



CROSS PLATFORM MOBILE APPLICATIONS



Author: Yaxiong Shen in June 2013, Son



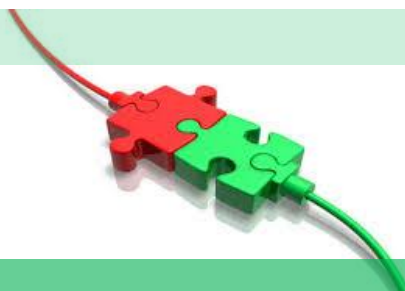
ICT Automatisering
ADVANCED THINKING

Company tutors:
Ton van Tetering
Patrick de Ru



Fontys

School tutor:
Bert van Gestel



GRADUATION / INTERNSHIP REPORT
FONTYS UNIVERSITY OF APPLIED SCIENCES

HBO-ICT: English Stream

Data student:	
Family name , initials:	Shen, Y
Student number:	2136278
project period: (from – till)	From 1 st of February to 28 th of June
Data company:	
Name company/institution:	ICT Automatisering
Department:	Machine & System
Address:	5692 EA Son
Company tutor:	
Family name, initials:	De Ru, P
Position:	Software engineer
Family name, initials:	Van Tetering, T
Position:	Software engineer
University tutor:	
Family name , initials:	Van Gestel, B
Final report:	
Title:	Cross platform mobile application
Date:	10-6-2013

Approved and signed by the company tutor:

Date:

Signature:

Preface

After experiencing a project with almost only programming in my first internship, I would like to look for a project that is more research orientated for my second internship. Also because of my personal interest in mobile applications, a mobile development project with a mobile apps related research component would be the best option for me. After almost three months of searching, when I was about to give up the idea, I stumbled on an advertisement of ICT Automatisering on the Fontys website; a research orientated internship with cross platform mobile app development. I was immediately interested and gladly I got this opportunity to spend the next 5 month at ICT Automatisering in Son.

I really enjoyed the five months I spent at ICT Automatisering. They offered me a professional and yet friendly work environment, which helped me integrate into the company very quickly at the beginning. There were a lot of employees who liked to help me to solve difficult issues during my internship, which helped me improve my technical level and personal social skills. Therefore I am really glad that I had the chance to do my graduation at ICT Automatisering.

I would like to offer a thank you to those people who contributed to my personal development during my graduation period. First let me present two of my company supervisors: Ton van Tetering and Patrick de Ru. Thank you for not only giving me technical support and feedback for my daily activities, but also teaching me how to act as a professional in a company. I also would like to say thank you to Jiska Van Herpen, who successfully guided the process of the interview and arranged the contract so quickly for me. Furthermore, I would like to thank all colleagues at ICT Automatisering in Son for the comfortable and pleasant experience within the company.

For Fontys Hogeschool, I would like to thank Bert Van Gestel for the process-oriented guidance during my internship. He ensured that I was in a right direction with both the wishes of school and the company.

Table of Contents

Preface.....	3
Summary	6
Glossary	7
1 Introduction	8
2 About the company.....	9
2.1 Company introduction.....	9
2.2 ICT guidance for interns.....	9
2.3 Company organization.....	9
3 Assignment overview	11
3.1 Problem definition.....	11
3.2 Project purpose.....	11
3.3 Project products.....	11
3.4 Working methods and tools	11
3.5 Objectives	12
3.6 Project plan.....	13
4 Research.....	14
4.1 Research description.....	14
4.1.1 Background.....	14
4.1.2 Research overview	14
4.2 Research aims.....	14
4.3 Research requirements	14
4.3.1 Criteria	15
4.3.2 Test application requirements	16
4.4 Research method	16
4.4.1 Theoretical comparison.....	16
4.4.2 Practical comparison.....	16
4.5 Research result	16
4.6 Research result	17
5 Methodologies and Techniques	18
5.1 System development method	18
5.1.1 Scrum	18
5.1.2 Object-oriented analysis and design	19
5.2 Research tools and techniques	19
5.2.1 First round tools	19
5.2.2 Test applications & “Domotica” application tools & techniques	19

5.2.3	Interactive method	20
5.2.4	Unobtrusive method.....	20
6	Analysis and design.....	22
6.1	PhoneGap	22
6.2	Titanium	22
6.3	MoSync	23
6.4	Domotica Frontend application	24
7	System development	26
7.1	Development process.....	26
7.1.1	Investigation.....	26
7.1.2	Application	26
7.2	Development difficulties and solutions.....	27
7.2.1	Investigation difficulties and solutions	27
7.2.2	“Domotica Frontend” project	28
8	System testing.....	29
8.1	Testing process.....	29
8.1.1	Project integration test	29
8.2	User acceptance testing.....	29
8.3	Testing priority	29
8.4	Testing result.....	30
9	Conclusion & Recommendation	31
9.1	Conclusion.....	31
9.2	Recommendation	31
	Evaluation.....	32
	Reference	33
	Appendix A Project survey.....	34
	Appendix B Project plan.....	37
	Appendix C Investigation report	50
	Appendix D URS document.....	98

Summary

ICT Automatisering is a Dutch company. The main expertises of ICT are portal & application development, embedded software development, and industrial automation. The company aims to have the leading-edge technology to ensure its competitiveness, which can be seen in this graduation project. The project has two popular software elements: Domotica and cross platform mobile frameworks.

At present, Domotica is widely used especially at home and in hospitals. It brings the convenience of controlling your house from anywhere. ICT also developed an Android prototype to control the lights in a dollhouse (a simulated house with lights). Due to the large amount of the mobile platforms in the mobile market, ICT would like to create a new Domotica application that can be used on different mobile platform.

It is quite difficult to directly choose one framework to develop the Domotica app. In this case, the intern firstly found nine potential frameworks through the internet. These are PhoneGap, Appcelerator Titanium, Xamarin, Rhodes, AppMobi, MoSync SDK, Rhodes, AirPlay SDK, Corona, and the combination of Xamarin + MonoCross. Then he got the requirements from the client and summarised them into 12 criteria. Based on these criteria, nine frameworks were analysed by scoring each of the criteria. After all frameworks were scored, he selected PhoneGap, Appcelerator Titanium and MoSync to create a few test applications to test their features. In the end, MoSync was selected for the company's further use, and Appcelerator Titanium was selected for the Domotica application development.

The intern had a meeting again with the client to collect the Domotica application requirements. After the meeting, the user requirements document was made. Then the design document was created based on the client's requirements. After having the design document confirmed with the client, the test plan was made. By the time this report is finished, the Domotica application has finished the first increment and half of the second increment. Right now, the Domotica application supports some basic functions such as turning on/off a light, changing light colour and intensity, and editing the server connection. Overall, although there were some issues with the layout of the application, Titanium is still quite a useful framework and made the Domotica application development easier.

Glossary

Domotica: Domotica is a home automation system, this word comes from the combination of the words domus (Latin, means house) and tica (Greek, meaning “that works by itself”). If you embed this system into all household appliances, you can, through PCs, mobile phones or tablets, easily control all lights, doors and curtains. Also you can monitor the working condition of all appliances at home.

Android: It is a Linux-based operating system, designed mainly for touch screen mobile devices such as smartphones and tablet computers. It is initially developed by Android, Inc. it is open source and released under Apache License^[1]

iOS: it was named iPhone OS. It is a mobile operating system developed and distributed by Apple Inc. Originally it was developed for iPhone. Now it has been extended to support more Apple device such as the iPod touch, iPad, and Apple TV^[2]

Framework: The framework mentioned in this report is a software framework. It is an abstraction in which software providing generic functionalities can be selectively changed by additional user-written code, thus providing application-specific software. It includes programs, compiler, code libraries, tool sets, and **Application Programming Interfaces (APIs)** that bring together all the different components to enable development of a project or solution^[3]

Subversion: It is often abbreviated SVN, which is a software versioning and revision control system. It is used to maintain current and historical versions of files such as source code, web pages, and documentation.^[4]

Eclipse Studio: It is a multi-language Integrated Development Environment (IDE). It is mainly used for Java development.

Xcode: It is an Integrated Development Environment (IDE) developed by Apple for developing software for OS X and iOS.

jQuery Mobile: It is a touch-optimized web framework. It is compatible with some mobile app frameworks and platforms such as PhoneGap^[5]

ARM: ARM architecture describes a family of RISC-based computer processors. It is mainly used in modern devices such as smartphones, digital TV's and mobile computers.^[6]

GCC: The full name is GNU Compiler Collection. It is a compiler system produced by the GNU Project to support multiple programming languages.

1 Introduction

Due to the growth of the smartphone market, mobile applications are required to offer excellent experience to all smartphone users. They can get all types of applications for their needs and should be able to make proper use of them. However, the market of mobile platforms is being fragmented due to the diversity of the mobile operating systems, this makes developing a mobile application for multiple platforms a time consuming task.

Three years ago, a company which had a mobile strategy was at an advantage. Two years ago, companies which didn't have a mobile strategy were already not competitive. From 2012 onwards, the condition is changing again when it comes to supporting multiple mobile platforms. Maybe it is not a big deal for some businesses which only care about one platform, but for most companies which would like to cover a myriad of device types it becomes difficult to get such applications to work on all these platforms. Fortunately, a new technology has come into existence and releases the pressure of people who need applications that work on different mobile platforms, which leads to a significant savings in time and money.

The market of cross platform mobile frameworks is growing quickly in the last five years. More and more IT companies are interested in this new trend. ICT Automatisering also realises the importance of cross platform mobile frameworks. There is currently a project called "Domotica Frontend", which already has a prototype (Android) application that can turn a light on or off in a dollhouse (a test environment), what the company would like to have is a cross-platform mobile application to communicate with the web service from multiple- platforms. Certainly they can easily find a lot of cross platform mobile frameworks to start with. However, each of them has its own features and advantages. In this case, finding out the best suited cross platform mobile framework for the "Domotica" project is extremely important. This is one of the reasons why an investigation is planned in advance. However, that is not the main goal of the investigation. ICT Automatisering, as an international company, has many mobile applications applied in different verticals such as automotive, logistics and healthcare. Some of them are required to run on multiple mobile platforms. So it is vital for the company to have a comprehensive view of all possible frameworks and which one is best suited.

This document is the final report of the project "Domotica Frontend" with explanations of all completed activities during the past five months. Before directly going into the project itself, it is also important to know the background of the project. Chapter 2 describes the company ICT Automatisering with sections about its organization and marketing position. The following chapter discusses the definition and working methods of the assignment. Subsequently, chapter 4 will be dedicated to the conducted research. The fifth chapter analyzes the application requirements and structure. After discussing the design in Chapter 6 and testing in chapter 7, the conclusion and recommendation will be given in chapter 8.

2 About the company

2.1 Company introduction

ICT Automatisering (ICT) started in 1978. It is the market leader in the field of embedded software. Since ICT supplies hardware, software and services, it is fully responsible for successful completion and future product availability. It also provides software for industrial applications and realizes technological complex automated solutions for managing and controlling business processes. Within the ICT Automatisering operation, there are six verticals: Logistics, Automotive, Machine & System, Industrial Automation, Energy and Healthcare. Where necessary, the verticals can work closely together so that the scale of the products and service required can be quickly established. ICT is also quite active in the Traffic & Transport and testing sectors. Because of its extensive specialist knowledge of automated processes, the company has a leading position in the knowledge-sharing market. Knowledge as the product of ICT is sold to the customers through staff or projects. In general, ICT services to its clients include well-educated staff, the realisation of system solutions on a project basis and the maintenance of ICT systems.

2.2 ICT guidance for interns

ICT normally assigns two people to the intern as coaches. The project coach assists with the project plan and ensures the overall process is in the right direction. The technical coach supports and guides the intern in relation to technical issues. The graduation thesis projects are structured via SCRUM method, which enables the intern to respond flexibly to changes in the project. Scrum is also the main working method within the company.

2.3 Company organization

There are approximately 1000 employees working at the company in three countries, the Netherlands, Germany, and Poland. It currently consists of four divisions: ICT Automatisering B.V. in the Netherlands (embed software forms part of and is incorporated into products, technical automation and services with a high added value), ICT Software Engineering GmbH in Germany (embedded system software and technical automation). For each region of ICT, it has a region manager, an OM (Operating management) team, a sales team, a service team and a group of professionals. The diagram below shows the general view of the company organization and where the intern functions.

The intern belongs to the professionals group. He is responsible for all documents and the project. In this group, there are two project managers to check the process of the project and give suggestions. They also give some requirements for the investigation and the “Domotica” application.

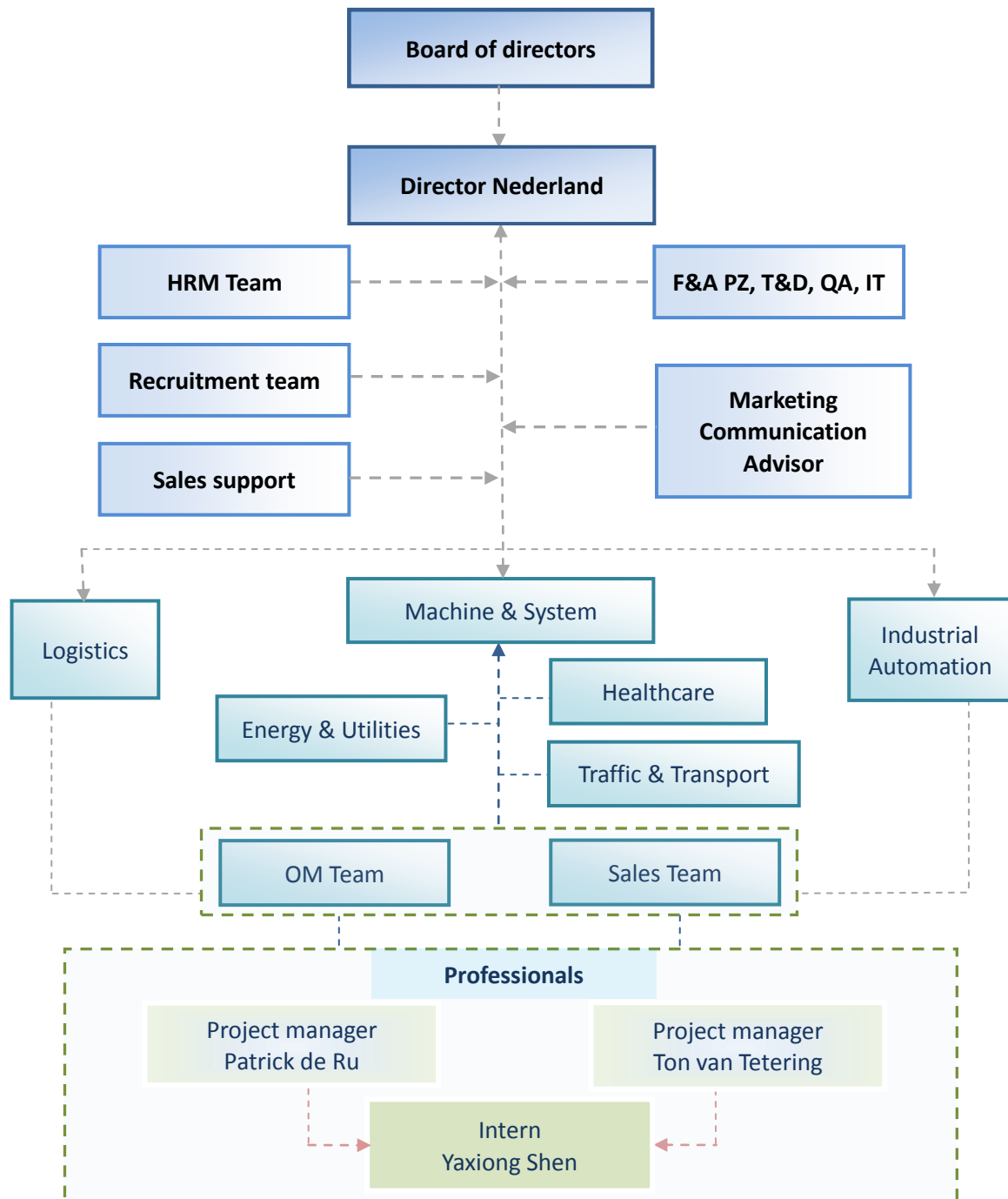


Diagram 2-1: company organization

3 Assignment overview

3.1 Problem definition

Within ICT they are currently building mobile applications for different platforms. As we all know, developing mobile apps for each platform is time and money consuming. That is why the company wants to find a cross-platform mobile framework that is well suited for mobile apps within ICT. Therefore the investigation result is really valuable and interesting for the company.

Since Domotica is growing in popularity to control devices in a house, the company has set up a project called “Domotica”. There is already an Android application with a basic user interface. The company would first of all like to improve the user interface of this app (that is why the project name for the internship is ‘Domotica Frontend’), and then use one of the possible cross platform mobile frameworks from the investigation result to turn the Domotica Android app into a cross platform app.

3.2 Project purpose

Smartphones are useful tools to control Domotica with; this involves creating a mobile application. In order to achieve the application purpose, which is to save money and time, a cross platform development framework can be used to develop a Domotica app for multiple mobile platforms at once instead of developing for each independent mobile platform. The purpose of the investigation in the project is to find the best framework for the company for later use and also for the frontend Domotica application development.

3.3 Project products

The end products of this project are: an investigation report with a few test applications for three frameworks and a “Domotica” cross platform mobile application.

Before having the final decision of the best framework in the investigation, some test applications will be developed to test a few required functionalities of the frameworks. After all, the result of the investigation will be evaluated and used by the company for their different mobile applications in different verticals. For example, in the healthcare vertical, there is an Android application named mPACSview that might also need to be used in different platforms. It allows specialists to retrieve and view diagnostic images directly from the PACS (Picture Archiving and Communication System) on a tablet; they can even view these medical images anywhere and at any time through an internet connection.

The “Domotica” cross platform mobile application belongs to Machine & System. This application will not directly be applied to the mobile market or for commercial use. Instead, it will be shown at exhibitions as a new mobile technical trend.

3.4 Working methods and tools

- Agile Scrum
- HTML, CSS, JavaScript
- C/C++
- Microsoft office word 2007 & Excel 2007
- Eclipse
- Xcode
- Titanium Studio

- MoSync SDK IDE
- SVN

3.5 Objectives

This part will give a general view about all objectives needed during the internship period, including the necessary contents and to which party it should belong to.

Corporate objectives:

- **Project plan:** The company tutor guides the student in this process and needs to approve the project plan.
- **Final presentation:** in the end of the project, the intern needs to give a presentation to the company. The school tutor is also welcome. In other words, it is the preparation for the presentation at school.

For the investigation:

- **User requirement specification:** It provides a high level description of the user's expectation of the investigation.
- **Backlog:** Gives general categories to show the plan for the investigation. Also it gives the time prediction.
- **Investigation report:** It gives the result of the investigation. This document needs to be approved by the company tutors and will be put into the final report as the research chapter at the end of the internship.
- **Test applications:** The test applications are developed to test the required functions for a certain framework.

For the Domotica application:

- **User requirement specification:** It gives an overview of the user requirements to the application. For example, how the client would like the application to look like, etc.
- **Design document:** It consists of the application's prototype, class diagram and sequence diagrams.
- **Test plan & test report:** The test plan describes how the application is expected to be tested. It mainly consists of the test phase, testing strategy and the test cases. The test report is the result after the intern tests the application followed by the test plan.

School objectives:

- **Project plan:** It consists of project statement, project phasing, and project management plan. It's delivered before the school tutor's first company visit.
- **Communication plan:** It describes how the intern intends to communicate with the university tutor (by telephone, email etc.) and when. It also gives a plan about which documents the intern will hand in to the tutor and when, and within which time the school tutor replies.
- **2-weekly update info:** Give a general view about the process of the project each two weeks. Since this project is controlled under Scrum, it will specifically describe what the intern has done during the two weeks, how he has done it, which problems the team (including the intern and the company tutors) have and what the team have done well and what can be the improved in the next sprint.
- **Final report:** The school tutor will receive three draft versions and one fixed version of the final report.

- *Final presentation:* In the end of the graduation internship, the intern needs to give a presentation at school. A final defense is also required during the presentation. The company tutors will also attend as the members of the jury.
- *Professional skills:* After half year internship in the company, the school would expect that the intern can work independently as a software engineer in a company.

Personal objectives:

- *Personal skill:*
 1. Have better communication skill with colleagues
 2. Dealing with problems independently
 3. Improve presentation skills
 4. Be ready to be a real employee in a company
 5. Think wider, not only think how to finish an assignment, but also the meanings and the effects of the project
- *Technical skill:*
 1. Improve the web based techniques
 2. Have more experience with mobile application development
 3. Be well-known with cross platform mobile application

3.6 Project plan

At the beginning of the internship, a project plan was built up and three parties (the company tutors, the school tutor, and the intern) agreed on this document. Thus the project would mainly follow the plans in this document. The project plan can be found in appendix B.

However, there are some changes that happened during the past five months. The actual project process is described in the system development chapter.

4 Research

This chapter gives an overview of the research required in this graduation project. It also explains the importance of setting up a research before programming the “Domotica Frontend” project.

4.1 Research description

4.1.1 Background

Consider the following scenario: a customer asks a developer to develop an iPhone application. Everything goes well until the customer realizes that an Android version of the same application is also necessary. Then the software developer has to recreate the same application for a different platform. However, what if the customer decides that Windows Phone should also be supported? Here starts an idea that developing a mobile application for multiple platforms at once would be a time saving task. It will help the developer save a lot of development time and decrease the cost for the customer. There are a number of frameworks offering to enable cross platform mobile application development.

