

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Smart Home Point for iOS
Bakalářská práce

Autor: Jaroslav, Kutnar
Studijní obor: K-AI

Vedoucí práce: doc. Ing. Ondřej Krejcar, Ph.D.

Hradec Králové

Říjen 2014

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 22.10.2014

Jaroslav Kutnar

Poděkování:

Děkuji vedoucímu bakalářské práce doc. Ing. Ondřeji Krejcarovi, Ph.D. za metodické vedení práce a jeho přínosné konzultace.

Anotace

Tato práce se zabývá návrhem, implementací a následným zlepšením vlastní aplikace pro mobilní platformu iOS. Aplikace je součástí projektu Smart Home Point, který se věnuje tvorbě automatického systému pro domácnosti pro mobilní platformy iOS, Android a WindowsPhone. Systém umožňuje centrální a automatické ovládání vybraných prvků domácnosti, jako je například nastavení teploty a světla v domě, spouštění zavlažovacího systému v pravidelných časových intervalech nebo ovládání garážových a vstupních dveří. Uživatel může svůj domov jednoduše ovládat přes tablet nebo mobilní telefon, kdekoli, kde je dostupný Internet.

V teoretické části je čtenář seznámen s projektem Smart Home Point a úvodními analýzami klíčových otázek, které vznikly před samotným začátkem vývoje aplikace.

Praktická část se věnuje návrhu (uživatelské rozhraní a grafika), implementaci samotné aplikace pro iPad a iPhone a následnému uživatelskému testování na vybraném vzorku uživatelů včetně prezentace a zapracování získaných výsledků.

Annotation

Title: Smart Home Point for iOS

This bachelor thesis deals with the layout, implementation and further improvement of the application created on my own for the iOS mobile platform. The application is a part of the project Smart Home Point that is devoted to the automation system for houses and functions for the following mobile platforms; iOS, Android and WindowsPhone. The system enables you a central and automated control of selected elements of houses such as temperature and lights control, irrigation machine scheduling or garage and main doors control. A user can easily

control his/her house via a tablet or mobile phone wherever an Internet connection is available.

The theoretical part presents the Smart Home Point project and the analyses of several key factors that were defined before the implementation itself.

The practical part presents the layout (usability and graphics), implementation of the application for iPad and iPhone and subsequent user testing with the selected sample of users including presentation and integration of received results.

Table of Contents

1	Introduction	1
1.1	Objectives	2
2	What is Smart Home Point.....	3
3	Design of Smart Home Point Solution	4
3.1	Components.....	5
4	Prerequisite Analyses	7
4.1	Market of Home Automation Systems.....	7
4.2	iOS Standards.....	9
4.2.1	Which Minimum iOS Version to Support	10
4.2.2	Which QR Code Reader to Use.....	11
4.2.3	How to Inform Users about Changes	13
4.3	Graphic and Usability Approaches in Relation to iOS Standards.....	15
4.3.1	Which User Interface Layout to Use.....	15
4.3.2	Which Views to Present	18
4.4	Requirements and Assumptions	18
5	Design of Applications.....	20
5.1	Functional Design.....	20
5.2	User Logic and Graphic Design	22
6	Implementation	25
6.1	Implementation Phase	25
6.1.1	Project Components	25
6.2	Testing Phase	30
6.3	Security.....	30
6.4	Hardware and Software Requirements	31
7	User Testing of Applications.....	32

7.1	Analysis of Target Group.....	33
7.2	Creating Testing Scenario	33
7.3	Selecting Testers.....	33
7.4	Testing	34
7.5	Analysis and Presentation of Test Results.....	34
7.5.1	Specific Findings	35
8	Discussion of Results	38
9	Conclusion and Recommendation	43
10	References	44
11	Appendix: Smart Home Point for iOS on CD.....	49

List of Images

Figure 1: Schema of Smart Home Point.....	5
Figure 2: Supported Appliances in Selected Companies	9
Figure 3: iOS Support Matrix.....	11
Figure 4: Matching Criteria for Selected Readers.....	12
Figure 5: Push Notifications Cycle.....	14
Figure 6: Table Layout with Drill-down Navigation.....	16
Figure 7: Collection View Layout with Drill-down Navigation	16
Figure 8: Single View Layout with Page Control.....	17
Figure 9: Class Diagram of Solution.....	21
Figure 10: iOS Principles Applied to iPhone and iPad Applications	22
Figure 11: Table Layout on iPhone	23
Figure 12: Views of Areas and Appliances on iPad.....	24
Figure 13: Project Tree	26
Figure 14: Control for Color Display	27
Figure 15: Scanning QR Code on iPhone.....	27
Figure 16: MVC Software Architecture.....	28
Figure 17: Layout Logic for iPhone.....	29
Figure 18: Login Page on iPhone	31
Figure 19: Improved Irrigation Part.....	40
Figure 20: Improved Temperature Part	41
Figure 21: New Overview Section	41
Figure 22: New ON/OFF ALL Control	42

1 Introduction

Just like a fire discovery brought extraordinary capabilities to mankind in a history also information technologies currently present revolutionary changes for modern society.

Everything is faster, easier and also automated. Not only the manufacturing is automated, we can also have our homes more convenient using automation. In fact, we can have centralized control of lightning, heating, ventilation and air conditioning, front doors and gates and other devices not only to improve comfort, security and efficiency of resources but also to significantly increase quality of life for such persons that would otherwise need help of others. [1]

Such environment and attitude of our society open a space for a new era of living and controlling our houses via mobile platforms. Most people still perceives an “intelligent house” as unaffordable luxury. [6] In contradistinction to paid systems, there is a free solution Smart Home Point that should provide a similar complex system and be affordable to the wider public. [2]

This bachelor thesis is a part of the project Smart Home Point from a single platform implementation point of view – the iOS platform.

The work is divided into five parts. The first part describes the whole project Smart Home Point and integration of the iPhone and iPad applications into this project.

The second part identifies prerequisite factors that are crucial to discuss prior to the implementation. The analyses require getting an overview of the market of the home automation systems, knowledge of iOS standards and approaches to graphic and usability layouts of iOS applications.

The third part is devoted to the graphic and design layouts of both applications based on the analysis done in the previous part.

The fourth part describes the implementation of the applications, solves technical aspects and requirements of the applications and presents programming components that will be either created or re-used as a free source.

The fifth part describes a process of a user testing of the applications that is of particular importance in a project lifecycle used by IT companies in present days. As a follow-up to the theory the real user testing will be organized with the aim to receive a specific feedback on the created iOS applications. The feedback will be presented and it will be discussed what suggestions can be immediately implemented and what can be used as open points where to further focus the effort.

1.1 Objectives

The aim of the work is to implement end-user applications for the iOS mobile platform, specifically for the iPhone and iPad devices, with the emphasis on the usability of the applications. To implement the applications, key prerequisite factors have to be determined and discussed, usability and graphic layout has to be prepared. Within the scope of usability there is a partial aim determined; to improve the applications based on a performed user testing and determine further development of the applications.

2 What is Smart Home Point

The Smart Home Point project is designed as an open source solution to provide “intelligent houses”. So-called intelligent houses are available to anyone who wants to remotely control his/her house via a mobile phone. [2]

As specified in the concept paper [2], *“The system enables monitor, control and socialize householders with their facilities, families and friends. The monitoring covers any useful sensor which could be connected via Ethernet network such as motion, light intensity, temperature and other ambient sensors as well as voice, video or another signal readers. The controlling enables door locking, heating, air-conditioning, lighting and networking management on spot or remotely from a mobile device or web browser. And mentioned socialization feature ability of system enables to create new interactive environment within home users who are identified by over their personal mobile devices and could share their social channels.”*

This project should fill in a gap on the market of sophisticated commercial systems where the supply of free solutions is not so high. [2]

To cover as wide audience as possible and ensure the project is beneficial, the most common mobile platforms have to be implemented. Thus, the project will support Android, iOS and WindowsPhone mobile clients. [2] This thesis will be devoted to the iOS mobile client, specifically iPhone and iPad.

3 Design of Smart Home Point Solution

The main functionality of the application that is the subject of this thesis is remote management of households, specifically for iPhone and iPad devices. Users will have the possibility to control their houses and surrounding places via simple and user-friendly environment.

The extent of the end-user application (e.g. what areas and appliances to control) will come out the prerequisite analysis as well as the final look. However, the aim is to design the applications as simple as possible so that they can be easily extensible with new appliances. I.e. the system can be deployed as a simple solution for heating, louvers and several lights and when a garage, garden or swimming pool are built they can be easily integrated into the current solution. This could also increase the value of real estates. [4], [5]

The thesis focuses on the process of design and implementation of the end-user applications on the iOS mobile platform, specifically iPhone and iPad. The work solves questions specific to the iOS platform only (e.g. what minimum iOS version to support, what QR code to use). The remaining questions (e.g. communication, security) are defined in the concept paper [2].

The end-user application is designed as “server independent”, thus it could be easily connected to any other server that supports the iOS platform.

