

MASARYK UNIVERSITY
FACULTY OF INFORMATICS



Cross-platform System for Time Scheduling

BACHELOR'S THESIS

Jan Tomášek

Brno, Fall 2015

MASARYK UNIVERSITY
FACULTY OF INFORMATICS



Cross-platform System for Time Scheduling

BACHELOR'S THESIS

Jan Tomášek

Brno, Fall 2015

Replace this page with a copy of the official signed thesis assignment and the copy of the Statement of an Author.

Declaration

Hereby I declare that this paper is my original authorial work, which I have worked out on my own. All sources, references, and literature used or excerpted during elaboration of this work are properly cited and listed in complete reference to the due source.

Jan Tomášek

Advisor: Ing. Libor Kubečka

Acknowledgement

Abstract

Keywords

Contents

1	Introduction	1
2	Time management	3
2.1	<i>Four generations of time management</i>	3
2.2	<i>Current applications</i>	4
3	Cross-platform application development	7
3.1	<i>Web applications</i>	7
3.2	<i>Native mobile applications</i>	9
3.3	<i>Cross-platform mobile frameworks</i>	10
4	Software Analysis	13
5	Back-end Implementation	15
6	Front-end Implementation	17
7	Testing an Publishing	19
8	Conclusion	21

1 Introduction

Time management helps people to be more productive and effective. Calendars, diaries, lists with notes, all of these are well-known organizers for time management. Less popular is the organizer for the fourth generation of time management, described by Stephen Covey¹, which differs from the others by its philosophy and construction.

From those who use some kind of time organizer, some still use classic paper organizers, but due to the evolution of computing technologies, more and more people use their devices to manage time electronically. Mobile phones are ideal devices for this purpose, because in time management, it is important to have an opportunity to modify a time plan whenever it is needed. However, the control of mobile phones is slower and the display size is smaller, compared to devices like PCs. Therefore, popular time management applications are cross-platform. There are many cross-platform applications for common organizers, but only few popular applications for Covey's organizer.

This thesis deals with the whole process of development of a cross-platform application tailored especially for the Covey's organizer. First part includes the summary of popular time management applications and the analysis of current possibilities of cross-platform development. Second part includes the system design, the implementation of user-friendly, attractive, cross-platform application, testing and publishing.

The final application, named Liferoles, meets all requirements and is capable to compete concurrent products. Application is cross-platform and allows user to access his data from more devices without loss of consistency. To achieve this, users' data are stored remotely on database server. User-friendliness is reached by splitting the application into two branches – mobile application² for mobile devices with Android or iOS and web application³ for devices like PCs or mobile devices with other OS. Both applications offer similar looks, but the layout and controls are customized according to the platform used.

1. Well-known leadership instructor and author of bestseller *The 7 habits of highly effective people*.

2. Available for free at Google Play and App Store e-shops.

3. Available at: <https://www.zatimjestenevimkdetobude.cz/>.

1. INTRODUCTION

In the next chapter time management and its four generations are described and popular applications are mentioned. After that the problematics of cross-platform development and current approaches used for cross-platform development are examined. The system design and the description of the back-end development follows. After that the front-end development, including GUI, authentication and description of both mobile and web application is described, followed by information about testing and publishing.

2 Time management

Stephen Covey divided time management evolution into four generations [1]. Each generation improves the previous one with additional points of interest.

2.1 Four generations of time management

The first generation used a simple list of tasks and notes which helped people to remember what to focus on and also gave them feeling of some kind of order which they should follow.

The second generation started to plan the future. People started to link their tasks with dates and record them into calendars and diaries. This was important because, apart of that, people were able to use time management for reaching long-term goals.

The third generation of planning is about priorities. People set their long-term or short-term goals and try to move towards them. In their daily planning they give priority to their tasks, which should lead them to their goals, sort them according to priority and presumed time spent, and then try to accomplish them in most efficient way. This is the most known time management approach nowadays.

