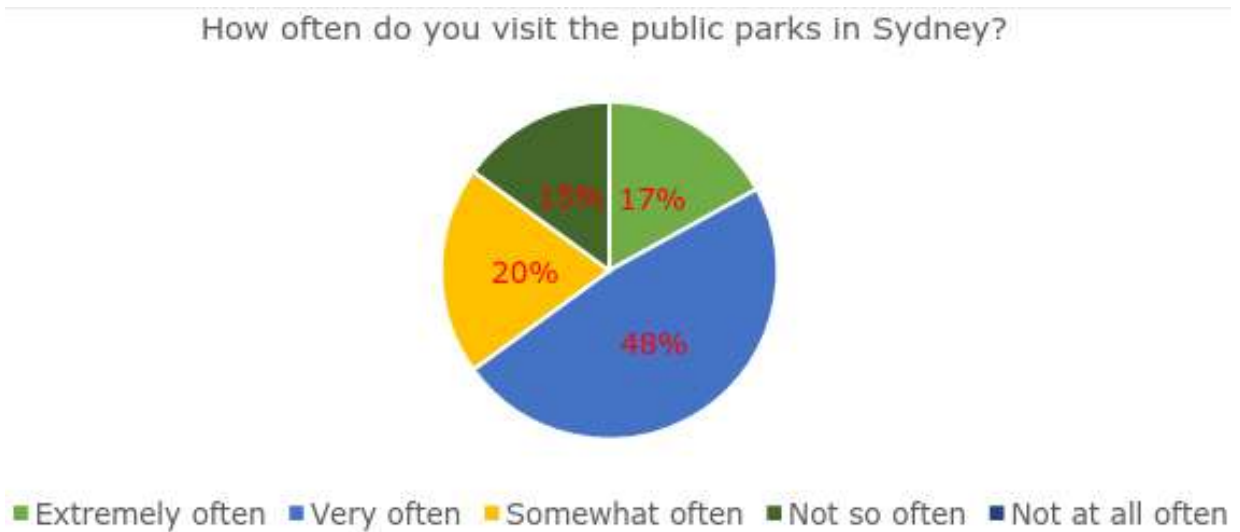


FIGURE 11 – Survey Monkey result – How often do you visit public parks in Sydney? – April/2017

When asked “Why do you go to the park?” the main response was: Time with family and / or friends with 84.86%, secondly having a picnic with 68.42%, followed by bringing the children with 65.79%. It is possible to see that the main reason of park visitations is related to social interaction, followed by relaxation and then physical activities.

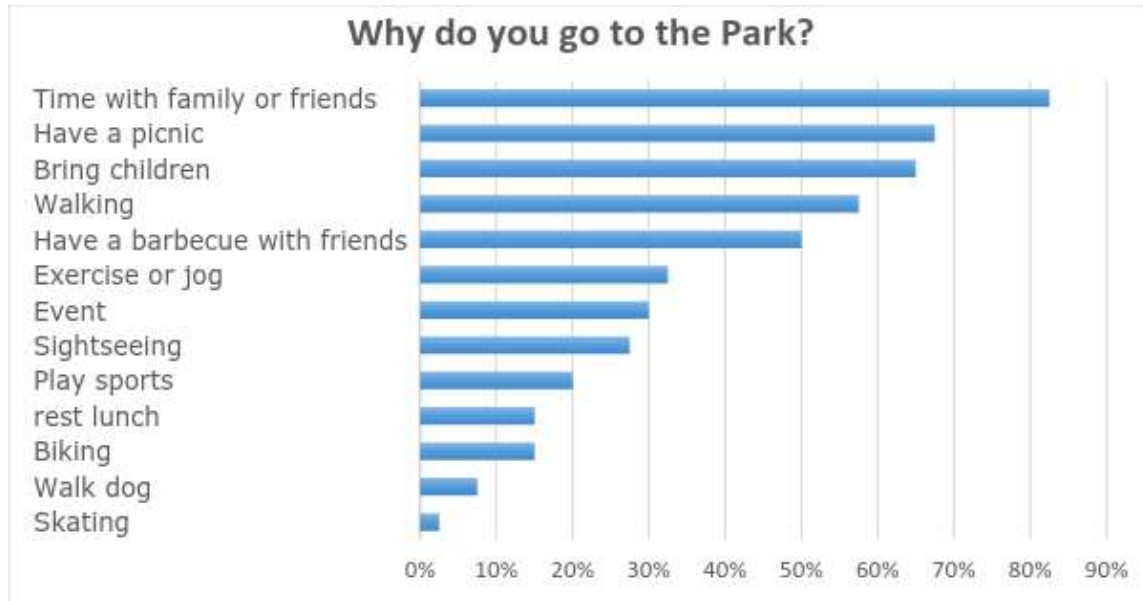


FIGURE 12 – Survey Monkey result – Why to you go to the Parks? – April/2017

In relation to how they research for park information, 52% try in Google, 31% in Social Medias, 13% in official websites and 4% in different fonts.

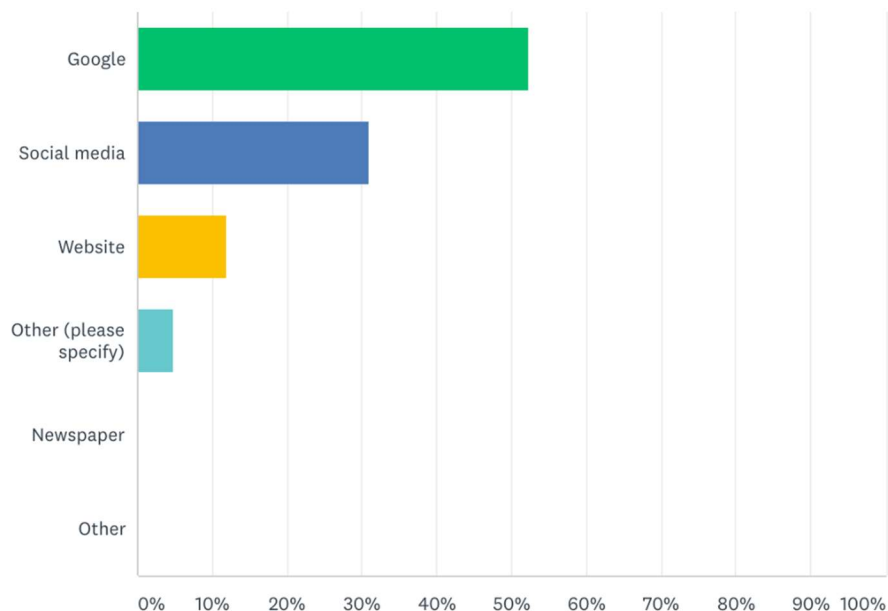
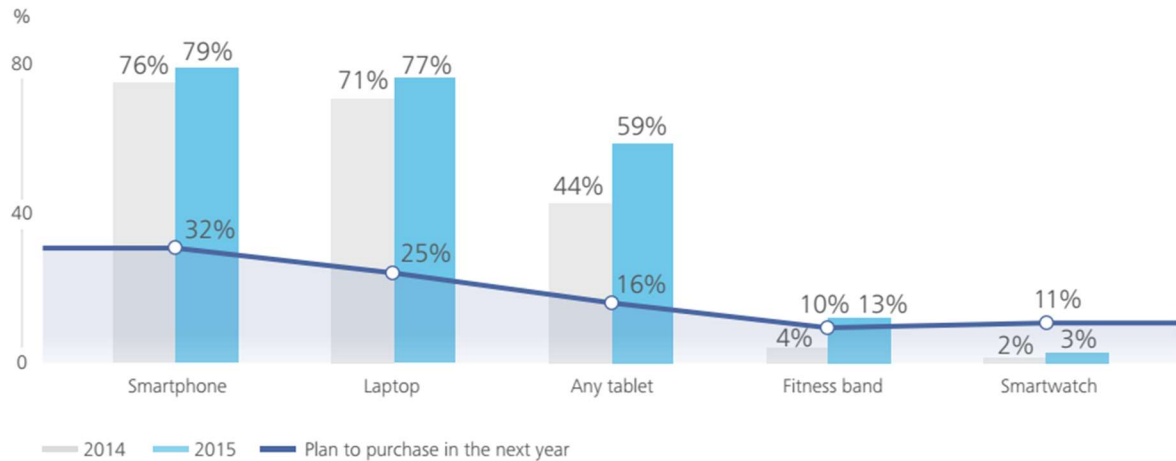