As shown on the schema below [Figure 1] one of the main components is the end-user application that ensures comfortable connection with a household. For now, the Smart Home Point project considers interface that will focus on mobile end-user applications. [2]

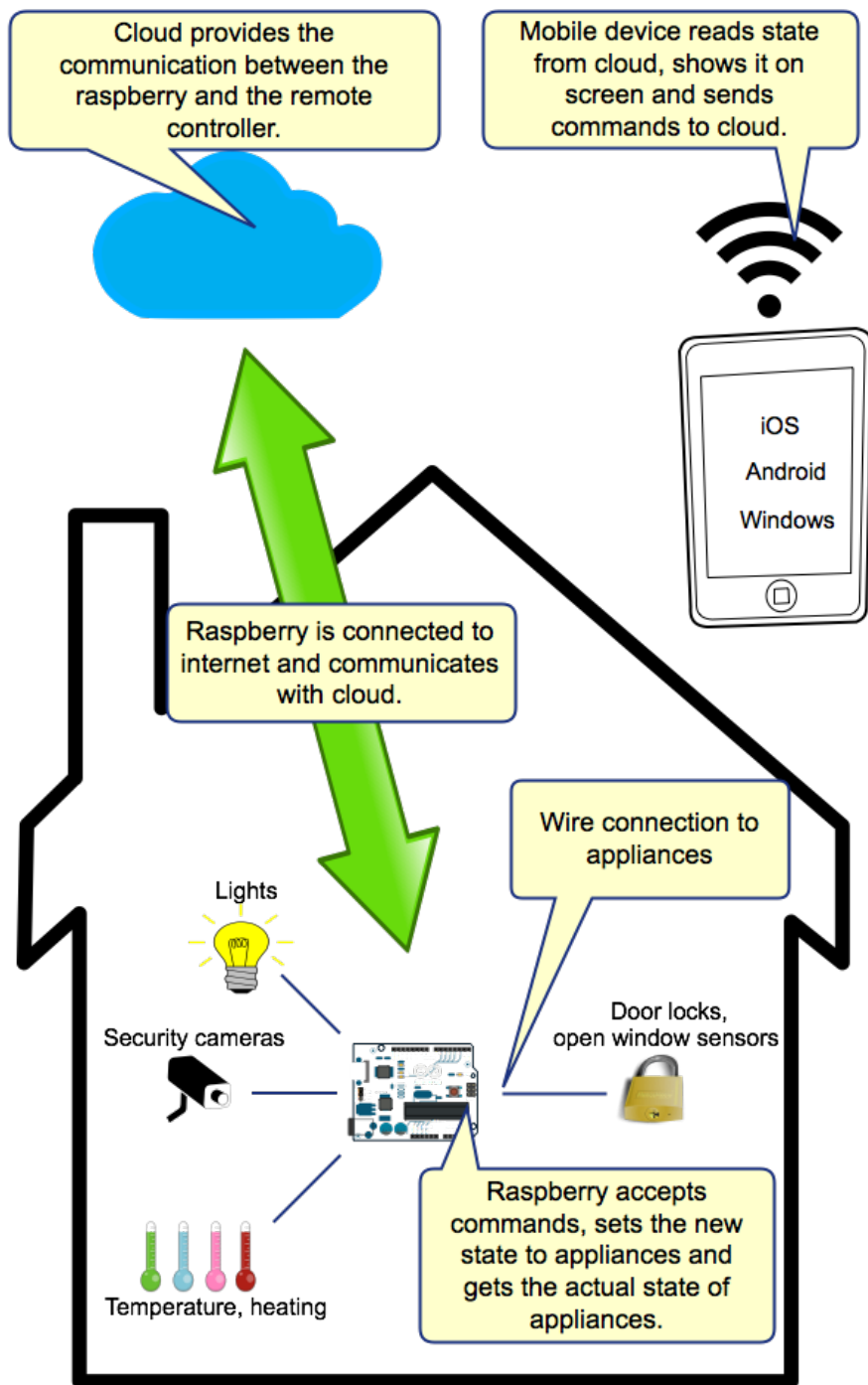


Figure 1: Schema of Smart Home Point
 Source: Own work based on [2] created in Evolus Pencil

3.1 Components

The detailed specification can be found in the concept paper [2], however, there is a simplified schema of the components and their relations [Figure 1].

The heart of the project is the raspberry computational server that “*provides monitoring, controlling and visualization of modular units in smart home place*” [2] and has the single appliances (lights, heating, air-conditioning, louvers, doors, alarm, ...) connected via wires.

Users manage their households remotely via end-user devices. Currently, the mobile platforms iOS, Android and WindowsPhone are supported only.

The whole communication is ensured via the cloud solution. [2]

4 Prerequisite Analyses

Prior to start developing the applications it is necessary to get an overview of the field of interest. For this project those penetrating fields need to be examined:

- Market of home automation systems
- iOS platform standards
- Graphic and usability approaches in relation to iOS standards

An analysis of the home automation systems market helps with better understanding of target user needs and currently available products and companies on the Czech market providing their solutions on the iOS platform.

An analysis of the iOS standards helps with getting better idea of implementing such a project as an iPhone and iPad applications, what is possible and desirable for such a platform from a technical point of view.

An analysis of the graphic and usability approaches used on the iOS platform gives the ideas of ways the information can be transferred to users in compliance with iOS standards.

The output of this chapter should be a list of requirements and assumptions serving as a base for further graphic and usability layout and implementation.

4.1 Market of Home Automation Systems

The trend of home automation systems in the Czech Republic is slowly becoming a standard for the premium segment of residential living and for newly built commercial projects as Prokop Svoboda from Svoboda&Williams stated in [6]. For cheaper flats and houses the trend is expanding very slowly as it significantly increases the final price of the property. [6] This segment rather waits for price reduction of such systems to make them more affordable. [6]

Anyway, the potential of this market, especially in the Czech Republic, is high as people are being convinced by new technologies and innovations every day. [7] However, it is quite difficult to get an overview about companies and

systems in this field as the market is not very segmented. [7] I.e. people actually have very little knowledge about companies, their products and related prices for these products. The companies unfortunately do not help to improve the situation as various companies think about “intelligent houses” in different ways.

Based on this finding, a comparison of selected sample of companies operating on the Czech market has been done to become clearer what kind of functionality should be covered in the project.

For the analysis, the web pages of the following companies operating on the home automation systems market in the Czech republic have been examined: Develop Tech, s.r.o [8], E System, s.r.o [9], SMARTTECH, s.r.o [10], Insight Home, a.s. [5], IBSolution, s.r.o. [11], ELKO EP, s.r.o [12], Loxone, s.r.o [13], Teco, a.s. [4] and Wanet, s.r.o. [14].

For most of the companies the “intelligent house” represents a flexible and modular solution with a high quality infrastructure, brings comfort, security, efficiency and economy savings, adjustable to a single client. [4], [5], [8], [9], [10], [11], [12], [13], [14].

The flexible and modular solution brings the question of the range of appliances that should be controlled and automated. See the table [Figure 2] containing the mentioned companies and supported appliances. Note that missing appliances for any company do not mean that the company is not able to provide it.

The table [Figure 2] will serve as a base for the appliances supported in the iOS applications for the Smart Home Point project.

		Company								
		Teco	Insight Home	Wanet	Loxone	IBSolution	iNELS	SMARTTECH	E System	Develop Tech
Appliances	Lights	x	x	x	x	x	x	x	x	x
	Heating	x	x	x	x	x	x	x	x	x
	Air-conditioning	x	x		x		x	x	x	x
	Louvers	x	x	x	x	x	x	x	x	x
	Electricity	x	x	x	x	x	x		x	x
	Home appliances		x				x			
	Garage doors	x	x	x		x	x			x
	Gates	x	x	x		x	x			x
	Irrigation	x	x		x	x		x		x
	Swimming pool	x	x			x		x	x	x
	Security/Alarm	x	x	x	x		x	x		x
	Video camera	x	x		x	x	x			x
	Smart metering	x								
	Biometric access		x							
	Meteorological station	x	x	x						
Sound and multimedia						x			x	

Figure 2: Supported Appliances in Selected Companies
Source: Own analysis based on [4], [5], [8], [9], [10], [11], [12], [13], [14]

For this project those appliances with 9, 7 and some with 6 crosses have been selected. As for the electricity with 8 crosses (presenting control of wall plugs), it would be difficult to cover usable scenarios thus it is not included in the list of supported appliances for the iOS applications of the Smart Home Point project.

4.2 iOS Standards

It is not so long time ago when Microsoft was considered as a gold standard in developing software for PC operating systems. However, for last few years Apple is convincing also Czech users that so called “horizontal and vertical integration” work and ensured Apple to be the world's most valuable publicly traded company. [15]

Simply speaking, Apple and its products are becoming a standard in the technology world as they helped bringing IT technology closer to average users with e.g. the concept of direct manipulation and multi-touch gestures. [16], [17]

In the world of gestures such as swipe, tap, pinch and sliders, switches, and buttons all have specific definitions within the context of the iOS operating system and developing has to be in compliance with iOS standards supported by the Apple iOS Developer Library [18]. [17]

When studying the iOS standards for the iOS applications the following questions have come out:

- Which minimum version of the iOS system should be supported.
- Which QR code reader should be used.
- How to inform users about changes.

4.2.1 Which Minimum iOS Version to Support

This question is important to clarify at the beginning because every iOS version provides and supports different range of functionality for developers (e.g. easier handling of user interface, Facebook and Twitter support).

The table below [Figure 3] shows the most commonly supported versions - currently iOS 6 and 7.

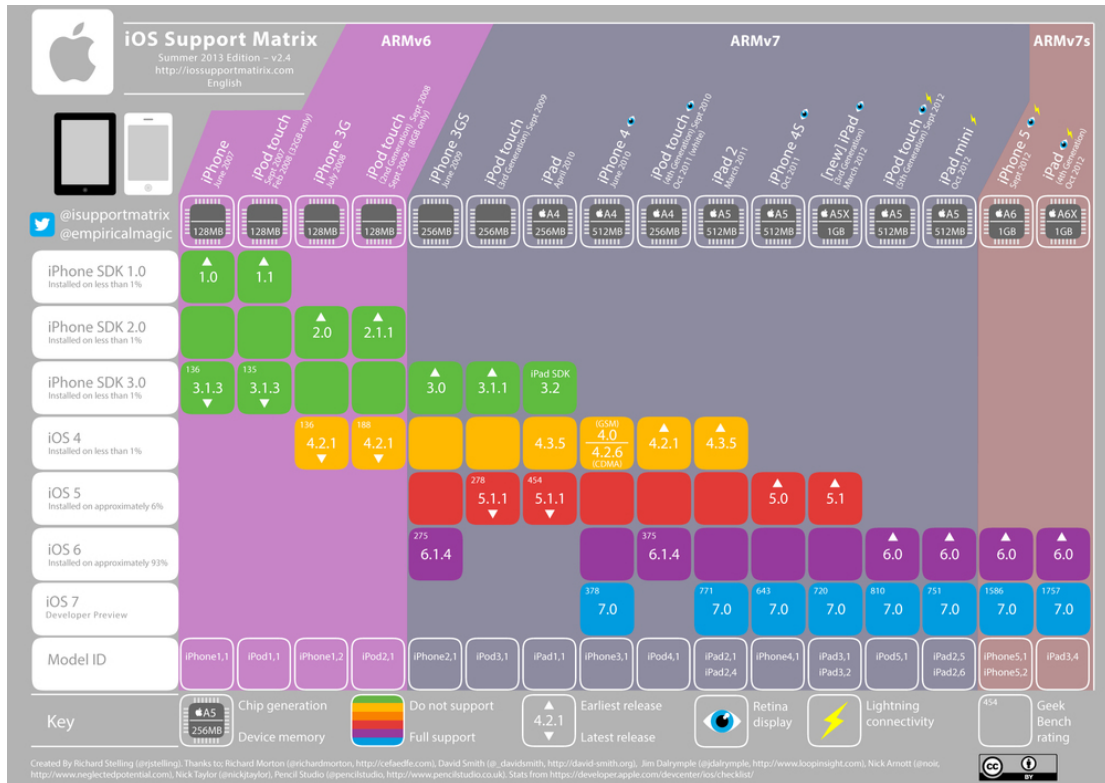


Figure 3: iOS Support Matrix
Source: [20]

At the time of developing these applications iOS 7 is still being developed (developer preview) and this version also brings the complete redesign of the look and feel of the interface compared to previous versions. It is a question whether such a big change will encourage the iOS 6 users to upgrade. [19]

With respect to these facts and certain uncertainty of the future development of the version iOS 7, the selected minimum version to support is iOS 6 (specifically iOS 6.0).

