Parking goes Mobile

A Research Proposal

Abstract—**This paper presents a research proposal to study the process to create a mobile application considering important restrictions on the development schedule and on the team size. The objective of the application is to help citizens to find parking spots in urban context. A theoretical framework is presented, based on several dimensions, such as the software development approach, the different mobile application's types and its implications, the specific design issues for mobile applications, the technological frameworks that could be used, and an analysis of the existent applications that have similar objectives. The research proposal is described: its objectives, the research question, and a discussion on technical options related with the dimensions of the theoretical framework. The following steps of the project are outlined.**

Keywords-parking, smartcities, mobile device, app, cross-platform framework, small team, development approach, usability, context, efficiency, management

I. INTRODUCTION

Urban parking is a common problem among cities throughout the world. Every day thousands of people leave their homes in their vehicles intending to park in a free zone or very close to the location they intend to visit. But finding a parking spot in a busy city is usually difficult, forcing drivers to wander around, slowing down whenever they see a possible parking spot. This can affect traffic and may even cause road accidents.

Our cities should be prepared and helpful to park more easily, allowing citizens to have a less stressful life and spend more quality time doing more productive activities. This idea was analyzed by several authors leading to the concept of smart cities, where citizens and objects of the city are transparently connected by the ubiquitous technologies in order to enhance the experience of life in urban environments. This ecosystem of smart cities is composed by Smart Utilities, Smart Transport, Smart Building and Smart Government [1].

Regarding urban parking, "cities large and small have learned that smarter parking management can be a valuable tool for revitalizing downtown districts, improving the costumer experience, and reducing both traffic and park congestion" [2]. The city of San Francisco, California, is a good example of smart parking. There, sensors connected via wireless are used to transmit information in real time about free parking spots, either through text message, online maps, dynamic parking signs, or through the mobile app developed for that purpose.

Although the concept of smart cities stimulates the appearance of new and interesting ideas about the future, the infrastructure and resources necessary to implement them are an important restriction to many cities. This study takes this into account and it aims to understand how to give citizens proper help in their parking process, so they spend less time and fuel, using only the already available resources in the city, namely the mobile phone network. The city of Aveiro, in Portugal, is the implementation ground for this project, whose main output is an app for drivers to use on their smartphones in their search for a parking spot. This project has two other restrictions, or limitations: it is designed and implemented by a very small team and in a very short period of time.

II. THEORETICAL FRAMEWORK

This section describes the theoretical framework that helps to make the necessary decisions throughout the design and implementation process, which are discussed during the presentation of the research proposal (section III).

The first theoretical issue is about the software development methodology approach to follow in order to get the app implemented in a short period of time. Then, different types of mobile apps' development strategies were examined. Finally, different components of mobile design and some cross-platform frameworks were also studied.

A. Software Development Approaches

When developing a product, it's important to follow a strategy that will keep the project within deadlines and prevent it from spending more money or resources than initially planned. Over time, some methodological approaches were proposed in order to help organize development teams, as well as promote communication between team members, clients and all stakeholders of a project.

The traditional approach, usually known as the waterfall approach [3], applies to projects that have well defined requirements that do not change over time. This model has sequential phases (analysis, design, coding and test) that should be followed in a strict order, and each phase can only begin when the previous one is finished. All these phases should be well documented for further reference. Likewise, there is a strict division of work within members of the development team, and each one usually have knowledge about a specific phase or technique. A functional version of the product can only be released once all phases are completed and, if it happens to detect a major error or to have an important delay during the development process, the whole project could be compromised in terms of time and resources [4].

On the other hand, if fast and multiple releases are needed, an agile method is recommended, because it is incremental and iterative. Each iteration is expected to be done in a short period of time, and includes analysis, design, coding and testing. At the end of each iteration, a functional release is presented to the client for approval [3]. The main system requirements are usually defined at an early stage, but it is possible to change them during the development process. However the advantages of this approach, it is not usually recommended to large teams or long term projects. Also, it's very difficult to manage and control costs, because requirement's changes during the

processed are allowed [3] [5]. Extreme Programming (XP), Scrum and Crystal are examples of agile methods.

B. Web, Native and Hybrid Apps

There're three general types of app, namely web, native, and hybrid app. The choice for the most appropriate to a project depends on the goals of the mobile app, its features, the time available for its development, among other factors.

A web app is implemented with the aid of three well established web technologies - HTML, CSS, and JavaScript [6]. A major advantage of this type of app is the compatibility between different platforms, content is accessible by any browser and by any mobile device with only one software version, although the end result may differ between different devices due to screen size and its resolution. In addition to being relatively simple and quick to develop, they are also easier and cheaper to maintain, since it is only necessary to implement the required changes in a single version [7]. Unless HTML5 is used, it's not possible to access the specific device's hardware and software, such as camera, GPS, Bluetooth, among others. Users can also experience some performance issues if the internet connection is weak or nonexistent [6].