4.1.2 Research overview

ICT Automatisering would like to have a cross platform mobile framework for their mobile development. The intern found out nine frameworks through the internet before the investigation started. This investigation then gives an overview of these nine potential frameworks. They are PhoneGap, Appcelerator Titanium, MoSync SDK, Rhodes, AirPlay SDK, Xamarin, AppMobi XDK, Corona, and the combination of Xamarin + MonoCross. All frameworks will be judged by a list of criteria. Initially this is done using only the information available online. Based on the results of this first round, a test application will be designed and created for the top three frameworks to test the framework support for device functionality and various other aspects. The top three will be judged with the criteria, but this time based on the experience of actually creating an application using each framework. In the end of the investigation, the intern will give an answer about which framework is the most suitable for the company.

4.2 Research aims

The result of the research is important for both the company and the intern, that’s why three months are planned for the investigation assignment. In this case, the result is of course subjective. The research report shows the company which cross platform mobile framework is the best option for a certain kind of mobile application. In the future, when the company has some mobile applications that need to be cross platform; the investigation report can be read and used as a reference.

At the same time, the intern also needs this investigation to get known with the functionalities of all potential frameworks and in the end chooses one of them for the “Domotica Frontend” project. This one must be best suitable for the Domotica project and at the same time the intern should be confident of using it to accomplish this project. The research experience is helpful for the intern to understand and use the cross platform mobile framework smoothly for the Domotica app.

4.3 Research requirements

It is important to make clear what the clients’ expectations and requirements about the investigation are. That is why the client meeting is necessary before the investigation starts. Naturally, all requirements are collected through the meeting with the client.

4.3.1 Criteria

After all requirements are collected, they are not structured, which makes it difficult to find out the most suitable framework based on the requirements. The intern summaries all requirements into 11 criteria to make sure that the investigation result is objective and comprehensive. The intern also gives the idea of using a weight score system. The company and the intern evaluate the 11 criteria and give a score to each criteria on a scale of 0 to 5 (where 0 is the lowest possible score and 5 the highest).

Table below (Table 4-1) is a list of criteria with explanation and the base weight for scoring.

Criteria	Explanation	weight
Included functionalities	What functionalities and device features does the framework support (e.g. camera, accelerometer and NFC)?	6
Support & documentation	Is the documentation of the framework good enough?(e.g. FAQ and official documentations)	8
Maturity of framework	How secure is the framework and what are the recent experiences of the other developers?	12
Code reusability	Is it easy to reuse the existing code in other projects?	3
Testability of framework	Is it easy to test the applications made using the framework? (e.g. debugging or profiling)	9
Flexibility & Source license	Is the framework flexible enough to be customized or expanded with other tools and functionalities? Is it open or close source?	4
License costs	How much does the license cost and is it allowed to use the created applications commercially?	11
Performance	What is the performance of the created applications (For example, what is the runtime compared to the native instructions?)	10
Security	Are created applications secure enough? Do they support Https connections? Do they offer code obfuscation?	5
Difficulties to create applications	How long and how much effort would it take to learn how to use the framework with enough skills to develop a good application?	7
Updatability	Is it easy and convenient to have a successful update for the development tools?	2
Distribution and Installation	Is it easy to distribute and install the created application? (Automatically signed? native installer format?)	1

Table 4-1: Investigation Criteria

4.3.2 Test application requirements

For each framework's test application, the following functionalities were required to be implemented in order to test if the framework supports the desired functionality.

- User interface: Test application should have basic UI elements and a table with 2 columns, and zoom in/out image in platform default way
- Hardware access: The test application needs to support camera access, file access, screen rotation
- Network access: The test application should be able to connect to the internet and get response.

4.4 Research method

The investigation consists of a theoretical comparison and practical comparison.

4.4.1 Theoretical comparison

This is the first round of the research. All available information online will be gathered for each framework.

4.4.1.1 Weight score system

All potential frameworks will be scored by the weighted score system. The final score of each criteria will be its base weight multiplied by the evaluated score. The final score of each framework will be the sum of its criteria scores.

Using a weight scored system is helpful to evaluate a framework in a more objective and comprehensive way. It avoids the author analyzing the framework using his personal opinions. It also lets the readers understand the functionalities of each framework in a more clear way.

4.4.2 Practical comparison

After the theoretical comparison, it is vital to prove the reality of the information gathered from the Internet. That is why the practical comparison is planned right before the end decision. First of all, a test plan will be created based on the functionalities that the company would like to test for the potential framework. After test applications are created, the list of criteria will be re-marked based on the result of the test application.

The framework which has the highest score should represent the best option for the company and will be chosen as final development tool for the "Domotica Frontend" project.

4.5 Research result

In the first round of the investigation, the top three frameworks are Appcelerator Titanium, PhoneGap, and AppMobi (ordered by score from high to low). However, there is an exception to the process described before. Since PhoneGap and AppMobi both function similarly, the company decided to do further investigation for MoSync (the fourth place in the 1st round investigation) instead of AppMobi. MoSync is interesting because it is the only framework that supports C/C++ development. Also switching the last framework gives a more diverse final comparison.

In the second round of the investigation, all required framework functionalities mentioned above have been successfully implemented in each framework's test application. In the end, Titanium has the highest score, followed by MoSync, and PhoneGap has the lowest score. However, the final recommendation for ICT Automatisering is MoSync. Although its score is in general lower than Titanium,

it supports more mobile platforms, has two IDEs and two programming languages to choose from and especially C++ is a good match with existing skills in the company. It also supports extra functions desired by the company, such as NFC. The result test app has native UI and runs smoothly. The lower score of the test application might be caused by the limited development time. For professional software engineers, it should be a better choice, since MoSync support more complicated UI and developing tools.

The final framework for “Domotica Frontend” project is also decided after the investigation. In contrast to the final recommendation made to ICT Automatisering, this project will be created with the Titanium framework. MoSync matches well the existing skills within ICT Automatisering and Titanium matches better with the existing skills of the intern. This mobile application will only be worked on by the intern and as such using Titanium will allow for faster and easier development.

4.6 Research result

To see more details of the investigation, please go to Appendix C for the investigation report.

5 Methodologies and Techniques

5.1 System development method

The process of the graduation project is controlled under Scrum. The applications (including test applications and “Domotica Frontend” application) are managed by the object-oriented analysis and design. The following two sub chapters will explain the definitions of the two methods, how it works, and why the project uses these two methods.

5.1.1 Scrum

5.1.1.1 What is it?

It is an iterative and incremental agile software development framework for managing software projects and product or application development. Unlike the traditional and sequential system development method, Scrum focuses on a flexible, holistic product development strategy where a development team works as a team to reach a common goal.

5.1.1.2 How it works?

Diagram 5-1^[7] shows how Scrum works.

Under Scrum, the project is split up into sprints. A sprint is a basic unit of development, which is a time boxed effort. It is restricted to a specific duration. This duration is fixed at the beginning of the project and is usually two weeks. Each sprint is preceded by a meeting. Before the sprint meeting, the intern finishes the tasks in the planning list. In the meeting, he gives a presentation with demo or PowerPoint. After that, the team set up the retrospective, where the team’s improvement points and highlights are discussed. In the end, it goes on to the planning part, where all tasks for the coming sprint are planned on the schedule and the estimation time for each task is made. Each day during the sprint, there is a daily standup meeting to update the project condition. During the meeting, the team member needs to tell the team about what he has done since yesterday, what he is planning to do today and if there are any bottlenecks.



Diagram 5-1: Scrum structure

5.1.1.3 Scrum tool

In this project, Scrumwise is used to make Scrum management simple and accessible. It is just a basic tool, which you can use to organize Scrum teams, software, or application developers. It is however not meant to handle high-level development projects. It is however quite easy for developers to use and suitable for smaller scale projects.

5.1.1.4 Why scrum?

Scrum makes the project development teams more effective and quicker to adapt and respond to changes that occur during a project. It assigns different roles in the team as it works toward incremental advances in development. It enables the creation of self-organizing teams by encouraging co-location of all team members, and verbal communication between all team members and disciplines in the project.

5.1.2 Object-oriented analysis and design.

5.1.2.1 What is it?

It is a software engineering approach that models a system as a group of interacting objects. Object-oriented analysis (OOA) applies object-modelling techniques to analyze the functional requirements for a system. It focuses more on what the system does. Object-oriented design (OOD) elaborates the analysis models to produce implementation specification. It focuses more on how the system does it.

5.1.2.2 Where is it used?

This development method is used during the development phase.

Normally object-oriented analysis model describes computer software that could be used to satisfy the customer's requirements. So during the analysis phase, the intern interviews with the project owners and then writes a user requirement document based on customer-defined requirements. The end result of the OOA consists of use cases, a class diagram, sequence diagrams, user interface design and a number of interaction diagrams. During object-oriented design phase, the intern applies implementation constraints to the end results produced in OOA. Such constraints could include not only constraints imposed by the chosen architecture but also any non-functional-technological or environmental – constraints.

5.2 Research tools and techniques

There are different tools used during the two rounds of investigation.

5.2.1 First round tools



In the first round, it was mainly theoretical focus. The following tools were used:



1. Microsoft office Word 2007: It was used during the whole investigation report.
2. Microsoft office Excel 2007: There are some forms, charts, and diagrams made with this tool.

5.2.2 Test applications & “Domotica” application tools & techniques

5.2.2.1 Tools


The second round was for the development of test applications. So besides using the same tools as in the first round, the following extra tools were used in the second round as well:

1.  Xcode 4.6.1: Used with the PhoneGap library to create PhoneGap test application for iOS device.
2.  Eclipse SDK 3.7.1: Used with the PhoneGap library to create PhoneGap test application for Android devices.

3.  MoSync SDK 3.2.1: This integrated development environment was used to create MoSync test applications for both iOS and Android.
4.  Titanium Studio 3.0.2: This integrated development environment (IDE) was used to create Titanium test application and Domotica Frontend application for both iOS and Android.

5.2.2.2 Techniques

Since the second round was for test application development, some development techniques were used. And depending on the tested frameworks, the techniques are quite different as well

1. Web-based techniques: PhoneGap and Titanium used this technique. Specifically for PhoneGap, which is just a set up project installed in Xcode or Eclipse SDK, it uses HTML, CSS, and JavaScript. For Titanium, it uses JavaScript for everything in Titanium Studio. For the MoSync framework, the main test application uses HTML5 and JavaScript.
2. C/C++: This technique was only used in MoSync camera test application, since most of the camera functions could not be supported by the HTML5 version of the MoSync library.
3.  JQuery Mobile: It was used to create the UI for PhoneGap test applications. Because PhoneGap only supports HTML to create simple and basic UI. JQuery Mobile can help quickly make more fancy and functional UI.

5.2.3 Interactive method

5.2.3.1 Interview

This method was widely used during this project. The intern needs to interview the project owners to get all requirements for the investigation, the test applications and the “Domotica Frontend” application. Compared to questionnaire or e-mail interaction, interview gives more direct reactions from the clients for certain questions. It also saves time and avoids the possible misunderstandings of written communication.



5.2.4 Unobtrusive method

5.2.4.1 Indirect method: Website


It occurs naturally in the investigation context. The intern used this method mostly in the investigation part and while having troubles with programming the applications. However, the range of the indirect method can also be quite narrow since it is decided only by the intern’s inventiveness and imagination. So it is better to combine the indirect methods with other methods to improve the quality of the search results.

5.2.4.2 Remaining tools

Besides all tools and techniques mentioned in the previous parts, there are some extra tools used during the project.

1.  TortoiseSVN: This tool is used to save all documents and applications created during the project. It makes sure that the correct version of work is backed up in a save place in case the current version needs to be recovered if it fails.
2.  Microsoft Lync: all team members who belong to the project use this tool to communicate

with others online. It saves time to visit other member's offices.

3.  Microsoft Office PowerPoint 2007: Since there is a presentation given in the end of each sprint (Scrum info see in Chapter 5.1.1), PowerPoint is used to create slides to show the progress of the sprint. It is also used for the final presentation in the company and at school.

6 Analysis and design

There are totally five mobile applications that have been created during this graduation project. Each of them uses a different framework. In this chapter, it explains how these frameworks are designed and how the applications functions.

6.1 PhoneGap

PhoneGap is used to create the PhoneGap test application. The diagram below explains how PhoneGap works in the test application

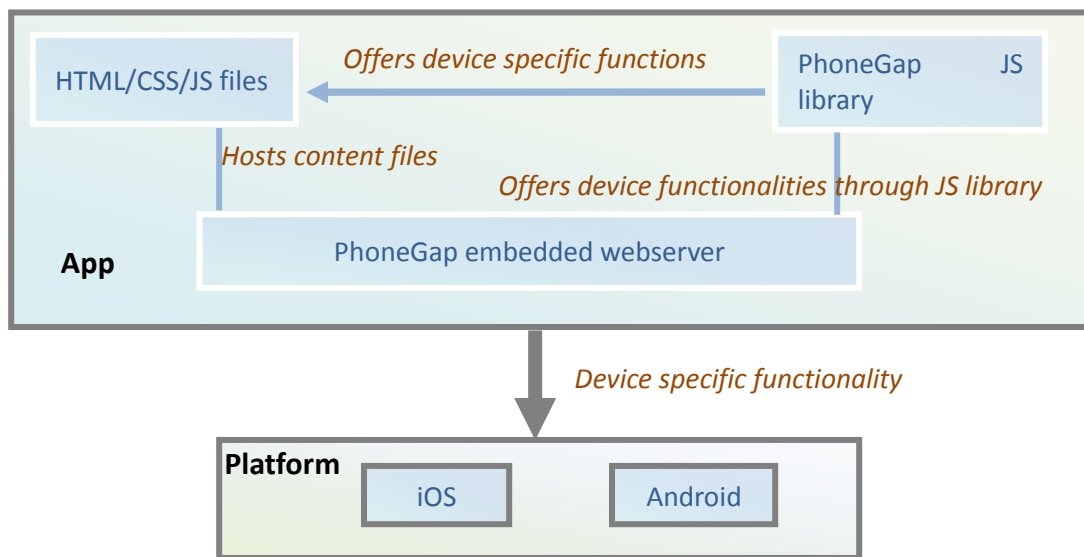


Diagram 6-1: PhoneGap test app

PhoneGap creates web based applications. The final application is bundled with a small webserver that hosts the contents of the application. The user interface is created with HTML and CSS. This of course means that PhoneGap is not able to produce native user interfaces and has no support for advanced graphics. Through a special JavaScript library and the embedded server PhoneGap is able to provide basic hardware and system access. Compared to native applications, PhoneGap apps have noticeably decreased performance. On the positive side the web based nature does allow PhoneGap to support almost every mobile platform in existence.

PhoneGap main features: Easy development difficulty, web based supports many platforms.

6.2 Titanium

Titanium is used to create the Titanium test application. The test application proved that Titanium is a powerful and yet stable framework. That is also one of the reasons why it is the final choice for the Domotica application.

The titanium test application is created using only JavaScript. Depending on the platform the application is either interpreted on runtime or compiled to executable intermediate code. Thanks to this feature apps don't perform noticeably slower than native apps. The user interface is also created in JavaScript and is completely native. This allows titanium applications to adept to each platform's look-and-feel; however it does not have any supports for advanced graphics. Titanium supports all basic platform

functionalities and can also be expanded using downloadable modules.

In diagram 6-2, it shows the work flow of the Titanium test running in iOS platform.

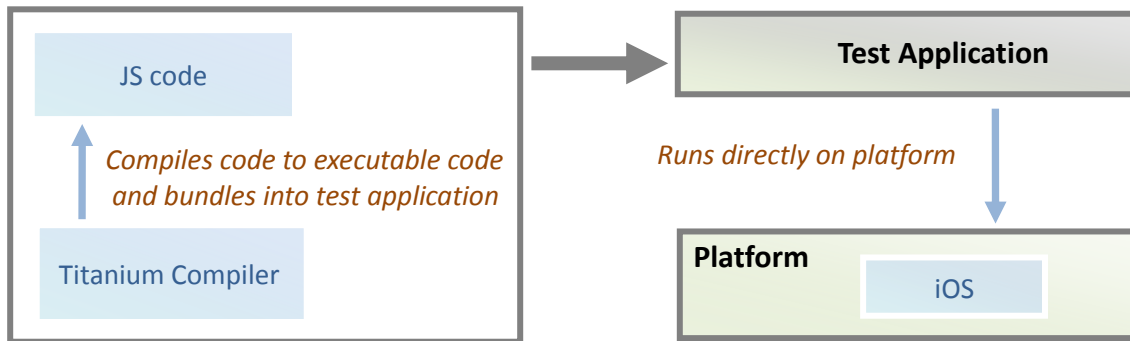


Diagram 6-2: Titanium test app in iOS

Compare diagram 6-2 and 6-3: because Apple doesn't allow frameworks to interpret code on runtime, there is a difference in the work flow between iOS and Android.

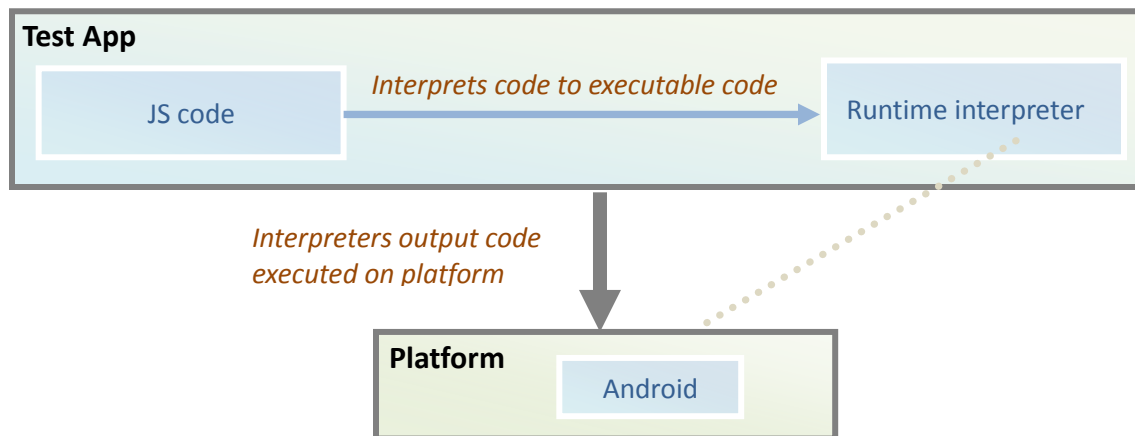


Diagram 6-3: Titanium test app in Android

Titanium main features: Medium development difficulty, native user interface, good performance.

6.3 MoSync

The MoSync camera test application is created with C++. MoSync comes with a custom C++ compiler that compiles to the target platform. This, of course, results in excellent performance. MoSync is unique because it can also do web based application in recent versions. This web based technique was also used in another MoSync test application. In this case MoSync basically functions the same as PhoneGap.

Because MoSync needs to be compiled, it only supports a limited number of mobile smartphone platforms (iOS, Android, and Windows Phone only). Thanks to its low-level C++ roots, MoSync is able to support advanced hardware functionalities and can do advanced graphics (using OpenGL).

Diagram 6-4 and 6-5 shows how different MoSync works with the test app in iOS and Android platform.

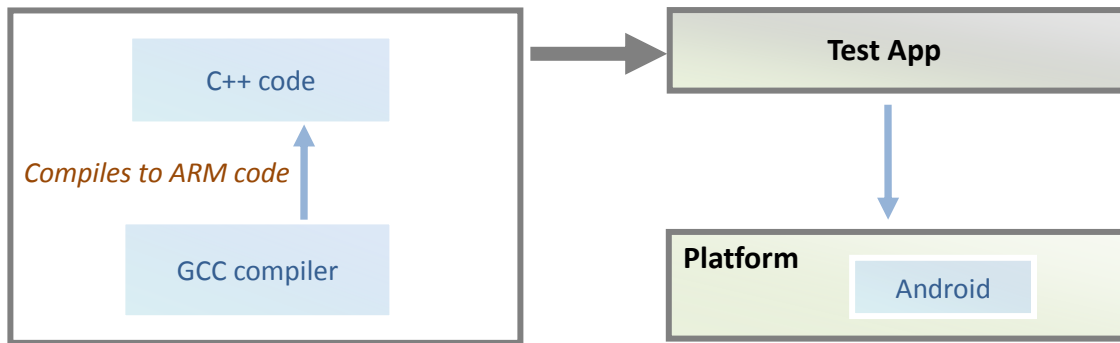


Diagram 6-4: MoSync test app in Android

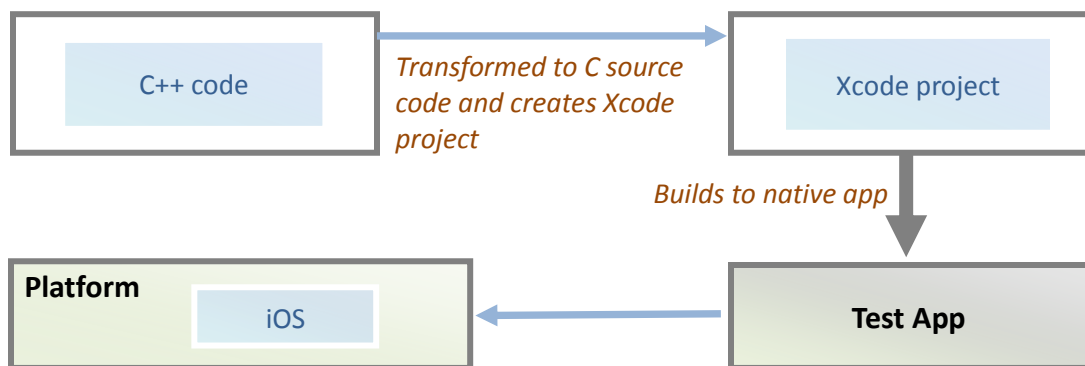


Diagram 6-5: MoSync test app in iOS

MoSync: High development difficulty, native and web based user interface, fast performance, advanced graphics.