4.2.2 Which QR Code Reader to Use

The question of the project security is primarily solved on the server side, however, it was necessary to secure the access of a single mobile device to the system. This issue was solved with the help of a QR code reader that serves as an easy way of authorization of a mobile phone to the Raspberry server.

This solution opens the space for determining the most suitable reader from the current supply of readers that are able to read QR code and are available for iOS [21]; ZXing [22], ZBar [23], Shopsavvy [24], Red Laser [25] and Softek [26].

To keep the idea of the open source project, it will help having the single software components free as well. As supportive criterions references available in relevant forums and discussions and own experience are used.

Summarizing the previous text, it is necessary to choose a suitable reader that will conform to the following criteria:

- Available for iOS
- Ability to read QR codes
- Licensing and pricing, optimum is open source license
- Good references/own experience (for indecisive results)

The table below [Figure 4] shows how the selected readers match the criterions defined above.

		Criterions			
		For iOS	QR Code	Open Source	Good References
Readers	Zxing	x	x	x	x
	Zbar	x	x	x	x
	Shopsavvy	x	x	x	
	Red Laser	x	x		x
	Softek	x	x		x

Figure 4: Matching Criteria for Selected Readers
Source: Own analysis based on [21], [22], [23], [24], [25], [26]

The table [Figure 4] shows that only the Zxing, Zbar and Shopsavvy readers fulfill the requirements of iOS availability, QR codes reading ability and optimal licensing. For them the supportive criterion – references and experience – had to be applied.

All three readers are discussed on forums in similar way thus the own experience had to be gained. The Shopsavvy reader was difficult to install and run and the results of scanning a sample set of QR codes were unsatisfactory (several

codes were not scanned correctly). Installations of Zxing and Zbar readers were smooth, sample sets were scanned correctly by both readers. Thus, the final criterion was the personal preference and previous experience with the Zbar code reader.

4.2.3 How to Inform Users about Changes

Users want to be informed when something has been changed in their houses, e.g. children are back from school, doors are open in the night.

Technically we are talking about local and push notifications. They work for the applications that do not run in foreground and let users know that there is some information for them. The information can be a message, an impending calendar event, or new data on a remote server. [27]

The following options how to detect a change and inform users have been discussed for this project [27]:

- services running on background (local notification)
- widgets in the notification center
- badges on the application icons (ideally it is a combination of local and push notifications, however, for this project only the push notification will be implemented as the background services are not supported)

The first two options are not supported for iOS 6.0, [29], thus, the third option will be applied. Badges on the application icons will be used and they will also be visible in the notification center.

The following schema [Figure 5] describes how the push notifications work.

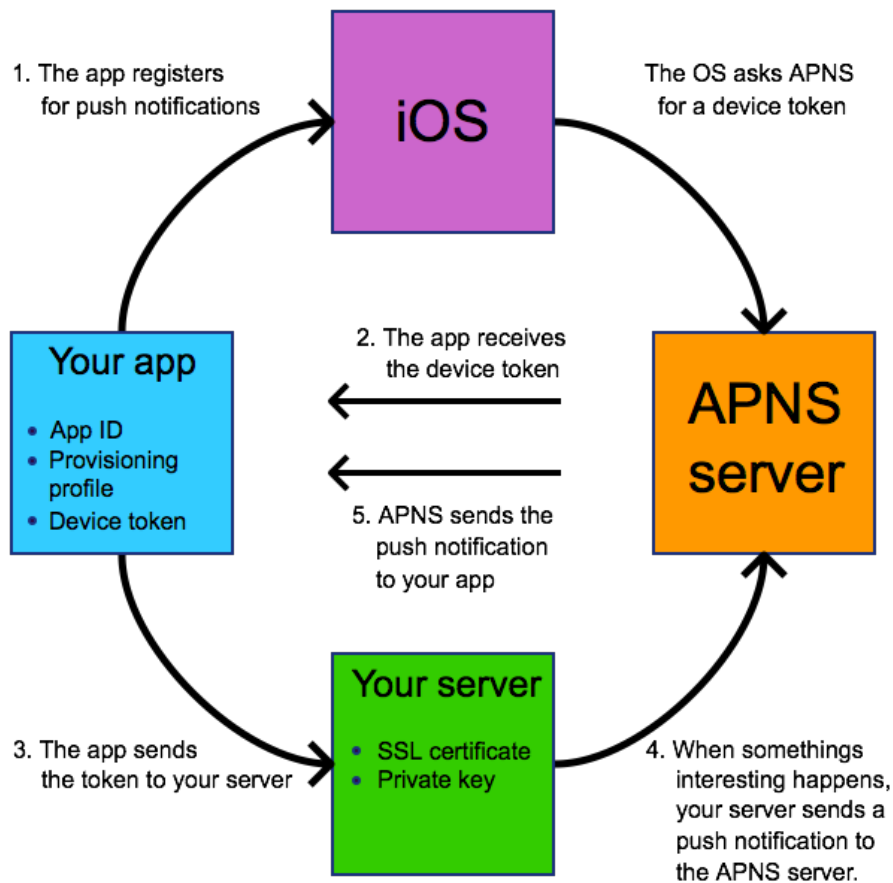


Figure 5: Push Notifications Cycle
Source: Own work based on [28] created in Evolus Pencil

1. An application enables the push notifications the user confirms the incoming notifications.
2. The application receives a “device token” which can be understood as the address that push notifications will be sent to.
3. The application sends the device token to your server.
4. When something happens your server sends a push notification to the Apple Push Notification Service (APNS) server.
5. APNS sends the push notification to the user's device (your application), the user's device shows an alert. [28]

4.3 Graphic and Usability Approaches in Relation to iOS Standards

It is often said that users spend just a couple of minutes to evaluate the application. This is also the rule that Apple fully follows. The graphic and usability layout show users what is important in the application, what their choices are and how single parts/components are related. [30]

The Apple iOS Developer Library [18] determines a set of principles that iOS applications should meet. Several principles can be directly adopted for the project (e.g. iOS App Anatomy, instant start with no redundant splash screen and with pre-filled values, launching in the device's default orientation, avoiding inconsistencies), several principles open a space for deeper thinking.

The following principles have to be discussed [30]:

- The navigation principle provides several options and it is up to the layout author to assess which user interface layout should to use.
- The views principle also provides several options but likely in a technical way, the project needs to solve this issue in a content view, i.e. which views should be present.

4.3.1 Which User Interface Layout to Use

The decision of the application structure and logic influences the impression that the application gives the user. Originally, the aim was to have the user interface layout unified across all mobile platforms to emphasize the application, however, it followed from the discussion that the single platform users have different expectations and are used to different logic.

Based on a research [31], [32], [33], [34] the following layouts are possible for iOS:

- Table layout with drill-down navigation
- Collection view layout with drill-down navigation
- Single view layout with page control

Table Layout with Drill-down Navigation

“Table views are versatile user interface objects frequently found in iOS apps. A table view presents data in a scrollable list of multiple rows that may be divided into sections.” [31]

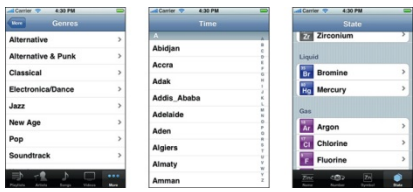
	<ul style="list-style-type: none"> + Hierarchy and structure of data + Display of detail information and controls in visually distinct groupings + Presenting a selectable list of options + Entire screen for presenting detailed information + Easy to implement using an integrated module 	<ul style="list-style-type: none"> - Less attractive design - For complex project less transparency in deeper levels - Information cannot be easily compared
---	--	---

Figure 6: Table Layout with Drill-down Navigation
Source: [31], [33]

Collection View Layout with Drill-down Navigation

“Collection Views allow content to be displayed using arbitrary layouts. They allow easily creating grid-like layouts out of the box, while supporting custom layouts as well.” [32]


	<ul style="list-style-type: none"> + Attractive design + Support of various designs (comparing to a table view) + Easy to implement using an integrated module (except for graphics) ? More suitable for iPads than iPhones (space issue) 	<ul style="list-style-type: none"> - Demanding graphic implementation - For complex project less transparency in deeper levels
---	---	--

Figure 7: Collection View Layout with Drill-down Navigation
Source: [32], [36]

Single View Layout with Page Control

A single view allows content to be displayed on one page and use page control to switch between single pages. *“A page control displays a horizontal series of dots, each of which represents a page or screen in an application.”* [34]

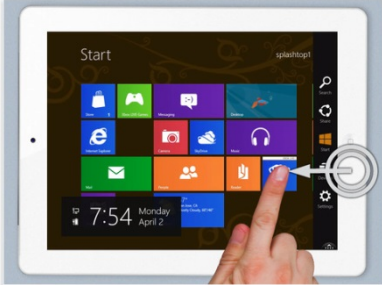
	<ul style="list-style-type: none"> + One page with no deeper levels + Entire screen for presenting detailed information + Easy to implement + Suitable for a small number of slides ? More suitable for iPads than iPhones (space issue) 	<ul style="list-style-type: none"> - Non-standard for iOS users as main navigation
---	---	---

Figure 8: Single View Layout with Page Control
Source: [34], [35]

The overview tables [Figure 6], [Figure 7] and [Figure 8] outline key advantages and disadvantages of each user layout approach. The key points can be generally grouped as graphics, structure, layout and implementation.

From the graphic point of view, the Collection view approach seems to be most attractive for users, however, it requires a skillful graphic designer.