FIGURE 13 – Survey Monkey result – How do you find park information? – April/2017

2.2.4 Mobile Devices Research

According to a Survey conducted by Deloitte in 2015 almost 80% of Australians surveyed have one smartphone. The total amount was around 15 million devices (Mobile Consumer Survey 2016).

Since 2014 the number of smartphones are larger than laptops and other devices.



Note: The responses above relate to the percentage of all consumers surveyed. All other questions use a base of smartphone users only.

FIGURE 14 – Comparative showing how many devices are owned by the interviewed – Courtesy by Deloitte

Regarding to which mobile operating system is most used in Australia, Apple comes first, followed by Android (Mobile Consumer Survey 2016).

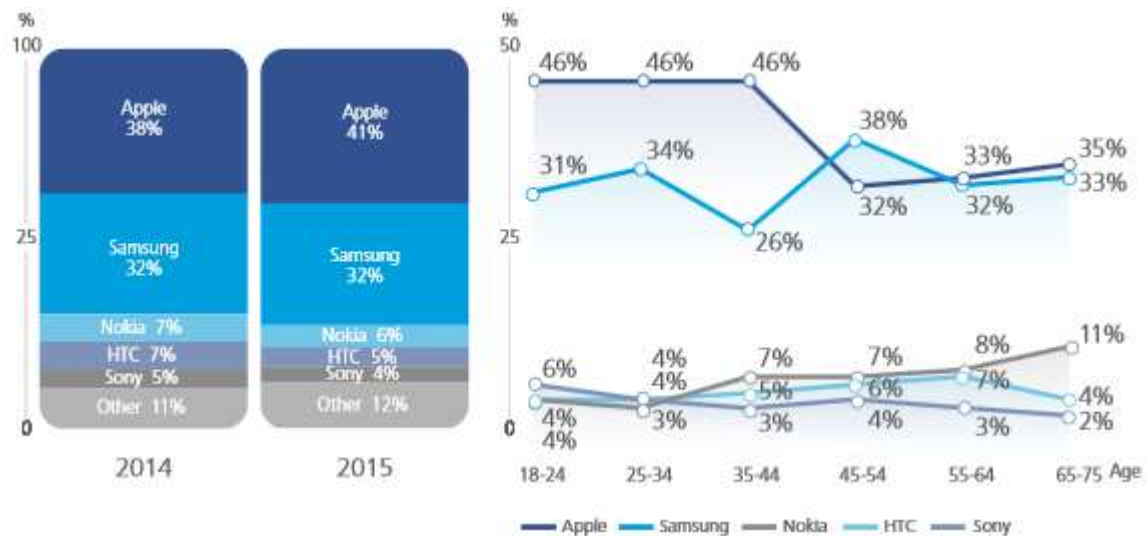


FIGURE 15 – Comparative showing which brand is most owned by interviewed – Courtesy by Deloitte

Apart from some differences in which one is most preferred in certain ages, both Apple and Android are the main devices used with 73% of the total devices.

2.3 SOCIAL RESEARCH

Social research is important to analyze if the proposed system is going to have social impacts and how to handle them. The result of the research demonstrated just a few minor impacts considering the proposed application and target audience.

As a result of the market research, the scope of the product considered only the English language for the proposed system and for the information provided on it. The decision was made base on the identified target audience: people living in Sydney where the majority is composed of English speakers or not-English speakers with good English knowledge. However, it should be considered the impact over another part of the target audience that are the immigrants and tourists that have a low level of English knowledge.

According to the Australian Bureau of Statistics (ABS), the 2016 Census showed that only in NSW there were 2 million people overseas-born from a total of 7.7 million (26%). Greater Sydney, with 5 million people (71% of NSW population) has 36% of the population of overseas-born where 29% came from not-English speaking countries. In another word, considering only the residents of Sydney, around 25% of the potential target users of the system (over 1 million people) are composed of people with a different main language (Census data 2016).

From this research, it was possible to identify two social impacts, one positive and one negative. The positive impact would be helping people with their English learning since they would face different words and sentences within the app related to their daily common activities. The negative impact would be people misunderstanding the content provided in the app and having a bad experience as a result.

In conclusion, those impacts were considered minor impacts since the English language is one of the main languages in the world and the percentage of people migrating or visiting Australia with at least a satisfactory English knowledge is very high.

2.4 RESEARCH CONCLUSION

In conclusion to all research made, the scope of the project could be defined.

The research demonstrated that the initial potential problems are real problems. Information about parks in Sydney are decentralized, split in many different places, making it hard to find it or consume too much time from people researching. With this outcome, the appeal to have a simpler solution is very high and the opportunity must be taken.

The proposed system will be initially designed for the city of Sydney and surround suburbs, the great Sydney. All parks, green areas and recreational places will be included in this solution. This decision was made considering the analysis of other existing solutions that consider local characteristics. Also, the potential problems observed initially was based on experiences in Sydney and it was not found a complete solution or a platform that offer a good experience for the user when looking for this information.

The solution will be a mobile application since the market research showed that this platform is the most used by people, especially in Australia.

The survey demonstrated that the main target audience will be composed of, families, tourists and sports person, with focus for ages between 26 and 45 years old of both genders. This public showed to be the most interested in an application such as MyPark app.

Considering that the target audience is composed of people that owns smartphones mainly with IOS and Android Operating Systems, and considering the time frame of the project, the size of the proposed system and the learning curve for development, it was defined the solution will be:

- a cross-platform application to reach as many users as possible;
- a Hybrid Application using Ionic Framework, which will speed up the development process;
- server side managed by Firebase, that will offer many services to manage data, security, authentication, reducing the time with development and making available time for more tests before deployment.

Among the several mobile hardware features, it was defined that MyPark app will integrate the GPS receiver functionality in part of its features.

The complete research also revealed some potential business model as follow:

- Sell of advertising spaces;
- Promote products from business partners that are in the parks or close to it;
- Establish partnership with government and councils to share data about park visitation, reviews and ratings.

All the decisions were based on characteristics of the project, like schedules and goals, and the product characteristics and features, trying to bring the best user experience and solve the problems identified.

3 SYSTEM DETAIL DESIGN

3.1 SYSTEM REQUIREMENTS

To have MyPark app installed, the minimum requirement for the devices are:

- For Android devices, Android Operating System 5.5 (Lollipop)
- For iPhone devices, IOS 10.
- For Windows Phone devices, Windows Phone 8.1.