Native app is an alternative to web apps. They are downloaded from the app stores, and subsequently installed on the user's device, allowing offline access. When developing a native app, it has to be considered which is the target operating system, since it's necessary to develop a new version in a new language for each of them – C and Objective C for iOS , C# and C++ for Windows or Java for Android and Blackberry, for instance [8]. To assist the development process, these companies have been releasing SDK's (software development kit) and API's (application programming interfaces) , including the necessary documentation to better understand the operation system in question, emulators that simulate the operating system, libraries, tutorials, debuggers and other tools. As it's required a different software version for each operating system, native applications become expensive to maintain and update, since each version has to be modified, tested and distributed [7]. One of the great advantages of this type of application is the ability to use the device components, creating a rich, compelling and immersive user experience.

Hybrid applications are the third type, which combines these two approaches already described. To the user, a hybrid application is much like a native app. However, the time and cost of development is significantly lower compared to the cost of native apps, since developers code once using web technologies and reuse the code on different devices [6], using a cross-platform framework, instead of rewriting the application for each operating system.

C. Mobile Design

The first contact users have with a mobile app is visual, and that determines whether they spend a minute or an hour using it. When a user seek for an app in an app store, besides reading other users comments, they also pay attention to the app's screenshots and even the icon's quality to decide whether they buy it or not. Thus, developing a quality design for a mobile app it's of critical importance [7].

Therefore, different constraints of mobile devices have to be considered before starting the design process, such as their small screen size (everything that takes too much space on the screen, like extensive menus and lists, large images, publicity among others, must be removed), specific use context (mobile devices are used under any circumstance and everywhere), and their portability (users move frequently when they're using their mobile phones, so different brightness has to be considered as well as some difficulties that may exist when they're typing or reading) [9].

Usability guidelines must also be considered, as they are "(…) a necessary condition for survival" [10]. If a user feels lost or confused operating an app, or even if he finds the app functionalities too hard to learn, he'll soon give up from using it. These frustration feelings affect the user's opinion about the application. Then, the quality of the user's experience have to be considered in order to get a positive user's impact and adoption [11].

All of these issues should be thought considering the end-user in order to develop an application that meets all of their needs.

D. Cross-platform frameworks

Cross-platforms frameworks are used to decrease development's time "(…) by achieving the 'write-once-deploy-everywhere' solution" [12]. These frameworks compile a web app (which is faster to implement) into a native or hybrid app, enabling developers to code in web technologies such as HTML, CSS and JavaScript with some native code to access device's components.

Several different frameworks of this kind have been developed and distributed, such as Sencha Touch, Apache Flex, Application Craft, Nimble Kit and many others. For this study, it were analyzed some of the best known frameworks, like jQuery Mobile, PhoneGap, Titanium and Rhodes, taking into account the project limitations mentioned above.

jQuery Mobile is a graphical user interface (UI) framework for mobile devices that allows developers to build web apps in an easy, quick and affordable way. Due to the simplicity of its structure, it is possible to develop pages using only HTML markups that are automatically styled by jQuery Mobile code that adds CSS elements and JavaScript features. This framework was designed to be as accessible as possible, including WAI-ARIA (Accessible Rich Internet Applications) attributes in order to make pages accessible to people with disabilities, or to all individuals who need some kind of help in a particular context [13].

PhoneGap is an open-source framework that developers use to build hybrid apps based on web technologies. It was downloaded over a million times, and used by over 400 million developers [14]. This framework allows developers to build their mobile apps in web technologies and compile them into a hybrid app. The compiled app is rendered through the web browser engine and it permits access to the device's components [15] [16]. PhoneGap should not be used in apps that require large mathematical calculations, 3D animations, or for data-driven applications [17].

Titanium is another framework that allows building mobile apps using web technologies, providing tools and support for those who want to use PHP, Ruby and Python [15]. It also has some JavaScript API's and libraries that allow access to UI components and device's native functionalities. To this end, developers code in JavaScript which is then compiled into

bytecode by Titanium, so that the desired platform's SDK compile into a native app, whether to test it in a native simulator of for final distribution [17].

The final studied framework was Rhodes, which by the end of 2012, was downloaded over 100,000 times and hundreds of mobile applications were created with this tool and put available in the app stores [15]. To build an app with this framework, developers can use Rhodes' IDE called RhoStudio, which provides tutorials, templates, emulators and debuggers, ou any other IDE that supports HTML, CSS, JavaScript and Ruby (Eclipse, Visual Studio, Netbeans, IntelliJ and TextMate are the best known IDE's) [16]. Mobile apps developed with Rhodes are compiled into Java bytecode to run on BlackBerry or any Java based operating system, and compiled into Ruby 1.9 bytecode for all other operating systems, being the end result a native app [17].