From the structure point of view, there is no obvious winner as all approaches may face difficulties with very complex projects. The content of this project is currently not very large, however, the extent is open and one of the assumptions is a flexibility and modularity of the solution.

From the layout point of view, the Table layout approach represents the number one choice as the other two approaches are mainly intended for iPad devices only. The iPhone devices may not provide sufficient space for such graphics.

From the implementation point of view, the Table layout approach provides the easiest way of implementation that could be managed within a one-person team, e.g. no demanding graphics. The Single view approach promises an easy

implementation as well, however, using this concept represents a kind of risk as it is not very standardized for iOS applications and it is not sure whether users would accept it.

Finally, it is a decision between an attractive design application with a layout limitation and demanding graphic implementation on one hand (Collection view) and a formally more user-friendly application that is easier to implement but with less attractive design on the other hand (Table layout).

For now, the decision is made with respect to the skills of human resource – Table layout. The correctness of the decision will be verified in the further user testing and if there are any major objections to this concept they will help to shape the project in the future development.

4.3.2 Which Views to Present

How users work, how they think when using the application? Do they want to see each room or each appliance? These questions should determine how the application should look like. The Apple iOS Developer Library [18] recommends to *“focus on the needs of 80 percent of your users”*. This rule helps with the decision on the content of the iOS applications (which views should be presented).

As shown in competitive applications [10], [12], [13], various users may require various views, e.g. some people want to turn off all lights, other want to set only kitchen appliances. With respect to the “80 percent” rule mentioned above this led into the decision of having both views in the application, however, displayed differently on each device (iPad/iPhone) to fully profit from the available space.

4.4 Requirements and Assumptions

The above discussed topics should help with defining requirements and assumptions for the iOS applications of the Smart Home Point project. The discussion focused on the following questions with the subsequent outcomes:

What appliances to control: lights, heating, air-conditioning, louvers, garage and main doors, irrigation, security/alarm, video camera

Which minimum version of the iOS system to support: iOS 6.0

Which QR code reader to use: Zbar code reader

How to inform users about changes: badges on the application icons

Which user interface layout to use: table layout with drill-down navigation

Which views to present: both views (Areas and Appliances) displayed differently on each device (iPhone and iPad)

In addition to the requirements and assumptions coming from the previous analyses, it is necessary to take into the account one more requirement coming from the architecture and concept of the whole Smart Home Point project. The decision of this question is not a subject of this thesis.

What kind of communication server/client to use: Apigee API

5 Design of Applications

This chapter helps bringing the idea of the project to a specific solution with a defined architecture, logic and design.

5.1 Functional Design

Designing a concept of the solution is a vital part of each project preceding user and graphic design and the implementation. This helps with clarifying the main purpose of the application, key functionality and target audience. [30]

Main purpose: Monitor, control and socialize households as defined in the concept paper [2]

Key functionality: Settings of lights, heating, air-conditioning, louvers, garage and main doors, irrigation, security/alarm, video camera as determined in the Market of Home Automation Systems analysis

Target audience: Owners of households that are interested in having their homes automated

This rough definition of purpose, functionality and audience is further extended and a detailed design of the solution is created as a class diagram shown on the schema below [Figure 9]. The diagram is used for further translating the model into the programming code.

The diagram designs the structure of the project including its classes, attributes, methods and the object relations. The diagram comprises of the following components:

- **DataManager** – the core that manages the whole project, there is the association relationship between DataManager and:
 - **DataProvider** that ensures the communication with the cloud solution. There is the generalization relationship between **DataProvider** and **UserGridDataProvider** that serves as API for the cloud communication.

- **Area** that represents locations controlled by the automated system (e.g. first floor, second floor, garden,...). **Area** is associated with **Room** (e.g. first floor has a kitchen, living room and bathroom, garden has a greenhouse). Room is associated with **Appliance** (e.g. kitchen has a defined set of appliances). There is the generalization relationship between **Appliance** and **Temperature**, **Louver**, **Irrigation**, **Camera**, **Lock**, **Security** and **Light**.

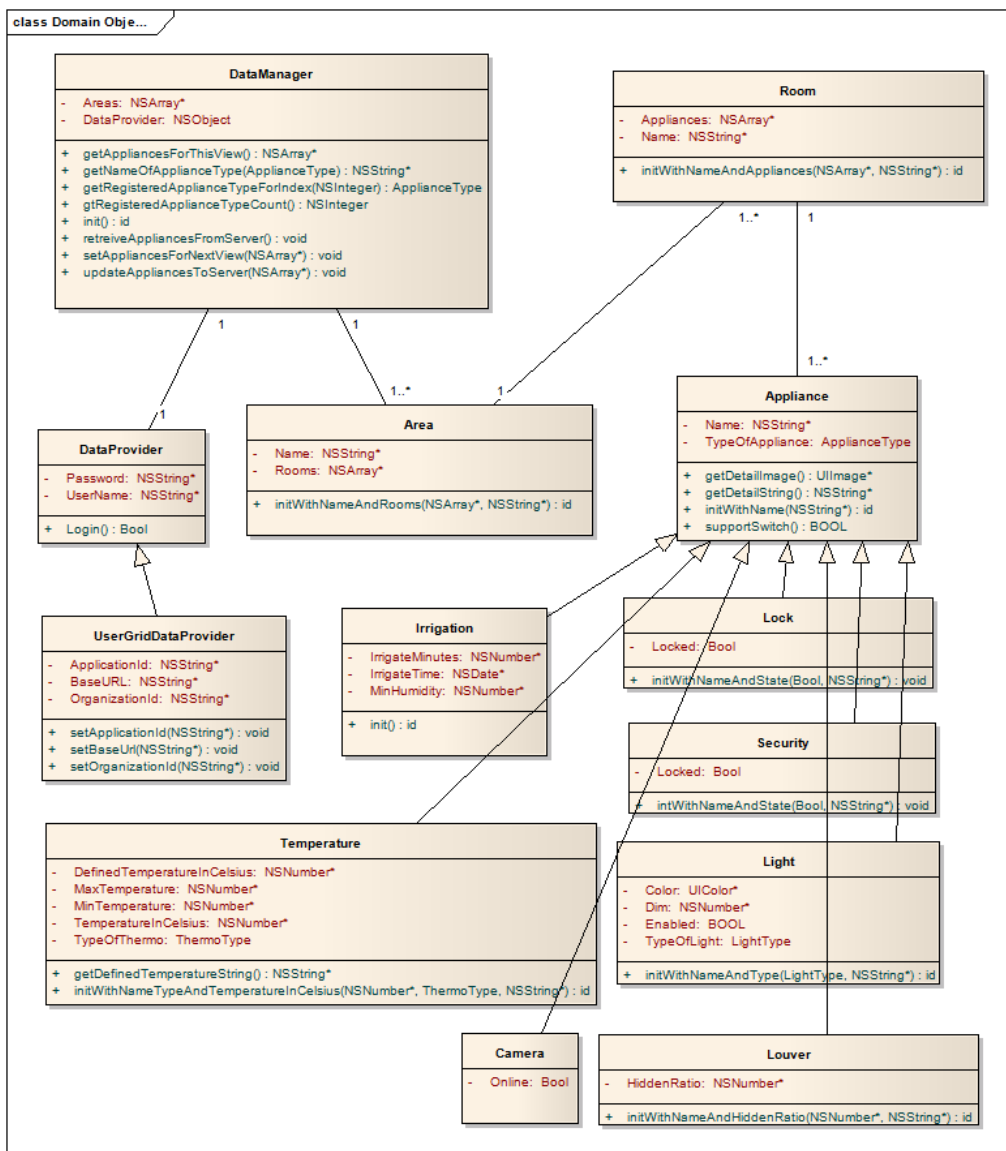


Figure 9: Class Diagram of Solution
 Source: Own work designed in OmniGraffle

The schema above [Figure 9] reflects the state of the project in the analysis phase. Due to a future user testing code improvements may be done resulting in

small changes in the schema. The whole process of the design continues with specific graphic and user layout that gives the project real shape.

5.2 User Logic and Graphic Design

Visual impression of the application plays the key role in the process of accepting and enjoying the product and together with the usability of the application they help bringing the product closer to users.

The Apple products are well known for great understanding of users. Thus also these applications were designed in compliance with iOS principles [30]. The main iOS principles applied are summarized in the table below [Figure 10]. Based on the analysis done in the chapter Which User Interface Layout to Use and with following the iOS principles, the following layout of the applications has been created:

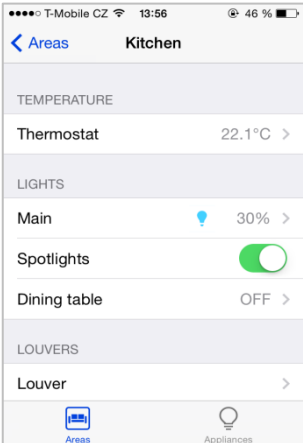
	<ul style="list-style-type: none"> - Both orientation support - Retina display support - Launched in device's default orientation - Proper navigation - Tab bars for different content - Direct manipulation (gestures) - Whole screen advantage used - Translucent elements applied (e.g. Control Center) - Text legible at every size - Single font in whole application - Icons precise and lucid - Tappable controls 44x44 pt - Adornments subtle and appropriate - Setup info supplied (default values, scenarios) - Easy to make choice (selecting controls instead of entering values) 	<ul style="list-style-type: none"> - No splash screen - No Close/Stop button - No disclaimers or agreements
---	--	--

Figure 10: iOS Principles Applied to iPhone and iPad Applications
Source: Own work, [30]

There will be the table layout with drill-down navigation. The concept of the application is similar to the global Settings part on iPhone/iPad that also uses the table layout thus the decision is supported with this “standard” concept for users experienced in iPhone/iPad logic. Also from the developing point of view, selecting this layout enabled to implement only one code shared for both views of iPhone and iPad.