3.2 SYSTEM FEATURES

The features of MyPark app are divided in two types: layouts and functionalities.

The Table 2 shows the functionality details and the Table 3 the layout details.

FUNCTIONALITY	DESCRIPTION
Login	The login system, that will authenticate the user
Register	The user can create a new account and have access to the content of the application
Search by Categories	This feature will present a list of categories for the user select, even more than one, to search the parks related to that category.
Share a Park	The user can share the details of a park with friend through email.
Map	The user can access the map to see information about location and distances of parks.
Add/Remove Favourites	The user can add a park as their favourite for further consult. It can be for visiting again, sharing or a collection of parks visited.
Create Review	The user can write a review of a park.
Rating	The user can rate the park using the five-star classification.

TABLE 2: MyPark app Features - Functionalities

LAYOUT	DESCRIPTION
Welcome View	Initial display to the user when the app is loading.
Login View	View where the user can enter email and password and execute login
Register View	View where the user enters its personal information to create an account
Search View	View with the list of categories where the user can select the preferred ones to perform a search
Search Result List View	The view with a list of parks that matched the search criteria
Search Result Map View	The view with a map showing the location of parks that matched the search criteria.
Park Details View	The view that show the details of a selected park
Profile View	The view that shows the user information details and list of favourite parks.
Review View	The view that shows the list of review written by users for a specific park
Add Review View	The view where the user can write its own review and rate a specific park
Share View	The view to write an email and send information about the park to specific persons.

TABLE 3: MyPark app Features - Layout

3.3 SYSTEM FLOW CHART

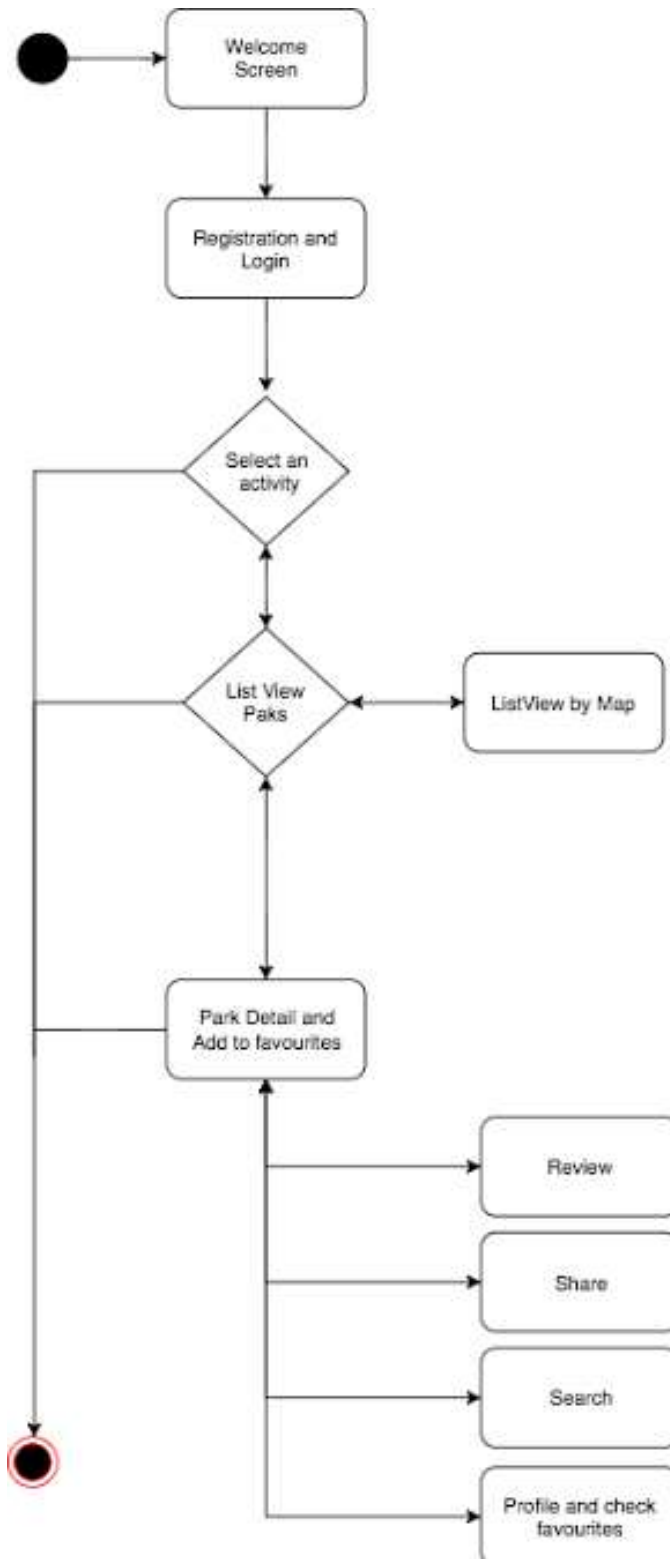


FIGURE 16 – MyPark App Activity Flowchart

3.4 DATA DICTIONARY

With the use of Firebase, it was defined the data dictionary for the proposed system as the table below:

FIELD NAME	TYPE	DESCRIPTION	MINIMUM LENTH	MAXIMUM LENGHT	FORMAT	RANGE
User_ID	varchar	User identifier authentication	1	256	sd07zdxnspP N3nU6mMuC kS7pl5V2	letters, numbers
User_Name	text	The name of the user	1	256	Aaaaaa	letters
User_Email	varchar	The user email	1	256	test@test.com	letters, numbers, char
Data-Created	date	Data created when user do the first registration	1	8	20/09/2017	numbers, char
Park_Name	varchar	The name of Park	1	30	Aaaaaa	letters
Park_Address	varchar	Full address of Park	1	30	2/3 Blackwood Avenue	letters, numbers, char
Park_Postcode	number	Postcode of Park	4	4	2008	numbers
Park_Category	text	Category of Parks	1	30	Aaaaaa	letters
Park_Image	varchar	Park pictures	1	30	Aaaaaa	letters
Park_Facilities	text	Facilities of Parks, e.g. Table, toilet, etc.	2	30	Aaaaaa	letters
Park_Favorites	Boolean	Save Park to favourites.	2	2	Aaaaaa	letters

Park_Rate	text	Give a star rate to a Park	1	5	Aaaaaa	numbers
Park_Review	text	Write a review about the Park	1	50	Aaaaaa	letters, numbers, char

TABLE 4: MyPark app – Data Dictionary

3.5 LAYOUT

3.5.1 Colours

The layout of MyPark was thought to be simple and convey the idea of relaxing and enjoying free time with motivation. The colours have been well selected so that the user want to come back to the application, doing more research and wanting to discover new parks.

The following colour scheme was defined to be used in the app.



FIGURE 17 – MyPark App Layout Colour Scheme

The blue and the yellow were chosen because they represent sky, water, and sun or light. They produce a good contrast with green present in most of the pictures. The dark grey was chosen to be the background color for the entire app, what is good for relaxation of eyes. The white was chosen for text colors and some image borders to create a good contrast with the dark grey in the background.

3.5.2 Wireframes

Wireframes are designed to help developers to quickly have a glance of the appearance of the application, defining position of texts, fonts, images etc. It is the first look to know if the navigability will be enjoyable, if the right content will be displayed in the right time and in the right place.