In the fourth generation Covey divided tasks into four quadrants [1] according to the importance and urgency:

1. important, urgent
2. important, not urgent
3. not important, urgent
4. not important, not urgent

It is important to focus on tasks from the second quadrant to prevent crisis tasks from the first quadrant. A necessary part of the fourth generation time management is the determination of roles which people play in their life – each person can have many of these e.g. a father, a husband, a teacher, a researcher, an athlete. Ideally, each planned task should be linked with some role. Time plan should be built for one week because, with the week planning, people have control

2. TIME MANAGEMENT

about how much time is dedicated to each role. The whole planning system should be flexible so that people can manage their tasks and change their priorities and time schedules. Another important thing is a personal mission, which should be written down in the system, so that people can read it, update it and motivate themselves.

2.2 Current applications

Fourth generation time management is not as popular as the previous ones, so as a result of this, most applications for time management don't support the above mentioned role-based time management. The main features of popular applications are calendars, to-do lists, notes, priority ordering and reminders.

My Effectiveness

My Effectiveness application enables users to record every feature of time management described by Franklin Covey. Users can write down their personal mission, roles and goals which should be achieved, actions and more. Action can be of three types – action, project, checklist. Each action must have assigned role and one of four quadrants and could have assigned priority, role goal etc. All these features were mentioned in Covey's book but it is not necessary to have them all in a time management application. The effort of maintaining much metadata made the application unnecessarily complicated and, therefore, not user-friendly. Another drawback is that the application is available only on Android devices.

ZZ Tasks

ZZ Tasks¹ is a cross-platform application which can be installed on devices with iOS or Android and it is also accessible at: <https://zztasks.com>. It supports the necessary parts for fourth generation time management – concept of roles linked with tasks, priority ordering, personal mission. Attractive feature is opportunity to synchronize tasks with Google apps or Microsoft Exchange accounts although

1. Formerly known as FranklinCovey Tasks.

this was not working when tested. It also supports many unnecessary features (reminder, task repetition, task delegation, customization), however, creation of tasks is fast and controls are simple so the application does not look confusing as the previous one. The application offers 30 day trial after which the support of history, synchronization and even access to web application are charged. Another drawback is that the mobile application doesn't use any specific mobile controls like side menus or drag and drop, which are appreciated by users. Everything is handled by buttons, forms and dialog boxes.

To compete current products, final application of this thesis is cross-platform and free. The functionality is simple and controls are intuitive. To make the application different, and therefore attractive, it displays graphs and statistics of tasks and roles, which should have motivational effect. Unlike other applications, it is very tied with planing system described in Covey's book.

3 Cross-platform application development

Cross-platform application should be independent on operating system (Windows, Linux, Android, iOS), type of device (smartphone, tablet, PC) and screen size. It is common that the final product of cross-platform development is an application which has different versions for different platforms. It is due to the fact that differences between device types and screen sizes are perceptible. The technologies which are widely used to achieve the goal are described below.

3.1 Web applications

Web applications are programs which allow users to communicate with the internet, where the content is stored, using their web browser. From the technical point of view, a web application is a dynamic website. Its structure and content is written in HTML (HyperText Markup Language) and look is described by CSS (Cascading Style Sheets) files. To make a website dynamic it is necessary to use some scripting language which handles events generated when users interact with their web browsers. Scripting languages can be server-side or client-side. The difference between these two is that server-side scripts runs on the server, whereas client-side scripts runs on the client side - in the web browser.

Web applications run on remote servers and in web browsers, which are contained in every OS (operating system), so every device with OS can use web application without installation. They are portable just by the share of a link and people can find them on the internet by using their web searchers. Developers can update websites instantly straight on servers, which is much more comfortable than maintaining updates to applications installed on clients' devices.