E. Existent Apps

In order to understand the user's needs, in terms of functionalities required and design interaction issues, it is important to examine other available mobile apps that pursue similar objectives.

Parking Karma [18] is a mobile app that helps users find parking spots based on information shared by other users. It also alerts the user before the parking meter expires as well as gives information about traffic incidents so users can avoid them. Park me Right [19] and ParkMe [20] were analyzed as well, and they allow users to find parking spots based on their present location or a specific address. They also save the location of the parked car so that users won't forget it. Parkopedia [21], besides doing all the functionalities mentioned above, provides parking spot's information in real time, allowing users to know where they can and can't park at the moment.

All these apps have clean and intuitive design, giving emphasis to well-design maps in contrasted colors, so that users can access the main features easily.

III. RESEARCH PROPOSAL

According to [22], an investigation should follow seven steps: starting question, exploration, problematic, construction of the analysis model, observation, information analysis and conclusion. This study will be conducted taking some of these steps into account.

This section describes the next phases to the study once the problematic was identified and the theoretical framework was analyzed. It will be pointed out the objectives of this study, the research question, the technical options that were made as well as the future tasks that will be done until a conclusion emerges.

A. Objectives and Research Question

As it was mentioned before, the context of this study is the development of a mobile app that helps drivers in Aveiro finding parking spots. The app has to be intuitive and easy to use in order to prevent distracting the driver from its main task, driving.

Since the beginning of this study, some restrictions to the project were known, such as being developed by an extremely small team and in a short period of time. This last restriction is increasingly common due to the market competition, forcing developers to build their projects as quick as possible in order to timely release innovative products. So, this study aims to understand how to development a mobile app taking into account a very specific context of use, a small development team and a short period of time for development, which leads to the following research question: *How to create mobile apps, considering small teams, a short period of time for development, and a specific context of urban parking?*

In addition to the development of this app, this study focuses in the research that precedes the development process itself, like the methodological approach that should be followed, the different types of apps that may be created, the mobile design components, the cross-platform frameworks, and the analysis of existent similar apps. Therefore, the objectives of this project are:

- To develop a mobile app that assists citizen in urban parking;
- To understand the design guidelines that should be used in this particular context;
- To understand which methods and tools that can be used in order to decrease the time spent in the app's development;
- To understand which software development methodological approaches can be used to have an effective development in a short period of time.

B. Technical Options

In order to follow the best path for an effective and efficient development process, options have to be made considering the topics described in the theoretical framework (section I).

Regarding the software development methodological approaches, agile methods and the waterfall approaches were analyzed. The later requires detailed documentation in all phases, including the development process itself. Furthermore, system requirements are defined at an early stage of the project, being rather difficult to modify them in the subsequent stages. In this project, it will be difficult to define all the application requirements at an early moment of the process, as new ideas and new knowledge are always emerging. Also, due to lack of time, the development process cannot be stopped to restart from the beginning if a major error is detected. Therefore, a traditional method is not suitable for this project. Agile methods, on the other hand, emphasize constant communication between the development team and the client, which allows updating system requirements at any time. So, on this project an agile method approach will be followed, as it allows changing and improving requirements during the process. Although none of the methods mentioned before can be directly applied, some of its practices can be used and combined into an adapted custom method, which will emerge along the development of this project.

Another decision that has to be made at an early stage is the type of app that will be developed. To this project, the time constraint is an important issue, so the app has to be easy to implement without the need to study new programming languages. Due to the specific context of use of this application, there are certain functionalities that require the use of specific device's components. So, in order to reduce the necessary development time, the choice is to create a hybrid app. Consequently, the app will be potentially available on different platforms from just one coded version, and native components could be accessed. A cross-platform framework

will be used. PhoneGap is the framework chosen because it uses web technologies and it provides a free user license agreement.

Regarding the design components, the app will have to be intuitive and easy to use, as users won't have much time to check information. As it can be used while users are driving, the app has to be as simple as possible, without any unnecessary functionalities that could distract the user from driving, and providing proper information with the least possible effort. Due to device's small screen size and the eventual use during movement, it will be used a sans-serif typo with a medium to large size, to help the user to consult information quickly.

C. Next Steps

As this is a research and development project, focus will be on mobile apps' development procedures rather than the existent products themselves. After a detailed analysis of the components mentioned before, the system requirements will be thought taking into account the project restrictions (a small development team, a very specific context of use, and a short period of time for development) and the needs of the users.

In order to better understand those needs, a survey will be done to question Aveiro drivers to uncover their opinion on the reasons that eventually lead to spend more time that it would be desired in finding parking spots, to get their opinion on the actions that can be done to overcome these problems, and to know which functionalities they consider more relevant in a mobile app that helps them finding parking spots. The survey results, along with the knowledge about other similar applications, are expected to be important contributions to the system requirements definition.