What defines, in large part, whether or not users will continue to use an App is the ability it has to offer a good content, in addition to being functional and easy to navigate through it. The visual part is very important at this time, but the product is not considered good if it display a beautiful view but it does not work well. The same happens if the product is ultra-functional, but does not have a visual appeal. Both parties are allied, and needs to receive the same amount of consideration during design.

The wireframes for MyPark were designed considering some mobile design patterns, to propitiate a nice user experience. The chosen text fonts, the size and position of images and position of menu and other icons was thought to make the app as intuitive and user friendly as possible.

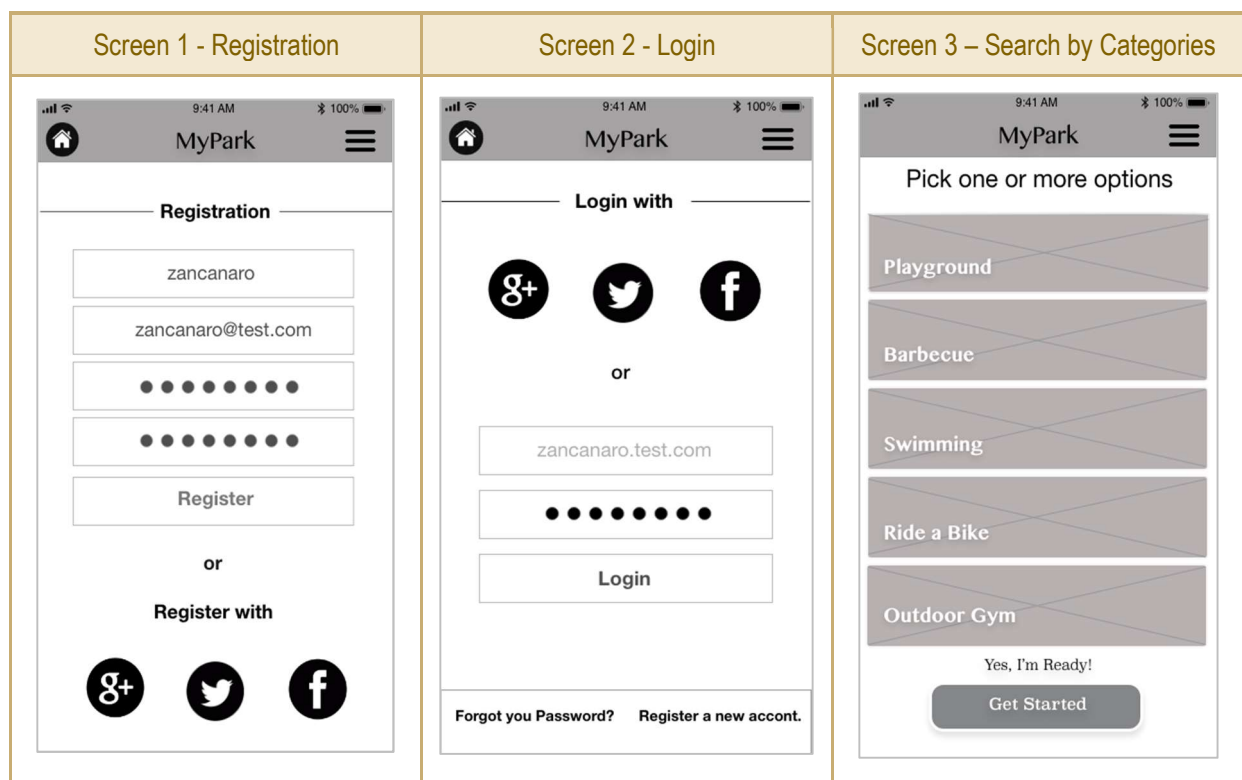


TABLE 5: MyPark app Wireframes - Registration, Login and Search



TABLE 6: MyPark app Wireframes – Search Result List and Park Details

3.5.3 Mockups

Mockups are representations of objects and products, in their natural size or on a large scale. It is a well-designed representation very close to the final design of the product in question.

The evaluation of the product is much easier when using this tool, which allows to developers to analyses the user feedback over the product before starting the implementation.

The MyPark mockups were designed using Adobe XD tool, which has a feature to link each image screen to each other simulating the navigability and functionalities the app. Using this resource, the test results of layout and usability become more accurate.

In the next Tables, it's presented a glance of some of the screens of the MyPark app.

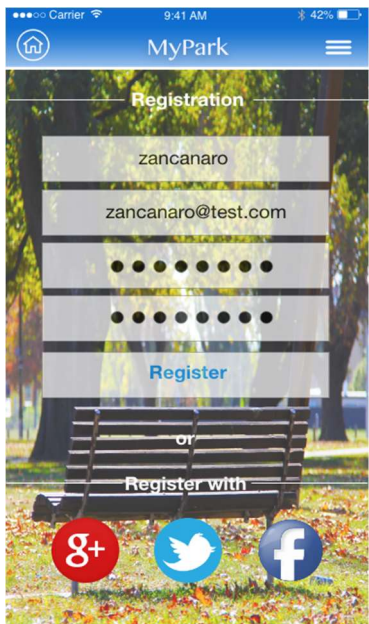

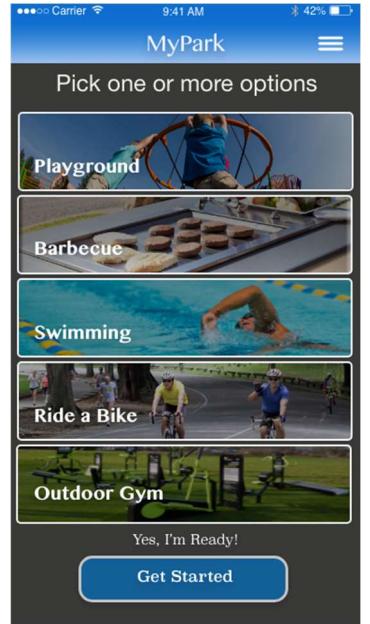
Screen 1 - Registration	Screen 2 - Login	Screen 3 – Search by Categories
		