6.4 Domotica Frontend application

This diagram below shows the system structure of the Domotica Frontend application

The final Domotica application was created with the Titanium framework. This framework was chosen due to the ease of development.

The app is based on a few functionalities: (Please refer to Appendix D for more URS document to see more requirements of Domotica application)

- Controlling lights
- Configuring scenes
- Scheduling scenes
- Server configuration

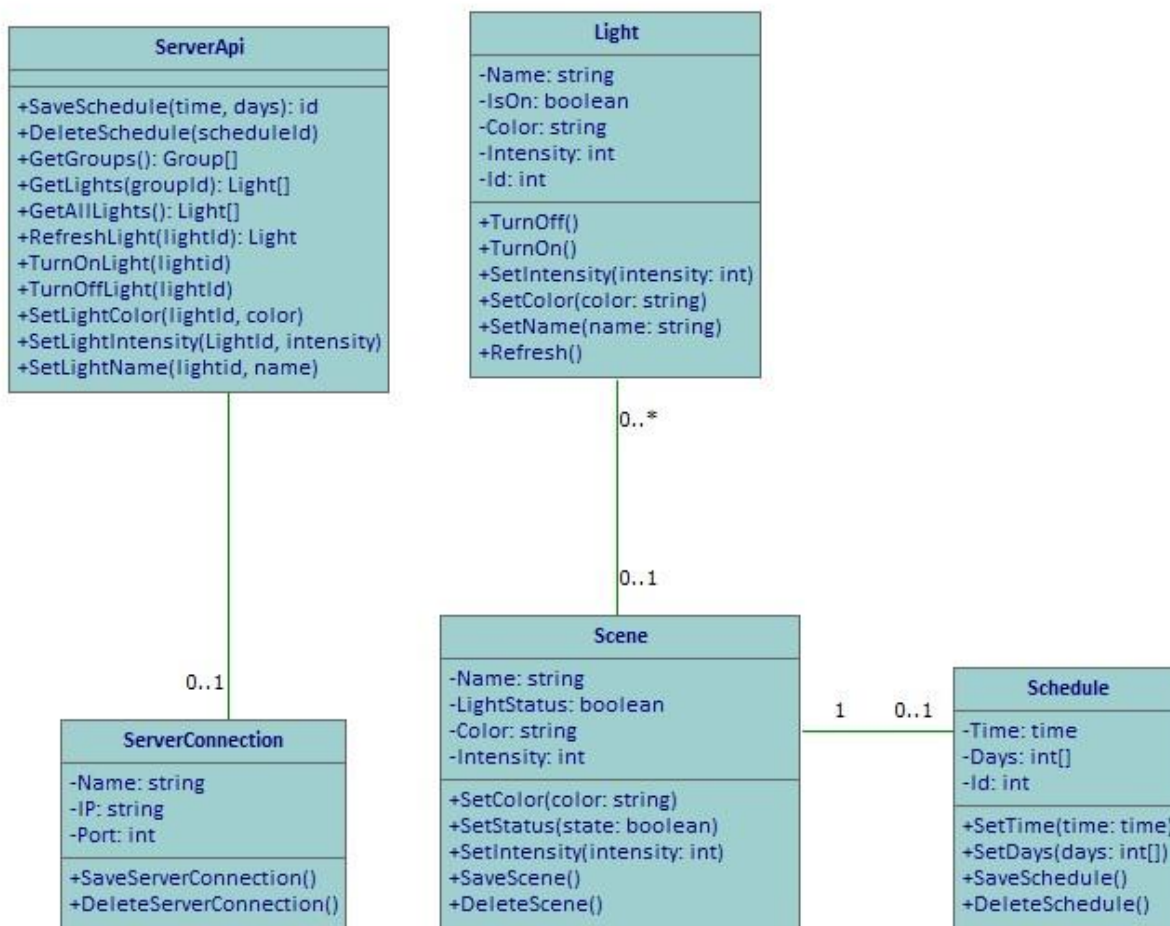


Diagram 6-6: Domotica app class diagram

The diagram above shows the class diagram for the Domotica Frontend app. The ServerApi class is the backbone of the application because it has functions that communicate with the backend server using the active serverConnection. The Light, Scene and Schedule classes have relationships with each other that form the basic structure of the application. A house has many Lights that can be identified by their id and name. A Light can be part of many scenes. A scene may contain many lights and may optionally have a schedule associated with it. All of these classes use the ServerApi when communication with the backend is required.

The user interface is based around editing information in the above classes and the relationships to each other. It will also use the ServerApi class to load data from the server.

7 System development

7.1 Development process

Since the whole project is controlled under the agile Scrum method, it is difficult to name the process on which each sprint's tasks are based. However, all tasks can be roughly summarised into seven phases. As mentioned before, this graduation project is made up of an investigation part and an application development part. Thus, phase 1-3 will show the process of the investigation and the rest will introduce the development flow of the application part.

7.1.1 Investigation

Phase 1: Initiative

Description: This is the beginning phase of this project. So first of all, a project plan is made to have a complete view of the general plan. After that, a meeting with the project owner is held to know the project owners' requirements and expectations of the investigation.

Deliverable: Project plan

Phase 2: Definition

Description: After the client meeting, a user requirement document is made. And to be sure that all requirements written in this document match the client's expectations, a confirmation meeting is held between the clients and the intern. The user requirements document is adjusted afterwards to be ready for the coming phase

Deliverable: User requirements document

Phase 3: Investigation

Description: The intern writes independent chapter for each framework and gives score in the initial round. The top three frameworks will have further testing. In the end, the recommendation is given for the company and the final framework for Domotica is decided as well. (Details please refer to Research chapter)

Deliverable: Investigation report, test applications for three frameworks.

7.1.2 Application

Phase 4: Definition

Description: This is the first phase in the application part. First of all, a client meeting is held to make clear the clients' requirements for the "Domotica Frontend" project. After that, the intern writes the use cases, functional and non-functional requirements for the user requirements document.

Deliverable: User requirements document

Phase 5: Design

Description: This design phase is meant for the design of the "Domotica Frontend" application. The intern draws the class diagrams and sequence diagrams, specifies data and functions per class, and designs the GUI with screenshots. After getting confirmation from the clients, each part is included into the design document.

Deliverable: Design document

Phase 6: Realization

Description: To implement the application is the first task. After that, a test plan is made to test the stability of the application. Then the intern can follow the test plan to test the application, and fix any bugs.

Deliverable: “Domotica Frontend” cross platform mobile application, Test plan document

Phase 7: Transformation

Description: After handing in the final version of the “Domotica Frontend” project. The intern gives the final report and holds a presentation to the company to present the final products.

Deliverable: Final report, Final presentation

7.2 Development difficulties and solutions

There were some unexpected problems that occurred during this project. Most of them were solved on time. Since there are two parts in the project, all difficulties and solutions will be summarized in two sections in this sub-chapter.

7.2.1 Investigation difficulties and solutions

In the 1st round:

Difficulty: The information timeliness played a little trick at the beginning. Two frameworks planned for the research were discontinued enduring the investigation. Also there was information gathered from the internet that was not up-to-date any more. This happened, for example, due to a newer version of a framework being released without updated documentation.

Method: There was one framework prepared as a backup plan, which saved some time in finding a new framework to use instead of the expired one. Also it was important to check the date of the information to be sure the information was up-to-date.

The 2nd round:

Difficulty: Framework installation is very annoying for people who just started programming with a cross platform IDE. The intern also experienced some problems with this. For example, the author worked behind a proxy connection, and Titanium IDE has a few bugs with proxy connections. Also the installation document has some small mistakes or misunderstandings, which also takes extra time trying to figure it out.

Method: Don’t completely trust the official website information; some online communities for developers also have useful information. At the same time, working with a proxy connection is not preferable for most frameworks that need to connect with the network and/or internet.

Difficulty: Unfamiliar programming language: Due to three frameworks having to be tested in the second round research, and each of them using a different programming language, it takes some effort to get familiar with each language.

Method: Reading online guides and starting with some examples helps to understand the new language quickly.

Difficulty: Unknown system error: There were a few unknown errors that happened while programming the MoSync test application, which took quite a few days of searching for a solution. For example, during development the MoSync test application for iOS, system error 15 occurred, which failed the packaging of the application for iOS. Unfortunately, there were no useful solutions to be found online.

Method: Register in MoSync website, and post a question with the error number in their developer community to ask for help. Unfortunately this does not lead to a quick development time.

Difficulty: Test applications perform differently on different mobile platforms. Since, with some frameworks, the user interface has a native look for different platforms, the layout sometimes does not perform as expected. There are also a few functions from certain frameworks that only work for a single platform.

Method: Need to adjust and test the user interface independently for each mobile platform. Also carefully read the framework API documentation to know clearly which function can be used on which platform.

Difficulty: Installing the test applications on an iOS device requires a profile certificate and provisioning. Setting this up correctly can be difficult sometimes. For example, in the Mosync project, the certificate didn't match the provisioning setup at the beginning.

Method: Figure out the correct match for the certificate and its provisioning. And make sure that they are not expired.

7.2.2 "Domotica Frontend" project

Difficult: The user interface of the Domotica application is more complicated than test applications. It makes it difficult to keep the user interface tidy. The main problem is the inconsistency of the UI in Android and iOS device.

Method: Make sure the measurement unit is platform specific. For example, in Titanium, 'px' is mainly for working on Android device, and 'pt' is for iOS device. Besides that, small adjustments are needed for iOS and Android application.

Difficult: Some UI elements are not supported by Titanium framework. For example, the color panel that was planned for the user to pick up a color is not supported. And this is one of the most important UI elements in this application.

Method: Instead of struggling with the color panel, a color picker was created manually with 20 views. Each view displays a basic color. So the user can click on a certain color view to choose a light color.

Difficult: The Domotica application is expected to have language localization based on the device language. Luckily Titanium supports this function. However in Android device, this function has some strict requirements

Method: Titanium uses official Android language localization method. It means that each property's name cannot have space, symbol or anything else except basic letters. In the application, there were some properties named with space or symbol, which was the reason the app didn't perform correct language localization. Once these requirements were met, the localization worked well.

8 System testing

8.1 Testing process

The testing is conducted after the design phase and the realization phase is done. The testing is only for the Domotica Frontend application and it will mainly focus on the functionality tests. There are two main stages of testing. They are project integration tests and user acceptance tests.

8.1.1 Project integration test

In this stage, there are three kinds of testing performed by different parties.

Functional testing: It is an objective to ensure that each element of the application software meets the functional requirements. In this testing field, the intern will also perform validation testing, which is intensive testing of the frontend fields. This includes following the GUI standards of the platform, valid, invalid or limited data input, the screen's look and appearance, and overall the consistency with the rest of the application.

Participant: This testing is done by the developer. The developer needs to make sure that all agreed functions are implemented in the final version of the Domotica application.

User testing: It is an objective to test if the program's functions perform as expected, and the UI elements are placed in the expected location.

Participant: The project owners can follow the test cases written in the test plan to do this testing.

Technical testing: It is an objective to make sure the program doesn't contain any critical errors, and try to minimize the bugs in the software as much as possible.

Participant: This testing is done by the developer

8.2 User acceptance testing

Description: It is an objective which will be performed by the client at the very end. Prior to that the developer also needs to perform these tests in order to ensure that the application achieves a fair percentage of "acceptance" by the client.

Participant: The client does this testing to test if it is an acceptable application.

8.3 Testing priority

The table on the right side shows the priority of the four kinds of tests. Technical testing is performed at first to make sure the application is stable. Followed by functional testing which ensures that all requirements are included and the user testing can double check from the project owner's point-of-view. After the integration test is done, the acceptance testing can be performed by the client to see if he is satisfied with the final version of the application

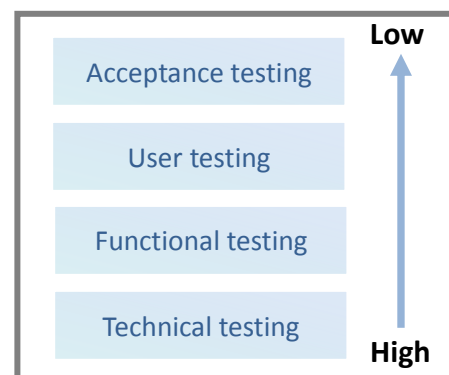


Diagram 8-1: Testing priority

8.4 Testing result

Due to the Domotica application not being fully programmed yet, the testing result will be mentioned in the final presentation if possible.

9 Conclusion & Recommendation

9.1 Conclusion

Titanium, PhoneGap, and MoSync, as the top three frameworks, were used to create test applications in the investigation second round. Overall, PhoneGap is a simple framework which supports many platforms but at the cost of performance. Titanium with its stable IDE is easy to use, and gives excellent application performance. Lastly MoSync is a flexible framework. It gives more options in programming languages and IDEs, while offering good application performance. In the end Titanium was selected for the Domotica application. And MoSync was recommended to the company.

When the final report is finished, most of the tasks in this graduation project are accomplished. First, the investigation, which is the most important and influential part in this project was completed in the middle of the 4th month. The estimated time for the investigation component before the project started was three months. So there would be two months left for the “Domotica Frontend” application and school work. However, when the investigation started, the client decided to extend the investigation by adding an extra round of testing. That is the reason why the Domotica application was postponed for half month. Nevertheless, the investigation report and four test applications have been delivered as promised. The Domotica application also has its “must do” requirements implemented. By the time this final report is done, the application will have most of its “should do” requirements finished. In general, the tasks formulated in each sprint were completely finished on time with good quality.

9.2 Recommendation

The mobile market is growing very fast and so are the cross platform mobile frameworks. Cross platform mobile frameworks release newer version frequently to fix some bugs and add more functionalities to become stronger and to remain of value. At the same time, some of the frameworks might be discontinued, and some new frameworks will arise as well. So it is wise to update the information for each framework regularly, or at least ICT Automatisering should keep an eye on the status of other frameworks and compare them against MoSync.

In the end, it should be mentioned that before deciding a framework, it is really important to know what kind of application you want to develop and which functions are expected to be necessary in this application. It is quite difficult to say which framework is the best one directly. They all have their own highlights and weaknesses. It is better to select a few frameworks with features that match the application needs. Then what really matters mostly are the stability and maturity of the framework.

Evaluation

Overall, I think I achieved most of my objectives that I made in the beginning of the project. In this case, I would like to say that in general I did a good job during this graduation project.

Unlike my previous internship, this graduation project was well organised and controlled under the Scrum method. This time I felt quite comfortable doing the tasks, because the specific tasks were only scheduled for two weeks. I didn't have to worry too much about anything except focusing on the current sprint tasks.

However, the investigation took a longer time than it was initially planned because of the extension. I was quite nervous and stressed about the limited time remaining. I was afraid that I couldn't finish all remaining tasks before the internship ended. At school, I always finished projects and assignments on time. In this case, nothing can be done, as what is still planned would make me overwhelmed. It also affected my working attitude and quality. Fortunately, my company tutors realised this problem and encouraged me to just focus on the current task to ensure its quality. I was also learned how to deal with projects that are delayed, since in reality, most of the projects cannot be delivered on time or experience some kind of delay. It was a good lesson for me to get ready to work as a professional after graduation.

Technically, I expanded my knowledge of mobile application. And more importantly, I learned the new trend technique "Cross platform mobile frameworks". It was an interesting subject and I think it will be more widespread in a few years. It is always good to learn new techniques in advance.

On the other hand, I also improved my presentation skill because of several sprint presentations. The key is to make clear who the audience is and what they would like to hear from the presentation.

One thing still needs to be improved is my social skill. I started to socialise with some colleagues at work sometimes, but most times I wasn't active enough in taking the initiative to start a conversation. I think I already made the first steps in this internship and I truly hope that I can improve more when I start to work as an employee in a company.

Reference

[1] Android definition

Link: [http://en.wikipedia.org/wiki/Android_\(operating_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))

[2] iOS definition

Link: <http://en.wikipedia.org/wiki/iOS>

[3] Framework definition

Link: http://en.wikipedia.org/wiki/Software_framework

[4] Framework definition

Link: http://en.wikipedia.org/wiki/Apache_Subversion

[5] JQuery Mobile definition

Link: http://en.wikipedia.org/wiki/JQuery_Mobile

[6] ARM definition

Link: http://en.wikipedia.org/wiki/ARM_architecture

[7] Make your projects more adaptive with Scrum

Author: Joe Wilson

Links: <http://blog.clickhere.com/category/production/>

[8] PhoneGap – How does it work

Author: Arnab

Link: <http://arnab.ch/blog/2011/12/phonegap-how-does-it-work/>

[9] Profile ICT

Link: <http://ict.eu/about-us/organisation/>

[10] Getting started with Titanium Studio

Author: Paul Dowsett

Link: <https://wiki.appcelerator.org/display/tis/Getting+Started+with+Titanium+Studio>

Appendix A

Project survey



Author name: **Yaxiong Shen**
Document type: **Investigation report**
Date: **29-January-2013**
Location: **Son**

Description of the graduation project:

Describe the problem analysis:

Domotica is growing in popularity to control devices in a house. Within ICT we have a simulation that we want to expand. We have a dollhouse with some Xbee modules in it that can be controlled from a server (for example to control lights). On this server a web service is running, that allows (mobile) applications to control the modules.

Currently, there is a prototype (Android) application that can turn a light on or off. What we would like to have is a cross-platform mobile application that can communicate with the web service, to control these modules. Functionality you can think of is:

Turn lights on/or off

Read the status of the lights

Change the colour of the lights

Implement a schedule which automatically turns lights on or off on a preconceived time, as long as the mobile device is within the WIFI range.

Implement specified profiles for different people

Control a motor to for example open or close curtains.

Describe the graduation assignment:

Creating of:

1. Documentation of the investigations with a conclusion which is the best option
2. First version of the mobile application which can at least control the lights
3. Prepared for further development
4. Design document of the application
5. Test application / results

What is the research component of this assignment?

Do an investigation to the possibilities of cross-platform mobile app development.

Focus on:

Supported platforms

Which frameworks/solutions are there on the market?

Possibility to build a native application

Performance

Use of device sensors

Use of hardware acceleration / OpenGL

What are the methods and tools?

Agile SCRUM

Object Orientation

UML

Programming language dependent of research (HTML5, JavaScript, CSS3, or Java, Objective C, C#)

How and by whom will the intern be guided by the company?

Patrick de Ru

(Technical coach)

patrick.de.ru@ict.nl

Ton van Tetering

(Process coach)

ton.van.tetering@ict.nl

What fields of Study play an important factor in realizing the graduation assignment?

Research

Information analysis

Requirements specification

Design

Realization

Testing

Appendix B

Project plan



Student Name: **Yaxiong Shen**
Student number: **2136278**
Internship type: **Graduation Internship**
Internship Company: **ICT Automatisering**
Date: **09-February-2013**
Place: **Son**

Table of Contents

Introduction.....	38
Project statement.....	39
Formal client.....	39
Project leader	39
Current situation	39
Project justification	39
Project product.....	39
Project deliverables and non-deliverables.....	39
Project constraints	39
Project risks	40
Project phasing.....	41
Management plan	43
Propose.....	43
Skill.....	43
Quality	44
Information.....	46
Time	48
Organization	48

Introduction

In the last year of study at Fontys, the intern needs to do a graduation project from February to the end of June. The company which offers the intern this opportunity is ICT Automatisering. The project at this company is Domotica frontend.

Domotica is a home automation system, this word comes from the combination of the words domus (Latin, means house) and policy (Greek, meaning “that works by itself”). If you embed this system into all household appliances, you can, through PCs, mobile phones or tablets, easily control all lights, doors and curtains. Also you can monitor the working condition of all appliances at home.

This assignment is composed of a research part and a programming part. Specifically, what the intern needs to accomplish is to set up an investigation to find out a suitable framework for the company and then program a tablet application using this framework that can be installed on different mobile platforms. (For instance, Android, IOS or Windows system.)

This is the intern’s second internship since last year; with the experience from the previous internship, the intern should have more confidence to start the graduation internship.

Project statement

9.3 Formal client

The formal client of this project is ICT Automatisering

9.4 Project leader

Patrick De Ru and Ton Van Tetering are the project leaders. Patrick is in charge of technical guidance. And Ton takes care of the project process-related matters.

9.5 Current situation

There is a dollhouse with some Xbee modules in it that can be controlled from a server. On this server, a web service is running that allows a mobile application to control the modules. Currently there is an Android prototype application that can turn the lights in the dollhouse on or off.

9.6 Project justification

Domotica is growing in popularity to control devices in a house. Within ICT they are building mobile applications for different platforms, and the company wants to have a cross-platform mobile application that can communicate with the web service to control these modules in the dollhouse. Therefore the investigation result is really valuable and interesting for the company

9.7 Project product

There should be an investigation report to give the research result to tell what the possible frameworks are for cross platform development and which the more suitable one is.