All types of web applications have drawbacks of performance and usability of platform specific controls compared to other technologies for cross-platform development. Another disadvantage is that the internet connection is needed to communicate with remote server. To suppress this disadvantage developers use web caching¹ and

1. A technology for storing web server responses locally.

3. CROSS-PLATFORM APPLICATION DEVELOPMENT

JavaScript language for client-side scripting. This approach minimize communication between client and server.

In the past, websites were developed mainly for desktops and other devices with large screens. Nowadays, due to powerful hardware and software, mobile phones enable people to browse websites and the percentage of page views from mobile phones is constantly growing. In May 2012 it was 10.11 % [2]. But there is still one big difference between desktops and mobile phones and that is screen size. Websites which are not optimized for mobile phones can still be viewed by users with zooming, but it is uncomfortable. Consequently, developers try to make both websites and web applications more user friendly with two approaches - responsive websites and mobile websites.

Responsive website

Responsive website adapts its layout according to platform used by visitor. Screen size is the key attribute in responsive website design. The layout viewed is wide and full of information for larger screens, while for smaller screens, more information is hidden under website navigation.

The power of responsive websites is the "one fits all" principle, which means that there is only one version of web application, with one source code, which works well on all platforms. Each device with internet browser and internet connection can use responsive website as an application. As the user connects, the layout and controls are set according to platform used. This approach is mostly used for informational websites to enable user to find information quickly from any platform. For more complex applications, full of controls and interactivity, this is not the best option, since it can't use full potential of the concrete platform. It takes more time to create good working responsive websites than just common websites. However, the development could be done by using a responsive web framework, e.g. Bootstrap, which makes it easier.

Mobile website

Another approach of creating cross-platform web applications is to have two separate websites – one optimized for larger screens and

one for mobile phones. When there is already a website optimized for desktops, it is easier to develop a mobile version of the same site than to transform it to responsive design. Mobile website can be tailored exactly for its purpose. However, mobile website needs its own domain and also code, so the whole maintenance is not so comfortable.

Pros and Cons

- + Portability
- + No installation
- + Instant updates
- + Can be found by web searcher
- Internet connection needed
- Weak performance
- Lacking support of native controls

3.2 Native mobile applications

A common way of development of cross-platform applications is to create a web application for devices with large screen and native applications for mobile phones. Native mobile applications are developed in specific programming languages according to specific OS of the device. For example native applications for devices with Android OS are developed in Java while iOS development is done in Objective C [3].

This approach provides the fastest performance of applications and allows the developer to utilize native API (application programming interface) of the OS for the use of camera, push notifications² etc. Native mobile applications can run on a client's side only, which means they can be fully used without the internet connection. Moreover, they

2. Notifications which are displayed to users even if they don't use the application at the moment.

can be published at mobile e-shop³ where most of mobile phone users look for specific applications.

A disadvantage is that to accomplish a cross-platform application, for each different OS (Android, iOS, Windows Phone, Blackberry etc.) a different application must be made. Various OS use different programming languages and all of them require more skills and are harder to learn than HTML and CSS used for web applications. Therefore, the development is challenging and expensive.

Pros and Cons

- + Access to every native API
- + The fastest performance
- + Mobile e-shop
- + Works offline
- Expensive and challenging development

3.3 Cross-platform mobile frameworks

Applications developed using cross-platform frameworks take advantages of both web applications and native applications. They are mostly written in one programming language so they are portable and the development process takes less time than in native mobile applications case. But, unlike web applications, they can use native API. Instead of publishing on the internet, they are installed locally on a device and they can be published on mobile e-shops.

Cross-platform frameworks are generally divided into three groups according approaches used to attain their purpose. [4].

Hybrid approach

Hybrid mobile applications are mainly written in HTML, CSS and JavaScript. They run in the native browser of the device where only

3. Most famous mobile e-shops are Google Play for Android OS and App Store for iOS.

the content of the application is viewed; the panel with URI (unique resource identifier) is hidden. The native browser can communicate with native API of the device, which allows the hybrid mobile application to use functionality like push notifications, camera and others, so the functionality of the application can be complex [5]. Differences between platforms are handled by specific hybrid mobile framework, so the application source code can be used for different platforms. With these features, the application written as a web application can behave like a native application. However, as the application depends on a native browser, it can be slower than native mobile application.