TABLE 7: MyPark app Mockups - Registration, Login and Search

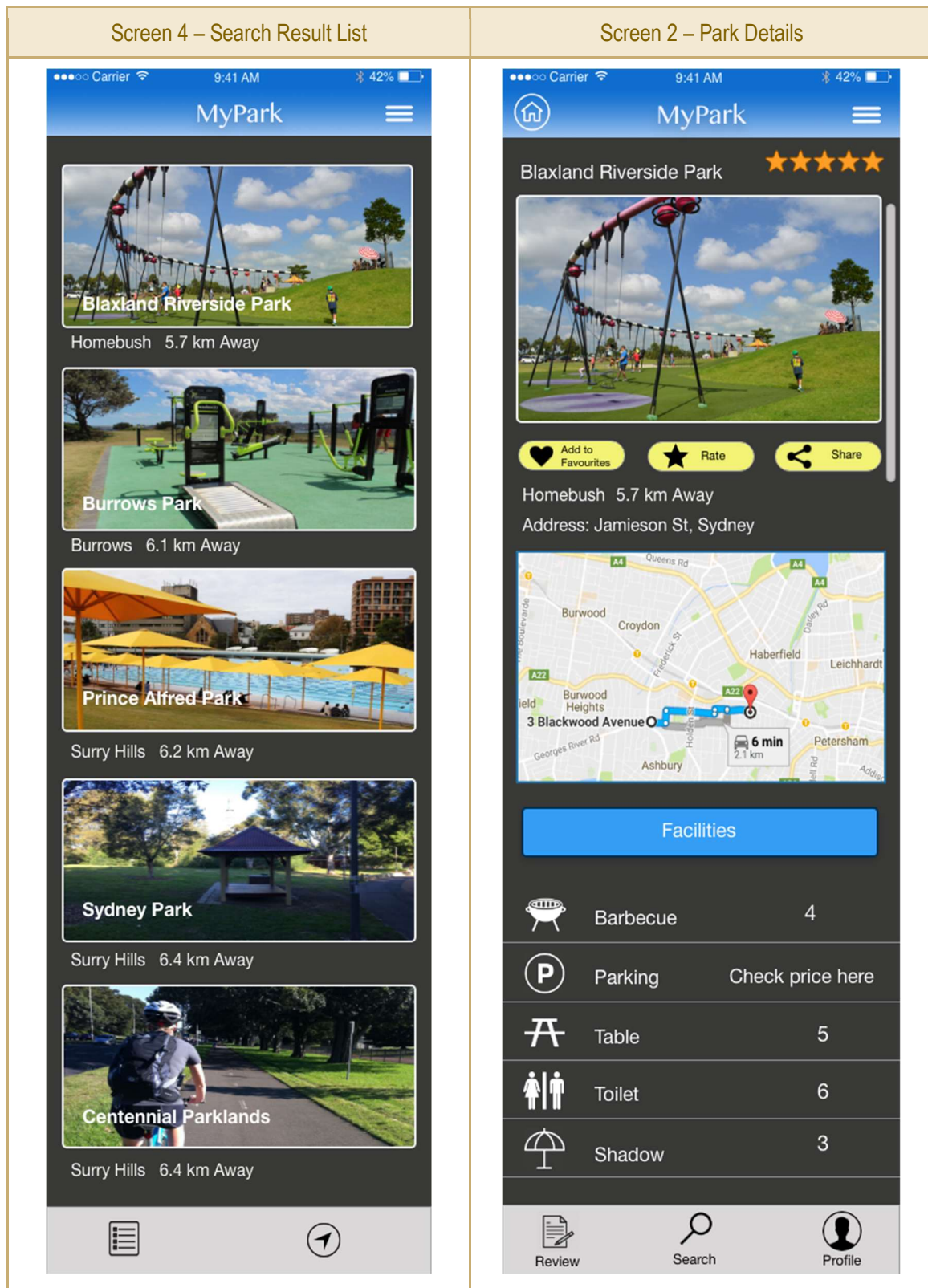


TABLE 8: MyPark app Mockups - Search Result List and Park Details

4 SYSTEM ARCHITECTURE

For the proposed system, the system architecture can be represented by the following image.

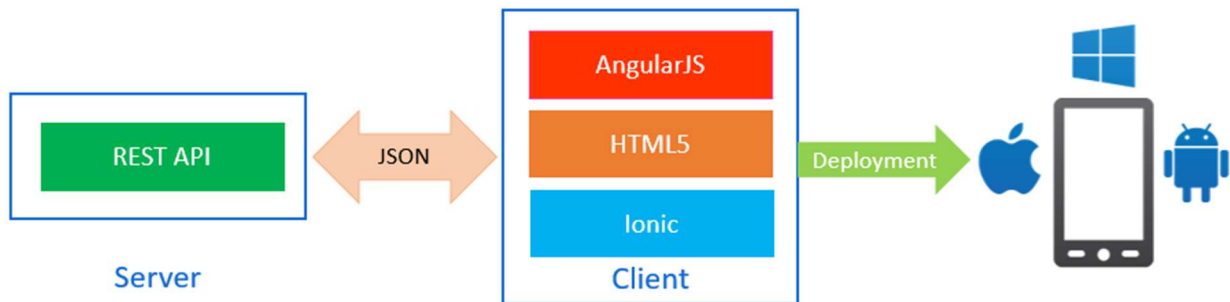


FIGURE 18 – MyPark App System Architecture representation – Image courtesy by dotnettricks.com

The server side will be all built over Firebase using their Database and service API's controlling user authentication, access to write and read data and security management.

The Client will receive all the application code developed using Ionic Framework, HTML5 and Angular JS.

The deployment of the app will target the most used operating system used by the target audience: IOS, Android and Windows Phone.

5 TECHNOLOGY REQUIREMENTS

5.1 SOURCE CODE MANAGEMENT

The tool chosen to help at the code source management and versioning was GitHub. The advantage of using it is the possibility for team member to implement the code locally and later share their implementation online, through GitHub. Also, it has some features to help developer in collaboration, such as task management, wikis, to create pull requests to merge codes and allow users to roll back to different points of the implemented code if necessary.

5.2 MOBILE DEVELOPMENT

To develop MyPark application using the Framework Ionic, it was decided for the GitHub's ATOM text editor.

Atom, GitHub's popular code editor allows programmers to access the semantics of various programming languages, integration with Git and GitHub itself. It has fully customizable themes and access to a community that is developing and building modules and extensions to it.

Atom is written in HTML and JavaScript and can easily have its interface customized using CSS.

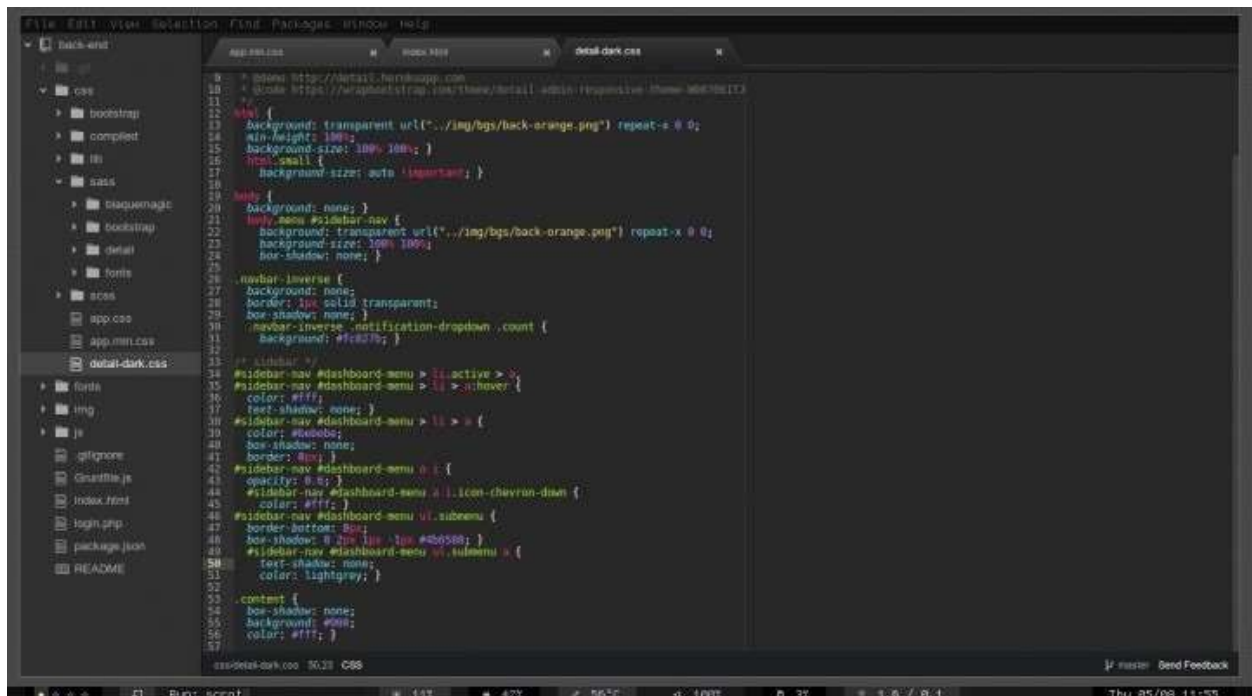


FIGURE 19 – GitHub's ATOM text editor view - Image courtesy by pplware

Other requirements involve installing on Atom the following packages and plugins:

SFTWARE	DESCRIPTION
NodeJS	Node.js is a platform for developing network-based server-side applications using JavaScript and the V8 JavaScript Engine, that is, with Node.js we can create a variety of Web applications using only JavaScript code
Ionic Framework + Cordova	Cordova is responsible for wrapping the application and offer access to native features of devices such as camera, GPS, accelerometer, among others. Ionic is a framework that takes care of the visual part of the application, how it is presented to the end user, creating interfaces very similar to native applications.

TABLE 9: Software Requirement for MyPark app implementation

5.3 HARDWARE

In order to run and test the application it will be used two devices, one Android and one iPhone, to visualize the application running in its target devices.

6 PROJECT PLANNING

6.1 LEARNING CURVE

This criterion aims to assess the time and cost required to learn how to develop an app using a particular approach, i.e. the lower the learning curve the better for those involved in application development.

Points that tell in favor of this are the documentation and support of the community. This criterion is most important when we are talking about beginners in a particular technology.

The learning curve for the web approach may be considered small since HTML, JavaScript and CSS are well-matured technologies and have a lot of documentation and support. In addition, the web developer can be taken advantage of having so many difficulties to adapt to the mobile world.

Technologies used for native development such as Android and IOS, also have a lot of documentation and support, but if you need to develop the app for more than one platform and the developer is not aware of any of them, this will be a little more work, to learn about all the platforms involved. Although the activities are distributed among other developers, although it reduces time, the cost will be affected.

In theory, the learning curve for the hybrid approach is very close to the web, since to develop hybrid applications HTML, JavaScript and CSS are used. However, you also need to be knowledgeable about packages apps, and in this regard, community documentation and support is not as strong as those of other approaches, although it also has a lot of documentation.

For this solution, the impact of the learning curve on the schedule is moderated. It will take two weeks of training and preparation to learn all the development details of the tools and the frameworks.

6.2 IMPLEMENTATION PLAN

The implementation of the system until its release will be split in four main phases: Training, Setup, Coding, and Testing.

6.2.1 Schedule

The schedule and the Gantt chart were created using the tool Microsoft Project.

Id	WBS	Tasks	Duration	Start	End	Resources
Id	WBS	Nome da tarefa	Duração	Início	Término	Nomes dos recursos
1	1	MySydneyPark - Implementation Phase	92 dias	Seg 19/03/18	Ter 19/06/18	
2	1.1	Training	17 dias	Seg 19/03/18	Qui 05/04/18	
3	1.1.1	Learn Ionic Framework	1 sem	Seg 19/03/18	Seg 26/03/18	Flavia, Cliverson
4	1.1.2	Learn AngularJS	1,43 sems	Seg 26/03/18	Qui 05/04/18	Flavia, Cliverson
5	1.2	Setup	2 dias	Qui 05/04/18	Sáb 07/04/18	
6	1.2.1	Install Atom Editor	1 dia	Qui 05/04/18	Sex 06/04/18	Flavia, Cliverson
7	1.2.2	Create MyPark project folder structure	1 dia	Qui 05/04/18	Sex 06/04/18	Flavia
8	1.2.3	Install NodeJS	1 dia	Qui 05/04/18	Sex 06/04/18	Flavia
9	1.2.4	Install Ionic/Cordova	1 dia	Qui 05/04/18	Sex 06/04/18	Flavia
10	1.2.5	Initiate Git version control	1 dia	Qui 05/04/18	Sex 06/04/18	Cliverson
11	1.2.6	Create a remote Github project and link to local Git	1 dia	Qui 05/04/18	Sex 06/04/18	Cliverson
12	1.2.7	Create a new Firebase Database	1 dia	Sex 06/04/18	Sáb 07/04/18	Flavia
13	1.2.8	Implement the Database	1 dia	Sex 06/04/18	Sáb 07/04/18	Cliverson
14	1.2.9	Configure the database in the project	1 dia	Sex 06/04/18	Sáb 07/04/18	Flavia
15	1.3	Coding	54 dias	Sáb 07/04/18	Qui 31/05/18	
16	1.3.1	FrontEnd	40 dias	Sáb 07/04/18	Qui 17/05/18	
17	1.3.1.1	Implement Welcome Page	2 dias	Sáb 07/04/18	Seg 09/04/18	Flavia
18	1.3.1.2	Implement Login Page	2 dias	Seg 09/04/18	Qua 11/04/18	Flavia
19	1.3.1.3	Implement Register Page	2 dias	Qua 11/04/18	Sex 13/04/18	Flavia
20	1.3.1.4	Implement Search Page	8 dias	Sex 13/04/18	Sáb 21/04/18	Flavia
21	1.3.1.5	Implement Search Result List Page	4 dias	Sáb 21/04/18	Qua 25/04/18	Flavia
22	1.3.1.6	Implement Search Result Map Page	4 dias	Qua 25/04/18	Dom 29/04/18	Flavia
23	1.3.1.7	Implement Park Detail page	6 dias	Dom 29/04/18	Sáb 05/05/18	Flavia
24	1.3.1.8	Implement Review Page	4 dias	Sáb 05/05/18	Qua 09/05/18	Flavia
25	1.3.1.9	Implement Add Review Page	2 dias	Qua 09/05/18	Sex 11/05/18	Flavia
26	1.3.1.10	Implement Share Page	2 dias	Sex 11/05/18	Dom 13/05/18	Flavia
27	1.3.1.11	Implement Profile Page	4 dias	Dom 13/05/18	Qui 17/05/18	Flavia
28	1.3.2	BackEnd	20 dias	Sáb 07/04/18	Sex 27/04/18	
29	1.3.2.1	Implement Login Functionality	2 dias	Sáb 07/04/18	Seg 09/04/18	Cliverson
30	1.3.2.2	Implement Registration Functionality	2 dias	Seg 09/04/18	Qua 11/04/18	Cliverson
31	1.3.2.3	Implement Search Functionality	3 dias	Qua 11/04/18	Sáb 14/04/18	Cliverson
32	1.3.2.4	Implement Map functionality	4 dias	Sáb 14/04/18	Qua 18/04/18	Cliverson
33	1.3.2.5	Implement Add Favorites functionality	2 dias	Qua 18/04/18	Sex 20/04/18	Cliverson
34	1.3.2.6	Implement Add Review functionality	2 dias	Sex 20/04/18	Dom 22/04/18	Cliverson
35	1.3.2.7	Implement Add Rating functionality	2 dias	Dom 22/04/18	Ter 24/04/18	Cliverson
36	1.3.2.8	Implement Share functionality	3 dias	Ter 24/04/18	Sex 27/04/18	Cliverson
37	1.3.3	Implement Integration	14 dias	Qui 17/05/18	Qui 31/05/18	
38	1.3.3.1	Implement Databind for Search Result List	4 dias	Qui 17/05/18	Seg 21/05/18	Cliverson
39	1.3.3.2	Implement Databind for Park Details	3 dias	Seg 21/05/18	Qui 24/05/18	Flavia
40	1.3.3.3	Implement Databind for Profile details	4 dias	Qui 24/05/18	Seg 28/05/18	Cliverson
41	1.3.3.4	Implement Databind for List of Reviews and Ratings	3 dias	Seg 28/05/18	Qui 31/05/18	Flavia
42	1.4	Testing	18 dias	Qui 31/05/18	Seg 18/06/18	
43	1.4.1	Execute First Round of Tests	5 dias	Qui 31/05/18	Ter 05/06/18	Flavia, Cliverson
44	1.4.2	Analyse Results and plan corrections	2 dias	Ter 05/06/18	Qui 07/06/18	Flavia, Cliverson
45	1.4.3	Implement the corrections	6 dias	Qui 07/06/18	Qua 13/06/18	Flavia, Cliverson
46	1.4.4	Execute Second Round of Tests	3 dias	Qua 13/06/18	Sáb 16/06/18	Flavia, Cliverson
47	1.4.5	Analyse results and fix minor bugs	2 dias	Sáb 16/06/18	Seg 18/06/18	Flavia, Cliverson
48	1.5	Launch	1 dia	Seg 18/06/18	Ter 19/06/18	
49	1.5.1	Deployment and Final Presentation	1 dia	Seg 18/06/18	Ter 19/06/18	Flavia, Cliverson