If the investigation result is positive, a tablet application which can be used on multiple platforms should be programmed. If the investigation result is negative, one or two independent tablet applications should be programmed. However, in both cases, the intern is responsible to accomplish the entire flow of the application from setting requirements, designing, development and testing.

9.8 Project deliverables and non-deliverables

Project deliverables:

1. URS document for investigation
2. Investigation report
3. URS document for application
4. Design document for application
5. Test document for application (Test plan + Test report).
6. Presentation and demonstration for the project

Non-deliverables:

1. No manual provided
2. No course provided
3. No related hardware and devices provided

9.9 Project constraints

Using agile Scrum (an agile software development framework to manage software projects that is good for projects which are difficult to plan ahead) is required during the internship period.

The programming tools depend on the framework investigation result; the required tools are different for each framework.

The programming language will be decided after the investigation. It could be HTML5, JavaScript, and CSS, Java, or C #).

9.10 Project risks

It is always safe to consider the potential risks in the project. Different risks can occur due to many reasons. However, it is more important to find out the solutions. The following table indicates the content, level, and frequency of the risks; it also gives the possible solutions.

Risk	Risk level	Frequency	Methods to avoid the risk
Tasks cannot be done on time which delays the current phase	High	Not often	To use Scrum correctly, attend the daily meeting to check the current process and what if there is any bottleneck. Adjust the phase period or the amount of the tasks in that phase if necessary
The project leaders cannot attend the daily meeting	Medium	Not often	Alternative ways can be used like communicating through E-mail or Lync
There is no suitable cross platform mobile framework after the investigation	High	One	Make a plan in advance if this risk happens. Decide the number of the independent applications to make instead.
The client adds new assignments while the basic phases are not done yet	Low	Not often	It is important to agree to accomplish basic phases first. Extra assignments will only be considered by the available time left and difficulty of the assignment.

Table 1: Project risks

Project phasing

This chapter shows all needed activities and expected documentation to get to the end of the project in a certain specified time area.

This assignment is made of two parts, an investigation part and a programming part. As you can see from the diagram below, phase1-3 belongs to the research part. After the research result gets approved by the clients, the programming part gets started; this is from phase 4 to phase 7. There are two phases both for definition, the first one is for investigation, and the other is for programming.

At the end of each phase, you can see all expected deliverables. It's really important for all participants to know what activities are completed and what deliverables are produced. In this way all participants can have a clear view on the state of the project.

Since the project is managed and controlled under the Scrum development model, there is no estimated time for tasks or phases at this point. In each two weeks, there will be some tasks expected to be completed. And after that, the new tasks will be planned depending on the state of the project at that time and the priority of the client.

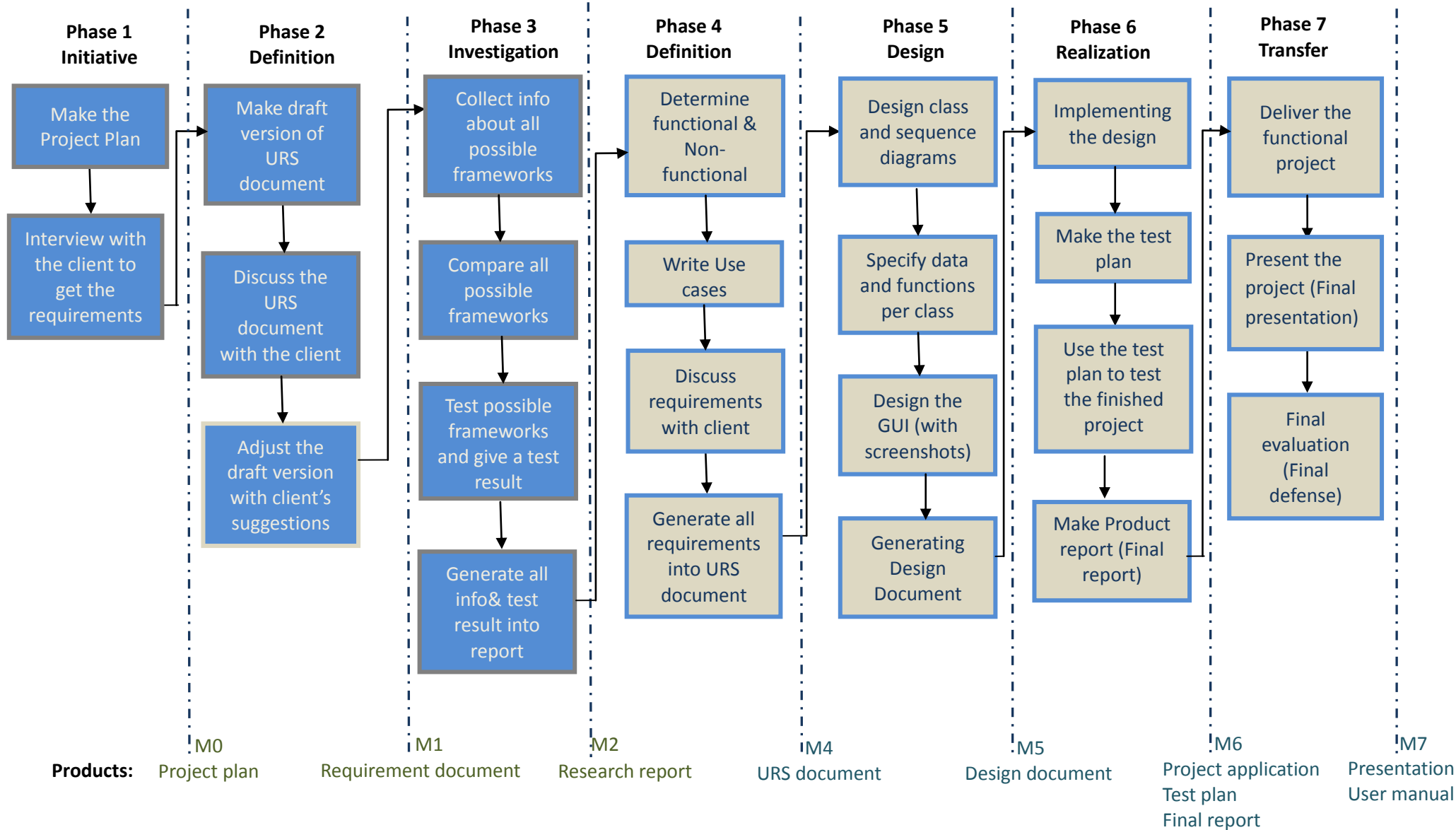


Diagram 1: project phasing

Management plan

Propose

This is a plan for the whole project to capture the entire project end-to-end, covering all project phases, from initiation through planning, execution and closure. A good management plan can make sure that the process and the final product of the project get a high quality. In this chapter, there are five elements that will be discussed.

Skills

This section describes what skills are needed during the different phases of the project. Analyzing the necessary skills in advance can offer the intern more time to get ready for each phase. For example, if there is any skill in a certain phase that the intern is not familiar with, he can learn and prepare for this skill before that phase get started.

Phase 1: Initiative & Phase 2: Definition & Phase 4: Definition of the application

Phase 1 is the planning part of the project. At the beginning, it is important to get familiar with the company, make the project plan, and get the investigation requirements from the client through an interview.

In phase 2, it's meant to write requirements document for the investigation, show this document to the client and adjust it with the client's suggestions.

Phase 4 gives the user requirements specification document for the application after the interview with the client.

These three phases have the same skill needed, as what you can see below:

1. Ability of making plans.
2. Can foresee problems
3. Be proficient of using Microsoft Word.
4. Good communication skill
5. Be able to make the use cases

Phase 3: Investigation

The intern needs to collect information of all possible frameworks and describe them in the final report. If necessary, the intern also needs to program some small applications to test the functions of some possible frameworks to give more specific analysis for the investigation report.

The following points are the necessary skills for phase 3:

1. Be proficient of using Microsoft Word
2. Capable of using programming language (depends on the specific framework, could be C#, HTML 5, or something else).
3. Have good conclusion skill and writing skills
4. Being objective
5. The skill to compare different frameworks.

Phase 5: Design

This is the last phase before programming. In this phase, there will be some diagrams and design that have to be accomplished.

This phase needs the following skills:

1. Be able to use a certain software (such as StarUML) to draw the class and sequence diagrams.
2. Knowledge of UML.
3. Capable of designing the GUI.

Phase 6: Realization

In realization phase, the implementation of the application should be done, following with a test plan to test the application whether it's functional or not.

This phase needs the following skills:

1. Be able to use a platform from a certain framework (which can be sure at the beginning of the project)
2. Familiar with the platform and the programming language of that platform.
3. Capable of programming
4. Can make suitable test plan to test the application

Phase 7: Transfer

Deliver the final product and give a final presentation for the company and school.

The following skills are required:

1. Be proficient of using Microsoft PowerPoint.
2. Be able to demonstrate the final product.
3. Good defense skill.

Quality

Quality management will ensure the effective design of processes that verify client needs, plan product life cycle and design, produce and deliver the product. There are 4 quality management principles mentioned in this session, it gives a general view of how the quality of the project can be guaranteed.

Principle 1: Customer focus

Since the client trusts the intern and offers him an opportunity to work on this project, the intern should understand the customers real needs, accomplish the client requirements for the investigation and application, and strive to meet the client's expectations.

Activities:

1. Two times interview with the client can help understand the client needs and expectations.
2. The requirements document can make sure the requirements the intern understand from the client is correct.
3. The use of bugtracking system: Trac is a web-based project management and bug tracking system. It allows for people to submit bugs and also has other project management related elements. It is easy to keep track of any issues that the application might have.

Using a bugtracking system can improve the application's quality because it can show the current state of the program in a clear way. The most important thing is that the users or other developers add bugs. It is

useful especially for the users to gather a list of which issues they might encounter when using the application. Often it might not be a real bug, but just some parts of the application working differently than the user expected. If the same bug/issue is reported by many users, then it may be necessary to consider changing something.

Principle 2: Leadership

Leaders (in this case, are company tutors) establish unity of purpose and give direction to the intern. They help the intern to check if the intern achieves the client expectations.

Activities:

1. They provide the intern with the required resources, training and freedom to act with responsibility and accountability.
2. Setting clear goals and target to the intern
3. Inspiring and encouraging the intern.

Principle 3: Process approach

The result of the final product can meet the client's expectations when the activities and recourses are managed as a process. In the project, the client uses Scrum approach to monitor the project process.

Activities:

1. Each two weeks, the finished tasks will be viewed and new tasks will be arranged. It helps to make sure the project process is in the correct direction.
2. The daily meeting checks if the intern has any bottlenecks that might delay the process
3. The process report for each two week ensures that the intern is in the correct flow of the project.

Principle 4: Factual approach to decision making

Effective decisions can save much more time for a project. It based on the analysis of the data and information

Activities:

1. The URS documents and the design document can ensure that all useful data and information are used in valid methods
2. The investigation report helps the intern make decision more easily to choose which framework is the more suitable one.
3. The presentation over each two weeks can help make more sufficient decision about the tasks for the next two weeks.

Principle 5: Development tools & Workflow

1. The usage of Subversion (often abbreviated SVN). It is version control software. It is used to maintain current and historical versions of files such as source code, web pages, and documentations. Trac has its ability to interface with Subversion.

This software is clear for the developers to keep track of different versions on the project. At the same time, it remembers all the previous versions, so it's easy to roll back to a last version of the project. In general, it improves development by keeping the code centrally organized.

Information

This section gives information about all documents needed during the internship period, including the necessary contents and deadline for each document, and to which party it should be sent. It also shows the planned meeting and presentations during these five months.

1. Intern-> School:

- **Project plan:** This report consists of project statement, project phasing and management plan. During the first company visit, the university tutor needs to approve this plan, and he will especially keep an eye on the content of the assignment, the structure of the project plan and the way in which the assignment will be tackled. Also whether the graduation project objectives will be achieved is one the aspects the university tutor will consider.
Deadline: the end of the second week (Before 15th of February).
- **Communication plan and invitation letter:** In the communication plan, it describes how the intern intends to communicate with the university tutor (by telephone, email etc.) and when. The invitation letter functions as confirmation. Both documents have to be sent with the project plan to the university tutor.
- **Final report:** it shows what the intern did and how and especially why.
Deadline: Before 14:30, 8-June-2013.

2. Intern-> Company

- **Project plan:** This report consists of project statement, project phasing and management plan. The company tutor guides the student in this process and needs to approve the project plan.
Deadline: The end of the second week (Before 15th of February).
- **Process report:** This document has to be written every two weeks to describe how the process going.
Time: Every two weeks
- **For the investigation:**
 - User requirement specifications:** This document gives the functional and non functional requirements from the client. It provides a high level description of the user's expectation of the investigation.
Time: It should start after the interview with the company tutors.
 - Backlog:** Gives general categories to show the plan for the investigation. Also it gives the time prediction.
Time: It should start right after the user requirement specification.
 - Investigation report:** it gives the result of the investigation. This document needs to be approved by the company tutors and will be put into the final report as research chapter at the end of the internship.
Time: Before the application gets started.
- **For the application:**
 - User requirement specifications:** This document consists of functional and non-functional requirements and use cases. It gives an overview of the user requirements to the application. For example, how the client wants the application to look like, the functions

Time: It should start after the investigation result gets approved and after the interview with Patrick and Ton.

b. *Design document:* It consists of the application's prototype, class diagram and sequence diagrams.

Time: After the user requirements specifications document is accomplished and approved.

c. *Test plan & test report:* The test plan describes how the client expects the application gets tested. It mainly consists of the test phase, testing strategy and the test cases. The test report is the result after the intern tests the application followed by the test plan.

Time: It should start after the design phase and coding phase is done.

Table 2 specifically shows the responsibilities of different party to different documents

Role	Project plan	Invitation letter	Day declaration	Process report	URS document	Design report	Test plan	Final report
Company tutors	RE,A	R	A	RE	RE,A	RE, A	RE, A	A
School tutor	A	R						RE, A
Intern	DR, S	DR, S	DR, S	DR	DR	DR	DR	DR, S
Fontys stagebureau	R		R, AR					R, AR
legend								
Full name	Approve	Archive	Review	Draw up	Send	Receive		
Short name	A	AR	RE	DR	S	R		

Table 2 Information table

3. Meetings and presentations:

- *Daily meeting:* it's an internal meeting with Patrick and Ton.
Time: Around 9:30 every day.
- *Internal presentation:* Gives the demo to show what have been done and what is missing for the past two weeks.
Time: At the end of the second week ever two weeks.
- *Company visit:* The school tutor will visit the company to talk about the project process with the intern and the company tutors two times during the internship.
Time: 1st time: it should be at latest in week 5 at the latest when the internship starts. The specific time will be informed through email.
2nd time: The second visit must be planned at the end of the graduation project period. The specific time will be informed through email.
- *Final presentation:* The presentation with demonstration will last not more than 30 minutes. The defense will start after the presentation. It can last approximately 20 minutes.
Time: it can be scheduled from June 21st until July 4th

Time

Time section gives the estimated time to a certain phases. Due to the usage of Scrum, it's difficult to estimate the working hours for specific tasks. Instead, the 7 phases mentioned before are concluded into 4 major aspects and give them estimated hours. The four aspects are preparation, investigation, development and documentation respectively. Table 3 shows the plan in details:

Major	Related Phase(s)	Estimated hours
Preparation	Phase 1: Initiative	<= 80 hours
Investigation	Phase 2-3: Definition & Investigation	~ 400 hours
Development	Phase 4-6: Definition & Design & Realization	~ 240 hours
Documentation	Phase 7: Transfer	<= 80 hours
Total working hours: Around 800 hours (40 hours/week* 20 weeks)		

Table 3 Estimated hours

Organization

This section gives a view of the organization through the whole company. It is essential for all participants to have a clear view over their own tasks and responsibilities. You can see the organization in Diagram 2 below.

Responsibilities:

project manager: Check the process of the project and give suggestions. Check the documentations. Give requirements about the investigation and the application.

Intern: Write all necessary documents and make the table application. Be responsible to contact the school tutor with the project leaders in the company.

School tutor: Contact with the intern and check from the project managers about the process of the project.

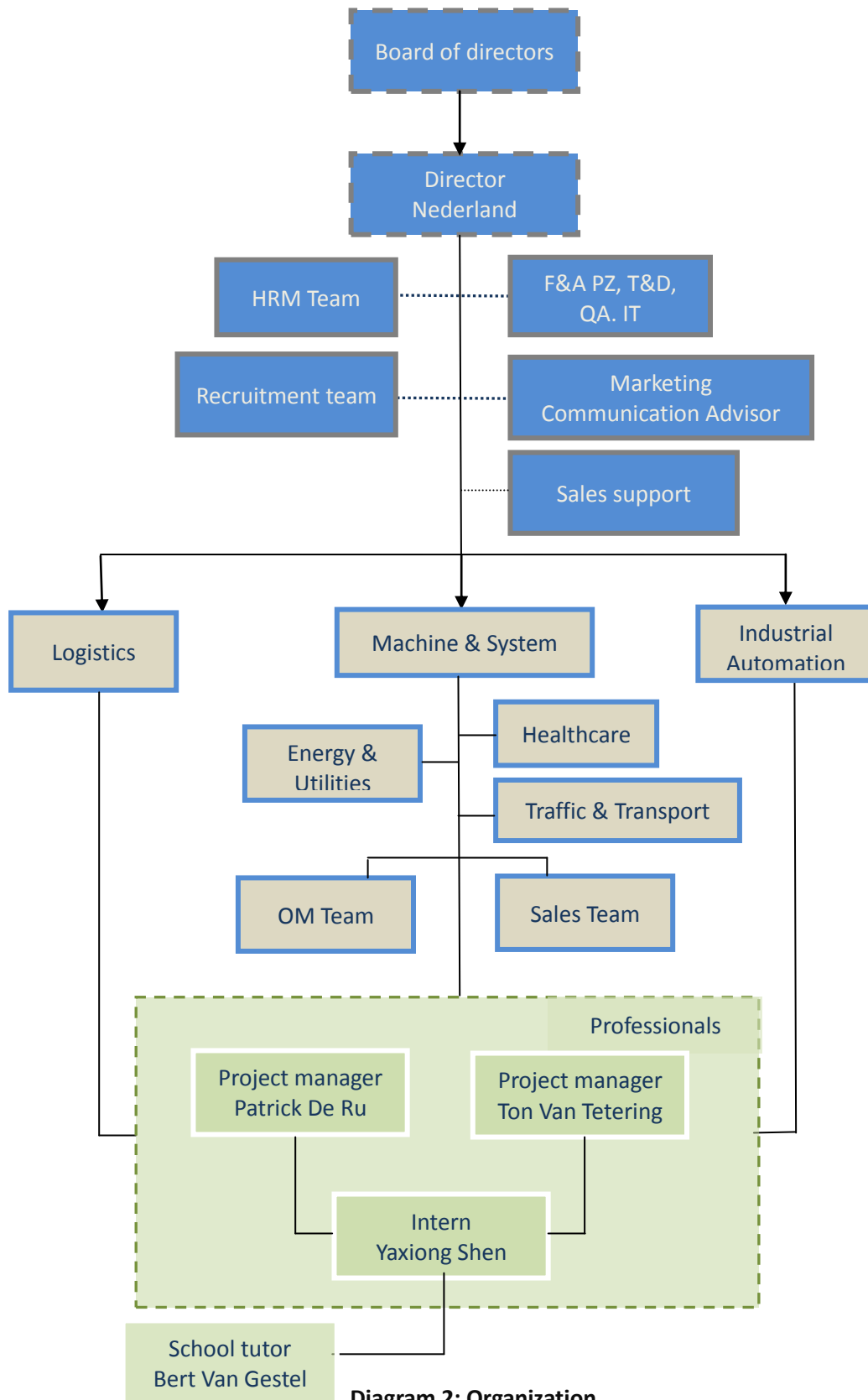


Diagram 2: Organization

Appendix C:

Investigation report



Student Name: **Yaxiong Shen**

Student number: **2136278**

Internship type: **Graduation Internship**

Internship Company: **ICT Automatisering**

Date: **17-May-2013**

Place: **Son**

Table of Contents

Introduction.....	55
About the author	55
Topic background	55
Topic description	55
Framework scoring.....	56
Introduction	56
Criteria.....	56
Framework overview with 1st round scoring.....	58
PhoneGap	62
Company	62
Price.....	62
Development & Environment.....	62
How it works	62
System support	62
Produc.....	62
Review.....	62
Titanium	64
Company	64
Price.....	64
Development & environment	64
How it works	64
System support	64
Product... ..	64
Review.....	64
MoSync	66
Company	66
Price.....	66
Development & Environment.....	66
How it works	66
Support system	66
Produc...t	66
Review.....	66
Rhodes.....	68
Company	68
Price.....	68
Development & Environment.....	68
How it works	68
Support system	68
Product... ..	68
Review.....	69
Marmalade	71

Company.....	71
Price.....	71
Development & Environment.....	71
How it works.....	71
Support system	71
Product.....	71
Review.....	72
Xamarin.....	73
Company.....	73
Price.....	73
Development & Environment.....	73
Support system	73
Product.....	73
Work architecture	73
Review.....	74
AppMobi.....	75
Company.....	75
Price.....	75
Development & Environment.....	75
Support system	75
Product.....	75
Work architecture	75
Review.....	76
Corona.....	77
Company.....	77
Price.....	77
Development & Environment.....	77
Support system	77
Product.....	77
Work architecture	77
Review.....	77
MonoCross	79
Company.....	79
Price.....	79
Development & Environment.....	79
Support system	79
Product.....	79
Work architecture	79
Review.....	79
First round result with score.....	81
1 st round description.....	81
1 st round Conclusion	81
For ICT Automatisering	81
For game development.....	81

For non web-based technology developers.....	82
Conclusion.....	82
Test round overview for PhoneGap.....	83
Testing application result.....	83
Criteria.....	83
Include functionalities	83
Support & documentation	83
Maturity of framework	83
Code reusability	83
Testability of framework.....	84
Flexibility.....	84
License costs.....	84
Performance.....	84
Security.....	84
Difficulties to create applications	84
Updatability.....	84
Distribution and installation.....	85
Review and conclusion.....	85
Test round overview for Appcelerator Titanium	86
Testing application result.....	86
Criteria.....	86
Include functionalities	86
Support & documentation	86
Maturity of framework	86
Code reusability	86
Testability of framework.....	87
Flexibility.....	87
License costs.....	87
Performance.....	87
Security.....	87
Difficulties to create applications	87
Updatability.....	87
Distribution and installation.....	87
Review and conclusion.....	88
Test round overview for MoSync	89
Testing application result.....	89
Criteria.....	89
Include functionalities	89
Support & documentation	89
Maturity of framework	89
Code reusability	89
Testability of framework.....	90
Flexibility.....	90

License costs.....	90
Performance.....	90
Security.....	90
Difficulties to create applications	90
Updatability.....	90
Distribution and installation.....	91
Review and conclusion.....	91
Recommendation & Conclusion	92
Final score	92
Recommendation & Final choice	92
Final conclusion.....	93

Introduction

About the author

Yaxiong Shen is an intern who is charge of Domotica frontend project. He comes from China, born in Nanjing which is the capital city of the Jiangsu Province. He came to the Netherlands when he was 18 years old. One year later, he started my information and communication technologies study at Fontys University. The reason why he chose the Netherlands to study is because it is an international country, its multi-culture makes him be more international and have more understandings about many cultures from different regions in the world. His hobbies are playing badminton, tennis, music in general, and photography.