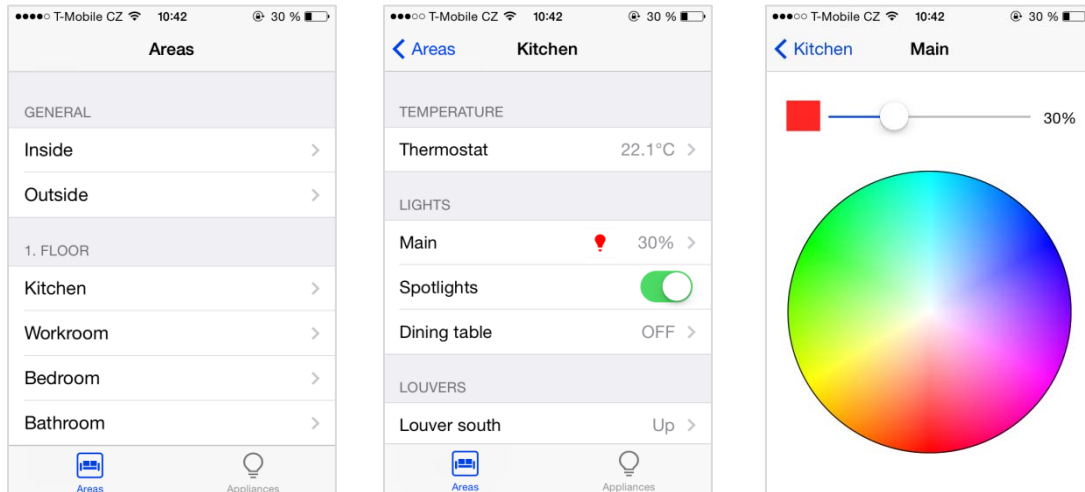


Figure 11: Table Layout on iPhone
Source: Own work displayed on iPhone

As for single views, there will be a different approach for each device:

- For iPhone – A separate view for Area and Appliances is created with respect to the screen size and comfortable control as shown on the screenshot above [Figure 11].
- For iPad – One view for Area and Appliances is created as the iPad screen provides enough space for comfortable control as shown on the screenshot below [Figure 12].

Both applications support the portrait and landscape views.

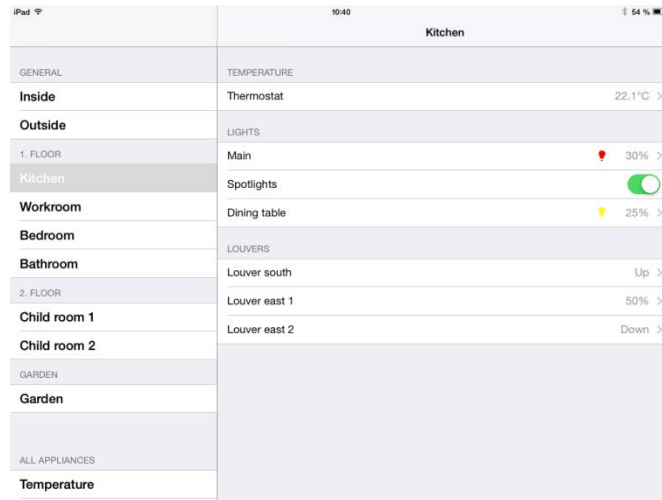


Figure 12: Views of Areas and Appliances on iPad
Source: Own work displayed on iPad

As for graphics, default iOS colors are used, i.e. for every iOS version the application is launched with its default colors. Size of icons (application icon, bar and navigation icons) and images are used in compliance with iOS principles [30].

6 Implementation

This part is devoted to the implementation of the iOS applications as a part of the Smart Home Point project [2]. The solution is based on the assumptions and requirements determined in the Prerequisite Analyses chapter and in the Smart Home Point concept paper [2] and user and graphic layout defined in the Design of Applications chapter.

The solution follows the simplified development cycle for smaller projects; planning, analysis and design, implementation, testing (including user testing), deployment and maintenance. [37] For the purpose of this thesis, analysis, design, implementation and testing phases are done. For the analysis and design, see the respective chapters above.

Moreover, the important questions for end users such as security and requirements for the application are discussed.

6.1 Implementation Phase

The project has been implemented in Xcode 5.1, MacOS X integrated development environment using the objective-c programming language. This set of features supports the whole process of developing the applications; from a user layout design, implementation, optimization and testing to the distribution via App Store. [38] As discussed in the chapter Which Minimum iOS Version to Support, the project is intended for iOS 6.0 and higher versions.

6.1.1 Project Components

The project has been designed in the way shown on the screenshot below [Figure 13]:

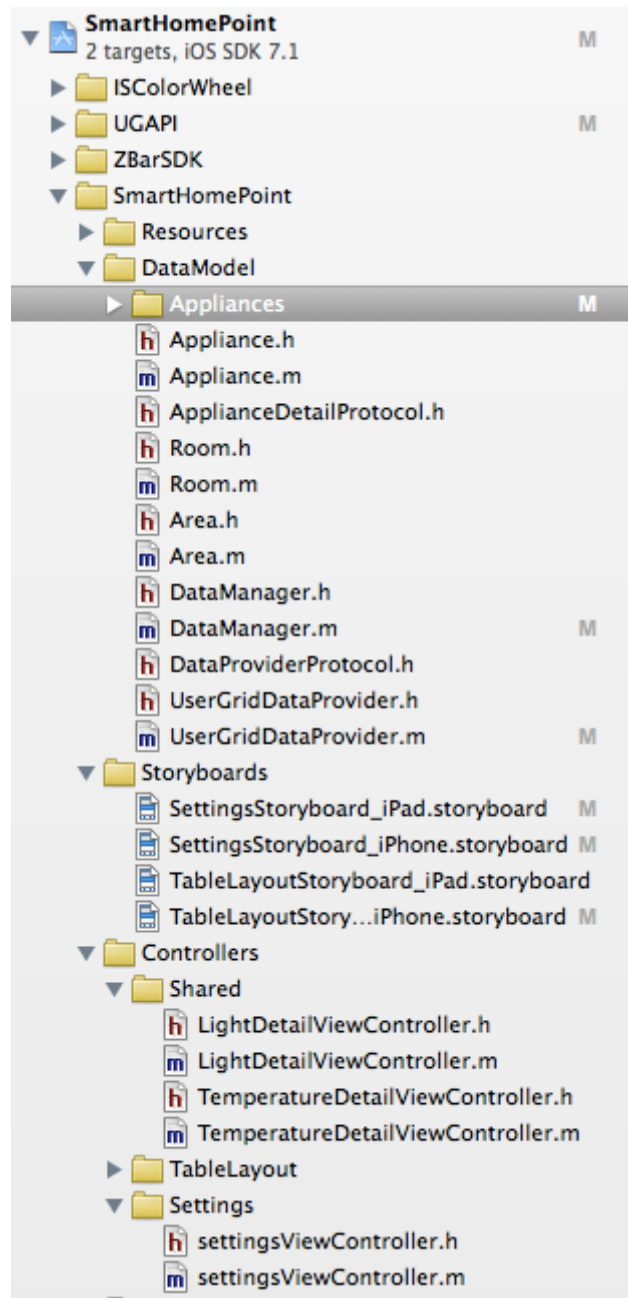


Figure 13: Project Tree
Source: Own work displayed in Xcode 5.1

ISColorWheel – This is an external library used for setting the color of lights. The library is available for free. [40]

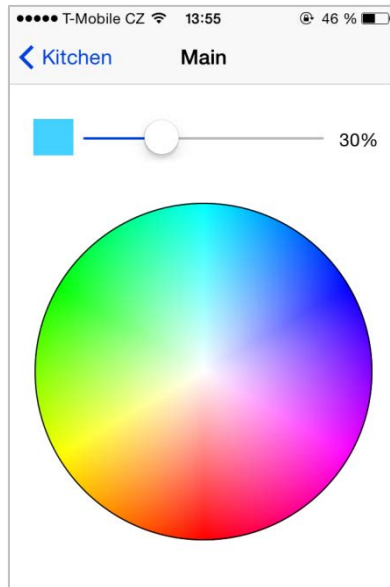


Figure 14: Control for Color Display
Source: Own work displayed on iPhone

UGAPI – This is an external library used as a component for easy communication with the Usergrid backend (the server from the Apigee company). The Usergrid backend server was selected by the leader of the project. The library is available for free. [41]

ZBarSDK – This is an external library used for scanning the QR code on the login page. The library is available for free. [23]



Figure 15: Scanning QR Code on iPhone
Source: [39]

SmartHomePoint – The whole project is based on the Model-view-controller (MVC) software architecture that divides the data model, GUI and the control logic

into three independent components. This makes the model easily changeable in one part without small or even any influence to other parts. [42]

This model requires creating of three components; Model (called DataModel in this project), View (called Storyboards in this project) and Controller (called Controllers in this project). Model represents domain specific information that the application works with. View converts data represented by the Model to the form suitable for the interactive presentation to users. Controller reacts to users events and ensures changes in the Model or View. The schema below [Figure 16] shows single components and their relations.

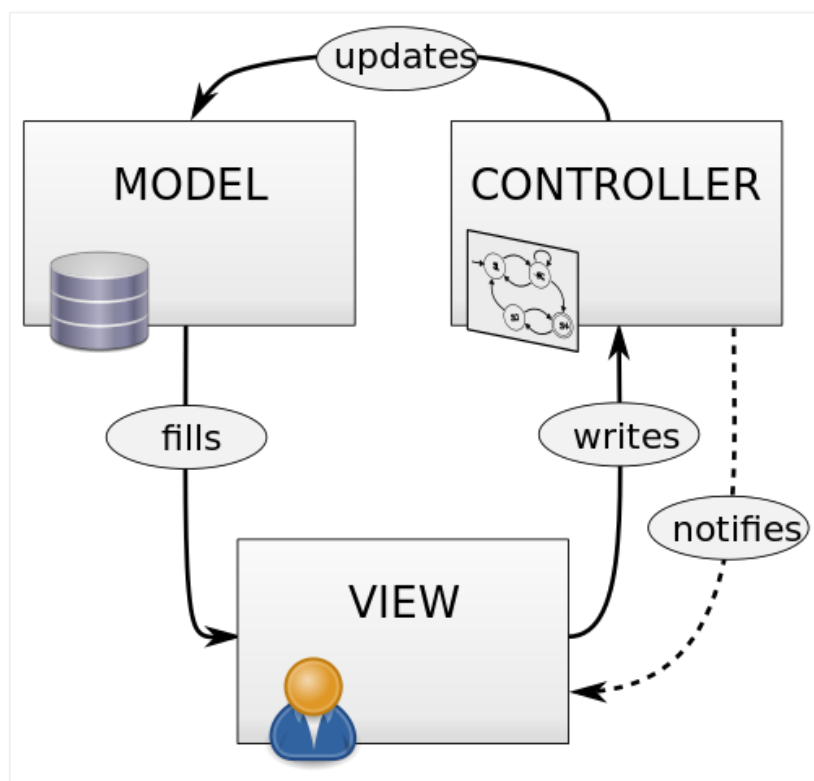


Figure 16: MVC Software Architecture
Source: [42]

- **Resources** – This folder contains images used for GUI graphics.
- **DataModel** – This folder contains data models of single appliances, i.e. structure of the whole application. Data models are independent on layouts.
 - **DataManager** – This class manages and provides data, i.e. receives data including states changes from DataProvider, provides data via Views and ensures correct settings (e.g. Celsius/Fahrenheit).