FIGURE 20 – MyPark App Implementation Phase Schedule

- dia(s) = day(s)
- Seg = Mon
- Ter = Tue
- Qua = Wed
- Qui = Thu
- Sex = Fri
- Sab = Sat
- Dom = Sun

6.2.2 Gantt

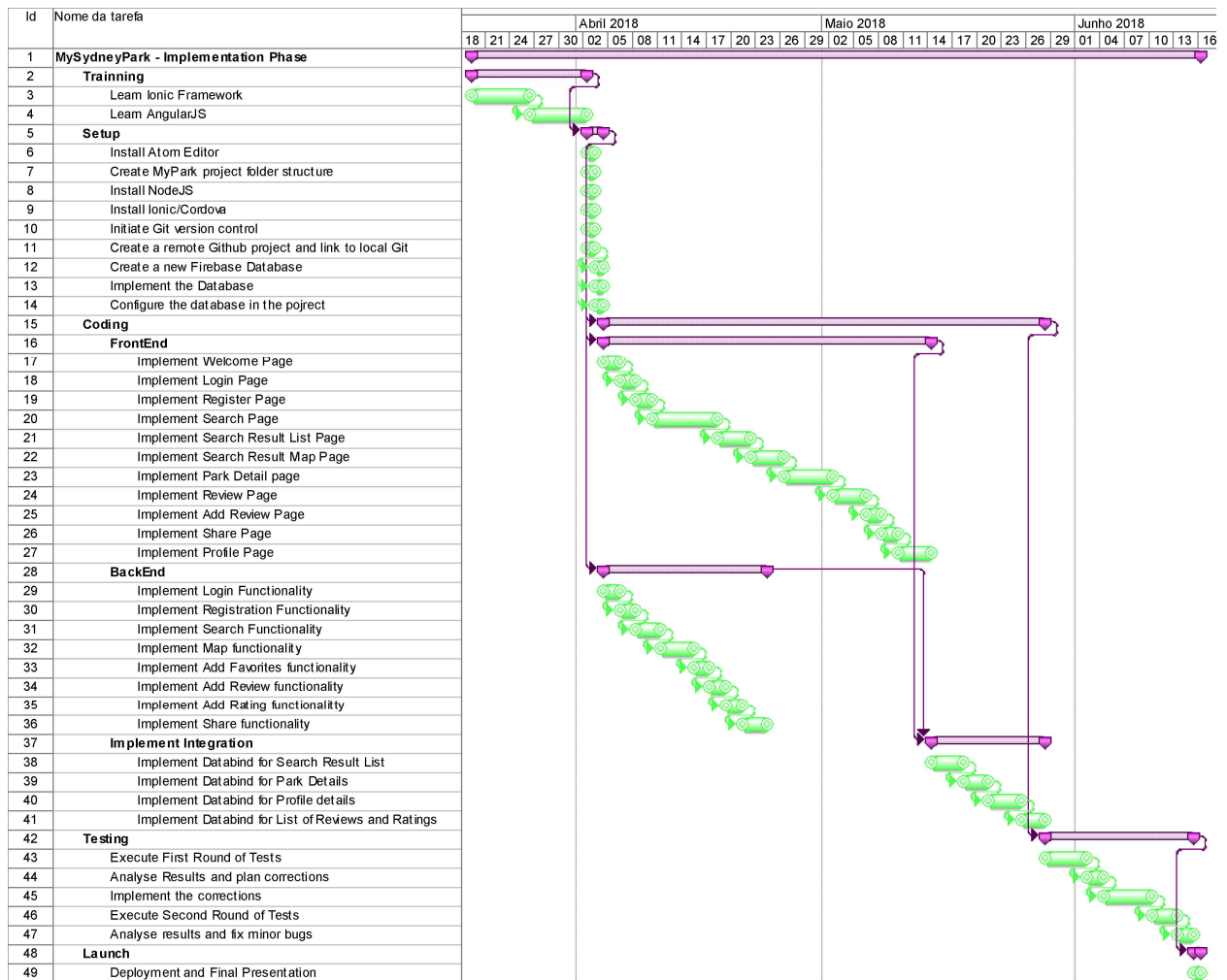


FIGURE 21 – MyPark App Project Gantt Chart

6.3 COMMUNICATION PLAN

The following table demonstrate the matrix of communication for team members during the implementation process.

PARTY	CHANNEL / TOOL	OBJECTIVE	FREQUENCY
Cliverson	Microsoft Project	Update the details about the project development progress.	Weekly
Cliverson, Flavia	Trello	Review the status of activities with the team	Daily
Cliverson, Flavia	Meeting	Face-to-Face meeting for decision making, discuss issues, and assess each other performance.	Weekly
Cliverson, Flavia	GitHub	Update the source code remote repository with the changes made in the project for versioning control	Daily
Cliverson, Flavia	Slack	Exchange of instant information, files, doubts, issues and other matters that need a quick approach	Daily

TABLE 10: MyPark app Project Plan - Matrix of Communication

6.4 CONTINGENCY PLAN

It is impossible to predict the future and be sure everything is going to happen as planned. The contingency plan approach tries to anticipate unexpected situations that could affect the project if it happens and that are not part of the project schedule. The proposed contingency plan is represented in the following table and shows what actions will be taken in case something happens during the development of this system.