Topic background

Due to the growth of the market of smart phones recently, mobile applications are getting extremely popular. The market is fragmented because of the large number of mobile platforms.

Consider the following scenario: a customer asks a developer to develop an iPhone application. Everything goes well till the customer realizes that an Android version of the same application is also necessary. Then the software developer has to recreate the same application for a different platform. However, what if the customer decides that Windows Phone should also be supported? Here starts an idea that developing a mobile application for multiple platforms at once would be a time saving task. It will help the developer save a lot of development time and decrease the cost for the customer. There are a number of frameworks offering to enable cross platform mobile application development.

Topic description

ICT Automatisering would like to have a cross platform mobile framework for their mobile development. This investigation project will give an overview of all potential frameworks. Then all frameworks will be judged by a list of criteria that are initially done using only the information available online. Based on the results of the first round, a test application will be designed for the top three frameworks to test the frameworks support for device functionality and various other aspects. These three will again be judged with the criteria, but this time based on the experience of actually creating an application in each. In the end of the investigation, the author will give an answer about which framework is the most suitable for the company.

Framework scoring

Introduction

The potential frameworks will be scored by the weighted score system. This system is based on a list of criteria which are summarized by the author and evaluated a score on a scale of 0 to 5 (where 0 is the lowest possible score and 5 the highest) by the company. The final score of each criteria will be its base weight multiplied by the evaluated score. The final score of each framework will be the sum of its criteria scores.

Using a weight scored system is helpful to evaluate a framework in a more objective and comprehensive way. It avoids the author analyzing the framework using his personal opinions. It also lets the readers understand the functionalities of each framework in a more clear way.

Criteria

The table below is a list of criteria, with short explanation and base weight for each:

Criteria	Explanation	weight
Included functionalities	What functionalities and device features does the framework support (e.g. camera, accelerometer and NFC)?	6
Support & document	Is the documentation of the framework good enough?(e.g. FAQ and official documentations)	8
Maturity of framework	How secure is the framework and what are the recent experiences of the other developers?	12
Code reusability	Is it easy to reuse the existing code in other projects?	3
Testability of framework	Is it easy to test the applications made by the framework? (e.g. debugging or profiling)	9
Flexibility & Source license	Is the framework flexible enough to be customized or expand the framework with other tools and functionalities? Is it open or close source?	4
License costs	How much does the license cost and can the created applications be used commercially?	11
Performance	What is the performance of the created applications (For example, what is the runtime compared to the native instructions?)	10
Security	Are created application secure enough? Do they support Https connections? Do they offer code obfuscation?	5
Difficulties to create applications	How long and how much effort would it take to learn how to use the framework with enough skills to develop a good application?	7

Updatability	Is it easy and convenient to have a good update for the development tools?	2
Distribution and Installation	Is it easy to distribute and install the created application? (Automatically signed or native installer format?)	1

Framework overview with 1st round scoring

The chapter gives an overview for all potential cross platform mobile frameworks that can be used for development.

Due to all information used for this interview being collected from the frameworks' official websites and the accompanying documentations, it is assumed that the information offered about a framework is accurate. The purpose of the initial overview is to select three more suitable frameworks to do further testing in the second round.

The list below shows all frameworks present in the overview:

- PhoneGap
- Appcelerator Titanium
- Mosync SDK
- Rhodes
- AirPlay SDK
- Xamarin
- Xamarin + MonoCross
- AppMobi XDK
- Corona

There are three tables in the appendix to show all frameworks' features and capabilities in different aspects. It helps the readers to see each framework's superiorities and weak points in an easy compare overview.

Please refer to Attachment A (Framework features), Attachment B (Framework system support) and Attachment C (Framework development & environment).

All frameworks mentioned above can actually be summarized into three main options for the cross platform mobile application development based on the different work architecture. The following three diagrams show these three options with some descriptions. The pro and cons will also be mentioned.

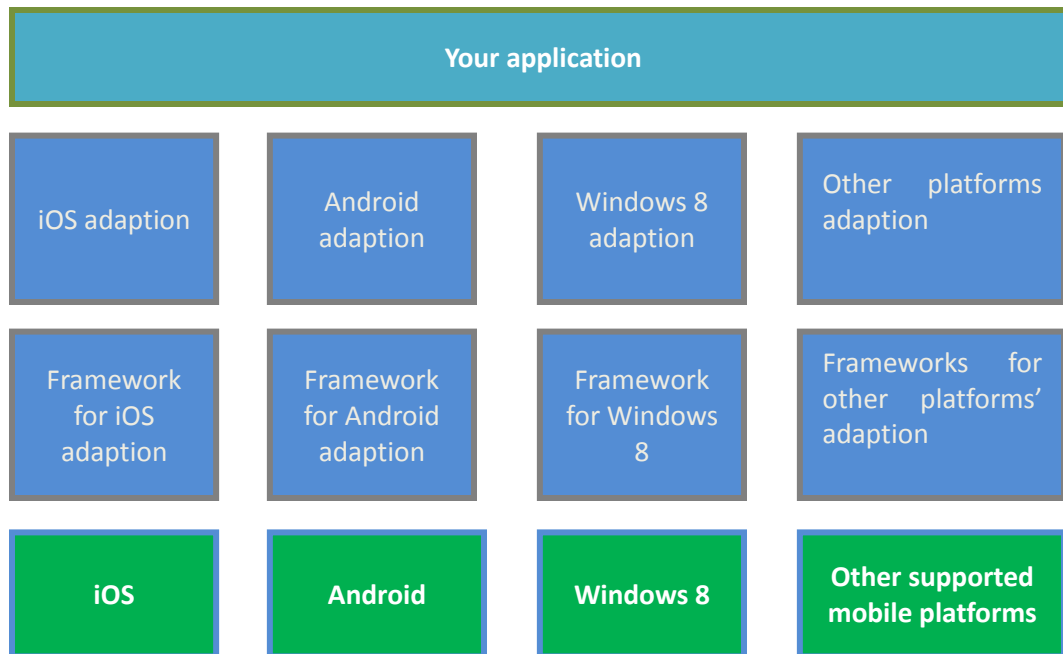


Diagram 1: Option 1

Option 1 shows the traditional method for cross platform mobile framework development. By using an adaption framework, the developers can code with the framework specific programming language such as C# instead of using a specific mobile platform language such as Java or objective C. For example, if a developer wants to develop a mobile app for both iOS and Android, he/she can use a certain adaption framework to program an iOS app using the framework specific language such as C#. Then because the iOS app is written by C#, the developer can reuse many parts of the code from the iOS app and develop an Android app using Android adaption framework which also supports C# development.

Pros:

1. No need to learn the mobile platform specific language
2. Most of the code can be reused for other mobile platform apps

Cons:

1. Have to compile and generate the code yourself if you want to reuse some code for another project
2. Pay twice if the developers use an iOS adaption framework and an Android adaption framework which are licensed separately.

Example frameworks:

1. Xamarin.iOS
2. Xamarin.Android.

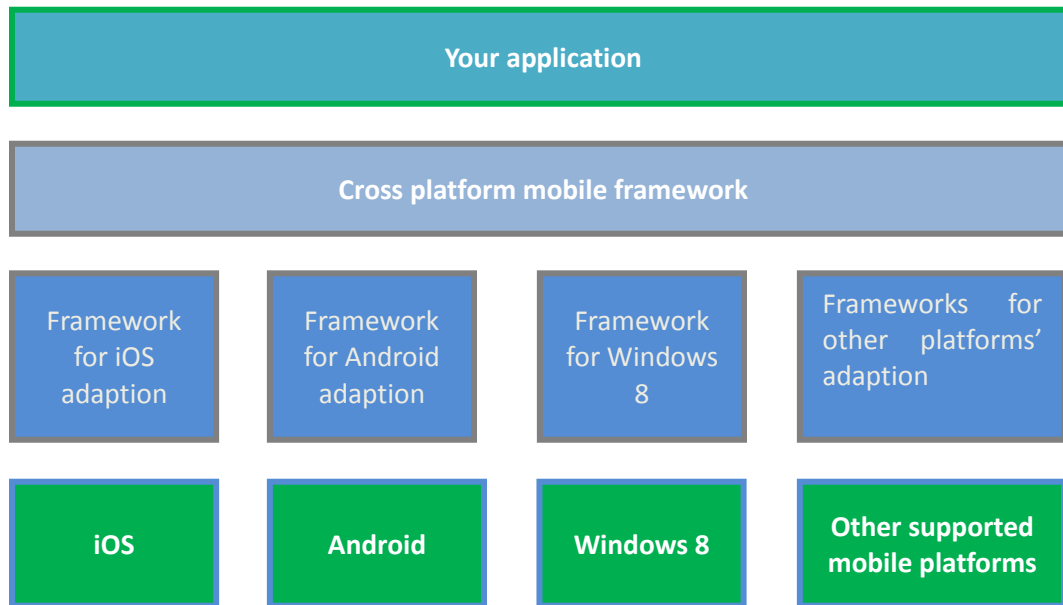


Diagram 2: Option 2

Option 2 (shown in Diagram 2) is a shared architecture, it is similar with option 1 but it is actually a simplified version. There are some cross platform mobile frameworks that are not standalone frameworks, but work with other adaption frameworks. With this kind of combination, developers don't need to rewrite code for each mobile platform themselves. The cross platform mobile framework does it all for you. For example, if a developer wants to develop an app for both iOS and Android, he just needs to install the framework and any tools or SDKs the framework depends on. Then he can develop an iOS/Android app using the framework specific language, which can be for each target platform.

Pros:

1. No need to learn the mobile platform specific language
2. Most of the code can be reused for other mobile platform apps
3. Save a lot of time developing an app for different mobile platforms
4. Quite flexible option. The cross platform mobile framework can be changed if it is discontinued or the developer is not happy with the performance of that framework.

Cons:

1. Could be high in cost if the mobile platform adaption frameworks and the cross platform mobile framework don't come from the same supplier.
2. Double integration could cause some technical issues, such as connection problems or library conflicts.

Example frameworks:

1. PhoneGap

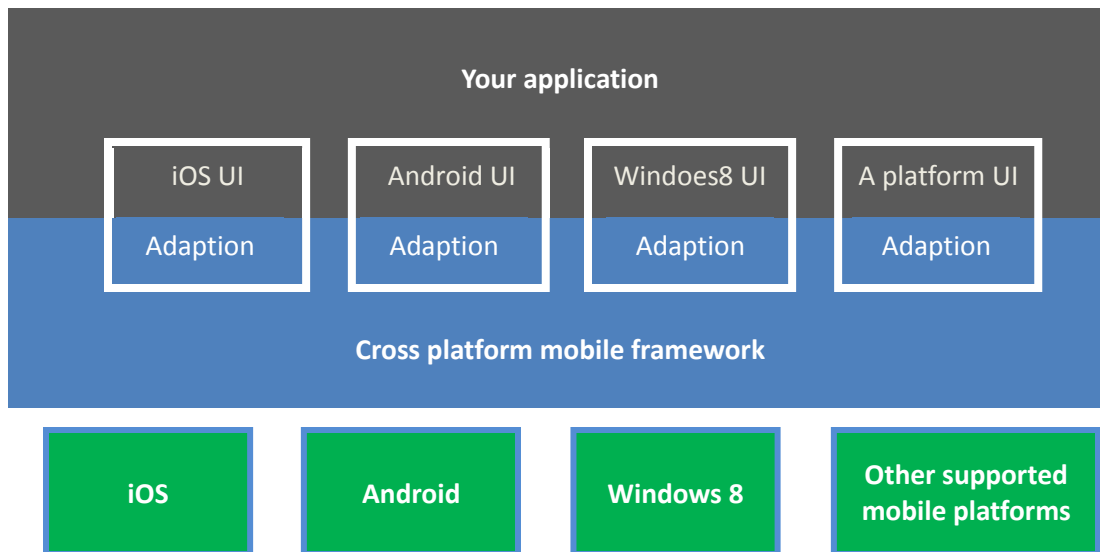


Diagram 3: Option 3

Option 3 is a simplified version of option 2. Basically, this option is more like “Write once, build to different mobile platforms” (although it is not entirely true, that will be discussed specifically later). This kind of framework supports developers to develop a mobile app only once using the framework specific language, which can be compiled for each target platform. There are in general two common kinds of frameworks. With the first kind of frameworks, developers only need to create one project for all target platforms, although sometimes they need to write different functions for specific platforms. For the other kind, developers need to create independent projects for each target platform, and write code separately, although most of the code can be shared for each platform. On the other hand, option 3 also supports the developers in designing the app’s UI. The frameworks in option 3 support this with the platform specific UI adaption. Usually it allows the developers create a cross platform app with a native UI for each platform.

Pros:

1. Much easier for developers, they can finish the whole development process (from coding to compiling) at once.
2. Easy deployment especially with the new cloud-based tools
3. Support native UI for supported platforms
4. More choices for development language

Cons:

1. Project completely relies on one framework, if the framework is discontinued, the project you created with this framework will be in a big trouble
2. Frameworks with native UI support less platforms

Example frameworks: Titanium, MoSync, Marmalade, Corona, MonoCross

PhoneGap

Company

PhoneGap is produced by Nitobi. The new features and bug fixes are committed frequently to the repository. This company is based in Canada.

Price

PhoneGap is a free and open source framework. Developers and companies can use PhoneGap for mobile applications that are free, commercial, open source, or any combination of these. The code was contributed to the Apache Software Foundation (ASF) under the name Apache Cordova. It will always remain free and open source under the Apache License, Version 2.0. It has extra support packages for individuals and enterprises.



Development & Environment

It uses straight up JavaScript, HTML5, and CSS3 for programming. The resulting applications are hybrid applications. Instead of an IDE provided, PhoneGap only offers a SDK.

How it works

A PhoneGap application is actually a native-wrapped web application. PhoneGap SDK packages the developer's web app into a native app. With the native app, the app's UI consists of a single web view that consumes all of the available space on the device's screen. When the app gets started, the app's startup page is loaded into the web view and then it passes control to the web view for the user's interaction with the web app. As long as the user has interaction with the web app, links or JavaScript code will load other content from the resource files, or connect to the outside network and pull content down from a web or app server. PhoneGap also includes a JS library to access the device functionalities.

System support

PhoneGap supports almost all current mobile operation systems (such as, iOS, Android, Blackberry, Windows Phone 7, Symbian and Bada). It also supports many features for all operation systems. However, there are some features that are not supported on all platforms. Compass, for example, doesn't work on old version of iOS and Blackberry

Product

More than 400,000 developers are using PhoneGap. Thousands of applications built using PhoneGap are available in mobile app stores and directories, such as BBC Olympics and Walk & Buy. Also as Apple store has said yes to version 0.8 of PhoneGap, more and more iOS developers will try this framework out.

9.10.1 Review

PhoneGap is one of the most widely-used frameworks. Since it only uses basic web technologies, it is quite easy for developers to build mobile applications. Also it supports the most platforms compared to other frameworks. However, because all layout rendering is done via web views, the resulting application doesn't have the native UI look.

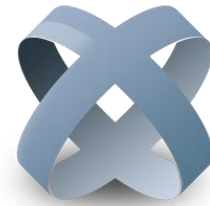
For the score of PhoneGap, please read the table below:

Criteria	Explanation	Score
Included functionalities	Support most features and systems	5
Support & document	Well-documented in its website	4
Maturity of framework	Widely-used and has good reputations	5
Code reusability	Easy to reuse the same code that is deployed with HTML5 enterprise desktop apps across other tablets and mobiles.	3
Testability of framework	Has its own test tool: Weinre	4
Flexibility	Plugin architecture: If it doesn't support the features the developer needs, the developer can easily write a Plugin himself	4
License costs	The SDK is free and open source, the support packages have costs	3
Performance	Different projects for different platforms makes it have less native performance. Using a web view makes it not have native performance	3
Security	Only support Https connection	2
Difficulties to create applications	Mainly basic web technology. Would be useful for people who have more experience with website design	3
Updatability	Regular version and Plugin updated. Some emergent bugs get fixed on time	4
Distribution and Installation	The resulting apps are packaged as the native apps. For example, PhoneGap Build is able to build "native app" which contains an .apk file.	4

Titanium

Company

Titanium is developed in 2006 by Appcelerator Inc., based in Mountain View, California. It has some investors: Storm Ventures, Mayfield Fund, Relay Ventures, TransLink Capital, Red Hat, INC., Sierra Ventures and eBay.



titanium™

Price

The price depends on the version. There are totally four versions available with different levels of support:

App Explore: No Appcelerator Enterprise extensions, only supports web. Working period is five day per week, and the response time is not estimated.

App Accelerate Standard: Extra supports chat, Critical bug fixes and 2-days response time

App Accelerate Enhance: Extra supports phone, critical bug fixed, full days working per week, and 1 day response time. Also offers 10 hours for advisory per year.

App Accelerate Premium: It is basically the same as Enhance version, just has 10 more advisory hours per year and 8 hours response time

Development & environment

It includes an open source SDK with over 5,000 device and mobile operating system APIs. The developers only need the knowledge of JavaScript to develop native apps in Titanium, instead of having to learn platform-specific languages. It also has its own Eclipse-based IDE called Titanium Studio, which is functionally comprehensive. It is with integrated mobile backend as a service (MBaaS) capabilities, as well as on-device debugging and an MVC framework (Alloy).

How it works

Titanium takes the JavaScript code, combines it with the Titanium API (which is written in the target platform's native language), and then uses a JavaScript interpreter which runs on the device's OS's to evaluate the code at runtime. When the app is launched, the interpreter executes the JS code. Titanium API acts as a bridge to allow direct access to the native experience, through actual native features and APIs.

System support

Titanium can create native apps across different mobile devices and operating systems including iOS, Android, Windows Phone and BlackBerry. For developing desktop applications, it supports Windows, Mac and Linux. It also supports access to most hardware functionalities.

Product

Titanium has thousands of enterprise customers such as eBay, Merck, Mitsubishi, NBC and PayPal. There are more than 50000 mobile applications developed by Titanium that have been deployed on 115.085.786 devices.

Review

Appcelerator has more focus on information-rich and interactive applications instead of graphics games.

Compare to most frameworks, Titanium is widely used by many companies and developers. It is creating a great platform for a growing community.

For the score of Titanium, please read the table below:

Criteria	Explanation	Score
Included functionalities	Support less platforms compared to the other frameworks	3
Support & document	Well-documented in its website and growing community	4
Maturity of framework	Mainstream framework with globally corporations	5
Code reusability	It offers 60%-90% code reuse for the mobile application while still having a native user interface	5
Testability of framework	Titanium Studio helps to test and debug the resulting apps	4
Flexibility	Quite flexible for small or simple apps, as more complex the project grows, it takes the save time in fixing and workarounds	3
License costs	Only the basic version is free to use	3
Performance	The resulting applications performance 60-70 times faster than other mobile application framework available in the market	5
Security	It supports HTTPs connection. The source code can't be easily found because it is complied with byte code. It is also possible to sign the package file to prevent fake version of the app.	4
Difficulties to create applications	The knowledge of Titanium architecture and the environment is required. Also it's necessary to learn the Titanium API	3
Updatability	The SDK is updated almost every month, with the new version of the documentation.	4
Distribution and Installation	It offers an automated way to package and distribute the apps to the platform's respective app stores through Titanium Studio	4

MoSync

Company

Mosync AB is the developer of Mosync. It was founded in late 2004 with global headquarters in Stockholm, Sweden and now has offices in Romania and Korea. The four main investors are STING Capital, David Axmark, Michael “Monty” Widenius, and John Wattin.