Native browsers are often used by native mobile applications, for example to refer to the documentation which is stored on the internet. This allows developers to update the documentation without needs to update the local version of the application installed on user's device. In this case the native mobile application occasionally uses the native browser. The hybrid mobile application works the other way round - the application runs in the native browser most of the time and if needed, it uses some native functionality of the device.

As the core of hybrid mobile applications is written in HTML5, CSS and JavaScript, a desktop web browser can display its content. However, web browsers don't necessarily support specific hybrid mobile framework technologies. The reason of possible incompatibility is that hybrid mobile frameworks add their own functionality, for example the use of native API.

Interpreted approach

In interpreted approach case the application source code is written in JavaScript. Unlike in the hybrid mobile application case the GUI (Graphical User Interface) is not written in HTML and CSS because the application typically does not run in the native web browser of the device. Instead, the JavaScript source code is interpreted at the runtime by JavaScript interpreter⁴. All GUI elements of interpreted mobile applications are native and together with native features are accessible from the JavaScript objects' method calls.

4. V8 for Android, JavaScriptCore for iOS

3. CROSS-PLATFORM APPLICATION DEVELOPMENT

In case of Appcelerator Titanium, which is the well-known framework for interpreted mobile applications, there is the Titanium API for this purpose. JavaScript objects are paired with native objects and shares properties and methods. With using Titanium API, methods called on JavaScript objects invokes native methods [6].

Compared to hybrid applications interpreted applications are faster and looks more native. However, to achieve native look and behaving, it is sometimes necessary to write platform specific code, because different platforms have different features and not all of them are handled by framework.

Cross compiled approach

Cross compiled approach is very similar to interpreted approach since the result is also native application. The difference is that cross compiled applications are written in C# and then compiled to native code. Well-known framework for this purpose is Xamarin.

Pros and Cons

- + Cross-platform
- + Works offline
- + Mobile e-shop publish
- Could be slower than native application but faster than web application
- Access to native API

4 Software Analysis

This chapter will contain diagrams (use-case and class)

5 Back-end Implementation

This chapter will contain info about technologies used for back-end (application server, database server, Hibernate ORM, REST API) and description of back-end development cycle.

6 Front-end Implementation

This chapter will contain info about technologies used for front-end (JS+HTML+CSS, AngularJS, Ionic Framework) and description of front-end development cycle including.

7 Testing an Publishing

If this chapter will be included it will contain info about testing and publishing application.

8 Conclusion

Bibliography

- [1] Stephen R. Covey. *The 7 habits of highly effective people*. 3rd ed. Praha: Managment Press, 2014. ISBN: 978-80-7261-282-6.
- [2] Mobiorge website, ed. *Global mobile statistics 2014*. 2014. URL: <https://mobiforge.com/research-analysis/global-mobile-statistics-2014-home-all-latest-stats-mobile-web-apps-marketing-advertising-subscriber> (visited on 11/22/2015).
- [3] Andre Charland and Brian Leroux. "Mobile application development: web vs. native". In: *Communications of the ACM* 54 (2011).
- [4] R. Raj and S.B. Tolety. "A study on approaches to build cross-platform mobile applications and criteria to select appropriate approach". In: *India Conference (INDICON), 2012 Annual IEEE*. 2012.
- [5] Patrick Rudolph. *Hybrid Mobile Apps. Providing A Native Experience With Web Technologies*. 2014. URL: <http://www.smashingmagazine.com/2014/10/providing-a-native-experience-with-web-technologies/> (visited on 11/22/2015).
- [6] Kevin Whinnery. *Comparing Titanium and PhoneGap*. 2012. URL: <http://www.smashingmagazine.com/2014/10/providing-a-native-experience-with-web-technologies/> (visited on 02/03/2016).