RISK	LIKELIHOOD (HML)	IMPACT (HML)	INTERVENTION	PLAN	RESPONSIBLE
Misunderstood project requirements	M	H	Mitigate	Validation with the client about the requirements	Team involved with the project
Introduction of new technology	M	M	Mitigate	Training for the team on the new subject	Project Manager
Schedule out of reality	H	M	Mitigate	Have a base on timelines of previous projects	Team involved with the project
Wrong development of requirements	M	H	Avoid	Make a daily meeting with the members involved in the project	Team involved with the project
Inadequate definition of roles and responsibilities.	M	L	Avoid	Define and document the roles of each team member	Team involved with the project
Lack of establishment of project processes/ procedures/ methodology / planning.	M	L	Accept	Follow the development plan	Team involved with the project
Conflict and lack of cooperation constantly	M	L	Avoid	Maintain a level of maturity and dedication to the project.	Team involved with the project
Issues related to the development machine	L	L	Mitigate	Use AIT's computers to keep working on the project since the code is keep it online.	Team involved with the project

TABLE 11: MyPark app Project Plan – Risks x Contingency

Summary of likelihood or probability that was mentioned at the Contingency plan.

SCALE (HML)	DESCRIPTION	COST	SCHEDULE
L	Low	< 5% Increase in budget	No change to schedule
M	Moderate	5-10% Increase in budget	1-2 weeks delay to schedule
H	High	10-20% Increase in budget	2-4 weeks delay to schedule

TABLE 12: Metrics for Probability and Impact of Risks

6.5 TEST PLAN

The test plan has the objective to certify that the application was built following the proposed specification, the functionalities are working as expected and the application is solving the problems identified at the research.

The list below identifies and prioritize the features to be tested.

ITEM	PRIORITY
Login	High
Register	High
Search by Categories	High
View the list of Parks	High
Park Details	Medium
Add to Favourites	Medium
Add a Review	Medium
Share	Medium
Get direction	Medium
List View on the Map	Medium
Search by Name or Postcode	High

Check Favourites	Medium
Settings Contact to developer	Medium
Allow access location	High
Check distance by GPS	Medium

TABLE 13: MyPark app Test Plan – Test Priorities

The types of tests showed in the Table below will be executed in order achieve the quality expected.

TEST TYPE	OBJECTIVE
Usability Testing	Ensure that the application is easy to use and provides a satisfying user experience for customers.
Compatibility test	Test the application on different mobile devices, browsers, screen sizes and Operating System versions, according to requirements.
Interface Testing	Test the menu options, buttons, bookmarks, history, settings, and application navigation flow.
Service Testing	Test online and offline application services.
Testing Resources	Test memory usage, automatic deletion of temporary files, local database development issues, etc.
Performance Testing	Test the performance of the application by changing the connection from 3G to 4G, 4G to WIFI, sharing documents, battery consumption, etc.
Operational Testing	Test backups and recovery plans (if the battery is dead, or data loss occurs while upgrading the application).
Installation Testing	Test the application by installing / uninstalling it on different device models
Security Testing	Test the application to validate whether the system protects the data or not.

TABLE 14: MyPark app Test Plan – Types of Tests to be performed

7 CONCLUSION

After a long research and planning we hope to have covered all details need to convert this propose in a real-life solution.

Since the beginning of this project our goal was not only create this application as developers, but also to bring joy and increase the quality of life of people that uses this system.

We are looking forward to receiving the approval and start implementing MyPark app.

8 REFERENCES

- InfoWorld 2017, Wayne Carter, viewed 26 September 2017,
<<https://www.infoworld.com/article/2887754/mobile-development/how-to-choose-a-database-for-your-mobile-apps.html>>
- Priyanka Varma 2016, viewed 26 September 2017,
<<https://www.quora.com/What-is-firebase>>
- TechTarget 2017, viewed 26 September 2017,
<<http://searchmicroservices.techtarget.com/definition/Web-services-application-services>>
- Cordova Apache Documentation 2017, viewed 27 September 2017,
<<https://cordova.apache.org/docs/en/latest/guide/overview/index.html>>
- Ritesh Patil 2016, viewed 05 September 2017,
<<https://www.infoq.com/articles/mobile-cross-platform-app-development>>
- Mobile Consumer Survey 2016, Deloitte Australia Technology, Media & Telecommunications, viewed 05 September 2017, report
<<http://landing.deloitte.com.au/rs/761-IBL-328/images/deloitte-au-tmt-mobile-consumer-survey-2015-291015.pdf>>
- Vensi, Inc. 2017, Annie, viewed 07 September 2017,
<<http://www.vensi.com/benefits-of-phonegap-cross-platform-mobile-app-development/>>
- TVS Next 2017, Jegan, viewed 07 September 2017,
<<http://tvsnnext.io/blog/advantages-of-ionic-framework-app-development/>>
- Census data 2016, City of Sydney, viewed 08 September 2017,
<http://profile.id.com.au/sydney/language>
- The PowToon Examiner - PowToon Blog 2017, viewed 03 September 2017,
<<https://www.powtoon.com/blog/>>
- Playground Finder 2017 – viewed 28 August 2017,
<www.playgroundfinder.com/about.pp>
- SurveyMonkey (2017), viewed 10 September 2017,
<<https://pt.surveymonkey.com/home/>>
- Zein Salleh, & Grundy J. 2016, A Systematic Mapping Study of Mobile Application Testing Techniques. The Journal of Systems and Software, p. 334-356.

APPENDIX A – SURVEY QUESTIONNAIRE

1. How old are you?

- ☐ 10-18
- ☐ 19-25
- ☐ 26-35
- ☐ 36-45
- ☐ 46-55

2. What is your gender?

- ☐ Female
- ☐ Male

3. Why do you go to the Park? (Check all that apply.)

- ☐ Biking
- ☐ Bring Children
- ☐ Event
- ☐ Sightseeing
- ☐ Skating
- ☐ Time with family and/ or friends
- ☐ Exercise or Jog
- ☐ Play Sports
- ☐ Walk Dog
- ☐ Have a barbecue with friends
- ☐ Have a picnic
- ☐ Rest lunch
- ☐ Walking

4. How often do you visit the public parks in Sydney?

- ☐ Extremely often
- ☐ Very often
- ☐ Somewhat often
- ☐ Not so often
- ☐ Not at all often

5. Which type of apps do you currently have on your digital devices (computer, tablets, phones, etc.)? (Check all that apply)

- ☐ Game apps (puzzles, charades, etc.)
- ☐ Sports apps (sports schedules, scores, headlines, etc.)
- ☐ Social networking apps (location check-ins, friend status updates, etc.)
- ☐ Travel apps (airplane tickets, tourist guides, public transportation info, etc.)
- ☐ Search tool apps (directions, phone numbers, recipes, etc.)
- ☐ Entertainment apps (movie trailers, celebrity gossip, radio station guides, etc.)
- ☐ News apps (local news, national headlines, technology announcements, etc.)
- ☐ Productivity apps (calendar, to do list, price checker, etc.)
- ☐ Utility apps (calculate, convert, translate, etc.)
- ☐ Weather apps (local forecasts, natural disaster updates, etc.)
- ☐ Other (please specify)

6. If any, what are the ages of the children in your household? (Check all that apply)

- ☐ 0 to 5 years
- ☐ 6 to 10 years
- ☐ 11 to 15 years
- ☐ 16 to 18 years

7. What is the distance that you spend to go to the park?

8. Where do you find information from?

- ☐ Newspaper
- ☐ Website
- ☐ Google
- ☐ Social media
- ☐ Other
- ☐ Other (please specify)

9. What is your nationality?

10. What do you think an application should have? (It will help you find Parks in Sydney)