Price

Mosync is an open source framework and licensed under GPL2. This means that it is completely free for private use, but if you want to sell the resulting app, or you want to distribute it to other people outside of your organization, you can choose either make your app open source or get a commercial license subscription. There are totally three kinds of subscriptions. The free version doesn't cover hot fixes, named support contact, email or phone support, and guaranteed response time. The basic pro version only offers email support. The gold pro covers all of the services the free version doesn't support.

Development & Environment

MoSync contains MoSync SDK and MoSync Reload. MoSync SDK is the IDE with C/C++ foundation, with the integration of MoSync Reload, which offers the developer to create apps using HTML/JavaScript. The MoSync SDK contains a syscalls library to let you dive down deep into the functionalities of the devices. The syscalls can talk to the phone's sensors, such as accelerometer and track movement, or even location-based features, like GPS, NFC and Bluetooth. There is another library called “Wormhole”, which contains JavaScript functions for handling native user interface widgets. This library makes the communication between JavaScript and C++ easy.

How it works

MoSync use a custom C++ compiler for Windows Phone and Android. It generates intermediate bytecode which then executed by the MoSync virtual machine. iOS doesn't allow dynamic code generation, so instead the MoSync C++ code is translated to C and then compiled using an Xcode project.

Support system

It supports nine platforms, including three mainstream platforms (iOS, Android, Windows Phone). However, there are some features and capabilities that can't be supported on all platforms by this framework. It also supports many features and hardware devices such as NFC and OpenGL. Please refer to the appendix in table A and B to see details.

Product

The amount of MoSync applications is not as much as the other frameworks. At the same time, for some reason, this framework is not the more popular choice for most global companies. You can barely find any cooperation between MoSync and big companies. The official website lists their customers such as Astando AB and E-Adept, Bloo AB and BlooFlirt, and Dalarnas Tidningar and MoPub LBS.

Review

MoSync uses C/C++ or HTML5/JavaScript, or a combination. Its resulting applications are normally hybrid

apps. Although it covers all platforms, the developers only need to create one project to cover all the platforms. Also because of the power of C++, the developers don't need to code in different languages for each of the platforms if there is a need for more performance.

For the score of MoSync, please read the table below:

Criteria	Explanation	Score
Included functionalities	Support 9 platforms but less features supported compared to the other frameworks	3
Support & document	Well-documented in its website, but the community is not that active	4
Maturity of framework	Not that widely used, could be because of having some potential bugs and the community is not really active	3
Code reusability	It is possible to reuse current web-based code when developing mobile apps	3
Testability of framework	Supports emulating applications for iOS, Android and Windows Phone, and includes a simulator if the developers want to test apps quickly	4
Flexibility	It is quite flexible. For the supported platforms, it allows the developers to add new input languages independently. On the other side, it is also possible to add support for new platforms independently of the input languages	4
License costs	It is released under a dual license: open source GPL and Commercial	3
Performance	The general performance is quite smooth	3
Security	Doesn't support Https connection	2
Difficulties to create applications	C/C ++ is the mainstream programming language, which supports the developers to start with this language quickly. Using APIs of MoSync, the application development is easier. For example, the MAUI class make it easy to develop MoSync apps	4
Updatability	A new release on average takes half a year. It isn't a very frequent updated framework. In most case the application can be used in the new version without changes in the app	3
Distribution and Installation	The installation package created by MoSync is completely independent (native package), which means that there is no need of any Plugins or libraries to be installed. At the same time, the distribution process becomes easy as well.	4

Rhodes

Company

RhoMobile was founded in 2008. It is a cross platform that focuses on business and enterprise apps. Rhomobile Inc. was the company behind Rhodes. However, Motorola Solutions bought it in July 2011 to continue developing and supporting the products. The main office still locates in San Jose, United State.



Price

Rhodes itself is open source released under the MIT License, which means that it is free to create mobile apps. For enterprise need, it offers Rhodes Enterprise Licenses. The price is not shown in the website. Instead, contact must be made with the sales department. RhoHub (cloud-based IDE) is also for free. Access to its own server RhoSync cost money.

Development & Environment

Rhodes is a Ruby-based framework that allows the developers to build applications in a similar way to how they normally build a Ruby application. It supports a Modal View Controller structure and allows the developers to separate the content from the presentation tier. This makes it easier to change and reuse most of the existing code.

How it works

Rhodes works mostly just like PhoneGap. It wraps a Ruby-on-Rails webserver in a native app and uses a Webview for the UI.

Support system

It supports all major smartphone operating systems: iOS, Android, RIM, Windows Mobile and Windows Phone 7. Symbian was dropped after Rhodes 1.2. It is capable at supporting many hardware devices such as GPS, camera, native mapping, Bluetooth and NFC (Near Field Communication).

Product

This framework has been downloaded many times. However, there aren't so many companies using the Rhodes. And there is no cooperation company information in the official website. After Motorola bought this company, it released a few useful tools, such as RhoElements, RhoConnect, and RhoStudio.

RhoConnect takes charge of enterprise database connections, pushes out updates to the device through RhoSync, and provides a synchronized data capability for transparent access to data, even when the device is offline. RhoHub is a cloud-based IDE; it comes with Git-based source control and team collaboration features that allows Mac-free iOS development. RhoGallery is an enterprise app store, managing apps with automatic provision, update push and de-provisioning capabilities. RhoStudio IDE makes development with Rhodes easier with build tools, templates, simulators and debuggers.

Review

This framework has a lot of sub-programming tools, which work well and give a lot of value to the platform. It's mainly for enterprise-focused apps. It provides a complete set of products managing an enterprise's mobile apps and data, since it is Ruby-based and has a lot of extra tools. On the other hand, extra tools may take longer time even though for simple applications.

For the score of Rhodes, please read the table below:

Criteria	Explanation	Score
Included functionalities	Covers most of the mainstream frameworks	3
Support & document	Doesn't have an independent official website. Only the sub-pages under the Motorola website. Quite difficult to find useful information. The documentation for reference and APIs has some improvement	2
Maturity of framework	It has existed for a few years now, so it is quite mature. However, after Motorola bought it, it doesn't have an independent website, also RhoStudio still have some bugs every now and then.	3
Code reusability	"Write it once and it runs on all smartphones", as you can see from the slogan. Rhodes aims to reuse the code as much as possible. It is true because of the MVC pattern. For example, the frontend HTML code could be used for an actual website as well.	4
Testability of framework	With RhoStudio, the developers can easily test and debug cross platform applications on one computer. It works in iOS, Blackberry, and Android	4
Flexibility	Rhodes supports a number of different extensions. The developers can add libraries for whatever functionalities they want. The native extensions allow the developers to write code in platform specific code	4
License costs	It is free for open source use. Commercial use is under Rhodes Enterprise License, which has a certain cost. The cost is not shown in the website	3
Performance	The resulting application doesn't have real native look and feel.	3
Security	It supports HTTPS connection as the other frameworks. It also protect the performance of the resulting apps, the mobile device and the wireless network	4
Difficulties to create applications	There will be a learning curve for people who don't know anything about Ruby. So for backend, they need to know Ruby and understand how Ruby on Rails works. For frontend, unlike some frameworks that use common language with backend, Rhodes development use web based languages for frontend development.	2
Updatability	Has regular new version releases, mostly for bugs fixes and performance improvement. Features are not really updated that often	3
Distribution and	Developers can use RhoElements to deploy both native and hybrid	4

Installation

apps on any mobile device that Rhodes supports.

Marmalade

Company

Marmalade SDK is developed by a private company named Ideaworks3D Limited. The company started in 1998 as a technology company focusing on graphics and networking. The main office of this company is located in London. It also has offices in Tokyo, San Francisco, and Seoul, South Korea. The main shareholder is ARM. It has connections with a number of global companies such as NOKIA, Apple, Google, NVIDIA, Samsung, and KONAMI. The technology behind Marmalade was developed between 2005 and 2008. It was launched as “AirPlay SDK” in late 2009. Ideaworks3D Limited has many shareholders.



Price

It is not an open source framework. It offers different pricing tiers, ranging from Community to Professional. Higher level tiers offer more platforms and include more support. Pricing is on a per seat basis, going up to 1500 dollar per seat/year. For the highest level of support the price must be discussed with the sales department of the company. There is a free 30-day evaluation period available.

9.10.2 Development & Environment

Marmalade supports two separate development approaches; the developer can make fully native apps using C++. The second approach is a hybrid app using HTML5, CSS3 and JavaScript that includes an implementation of the Adobe PhoneGap specification or get better performance with Lua (Marmalade Quick). It offers a unique architecture abstracts platform variations while providing direct access to the CPU and GPU, so that the apps performance can be maximized across most smartphones and tablets.

How it works

Developers use C++, OpenGL ES and Marmalade APIs that abstract the native device APIs. Then the compilers turn this to ARM-level machine code, in addition to platform-specific code which is executed by native abstraction layer packaged in the runtime.

Marmalade SDK includes many development tools. There are a simulator, ARM debugger, deployment tool, UI builder, remote control facility for device controls testing, cross-platform and extension APIs, and an extension development kit. All procedures are done through this SDK.

Support system

Marmalade supports iOS, Android, BlackBerry PlayBook OS, bada, Windows desktop, MAC OS X desktop, and LG Smart TV. It supports almost all (Hardware) functionalities for the supported platforms, such as landscape screen switching, audio, radio, OpenGL, and accelerometer

Product

There are a huge number of Marmalade apps in App Store, Google Play, and Amazon Appstore. There are some famous mobile games have been made with Marmalade such as Plants vs. Zombies and Cut the Rope.

Review

Marmalade SDK allows developers to maintain a single codebase, while allowing full use of native platform APIs on iOS and Android. It also supports OpenGL. In general, it is more suitable for game development.

For the score of Marmalade, please read the table below:

Criteria	Explanation	Score
Included functionalities	Support all mainstream platforms and some mobile platforms which are not the main targets	4
Support & document	The documentation page in its website sometimes doesn't have response, also the community is not really active	3
Maturity of framework	It has been used by many companies to develop cross platform apps. However, most of the resulting apps are game-related	3
Code reusability	Since Marmalade use C/C++ standard libraries, it's quite easy to port existing C/C++ code	3
Testability of framework	Marmalade Simulator is a desktop simulator which can test actual ARM code without deploying to an ARM based device. It also allows the developers to test across an unlimited set of screen resolutions and simulate access to the hardware devices, such as GPS, Audio, and Compass	5
Flexibility	It is quite extendable to native APIs. Marmalade has its own Extensions Development Kit (EDK) which allows the developers to design their own extension APIs, and then implement them on whatever platforms they want with using each platform's native software development kit	4
License costs	There are totally four versions of this framework, but none of them is for free. Instead the company only offers a 30-day trial	1
Performance	For the native approach mentioned earlier, Marmalade delivers good performance for games and apps. However, the textures doesn't perform that well. For example, if you need to enlarge the resources such as a text or an image, the resources look really pixilated	3
Security	It doesn't support HTTPs connection yet	2
Difficulties to create applications	Since there is no need to learn multiple languages or SDK's, it's easy to start programming for developers using C++ with a single development environment. It also saves a lot of time for development and deployment	4
Updatability	There are regular updates for the SDK with some new features and capabilities.	3
Distribution and Installation	Because of the Marmalade's patented single-binary architecture, the distribution is easier for the developers. Also since it has supports for iOS and Android specific features such as App Store billing, Game center, and Android Market Billing, it is faster to put the apps in the app stores compared to other frameworks	4

Xamarin

Company

Xamarin was originally led by Ximian, also formerly sponsored by Novell. The first release of Xamarin was shipped in June 2004. Now it is sponsored by Xamarin. Xamarin is a software company created in May 2011. The company also created Xamarin.iOS in 2009 (previously known as Mono Touch) and Xamarin.Android in 2011



(Previously known as Mono for Android) based on the mono open source project. It is located in Boston, Massachusetts.

Price

For both Xamarin.iOS and Xamarin.Android, there are totally four versions. The starter is for free with limited app size. The indie version cost 299 dollar with unlimited app size and licensed for an individual. The business versions takes dollar with all indie support, extra visual studio support and licensed per seat. The enterprise version cost 1899 dollars with all business features and extra priority support with SLA.

Development & Environment

Xamarin is a free and open source cross-platform, implementation of Microsoft's .NET Framework based on the ECMA (an international organization for information and communication technology and consumer electronics) standards for C# and the CLR (Common Language Runtime). It is for web and desktop apps development. Xamarin.Android and Xamarin.iOS are the two cross platform frameworks for mobile apps development.

Support system

Xamarin.iOS and Xamarin.Android are both built on top of Mono. With this combination, Xamarin supports iOS and Android. Mono is an open-source version of the .NET Framework based on the published .NET ECMA standards. Additionally, Xamarin doesn't have a specific platform that supports Windows Phone, but since C# is also Windows Phone's indigenous language, it can be shared with across Windows Phone using Xamarin's tools.

Product

Xamarin has some customers such as CISCO, hp, AT&T, and Microsoft. There is also a big amount of developers using Xamarin. Furthermore, many developers also choose to use Xamarin for their cross platform game application development.

Work architecture

iOS – C# is ahead-of-time (AOT) compiled to ARM assembly language. The .NET framework is included, with unused classes being stripped out during linking to reduce the application size. Apple does not allow runtime code generation on iOS, so some language features are not available (see Xamarin.iOS Limitations).

Android – C# is compiled to IL and packaged with MonoVM + JIT'ing. Unused classes in the framework are stripped out during linking. The application runs side-by-side with Java/Dalvik and interacts with the native types via JNI (see Xamarin.Android Limitations).

Review

Xamarin is actually a comprehensive cross platform mobile framework. It supports full access for both iOS and Android APIs. It also presents a familiar interface from the native UI toolkits that can perform as fast as any other native application.

For the score of Xamarin, please read the table below:

Criteria	Explanation	Score
Included functionalities	For mobile apps development, it only supports iOS and Android. For Windows Phone, it doesn't support directly. However, since C# is also Windows phone native language, most of the existing code can be reused.	3
Support & document	The community; named Monologue, is often updated. It also has documentation of APIs on its official website. The website stack overflow.com shows that developers of Xamarin are active	4
Maturity of framework	The company Xamarin was created 8 years ago. It is quite mature now. The framework itself is stable as well.	4
Code reusability	For iOS and Android, up to 90% of the existing C# code can be reused. The Xamarin Component Store offers a lot of third-party libraries packaged for instant adoption	4
Testability of framework	Quite easy to test and debug. For example, developers can just use the native methods of testing Android apps for Xamarin.Android apps. They can also configure the apps to use an x86 emulator. Debugging is possible over a USB connection.	4
Flexibility	It uses GPLv2 license that supports to expand on the existing tools.	3
License costs	The starter is free to use. For bigger and enterprise applications, Xamarin has a certain cost from 299 to 1899.	2
Performance	The developers are able to design the UI with Xamarin Studio; they will perform as fast as native applications. Since Xamarin.Android and Xamarin.iOS both offer full access to the Android APIs and iOS APIs, so the performance should be nearly native as well. However, there have been some issues with memory management	3
Security	It supports HTTPS connections and any other features that native APIs support	4
Difficulties to create applications	The coding part doesn't have a big learning curve. However, two independent apps are necessary if iOS and Android apps are both needed, which is time consuming. Also iOS development requires an Apple machine unless MonoCross is used	3
Updatability	It is frequently updated and most some platform new features can be supported with newer version of Xamarin. Most of the bugs reported by the developers can be expected to be fixed for a new release	4
Distribution and Installation	The distribution part cannot be automatically done. For example, developers need sign the apk file using Xamarin Studio or Visual Studio themselves.	3

AppMobi

Company

AppMobi platform is created by the company appMobi Inc. It is an outgrowth of FlyCast, which was the first software company delivering streamed entertainment apps to the iPhone. Many of the top ten apps on different smartphone platforms come from this company as well. AppMobi Inc. is a mobile technology company based in Lancaster, Pennsylvania, United States. It was founded on January, 2006. The company was named “Most Promising Company for 2012”. In February, 2013, all of appMobi’s HTML5 development tools had been acquired by Intel. Right now the company is a mobile cloud services provider.



The company also has other development tools. For example, the appMobi Impact XDK is for game development, the resulting game apps can be used from mobile devices to desktops. It also released PhoneGap XDK in 2011, supporting HTML5 development using the open source PhoneGap API.

Price

It is free to start building an app. The cost will be charged according to the MAU (Monthly Active Users). Specifically, the completely free version is for hobby apps, the limitation is no cloud services support. The version for “Money Making Apps” supports full access to all appMobi Cloud Services, it will be charged if the MAU is over 10,000, which is \$0, 10 per MAU.

The price mentioned before including all development tools belonging to the appMobi platform, such as appMobi XDK, appMobi Impact XDK, and appHub.

Development & Environment

AppMobi provides some functions and features that HTML5 doesn’t support, such as in-app purchasing, user authentication, and live app updates. The resulting apps are mostly hybrid apps. With appMobi’s cloud-based build system, the HTML5-based apps can be turned into native API compliant mobile apps for iOS and Android. AppMobi XDK (cross platform development kit) is an integrated development environment using HTML5.

Support system

AppMobi XDK supports iOS, Android, and Windows Phone 8. In addition to its support of these three major smartphone platforms, it also supports web development. For the store-ready appMobi apps, the developers can build them for Facebook, Chrome, Amazon, and Mozilla app stores as well. For the capabilities, it supports many native capabilities such as camera, GPS, touch gestures, vibration, and more.

Product

AppMobi has many kinds of partners. It has some technology partners such as Amazon Web Service, and Impact. Companies like Seisan, LLC, and Postage, Inc. use appMobi tools for cross platform app development. There are more than 50,000 active developers using appMobi for mobile and desktop app development.

Review

AppMobi platforms consist of many useful and powerful tools. Because of its special cloud-based system, it offers the developers the ability to develop mobile apps within a web browser without any type of installation. The StoreView (cloud-based analytics service) from appHub allow the developers to follow their application's performance after the apps get distributed to each platform.

For the score of appMobi, please read the table below:

Criteria	Explanation	Score
Included functionalities	For mobile operating systems, it only supports iOS, Android and Windows 8. It however supports many native platform capabilities and features	3
Support & document	The API documentation in the official site looks well-organized. With some simple code examples helping developers get started. It also has some good technology partners	4
Maturity of framework	The technology of appMobi was originally based on FlyCast. In 2011, appMobi XDK had been downloaded from the Chrome Store more than one thousand times per week. In 2013, Intel bought all HTML5 based tools; this proves appMobi's abilities as well.	4
Code reusability	Developers can reuse the codes in the other projects instead of rewriting them.	3
Testability of framework	AppMobi XDK supports debugging apps on PC, device, and emulation. The deployment is much faster as well compared to use some other frameworks. Especially it supports testing the app through Wi-Fi or cloud-hosting service	5
Flexibility	Developers can easily access to plugins and modules for additional specific functions	4
License costs	The free version has limitations. The paid version depends on the MAU, and cost is quite reasonable compared to other frameworks	3
Performance	AppMobi apps generally perform well. There are two reasons: first is because of using the ImpactJS gaming engine in appMobi XDK, this improves the game apps' performance a lot. Also the StoreView ensures the apps perform normally after release	3
Security	Private Stack (another tool from appMobi) is the secure mobile ecosystem for enterprises, which adds important security features to appMobi apps	4
Difficulties to create applications	HTML5 makes the learning curve disappear for web developers. However, since it has many independent development tools, it may take a while to get used of them all	4
Updatability	The appMobi XDK offers over-the-air update service. If the developers update the XDK version, it will automatically update the app to reflect any changes made to the code	3
Distribution and Installation	The distribution and installation can both be done by appHub. AppHub creates store-ready binary files and deploys them in app stores. With the StoreView, the developers can be informed on how their apps perform on each platform.	4

Corona

Company

Corona SDK is created by Walter Luh, who co-founded Corona Labs Inc. (formerly known as Anscamobile), together with Carlos Lcaza in 2008. Corona Labs Inc. is a mobile software company based in Palo Alto, California.



Price

The trial version of the SDK is free. It starts to charge when developers want to publish their apps to the store. The subscription price depends on the version of the Corona SDK. The supported features and capabilities of different versions are all the same except for the platform support. Under the Indie category, the Corona SDK for iOS or Android costs \$199. The Pro version costs \$349 and supports extra two platforms (Nook, and Kindle Fire). The subscriptions are yearly.

Development & Environment

It supports development by using integrated Lua, layered on top of C++/OpenGL to build mobile applications. The mobile apps created by Corona can be native apps, but Corona can also deliver hybrid apps using a combination of HTML5 and Open-GLES based functionality.

How it works

It has a platform abstraction runtime. The Lua code is compiled to a Lua bytecode packaged in a native runtime shell that provides the bridge to the underlying operating system. The Corona SDK consists of many tools including app templates, API libraries (over 500), sample code, a debugger, a simulator and APIs for the Flurry analytics platform. It also implements native UI elements using a native OpenGL engine packaged within the runtime.

Support system

It supports iOS, Android, Nook, and Kindle Fire from one code base. With Corona Enterprise, the developers can even call any native library on the iOS and Android. It supports some hardware such as accelerometer, GPS, compass, and camera. It is built on mobile and industry standards including OpenGL, OpenAL, Google Maps, game center, and in-app purchases. Some features available for iOS are not implemented for Android, such as camera and activity indicator.

Product

Corona itself offers the clients three products and training support. There are Corona SDK, Corona Enterprise, and Corona Cloud. Corona Enterprise takes full advantages of Corona SDK while calling any native C-based or Java library. Corona Cloud is currently still in closed beta.