- **DataProviderProtocol** – This class represents an interface for linking any implementation of a server for Smart Home system. Currently, the UserGridDataProvider implementation of a server is used.
- **UserGridDataProvider** – This class represents an implementation of a server created using apigee/usergrid-ios-sdk. The server ensures communication with iPhone/iPad including security.
- **Storyboards** – This folder contains single layouts for iPhone and iPad. This implementation enables flexible settings – any type of layout can be added later on. The schema below [Figure 17] shows visual representation of the layout logic for the iPhone device.

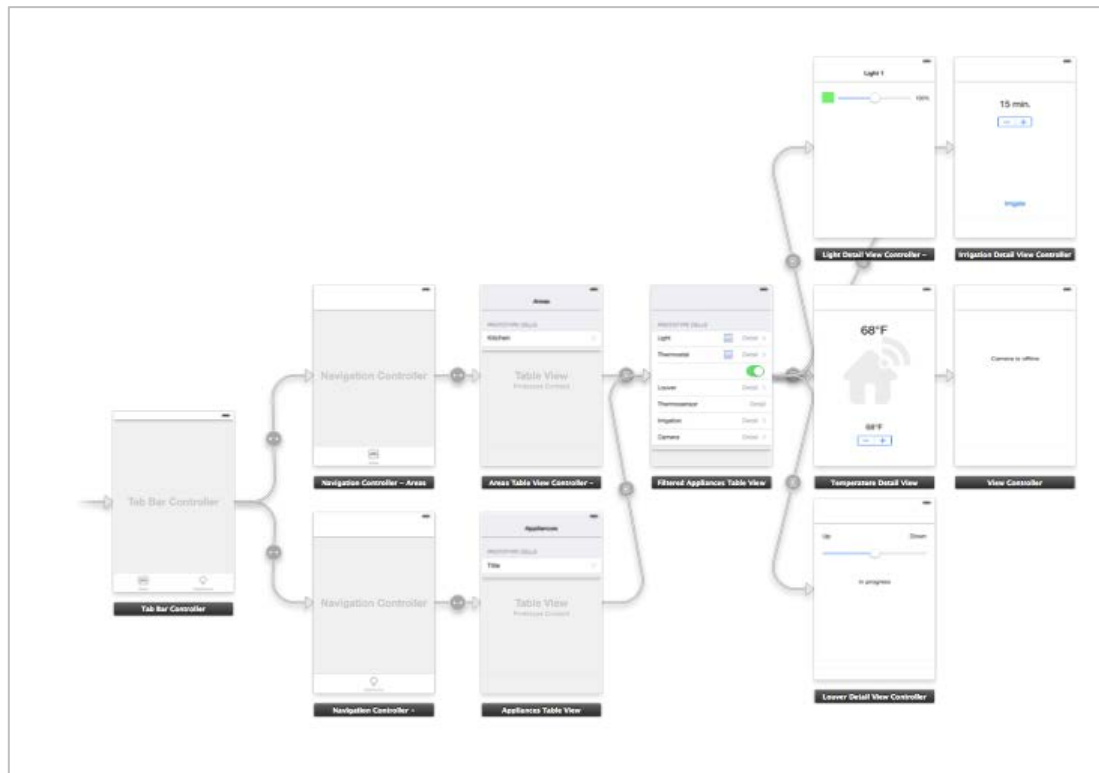


Figure 17: Layout Logic for iPhone
Source: Own work displayed in Xcode 5.1

- **Controllers** – This folder contains controllers that manage what data will be displayed on iPhone/iPad.

6.2 Testing Phase

When delivering the project, it is the most important responsibility of the team to have the project without functional and formal errors. For this project the testing has been done in two phases:

- Developer testing phase
- User testing phase

Developer testing phase was performed in parallel to developing. After a certain part was developed, the testing was done using the iOS Simulator iPad/iPhone integrated in Xcode.

This approach enabled quick and effective development with zero defects. As developing and testing were done by one person, another insight was more than welcome. Thus, user testing has been suggested to help improving the application and bring it closer to the target audience. For the user testing purposes, the application will be available directly on iPhone and iPad. This is only possible if you have the paid account in Apple [43]. User testing is discussed in the User Testing of Applications chapter.

6.3 Security

With the growing number of threats that IT world faces every day the security is becoming a key feature of each project and has to be assured.

Security for the Smart Home Point project is implemented on the server side which means that correctness of data sent to iPhone/iPad is verified on the server. The only security element for iPhone/iPad is logging in the application using the login name and password. Data is sent to the server to be authenticated.

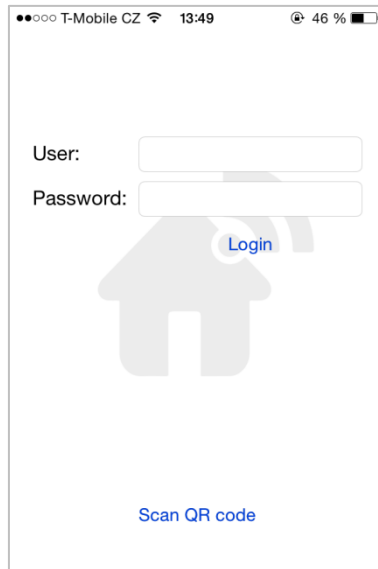


Figure 18: Login Page on iPhone
Source: Own work displayed in Xcode 5.1 iOS Simulator

6.4 Hardware and Software Requirements

With every release of software it is necessary to inform users about hardware and software requirements to be guaranteed.

As for hardware, the only requirement is the device that supports at least the 6.0 version of iOS as defined in the chapter Which Minimum iOS Version to Support. According to the table [44], the following devices are supported:

- iPhone 3GS and higher
- iPad 2, iPad 3rd generation, iPad 4th generation, iPad Air

As for software, the only requirement is Internet connection to be able to connect to the server that provides authentication and data.

7 User Testing of Applications

The importance of user feedback is growing and no company would afford to overlook user's point of view and their comments. Usability often determines how successful the application can be on the market and customers become considering user-friendly applications as standard. [45]

The theory presents several ways of improving usability of the applications; user testing, personas, prototyping, heuristic evaluation, interviews, user stories etc. [46]

For this project a user testing based on *“watching users working with the application to help to discover “mistakes” hidden to developers and designers was chosen”*. [46]

A user testing simply answers the questions; how easy is to use the application, how easy is to learn the application and how convenient is the application to the end user and focuses on the parameters flow – navigation – layout – speed – content. [47]

A user testing has the following phases [48]:

- Analysis of the target group of the application and needs of such a group
- Creating a testing scenario
- Selecting testers
- Testing
- Analysis of test results
- Presenting results

Speaking about benefits, this technique should result in better quality of the application that is easier to use and more readily accepted by users. Moreover, it should shorten the learning curve for new users. [47]

7.1 Analysis of Target Group

Prior to testing, it is necessary to know who the users of the application are and what the application should offer them. Only defining correct target users will ensure relevant and useful results.

It is recommended to recruit participants (testers) similar to the application users. [49] In this case, the target group comprises of owners of households who have an experience with home automation or who plan to have their houses automated. Regarding the level of computer knowledge, an average end user is expected.

7.2 Creating Testing Scenario

Defining a suitable task presents a good way of finding how users work with the application, especially so called “task-based scenarios” that only define what users should do but not how to achieve the goal. The scenarios should be brief and accurate. [50]

Scenario 1

You are coming from holiday and you want to have your house cosy before you enter it. So you want the temperature 22 degrees of Celsius in the kitchen and bathroom, yellow lights on in the bedroom and all louvers up.

Aim of this scenario: to see how a user searches for specific settings.

Scenario 2

Tell us the current setting of the whole house.

Aim of this scenario: to see how a user thinks on the global level.

In addition, each user will be asked about the user layout defined in the chapter Which User Interface Layout to Use as table layout with drill-down navigation.

7.3 Selecting Testers

Testers should be selected within a target group defined in the chapter Analysis of Target Group. As Jacob Nielsen, a famous Danish usability consultant,

recommends there will be 5 testers. Such a number will help finding almost as many problems as it would be possible with more testers. [51] Each tester will conduct the same test on each device (iPhone and iPad).

7.4 Testing

Each tester has the application installed on his/her iPhone/iPad. Then he/she will try to complete given test scenarios. The behavior of each tester is observed by the author of the applications to get the real experience how testers work. The testers will not get any help during testing. Testing is performed with one tester, the rest of tester is not present to avoid affecting the results. During the testing testers are encouraged to comment the work.

The observations as well as comments coming from testers and any other possible issues e.g. testers confusion are documented. The given results will be analyzed and presented in an appropriate form. [52]

7.5 Analysis and Presentation of Test Results

The user testing was beneficial. It discovered several findings, missing functionalities and testers had useful comments that helped to improve the applications and also proved given questions.

As defined in the introduction of the chapter, the user testing focused on the following parameters: flow – navigation – layout – speed – content. Observations and comments are divided according to the defined parameters.

Flow (Testers are able to follow the logic of applications)

- Clear logic hierarchy
- Intuitive management

Navigation and Layout (Testers are able to navigate fluently and find what they expect where they expect)

- Clear visual hierarchy
- Intuitive management

- User testing did not prove Table view layout as a good or bad solution. However, availability of more attractive design would be appreciated by testers.
- Testers agreed on Table view layout as sufficient for a free solution.

Speed and Content (Testers are able to set what they want in reasonable time)

- All testers were able to finish the scenarios in reasonable time.
- Speed of testing was reasonable for most of functionalities expect for those with unclear settings (Thermostat, Irrigation). See the specific findings below.