When most of the system requirements and functionalities are reasonably defined, the design process of certain features will be started followed by its development. Once a functional version is finished, it will be tested internally and approved by the stakeholders involved in this project. Then, necessary corrections will be done and it will be started the next phase of design and development of other functionalities. This will be done in small iterations until a stable prototype of the app is ready. Then, it will be conducted usability tests, in order to understand if the system is as intuitive and simple as it should be. As this app will be used before or, more likely, while driving, it is impossible to test it in its real context of use in a safe way. So, this scenario will be simulated using a racing game and a game driving chair, so that users can test the app within a safe environment. After analysis of these tests, all errors that were identified will be fixed in order to release a beta version of this app.

IV. Final Remarks

This study aims to contribute to the resolution of the urban parking problem in Aveiro, without spending money implementing new components to the city, and considering the specific constraints to the project (small team, short period of time for development, and a very specific context of use). In addition to the final product, it is important to understand some design guidelines for this specific context, as well as the software development methodologies and tools that can help to create the application in a fast and effective development process. The expected outputs of the project are a functional version of the app, as well as learning outcomes about the design and the development issues of developing a mobile app considering the restrictions mentioned. Finally, lessons learned from the methodological approach followed will be systematized and shared.

References

[1] M. Dohler, I. Vilajosana, X. Vilajosana, and J. Llosa, "Smart Cities: An Action Plan," Smart City Expo World Congr., pp. 1–6, 2011.

[2] B. J. Nelson and J. Schrieber, "Smart Parking Revisited," Planning, vol. 78, no. 5, pp. 25–29, 2012.

[3] M. S. Palmquist, M. A. Lapham, S. Miller, T. Chick, and I. Ozkaya, "Parallel Worlds: Agile and Waterfall Differences and Similarities," no. October, 2013.

[4] Ruparelia, N. B. (2010). Software development lifecycle models. *ACM SIGSOFT Software Engineering Notes*, *35*(3), 8.

[5] M. R. S. Tomás, "Métodos ágeis: características, pontos fortes e fracos e possibilidade de aplicação," 2009.

[6] Lionbridge, "Mobile Web Apps vs . Mobile Native Apps: How to Make the Right Choice," 2012.

[7] B. Fling, Mobile Design and Development, First edit. 2009.

[8] D. Seven, "What is a Hybrid Mobile App?," 2012. [Online]. Available: http://www.icenium.com/blog/icenium-team-blog/2012/06/14/what-is-a-hybrid-mobile-app-. [Accessed: 20-Nov-2013].

[9] J. Tidwell, Designing Interfaces, Second Edi. 2010, pp. 441 – 476.

[10] J. Nielsen, "Usability 101: Introduction to Usability," 2012. [Online]. Available: http://www.nngroup.com/articles/usability-101-introduction-to-usability/. [Accessed: 26-Dec-2013].

[11] M. Gibson, "UX:Psychology of great design," Web Designer Issue212, pp. 42–47, 2013.

[12] R. N. Sansour, N. Kafri, and M. N. Sabha, "A Survey on Mobile Multimedia Application Development," 2013.

[13] M. Doyle, Master Mobile Web Apps with jQuery Mobile Third Edition. 2012.

[14] "PhoneGap | About," 2013. [Online]. Available: http://phonegap.com/about/. [Accessed: 26-Nov-2013].

[15] A. Ribeiro and A. R. da Silva, "Survey on Cross-Platforms and Languages for Mobile Apps," 2012 Eighth Int. Conf. Qual. Inf. Commun. Technol., pp. 255–260, Sep. 2012.

[16] M. Palmieri, I. Singh, and A. Cicchetti, "Comparison of cross-platform mobile development tools," 2012 16th Int. Conf. Intell. Next Gener. Networks, pp. 179–186, Oct. 2012.

[17] S. Allen, V. Graupera, and L. Lundrigan, Pro Smartphone Cross-Platform Development. Berkeley, CA: Apress, 2010.

[18] Parking Karma, "Parking Karma | The social app that will help you finding free parking spots and much more!," 2013. [Online]. Available: http://uwannabee.com/. [Accessed: 14-Feb-2014].

[19] SurfStudio, "Park Me Right Android App: AR Car Locator & Parking Search App." [Online]. Available: http://parkmeright.com/. [Accessed: 14-Feb-2014].

[20] ParkMe Inc, "ParkMe | The Best Way to Find & Compare Cheap Parking Rates." [Online]. Available: http://www.parkme.com/. [Accessed: 14-Feb-2014].

[21] Parkopedia, "Parkopedia - Helping you park." [Online]. Available: http://en.parkopedia.com/. [Accessed: 14-Feb-2014].

[22] R. Quivy and L. Van Campenhoudt, Manual de Investigação em Ciências Sociais, 4a ed. 2005.