Review

Since Lua is a high-level language which helps to develop games much more quickly compared lower-level frameworks such as OpenGL; it is widely used among game developers for 2D games. It offers a built-in physics engine. The slogan of Corona is "Write once, build to both iOS and Android", which is

similar with some other cross platform framework's slogan. The difference is that Corona actually offers the possibility for the developers.

For the score of Corona, please read the table below:

Criteria	Explanation	Score
Included functionalities	Corona SDK only supports Android, iOS. For tablet operating systems, it also supports Kindle fire and nook	2
Support & document	It is well documented. There are more than 200 thousand developers in its community. It includes more than 500 APIs, which gives many kinds of support for the functions and capabilities	4
Maturity of framework	It has been widely used, especially among Flash coders. Many mature apps have been published using Corona	4
Code reusability	The code of Corona apps use Lua which, as native apps, is less reusable; The code of apps using a combination of HTML5 and OpenGL can offer a higher chance of being reusable	2
Testability of framework	The company offers a command-line debugger	2
Flexibility	No need to compile the app, changes can be seen instantly using refresh function on the simulator Since Corona uses its own engine to generate the apps, the developers don't have access to the latest platform APIs. Developers cannot extend it to support any missing functions. Also the apps are compiled on Corona server, even though the apps get distributed, which makes this framework less flexible	3
License costs	It has free trial version. Only distribution cost money. The developers get two seats (two laptops authorized) if they buy Corona for Android or iOS. However, if you pay double to get the combined iOS and Android license, you still have only two 2 seats, that actually makes the price more expensive	3
Performance	The speed of Corona SDK apps are on average slower than other frameworks apps. The Android app made by Corona performs worse than iOS. On the other hand, it does offer high-performance graphics.	3
Security	No IP address protection	2
Difficulties to create applications	Based on Lua, so actually simple and easy for coding. And using Lua for programming can be quite productive. However, it is not a big advantage for people who are familiar with C/C++ instead of Lua.	4
Updatability	Have daily build update. Most of bugs get fixed quite quickly	4
Distribution and Installation	The distribution process is not done automatically. XCode is needed if developers want to submit their Corona iOS apps to the App Store.	2

MonoCross

Company

ITR Mobility started MonoCross as a mobile framework in 2010. It is a management and IT consulting company.

Price

MonoCross is an open source mobile framework. It is licensed under MIT, which means that it is free for developers to download and use. Commercial use has a certain cost.

Development & Environment

MonoCross is a C# .NET framework. It is based on a modified Model-view-controller (MVC) pattern that separate the presentation layer (View) from the application code (Model + Controller). It supports the developers to develop both native and HTML5 based web apps that share business logic and data code.

Support system

It supports the developers to create native output apps for iOS, Android, Windows Mobile, and Windows Phone. It also supports web/hybrid apps development for iOS, Android, Blackberry, and Symbian.

Product

This framework is being used by large Fortune 500 enterprises in production. No further information can be found in its official website.

Work architecture

MonoCross uses MVC pattern to separate user interface from business and data layer so that the code in the business and data layer can be reused across supported platforms such as iOS and Android. And then the UI layer can be developed independently for different platforms to reach the target platform native look.

Review

This framework mainly focuses on enterprise application. C#. NET can maximize the existing investments and skillsets so that the company can save the cost and effort. MonoCross is a library built on top of Xamarin.iOS and Xamarin.Android

For the score of MonoCross, please read the table below:

Criteria	Explanation	Score
Included functionalities	It supports the major mobile platforms. It also support both native and hybrid apps. it offers the access to the native API's	3
Support & document	It is not well documentation. The community is not easy to find and not active as well	2
Maturity of framework	It starts in 2010, which was for Linux only. It is normally used with Xamarin.iOS and Xamarin.Android that makes it less independent	2
Code reusability	The code in data and business layer can be reused across different platforms	4
Testability of framework	Share the Mono Develop with the Xamarin.iOS and Xamarin.Android, which can do debugging and testing	3
Flexibility	Quite flexible. Developers can write one app for the target platforms and custom the UI for each platform themselves. It reduce any subsequent changes and enhancements since the codebase is only updated once instead of for each target platform	4
License costs	It is open source and released under MIT license	4
Performance	The performance is the same as the native written apps since the MonoCross app is compiled to a native executable	4
Security	The abstract utility interfaces offers access to all functions developers need to secure their apps	3
Difficulties to create applications	C#.NET is quite easy for coders to develop mobile apps. The learning curve is quite small for MonoCross development	4
Updatability	Quite frequent updates. New features and capabilities are developing	3
Distribution and Installation	The abstract application model makes deployment easier	3

First round result with score

This chapter gives a general view of the 1st round investigation result and different options for different perspectives.

1st round description

For some of the frameworks certain criteria can't be judged comprehensively because the required information is not available or enough. These criteria will be resulted in a lower score and reflects partly on the maturity of that framework.

The table below shows the result after the initial round investigation:

Framework	Final score
Appcelerator Titanium	315
PhoneGap	301
appMobi	286
MoSync	258
Xamarin	256
Marmalade	248
MonoCross	243
Rhodes	239
Corona	233

Table 1: Final scores after initial judging

For a complete view of the results for all frameworks, please refer to the table in attachment A.

1st round Conclusion

For ICT Automatisering

It is pretty noticeable that in general frameworks using web based technology (Titanium, PhoneGap and appMobi), such as HTML5, JS, and CSS3, get higher score, while the C/C++ frameworks are in the bottom of the list. This is mainly because of the higher learning curve and longer development time required for C++ and the game focus that those frameworks seem to have (especially Marmalade and Corona). Rhodes has the similar condition, it is a good framework with extra useful tools, but it will not be a good choice for ICT Automatisering mobile development team due to its requirement to learn Ruby. Another reason why Rhodes has the second lowest score is because of a lack of information. It doesn't have its own official website, which makes it difficult to find out the required information for the criteria.

For game development

For companies and developers who mainly focus on mobile game development, it is better to choose frameworks which are also for game development. There are three frameworks listed in the table above that focus on games. They are appMobi, Marmalade, and Corona. To ignore the order in that table, Corona will not be a good choice, although Lua development could be quite productive for developers who are familiar with it. However, it doesn't give fully support for the development procedure. On the

other hand, Marmalade includes many development tools that offers fully support from development, testing, to deployment. Native app and hybrid app are both possible by Marmalade as well. AppMobi is a good choice for developer who doesn't want to deal with C++. It has a unique IDE for game development using HTML5.

For non web-based technology developers

For developers who aren't familiar with web-based technology, the option is different again. There are four frameworks available for them: MoSync, Marmalade, Xamarin, and MonoCross respectively. MoSync and Marmalade are for C/C++ developers, while the rest is for C#.NET developers.

Conclusion

It is almost impossible to directly name one or two best ones among all these nine frameworks. As you can read earlier that different frameworks have different focus, while companies and developers have their own development preferences as well. So it is vital to know exactly what you expect from a cross platform mobile framework and what it can pay you back at the meanwhile.

Overall, Appcelerator Titanium, PhoneGap, and appMobi as the top three frameworks will bring to the second round for functions and capabilities testing. The criteria will be re-marked for each of the framework afterwards.

Test round overview for PhoneGap

Testing application result

PhoneGap supports two tested functions of the test application. However, native UI is not supported. Most of the development time was spent learning JQuery mobile and solving bugs in both frameworks. PhoneGap is really easy to develop for. You can directly install the library into Xcode or Eclipse and start programming.

There were a few minor points of frustration, but in the end everything was working.

As mentioned above, JQuery Mobile was used for the user interface since PhoneGap just provides a web environment and as such is not responsible for the user interface. This means that any UI made within PhoneGap is not native UI. Combined it with a framework like JQuery Mobile makes the application performance worse than native apps. It is worthy to keep this in mind because it shows one weakest point of PhoneGap. It is relevant when comparing to other frameworks.

Criteria

Include functionalities

PhoneGap supports almost all the features tested by the test application. The only exception being native UI. Nevertheless, by being based on HTML it allows many developers to use existing skills.

Score: 4

Compared with first round score: **-1**

Support & documentation

As one of the most popular cross platform frameworks, PhoneGap has a large community with a number of developers.

The API documentation is not that helpful as expected. For example, there is a matrix showing all supported features across different platforms, but the support claimed by this matrix doesn't match with what API documentation claims. Also there are some errors written in their API documentation. The developers need to keep it in mind to not completely trust the example code written in the API documentation.

Score: 2

Compared with first round score: **-2**

Maturity of framework

As mentioned above, it is really popular with a large community, and the library itself is quite stable despite the low score of the documentation.

Score: 5

Compared with first round score: **0**

Code reusability

JavaScript libraries and HTML file can be reused and shared for different platforms. However, if you want to make the UI look as native as possible, the CSS files need a lot of changes since different mobile platforms have different screen layout and size and different interface style.

Score: 2

Compared with first round score: -1

Testability of framework

It is possible to test the application on the actual device. You can also test it in the native platform emulator.

Score: 4

Compared with first round score: 0

Flexibility

PhoneGap is actually a JavaScript library, which means it is easy to be modified or extended. Many developers in the PhoneGap community make changes during the PhoneGap installation to add some extra functionalities.

Score: 4

Compare with first round score: 0

License costs

No extra cost during test application development. Not that relevant in the test applications, so the score is maintained from the first round.

Score: 3

Compared with first round score: 0

Performance

The performance of the PhoneGap test application is really bad. PhoneGap doesn't do anything for UI, if you want to have a nice UI, then you need extra tool for that. The response time is really slow. And since it uses web views, there is barely improvement possible for this weakness.

Score: 2

Compare with first round score: -1

Security

Not relevant in the test applications. The score maintains the same as it is in the first round investigation

Score: 2

Compare with first round score: 0

Difficulties to create applications

The test application development was quite simple and it only has a small learning curve for the JQuery Mobile. However, the biggest annoyance is the inconsistency in the order of function parameters. For example, the cameraSuccess function has its parameters in a different order than the photoGallerySuccess function.

The development skills are just web based techniques.

Score: 3

Compare with first round score: 0

Updatability

Not relevant during the test applications period.

Score: 4

Compare with first round score: 0

Distribution and installation

PhoneGap use specific platform tools to build the projects, so the test application is built and package in the same way as native applications.

Score: 4

Compare with first round score: 0

Review and conclusion

PhoneGap should be an excellent choice for web developers. It is basically the same as developing a website, with an extra library that offers mobile device specific functionalities. It also explains why the performance is not as good as the other frameworks' applications.

In general, it is really easy to develop with web skills. Although the documentation is not really complete, because of the simplicity of the framework, learning PhoneGap framework is still not a big deal.

Test round overview for Appcelerator Titanium

Test application result

All required features were implemented. Although the structure of a Titanium application is a bit complicated, once you make it clear, it is rather easy to use. The Titanium Studio is a good IDE, except the property button in the drop-down menu didn't work most of the time. One more annoyance about Titanium IDE was that errors in the code could not be found directly until the project was actually running.

Criteria

Include functionalities

It supports all the required test features in the test application. Though Titanium use web base technique as PhoneGap does (JavaScript), it creates native UI without any extra third party libraries.

Score: 5

Compare with first round score: +2

Support & document

The community is quite active and has more than 2 million developers. It is easy to find support and documentation for many problems.

The Titanium API documentation is quite useful. For different version of the Titanium, it has different pages for the library. Also the explanation for the API methods is quite clear; some of them even have example code for extra explanation. The kitchenSink demo application is offered as an example for starters.

Score: 5

Compare with first round score: +1

Maturity of framework

No matter Titanium API or Titanium IDE, they are all quite mature and stable. During the 2 month development period, the Titanium Studio only crashed once. There are a few bugs as mentioned before, but for the most part it's very useable. Only one minus point is the preference, it can't be shown most of times.

Score: 4

Compare with first round score: -1

Code reusability

It is possible to use the code in different libraries that can be reused in other projects. Since the UI is created by JavaScript, the UI components or entire pages could be shared between projects.

It is possible to share JavaScript files between mobile and web projects, if the project is designed in the right way.

Score: 5

Compare with first round score: 0

Testability of framework

It is quite easy to test the app in the device or the native emulator. It also offers Titanium emulator for iOS and Android. However, test the app in iOS device need to update it through iTunes.

Compare with first round score: -1

Flexibility

As mentioned before, Titanium is quite flexible for simple or small applications. Since the test application is not a big application, this point doesn't get proved that much. The score maintains the same

Score: 3

Compare with first round score: 0

License costs

Not relevant during the test applications period.

Score: 3

Compare with first round score: 0

Performance

The Titanium test application had quite good performance, it maps native UI elements, even with taking camera or pick up an image from the local library, the app still kept running smoothly. The underlying JavaScript engine also seems to be quite capable.

Score: 5

Compare with first round score: 0

Security

Not relevant in the test applications. The score maintains the same as it is in the first round investigation

Score: 4

Compare with first round score: 0

Difficulties to create applications

Titanium as a framework is consistent and not very complicated and its event-based system is easy to use. Only JavaScript is mostly used during the development time. Its IDE has quite clear structure and friendly UI for developers. Because it has decent auto complete plug-in to remind the developers about the Titanium API methods.

Score: 4

Compare with first round score: +1

Updatability

There was one new version released during the 2 months development. It seems like that Titanium gives a new release usually for every 2 months to fix some bugs and add some new functions.

Score: 4

Compare with first round score: 0

9.10.3 Distribution and installation

Titanium builds the applications to the native installer format for each platform automatically. The

researcher didn't distribute the test application to the real store. However, the test application that gets installed in the device is actually a store-ready app.

Score: 4

Compare with first round score: 0

Review and conclusion

Titanium offers quite a wide range of functionality support, but its range of supported mobile platforms is limited compared to other frameworks. The framework itself is quite mature, with big active community support. The IDE is also useful, despite a few bugs, which doesn't matter that much for the programming and testing stage.

In short, after two rounds investigation, Titanium seems like a quite serious framework that you can certainly use it as a professional mobile development tool. Especially if you see the large amount of companies already using it.

Test round overview for MoSync

Test application result

The plan was to create a MoSync test application using their HTML5 support. However, the camera test case could not be implemented successfully in HTML5 version of the MoSync test application. In the end it was decided to create an independent test application for camera test case using C++.

Both HTML5 and C/C++ MoSync support native UI. However, the orientation only works in C/C++ test application. Both versions offer network access.

The MoSync SDK (Its own IDE) in general works fine except it stopped working a few times, and after restarting the IDE, some configuration settings were lost.

Criteria

Include functionalities

MoSync supports both web-based techniques and C/C++ development. In theory, it should support the most the functionalities of the second round frameworks. However, it supports the least test functionalities.

Score: 2

Compare with first round score: -1

Support & document

The community is quite big, but not really active. The researcher had an IDE setup problem and posted it the community, but there were no replies. Also for the post in a recent week, people barely got replies as well.

The API documentation is not really clear and difficult to find the correct category. The explanation for each method or class is quite simple, but there are no direct code examples for most methods. Fortunately, it does have some demo examples for both HTML5 and C/C++ development.

Score: 3

Compare with first round score: -1

Maturity of framework

Some API methods don't function as what it is explained in the documentation. For example in the camera test application, the set Picture method doesn't actually show the picture picked from the phone's local library.

The MoSync SDK normally works fine, except sometimes it has random crashes.

Score: 4

Compare with first round score: -1

Code reusability

For the C/C++ version of test application, it is quite difficult to reuse the entire UI page or even some UI components in other projects, since the UI is also created by C++.

The HTML5 version has certain code reusability like the other frameworks.

Score: 4

Compare with first round score: +1

Testability of framework

Quite easy to test the application. It supports Over The Air (OTA) testing on iOS devices. Debugging in iOS and Android emulator is possible.

Score: 4

Compare with first round score: 0

Flexibility

Having two programming languages makes it more flexible for the developers. Both of them support quite a lot of API functions as well.

Score: 4

Compare with first round score: 0

License costs

Not relevant during the test applications period.

Score: 3

Compare with first round score: 0

Performance

The MoSync test application had quite good performance, it maps native UI elements, even with taking camera or pick up an image from the local library, the app still kept running smoothly.

Score: 4

Compare with first round score: +1

Security

Not relevant in the test applications. The score maintains the same as it is in the first round investigation

Score: 4

Compare with first round score: 0

Difficulties to create applications

Two programming languages already make the development less difficult. The web developers can still use web-based techniques to develop the application, while the others still have an alternative option.

As long as you understand how the MoSync application works (No matter C/C++ or HTML5), it will be no problem to continue.

One minus point is that for different language, it supports different functionalities. For example, the file access to the local photo library is only possible using C/C++. So developers using HTML5 in MoSync who at the same time want to implement this function will be unable to do so.

The iOS signing and provisioning sometimes doesn't get matched to the correct license, causing unusual build failures.

Score: 3

Compare with first round score: +1

Updatability

There was no version update during the 2 month development. But based on the update history it has seasonal new version releases.

Score: 3

Compare with first round score: 0

Distribution and installation

MoSync builds the applications to the native installer format for each platform automatically. It also supports the Bluetooth installation in the actual device, even though iOS Bluetooth doesn't function well normally.

Score: 4

Compare with first round score: 0

Review and conclusion

Both two versions of MoSync support many API functions. It also has two IDE that the developers can choose. In general, it is actually quite a comprehensive framework. The result application performs smoothly. Also the community is growing.

However, it is not really mature and stable framework, such as unclear documentation and support and a few development bugs.

Recommendation & Conclusion

Final score

Based on the experience of creating a few test applications, the three chosen frameworks were judged again. The final scores are as follows:

Framework	Final score
Appcelerator Titanium	315
MoSync	275
PhoneGap	266

Table 1: Final scores after the second round

The order of the scores is quite different compared the results from the first round of scoring; only Titanium maintains the same score. The switch of PhoneGap and MoSync is really interesting because the general impression of MoSync was not very positive. It was actually the alternative framework of appMobi. On the other hand, PhoneGap had good score for the individual criteria in the first round, but after test application experience, it fell to the third place.

Recommendation & Final choice

The final recommendation is not only based on the scores judged in the second round of testing. The quality of the test applications also comes into play as well as the author's general impression of the frameworks, which is mainly reflected by the framework maturity.

One of the most important decisions that must be made when choosing a framework is the decision of native versus web based applications. For web developers, it is of course very easy to choose web based framework since it basically just wraps around a browser element and use standard web development tools. One plus about web based frameworks is that they normally support more mobile platforms than native apps.

Native frameworks offers the native UI elements as provided by the native platforms, that is why the result apps looks just like normal native apps. Web based apps could mimic this with CSS, but still it is quite difficult to make an app look exactly the same as the native UI of the targeted platform. The result apps made by native frameworks generally have better performance as well, especially in UI related matters.

For each type of applications the final choice is obvious: for native applications, Titanium is the best choice for web developers, and MoSync is the alternative option for non-web developers who want to use real programming language such as C/C++. PhoneGap is recommended for web based application development.

In the end, MoSync should be the best framework for ICT Automatisering. Although its score is in general lower than Titanium, It supports more mobile platforms, has two IDEs and two programming languages to choose from and especially C++ is a good match with existing skills in the company. It also supports extra functions desired by the company, such as NFC. The result test app has native UI and runs smoothly and native. The lower score of the test application might be caused because of limitation of the time for test application. For serious applications, it should be a better choice.

Final conclusion

This investigation result might not be up-to-date by the time it is finished. The mobile market is moving very fast and so are the cross platform mobile frameworks. As mentioned earlier, cross platform mobile frameworks release newer version frequently to fix some bugs and add more supported functionalities. Some of the frameworks might discontinue, and some new frameworks will arise as well. So it is wise to recheck the results of this investigation project in a few months, or at least ICT Automatisering should keep an eye on the status of other frameworks and compare them against MoSync.

In the end, it should be mentioned that before deciding a framework, it is really important to know what kind of application you want to develop and which functions you want to have in the application. It is quite difficult to say which framework is the best one directly. They all have their own highlights and weaknesses. You need to find out a few frameworks with features that match your application needs. Then what really matters mostly are the stability and maturity of the framework.