7.5.1 Specific Findings

During the user testing 11 specific comments were documented. 7 of them have been considered as possible and easy to implement immediately. 4 of them were considered as useful as well, however, they do not have either quick solution or the solution goes beyond the scope of this thesis.

The following findings will be implemented:

- 1) Login page without possibility to rotate. It is launched in a device's default orientation.
→ **It will be possible to rotate login page on iPhone and iPad devices.**
- 2) Missing captions for Irrigation settings. It is not clear what is being set.
→ **The Irrigation part will be completely reworked for iPhone and iPad devices.**
- 3) The Irrigate button is not obvious.
→ **The Irrigation part will be completely reworked for iPhone and iPad devices.**
- 4) Missing timing for Irrigation.
→ **The Irrigation part will be completely reworked for iPhone and iPad devices.**
- 5) Missing caption for Thermostat settings. It is not clear what is being set.

- **Captions will be added to the Temperature part for iPhone and iPad devices.**
- 6) Missing a quick overview of house settings.
→ **System of quick alerts/warnings will be added for iPhone and iPad devices.**
- 7) Missing global settings for some appliances to switch on/off all (e.g. all lights, louvers).
→ **A control ON/OFF ALL will be added to the selected appliances for iPhone and iPad devices.**

See the Discussion of Results chapter for screenshots of implemented findings.

The following findings will not be implemented, however, they will be kept for further development:

- 1) Missing possibility to set a scenario, a set of subsettings, e.g. for holiday, for evening (e.g. lock doors at 9 pm and start irrigation at 8 pm).
→ **The functionality should be handled on a server to ensure up-to-date settings for all end-user applications (regardless of a platform) used within a household.**
- 2) Settings of lights and louvers are too complicated. ON/OFF is sufficient for most of times, other settings (e.g. intensity of lights) is typically done occasionally.
→ **The settings are done in compliance with iOS standards for the selected table layout. Investing to a more attractive collection view design with icons would solve this issue.**
- 3) Having a visual representation of each appliance could help testers to navigate quicker in the application, i.e. seeing an icon of a lock is faster than reading the text “lock”.
→ **The issue is a limitation of the selected table layout. Adding icons to texts would look nice for iPad with sufficient space, however, for iPhone it would be overfull. Investing to a more attractive collection view design with icons would solve this issue.**

4) Too much free space for some controls on iPad.

→ **The issue is a limitation of the selected table layout. Investing to a more attractive collection view design with icons would solve this issue.**

8 Discussion of Results

The main aim of this thesis was to design and implement end-user applications for the iOS mobile platform, specifically for the iPhone and iPad devices, as a part of the complex project Smart Home Point [2] with the emphasis on the usability of the applications.

To implement the applications, key prerequisite factors have been determined and discussed. Based on the examination of the following fields; market of home automation systems, iOS platform standards, graphic and usability approaches in relation to iOS standards, the specific questions have been opened and answered. Results of these analyses served as requirements and assumptions for further graphic and usability layout and implementation.

Within the scope of usability, a partial aim has been determined – to improve the applications based on a performed user testing and determine further development of the applications.

The resultant applications fulfill the requirements and assumptions defined at the beginning. The applications support the iOS version 6 and higher. The security of the applications is ensured by scanning the access information to a server via a QR code using the Zbar code reader. However, the assumption of the minimum supported version iOS 6 does not show clear results. On one hand, the range of the supported version is large and covers most of the iOS users. On the other hand, the iOS 6 version contains a limitation for notifications described in the chapter How to Inform Users about Changes. Extended notifications features will be available in the version iOS 8 that should be launched in October 2014. [56] This disadvantage could be partly solved by implementing an extended support of notifications. This would be available only for iOS 8 users. Such an implementation is left for further development.

The whole layout is defined as a table with drill-down navigation providing both types of information; areas and appliances to control.

For a start-up implementation, the following appliances were selected: lights, heating, air-conditioning, louvers, garage and main doors, irrigation,

security/alarm, video camera with ways of control (suggestions from user testing are implemented):

- **Lights** – distinguish between spotlights that are ON/OFF and other lights that can have a color and intensity set. A possibility to switch all lights on/off is available. This feature is available in the menu Lights.
- **Heating/Air-conditioning** – set via a stepper control that enables to add a temperature in °C/°F depending on the settings in the iPhone/iPad Settings part. One GUI control is used for two appliances as both influence temperature in a household. This feature is available in the menu Temperature.
- **Louvers** – set a position of louvers, fully pulled up/down, everywhere between showing progress. A possibility to set all louvers up/down is available. This feature is available in the menu Louvers.
- **Garage and main doors** – set ON/OFF. A possibility to set all doors on/off is available. This feature is available in the menu Locks.
- **Irrigation** – set a number of minutes that the irrigation will work, the humidity threshold value that triggers irrigation and timing when the irrigation should start. This feature is available in the menu Irrigation.
- **Security/alarm** – set ON/OFF. A possibility to set all locks on/off is available. This feature is available in the menu Alarms.
- **Video camera** – display a selected camera screen. This feature is available in the menu Cameras.

Due to a limitation of iOS 6.0 [29], notifications are solved using badges on the application icons and in the notification center.

The whole implementation is in compliance with iOS standards and principles [30] that were proved as sophisticated during the user testing. The implementation was based on the Model-view-controller (MVC) software architecture that made the project easily changeable in one part without small or even any influence to other parts. Also usage of external libraries ensured convenient work.

The user testing proved that usage of the applications is self-evident, just by looking at the application the average users knew what to do and how a task should be done. The visual and logical hierarchy is clear, things that are related logically are also related visually. As most of the testers are experienced in home automation, they either have a competitive paid solution or are interested in getting one, they pointed to 11 specific issues that were either of cosmetic character, unclear or missing at all. All of them were considered as beneficial and making the application more user-friendly, however, not all of them were possible to implement. 7 of them were quickly and easily implemented. 4 of them were kept for future development as the solution goes beyond the scope of this thesis or requires extensive graphic changes. See the following overview of implemented findings:

Issue 1: The login page can be rotated.

→ An obvious solution, no need to add a screenshot.

Issue 2, 3, 4: The Irrigation part is reworked.

→ It is now possible to set a number of minutes that the irrigation will take, the humidity threshold value that triggers irrigation and timing when the irrigation should start [Figure 19].

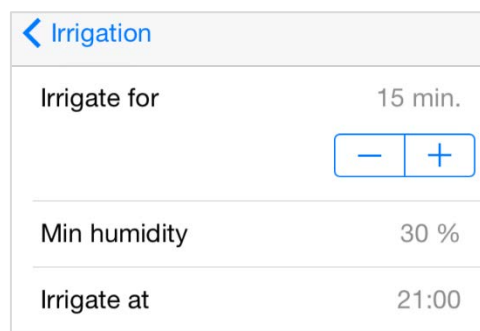


Figure 19: Improved Irrigation Part
Source: Own work displayed on iPhone

Issue 5: Captions for the Temperature part are added.

→ It is now possible to easily set a temperature as hints are added [Figure 17].

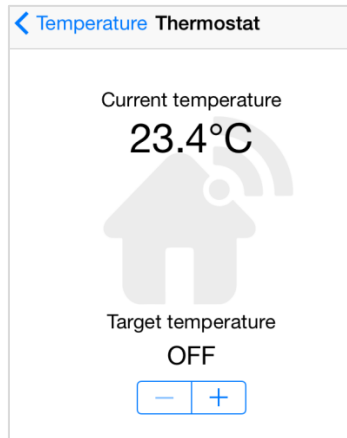


Figure 20: Improved Temperature Part
Source: Own work displayed on iPhone

Issue 6: A quick overview of house settings is added.

→ It is now possible to view inconsistencies in the household (e.g. doors cannot be closed), adjust the view with user's favorite features (e.g. alarm is the first the user wants to check). This view is available on the new tab Overview and is displayed when the application starts by default [Figure 21]. Favorite items can be modified on the server.

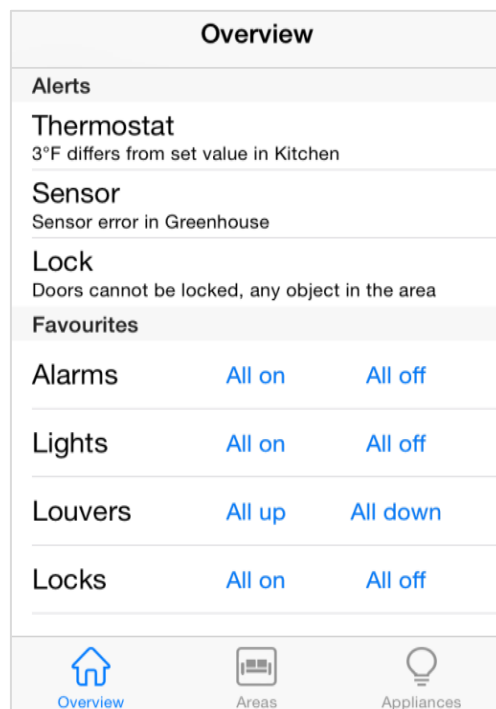


Figure 21: New Overview Section
Source: Own work displayed on iPhone

Issue 7: Lights, louvers, alarms and locks have the possibility to switch them all on/off.

→ It is now possible to easily switch on/off all lights, louvers, alarms and locks either within a selected room (as shown on [Figure 22] below) or within a whole household on the Appliances tab or on the new tab Overview. The feature is solved by single buttons All on and All off (All up/All down depending on the feature).



Figure 22: New ON/OFF ALL Control
Source: Own work displayed on iPhone

The implemented improvements ensure more convenient work that is done on a daily basis with the applications. However, no further user testing was done with new functionality. This is left for further development.

The question of the table layout design was not clearly proved as a good or bad solution. However, most of testers accepted the layout easily and commented that design is sufficient for a free solution.

9 Conclusion and Recommendation

iOS applications, especially iPads, learnt us to manage and control complicated systems in an easy and intuitive way. This was also the idea that accompanied the whole project.

The main aim of this thesis was to design and implement end-user applications for the iOS mobile platform, specifically for the iPhone and iPad devices, as a part of the complex project Smart Home Point [2] with the emphasis on the usability of the applications.

The thesis succeeded in implementing the end-user applications (for iPhone and iPad) that help moving the free Smart Home Point solution towards its real deployment. This was also proved by the user testing; parts that are based on the iOS principles are self-explanatory and usage is straightforward. In addition, several cosmetic ambiguities and functional deficiencies were found by testers. All findings were considered as reasonable and beneficial. Those that were able to quickly and easily implement are fixed, the rest of findings (4 issues) requires either extensive graphic development or is beyond the scope of this thesis. With growing the project, the team may hire a graphic designer to ensure more attractive design.