Attachment B 1st round score

	Base weight	Framework								
Criteria		Titanium	PhoneGap	AppMobi	Mosync SDK	Xamarin	Marmalade	MonoCross	Rhodes	Corona
Included functionalities	6	3	5	3	3	2	4	3	3	2
Support & document	8	4	4	4	4	4	3	2	2	4
Maturity of framework	12	5	5	4	3	4	3	2	3	4
Code reusability	3	5	3	3	3	4	3	4	4	2
Testability of framework	9	4	4	5	4	4	5	3	4	2
Flexibility & Source license	4	3	4	4	4	3	4	4	4	3
License costs	11	3	3	3	3	2	1	4	2	3
Performance	10	5	3	3	3	3	3	4	3	3
Security	2	4	2	4	3	4	3	3	4	2
Difficulties to create applications	1	3	3	4	4	3	4	4	2	4
Updatability	5	4	4	3	2	4	3	3	3	4
Distribution and Installation	7	4	4	4	4	3	4	3	4	2
		315	301	286	258	256	248	243	239	233

Attachment B

---Frameworks platforms support:

Support system Framework	iOS	Android	Windows Phone 7	Windows 8	Blackberry	Web OS	Symbian	Others
PhoneGap	Yes	Yes	Yes	No	Yes	Yes	Yes	Bada, Web OS
Appcelerator Titanium	Yes	Yes	Yes	No	Yes	No	No	-
MoSync	Yes	Yes	Yes	No	Yes	No	yes	Java ME MIDP, Moblin
Rhodes	Yes	Yes	Yes	Only latest version	Yes	No	No	RIM
Marmalade	Yes	Yes	No	Yes, in beta	Yes	No	No	Bada, Windows desktop, Mac OS X desktop LG Smart TV
Xamarin	Yes	Yes	Yes	No	No	No	No	-
appMobi	Yes	Yes	No	Yes	No	Yes	no	-
Corona	Yes	Yes	No	No	No	No	No	Nook, Kindle Fire
MonoCross	Yes	Yes	Yes	Yes	Yes	Yes	Yes	-

Attachment C

---Frameworks features support:

1. Hardware access: including accelerometer, camera
2. Geolocation: includes compass, geolocation
3. Media: including sound, Streaming audio, recording Audio, video
4. Connectivity: includes access to Bluetooth, SMS, and Telephone
5. NFC: Near Field Communications

Features Framework	Hardware access	Media	Contacts	File access	Connectivity	Geolocation	Notification	NFC
PhoneGap	Yes, except for iPhone 3G and Blackberry 5x	Yes, except for Blackberry	Yes	Yes	No	Yes	Yes	No
Appcelerator Titanium	Yes	Yes	Yes	Yes	Yes, except Bluetooth	Yes	Yes	No
MoSync	Camera yes: for iOS, Android and Windows 7, which can't do recording	Yes, except audio only for iOS and Android	No	Yes	Bluetooth yes: not for iOS	Yes, for iOS, Android, and Windows 7	Only for iOS and Android	Yes
Rhodes	Only camera is available	No	Yes	Yes	Only Bluetooth	Yes	Yes	Yes
Marmalade	Yes	Yes, except recording	Yes	Yes	Yes, except Bluetooth	Yes	No	No
Xamarin	Camera only	Yes	Yes	Yes	Bluetooth	Yes, except Compass	No	No
appMobi	Yes	Yes	Yes	Yes	Only SMS	Yes	Yes	No
Corona	Yes	Yes	No	No	No	Yes	Yes	No
MonoCross	Yes	Yes	Yes	No	Only telephone	yes	Yes	No

Attachment D

---Frameworks development comparison:

1. IDE: Integrated development environment

Framework	Programming language	Output app	User interface	IDE	Debug support
PhoneGap	HTML5, CSS3, JavaScript	Hybrid	Web view	No IDE, only web development tools	Yes, Weinre
Appcelerator Titanium	JavaScript (HTML)	Native/ Hybrid/ Mobile web	Native	Titanium Studio (Eclipse based)	Yes , with its IDE
MoSync	C/C++ or HTML5	Native/ hybrid	native	MoSync SDK	Yes, support emulator for iOS, Android and Windows Phone
Rhodes	Ruby, HTML	Native/Hybrid	Combination of native and webview	RhoHub (Cloud-based IDE)	RhoStudio, RhoHub
Marmalade	C/C++ HTML5, CSS3, JS	Native hybrid	native	Xcode, Microsoft Visual C++/ Studio	Yes, Marmalade Simulator
Xamarin	C#.NET	Native	Webview	Visual Studio, Xcode, MonoDevelop	MonoDevelop or Native simulators
appMobi	HTML5	hybrid	Webview	AppMobi XDK, appMobi Impact XDK	Yes, AppMobi XDK,
Corona	Lua HTML5 & OpenGL	Native Hybrid	Native	Tester editor	Command-line debugger
MonoCross	C#.NET HTML5	Native Mobile web	Native	Visual Studio, Xcode, MonoDevelop	MonoDevelop

Appendix D

URS document



Student Name: **Yaxiong Shen**
Internship type: **Graduation Internship**
Internship Company: **ICT Automatisering**
Date: **23-May-2013**
Place: **Son**

Table of Contents

Introduction	100
Functional requirements	101
Non-functional requirements	102
Use case	103

Introduction

This requirement document is the preparation for the “Domotica Frontend” project. All requirements are collected through an interview with the client. This document is a confirmation to make sure the final application functions as the client expects.

There are three increments during the progress of this application. The first increment contains all “must do” functions. All “should do” functions will be done in the second period. The last increment is for the “could do” functions.

This document includes functional requirements, use cases and non-functional requirements.

There are two main roles in this app: user and server. User is the person who uses the “Domotica Frontend” app to control the lights in the house. The server is the computer installed in the house that is connected to the lights and can control them. The app sends commands to the server.

Functional requirements

MOSCOW LIST

➤ MUST DO:

- ❖ The application has a splash screen with ICT logo and Domotica photo
- ❖ Support more than one phone/tablet to control the same house at the same time
- ❖ The app supports multiple languages, such as English, and Dutch
- ❖ The user will be informed by a pop-up box if a certain function is not yet implemented by the server
- ❖ The user is able to control the house over the internet connection

Lights

- ❖ The user is able to turn on/off each single light
- ❖ The user is able to change each light color
- ❖ The user is able to change the light intensity
- ❖ The user is able to request the status of the light. The status includes on/off, current color and intensity.
- ❖ When the app starts, the lights status in the app must be the same as the current lights status in the house

Server connection

- ❖ The user is able to update a selected server connection

SHOULD DO:

Lights

- ❖ The app shows as many lights controls as the lights in the houses
- ❖ The user can set each light's name on the server
- ❖ The user is able to get all lights' names from the server
- ❖ When the user changes a light's color, he is able to change all lights that are on to be the same color as that light

Server connection

- ❖ The user is able to select whichever server connection he wants to be active.
- ❖ The user is able to set up a new server connection
- ❖ The user is able to remove a selected server connection

Custom scene

- ❖ The user is able to create custom scene
- ❖ The user is able to execute a scene
- ❖ Have one default scene: Turn off all lights at once
- ❖ The user is able to delete a scene

COULD DO:

Custom scene

- ❖ The user is able to schedule a scene
- ❖ The user is able to delete a scheduled scene

Non-functional requirements

- ❖ The application supports iOS and Android platforms only
- ❖ The development environment is Titanium Studio 3.1.0
- ❖ The programming language is JavaScript and HTML
- ❖ The app needs to have connection to the server
- ❖ A iOS/Android tablet/phone is needed
- ❖ The user who is going to use the app should have the Domotica server installed in the house
- ❖ The extension of the app is possible for later development
- ❖ No training course is provided
- ❖ The scene is only saved locally in the app
- ❖ A scheduled scene is saved in the server connection

Use cases

Use case diagram:

Diagram 1: Lights use cases diagram

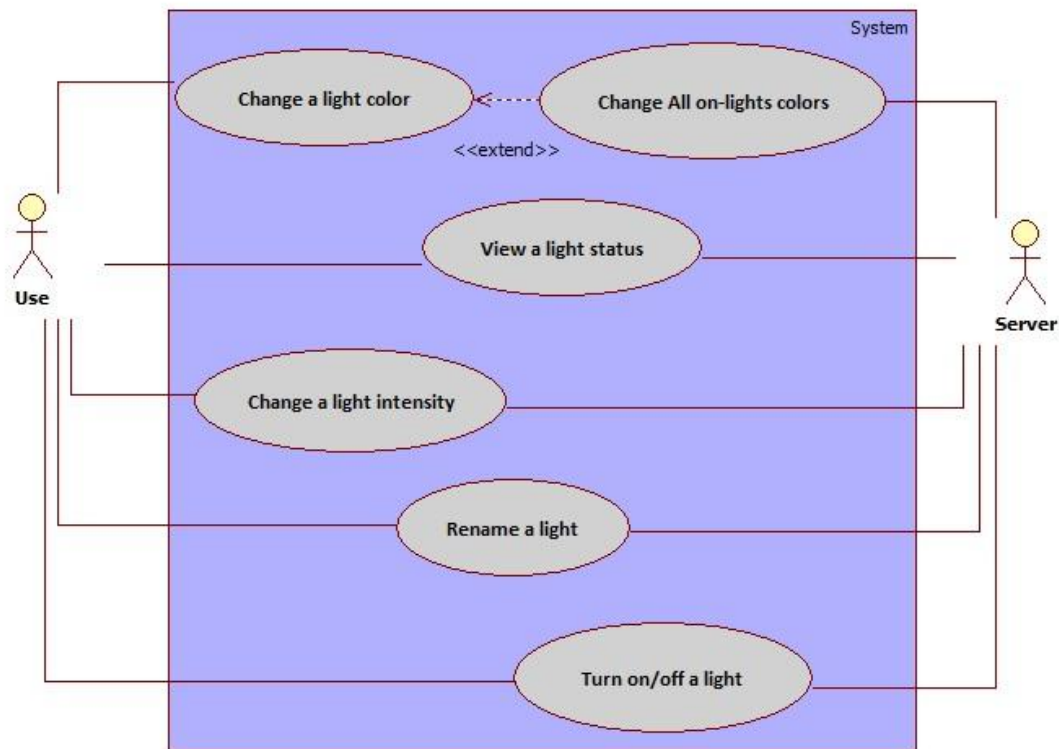


Diagram 2: Server connection use cases diagram

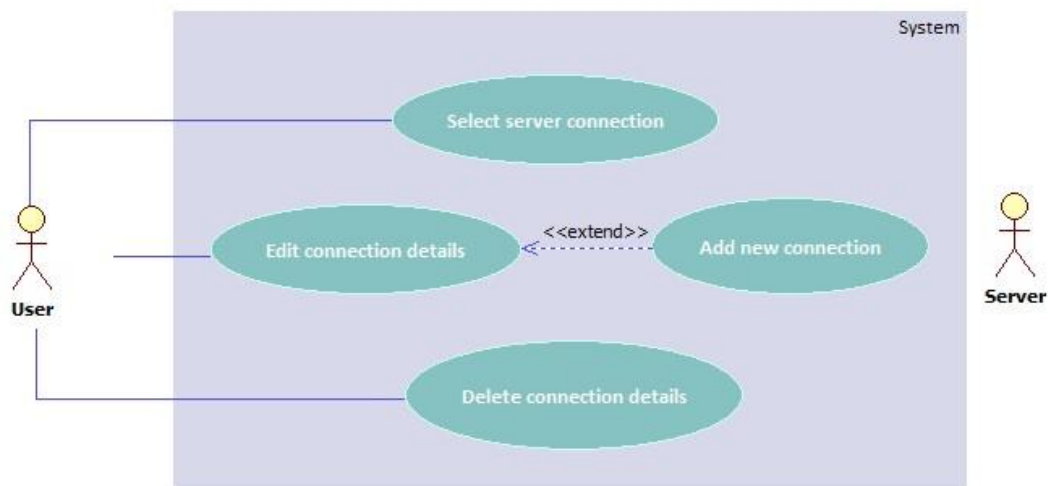
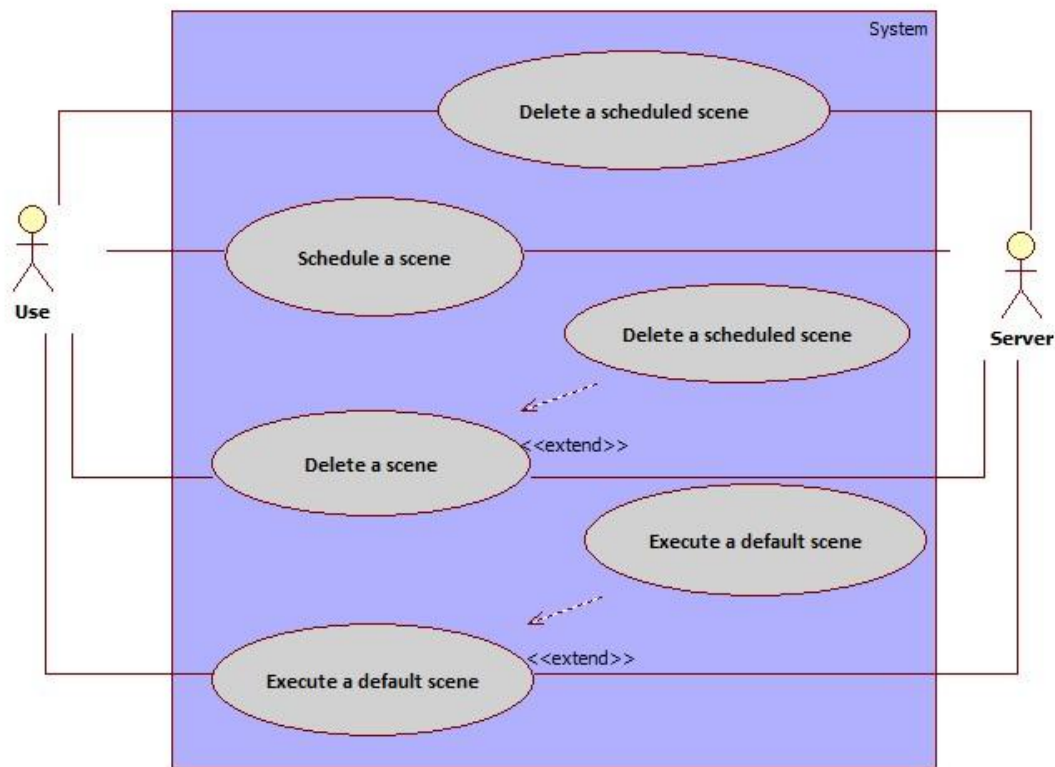


Diagram 3: Custom scene use cases diagram

**Use cases list:****Use case1: A user is able to turn/off each single light****Goal:** To turn on/off light(s)**Name:** Turn on/off a light**Actor:** user**Pre-condition:** Has internet connection for the house and the tablet/phone. The app is running in the table/phone.**Main success scenario:**

- 1 In the Lights page, the user clicks on a light number in the lights list
- 2 The user clicks on "Select" button
- 3 The app shows the Light control page
- 4 The user chooses "Light" option
- 5 The user clicks on the light icon
- 6 The light icon is turned on
- 7 The user clicks on the light icon again
- 8 the light icon is turned off

Exception:

None

Post-condition: The user successfully turns on/off each light**Use case2: A user is able to change each light's color**

Goal: To change each light color

Name: Change a light color

Actor: User

Pre-condition: Has internet connection for the house and the tablet/phone. The app is running in the table/phone.

Main success scenario:

- 1 In the Lights page, the user clicks on a light number in the lights list
- 2 The user clicks on "Select" button
- 3 The app shows the Light control page
- 4 The user chooses "Light" option
- 5 The user selects a color in the color panel and clicks on the empty space in the screen
- 6 The user clicks on the light icon
- 7 The color of the selected light is changed to the color which the user just selected in the panel

Exception:

None

Post-condition: The user successfully changes the color of the selected light

Use case3: When the user changes a light's color, he is able to change all lights that are on to a same color

Goal: To change all on-lights' colors to the same color

Name: Change all on-lights color

Actor: User

Pre-condition: Has internet connection for the house and the tablet/phone. The app is running in the table/phone. There is at one light in the app

Main success scenario:

- 1 In the Lights page, the user clicks on a light number in the lights list
- 2 The user clicks on "Select" button
- 3 The app shows the Light control page
- 4 The user chooses "All lights" option
- 5 The user selects a color in the color panel and clicks on the empty space in the screen
- 6 The user clicks on the light icon
- 7 The color of the all on-lights is changed to one color which the user just selected in the panel

Exception:

None

Post-condition: The user successfully changes the color of the all-on lights

Use case 4: A user is able to set each light's name in the app

Goal: To set up each light's name

Name: Rename a light

Actor: User

Pre-condition: Has internet connection for the house and the tablet/phone. The app is running in the table/phone. There is at least one light in the app

Main success scenario:

- 1 In the Lights page, the user clicks on a light number in the lights list
- 2 The user clicks on "Select" button
- 3 The app shows the Light control page
- 4 The user clicks on "Edit" button
- 5 The app shows the one text field with two buttons
- 6 The users fills in the light name in the text filed and clicks on "Save" button
- 7 The selected light name has been saved

Exception:

- 7a the text field is empty
- 7b the length of the text field is longer than 10.
 - ➔ The app shows a warning message:"invalid light's name, please type it again."

Post-condition: The user successfully renames the light's name in the app

Use case 5: A user is able to able to change the light intensity

Goal: To change the light intensity

Name: Change a light intensity

Actor: User

Pre-condition: Has internet connection for the house and the tablet/phone. The app is running in the table/phone. There is at least one light in the app

Main success scenario:

- 1 In the Lights page, the user clicks on a light number in the lights list
- 2 The user clicks on "Select" button
- 3 The app shows the Light control page
- 4 The user chooses "Light" option
- 5 The user drags the dot in the slide to choose the percentage of the intensity
- 6 The user clicks on the light icon
- 11 The selected light in the house has intensity based on the current percentage of the light's intensity

Exception:

None

Post-condition: The user successfully changes the light intensity

Use case 6: A user is able to set up a new server connection

Goal: To create a new server connection

Name: Add new connection

Actor: User

Pre-condition: Has internet connection for the house and the tablet/phone. The app is running in the table/phone

Main success scenario:

- 1 The user clicks on "Server connection" button in the main page

- 2 The app shows the Server connection page
- 3 The user clicks on “Add” button in the Server connection page
- 4 The app shows a separate page to ask for the new server connection information
- 5 The user fills in the server connection info. (The name of the new server connection, the IP address of new server connection, the gate number, and the password)
- 6 The user clicks on “Save” button in the separate page
- 7 The new server connection has been created and displayed in the selection list in the app

Exception:

6a the information of the server connection is not correct.

- ➔ The server connection can’t be set up; the app shows an error message box: “invalid server connection info. Please try again.”

Post-condition: The user successfully set up a new server connection in the app

Use case 7: A user is able to select a server connection

Goal: To select a server connection in the app

Name: Select server connection

Actor: User

Pre-condition: Has internet connection for the house and the tablet/phone. The app is running in the table/phone. In the app, there is at least one server connection in the selection list

Main success scenario:

- 1 The user clicks on “Server connection” button in the main page
- 2 The app shows the Server connection page
- 3 The user clicks on “connect” button which belongs to the server connection he/she wants to connect to
- 4 The app is connected to the selected server

Exception:

3a The server connection is no long available anymore

- ➔ The app shows an error message box: “invalid server connection.”

Post-condition: The user successfully selects a server connection in the app

Use case 8: A user is able to update a server connection

Goal: To update server connection information in the app

Name: Edit connection details

Actor: User

Pre-condition: Has internet connection for the house and the tablet/phone. The app is running in the table/phone. In the app, there is at least one server connection in the selection list

Main success scenario:

- 1 The user clicks on “Server connection” button in the main page
- 2 The app shows the Server connection page
- 3 The user clicks on “Edit” button which belongs to the server connection he/she wants to connect to
- 4 The app shows a separate page called “Edit server connection”

- 5 The user fills in the server connection info. (The name of the new server connection, the IP address of new server connection, the gate number, and the password)
- 6 The user clicks on "Save" button.
- 7 The selected server connection's information has been updated

Exception:

6a The test field is empty and the length of the test filed is longer than 40

➔ The app shows an error message box: "invalid type, please try it again."

Post-condition: The user successfully updates server connection information

Use case 9: A user is able to create custom scene

Goal: To create a custom scene

Name: Create a custom scene

Actor: User

Pre-condition: Has internet connection for the house and the tablet/phone. The app is running in the table/phone. There is at least one light in the app

Main success scenario:

- 1 In the main page the user clicks on "Scene" button
- 2 The app shows a separate page named Scene
- 3 The user clicks "Add" button
- 4 The app shows a separate page named Create new scene
- 5 The user clicks on "Add" button
- 6 The app shows a list of all lights in the app
- 7 The user selects one of them and clicks on "Add" button (step 5 repeats as many time as the user needs)
- 8 The user adds the scene name and description
- 9 The user set the status of the scene. (The property includes on/off, color and intensity)
- 10 The user clicks on "Create" button

Exception:

10a The scene name is existing.

➔ The app shows the pop-up box: "Name existing, please try again."

Post-condition: The user successfully creates a custom scene

Use case 10: A user is able to schedule a scene

Goal: To schedule a custom scene

Name: Schedule a scene

Actor: User

Pre-condition: Has internet connection for the house and the tablet/phone. The app is running in the table/phone. There is at least one custom scene in the Custom Scene page.

Main success scenario:

- 1 In the main page the user clicks on "Scene" button
- 2 The app shows a separate page named Scene
- 3 The user selects the scene he/she wants to schedule
- 4 The user clicks "Schedule" button

- 5 The app shows a separate page named Schedule scene
- 6 The user fills in the scheduled time, day, and repeat status.
- 7 The user clicks on “Create” button
- 8 The selected scene has been selected

Exception:

7a The time scheduled by the user is not in correct order.

➔ The app shows a pop-up box: “invalid time, please try it again.”

Post-condition: The user successfully schedule a scene

Use case 11: A user is able to delete a scheduled scene

Goal: To delete a scheduled scene

Name: Delete a scene

Actor: User

Pre-condition: Has internet connection for the house and the tablet/phone. The app is running in the table/phone. There is at least one custom scene in the scheduled scenes list.

Main success scenario:

- 1 In the main page the user clicks on “Scene” button
- 2 The app shows a separate page named Scene
- 3 The user clicks “Schedule” button
- 4 The app shows a separate page named Schedule scene
- 5 The user selects the scheduled scene he/she wants to delete
- 6 The user clicks on “Remove” button
- 7 The selected scene is removed from the list

Exception:

None

Post-condition: The user successfully deletes a scheduled scene