Despite of quite positive impression of the applications coming from the user testing, a good application will only be shown by continuous using in time.

The work can serve as a handbook for readers that are interested in managing a smaller project for the iOS mobile platform. The applications as parts of free solution should fill the gap on the commercial market of home automated systems.

“Technology is nothing. What's important is that you have a faith in people, that they're basically good and smart, and if you give them tools, they'll do wonderful things with them.” (Steve Jobs)

10 References

- [1] Wikipedia. Home Automation. [online]. [cit. 2013-07-20]. Available at: https://en.wikipedia.org/wiki/Home_automation
- [2] BEHAN, Miroslav, KREJČAR Ondřej. *Open Source Based Concept of Intelligent House*. [online]. [cit. 2013-07-31]. Available at: <http://ebooks.iospress.nl/publication/33882>
- [3] SMYTH, Neil. *iPhone iOS 6 Development Essentials*. First Edition. 2012. ISBN-13 978-1479211418.
- [4] Teco. Ovládej svůj dům. [online]. [cit. 2014-04-21]. Available at: <http://controlyourhouse.com/cs/>
- [5] Insight Home. [online]. [cit. 2014-04-21]. Available at: <http://www.insighthome.eu/inHome.html>
- [6] Insight Home, a.s.2013. Realitní kongres 2013 – Insight Home partnerem [online]. [cit. 2014-05-07]. Available at: <http://www.insighthome.eu/news/2013-Realitni-kongres-Insight-Home.html>
- [7] Digitální domácnost.cz. Inteligentní dům? Ano, ale není inteligentní dům jako inteligentní dům. [online]. [cit. 2014-05-07]. Available at: <http://www.digitalnidomacnost.cz/inteligentni-dum-ano-ale-neni-inteligentni-dum-jako-inteligentni-dum/>
- [8] Develop Tech. [online]. [cit. 2014-05-09]. Available at: <http://www.developtech.cz/>
- [9] E System. [online]. [cit. 2014-05-09]. Available at: <http://www.esystem.cz/>
- [10] SMARTTECH. [online]. [cit. 2014-05-09]. Available at: <http://i-smarttech.cz/>
- [11] IBSolution. [online]. [cit. 2014-05-09]. Available at: <http://www.ibsolution.cz/>
- [12] iNELS. [online]. [cit. 2014-05-09]. Available at: <http://www.inels.cz/>
- [13] Loxone. [online]. [cit. 2014-05-09]. Available at: <http://www.loxone.com/cscz/start.html>
- [14] Wanet. [online]. [cit. 2014-05-09]. Available at: <http://www.wanet.cz/>
- [15] SPARKS Daniel. *How Apple Set a New Standard in Tech*. The Motley Fool. [online]. [cit. 2014-05-27]. Available at:

<http://www.fool.com/investing/general/2013/10/15/how-apple-set-the-new-standard-in-tech.aspx>

- [16] DANIEL P. *iOS 6 vs iOS 7 design differences comparison: the end of skeuomorphism*. Phone Arena. [online]. [cit. 2014-05-27]. Available at: http://www.phonearena.com/news/iOS-6-vs-iOS-7-design-differences-comparison-the-end-of-skeuomorphism_id47545
- [17] Wikipedia. iOS. [online]. [cit. 2014-05-27]. Available at: <http://en.wikipedia.org/wiki/IOS>
- [18] Apple Inc. 2013. iOS Developer Library. [online]. [cit. 2014-05-27]. Available at: <https://developer.apple.com/library/ios/navigation/>
- [19] Bennett Madeline. *iOS 7 vs iOS 6 head-to-head review*. V3. [online]. [cit. 2014-05-27]. Available at: <http://www.v3.co.uk/v3-uk/review/2295962/ios-7-vs-ios-6-head-to-head-review>
- [20] Apple Inc. 2013. iOS Support Matrix. [online]. [cit. 2013-07-25]. Available at: <http://iossupportmatrix.com>
- [21] iOS Library for Barcode Reader. Stackoverflow. [online]. [cit. 2013-07-20]. Available at: <http://stackoverflow.com/questions/14513553/ios-library-for-barcode-reader>
- [22] GitHub Inc. 2014. zxing. [online]. [cit. 2014-06-02]. Available at: <https://github.com/zxing/zxing/>
- [23] BROWN, Jeff. *ZBar Bar Code Reader*. [online]. [cit. 2014-04-21]. Available at: <http://zbar.sourceforge.net/iphone/index.html>
- [24] ShopSavvy, Inc. Developers. [online]. [cit. 2014-06-02]. Available at: <http://shopsavvy.com/developers>
- [25] Red Laser Developers. RedLaser Barcode Scanning SDK. [online]. [cit. 2014-06-02]. Available at: <http://redlaser.com/developers/>
- [26] Barcode.com. Barcode Reader Toolkit for Mobile Platforms. [online]. [cit. 2014-06-02]. Available at: <http://www.bardecode.com/en1/app/mobilesdk/>
- [27] Apple Inc. 2013. Local and Push Notification Programming Guide. [online]. [cit. 2013-07-25]. Available at: <https://developer.apple.com/library/ios/#documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/Introduction.html>
- [28] HAFIZJI, Ali. *Apple Push Notification Services in iOS 6 Tutorial: Part 1/2*. In Raywenderlich. [online]. [cit. 2013-07-25]. Available at:

<http://www.raywenderlich.com/32960/apple-push-notification-services-in-ios-6-tutorial-part-1>

- [29] PRABHU, Gautam. *6 iOS8 Features Inspired by Jailbreak Apps and Tweaks*. In iPhone Hack. [online]. [cit. 2014-07-25]. Available at: <http://www.iphonehacks.com/2014/06/ios-8-features-inspired-jailbreak-apps-tweaks.html>
- [30] Apple Inc. 2014. Designing for iOS. [online]. [cit. 2014-06-06]. Available at: https://developer.apple.com/library/ios/documentation/userexperience/conceptual/mobilehig/index.html#//apple_ref/doc/uid/TP40006556-CH66-SW1
- [31] Apple Inc. 2014. Table View Programming Guide for iOS. [online]. [cit. 2014-04-22]. Available at: http://developer.apple.com/library/ios/documentation/userexperience/conceptual/tableview_iphone/AboutTableViewsiPhone/AboutTableViewsiPhone.html
- [32] Xamarin. Introduction to Collection Views. [online]. [cit. 2014-04-22]. Available at: http://docs.xamarin.com/guides/ios/user_interface/introduction_to_collection_views/
- [33] Android Patterns. Drill Down Navigation. [online]. [cit. 2014-04-22]. Available at: http://www.androidpatterns.com/uap_pattern/drill-down-navigation
- [34] Apple Inc. 2014. Page Controls. [online]. [cit. 2014-04-22]. Available at: <https://developer.apple.com/library/ios/documentation/userexperience/conceptual/UIKitUICatalog/UIPageControl.html>
- [35] BRIDGWATER, Adrian. *Splashtop Launches Win8 Metro Testbed for Android Tablets*. In Dr. Dobb's. [online]. [cit. 2014-04-22]. Available at: <http://www.drdoobs.com/mobile/splashtop-launches-win8-metro-testbed-fo/240001802>
- [36] Smart Home iPad App. [online]. [cit. 2014-04-22]. Available at: <http://joshcarpenter.ca/#Smart-Home-iPad-App>
- [37] Wikipedia. Software Development Process. [online]. [cit. 2014-06-12]. Available at: http://en.wikipedia.org/wiki/Software_development_process
- [38] Apple Inc. 2014. Xcode. [online]. [cit. 2014-06-21]. Available at: <https://developer.apple.com/xcode/>
- [39] Master the New Net. 5 Rules to of QR Codes. [online]. [cit. 2014-06-21]. Available at: <http://masterthenewnet.com/the-5-rules-of-qr-codes/>

- [40] GitHub, Inc. 2014. *narpas/iOS-color-wheel*. [online]. [cit. 2014-04-21]. Available at: <https://github.com/narpas/iOS-color-wheel>
- [41] Apigee. UGAPI. [online]. [cit. 2014-04-21]. Available at: <https://apigee.com/about/>
- [42] Wikipedia. Model View Controller . [online]. [cit. 2014-04-21]. Available at: <http://cs.wikipedia.org/wiki/Model-view-controller>
- [43] AppleInc. 2014. iOS Developer Program. [online]. [cit. 2014-04-21]. Available at: <https://developer.apple.com/programs/ios/>
- [44] Wikipedia. List of iOS Devices. [online]. [cit. 2014-04-21]. Available at: http://en.wikipedia.org/wiki/List_of_iOS_devices
- [45] NORMAN, Donald A. *Design pro každý den*. 1. vyd. v českém jazyce. Praha: Dokořán, 2010, 271 s. ISBN 978-80-7363-314-1.
- [46] Wikipedia. Uživatelské testování. [online]. [cit. 2014-01-19]. Available at: http://cs.wikipedia.org/wiki/U%C5%BEivatelsk%C3%A9_testov%C3%A1n%C3%AD
- [47] ISTQB Exam Certification. What is Usability testing in software and it's benefits to end user?. [online]. [cit. 2014-06-20]. Available at: <http://istqbexamcertification.com/what-is-usability-testing-in-software-and-its-benefits-to-end-user/>
- [48] Dobrý web. Uživatelské testování použitelnosti. [online]. [cit. 2014-01-19]. Available at: <http://www.dobryweb.cz/uzivatelske-testovani>
- [49] Usability.gov. Recruiting Usability Test Participants. [online]. [cit. 2014-06-21]. Available at: <http://www.usability.gov/how-to-and-tools/methods/recruiting-usability-test-participants.html>
- [50] Usability.gov. Scenarios. [online]. [cit. 2014-06-21]. Available at: <http://www.usability.gov/how-to-and-tools/methods/scenarios.html>
- [51] Nielsen Norman Group. How Many Test Users in a Usability Study?. [online]. [cit. 2014-06-21]. Available at: <http://www.nngroup.com/articles/how-many-test-users/>
- [52] Web Design & Development I. Lesson 1: Conducting a Usability Test. [online]. [cit. 2014-06-21]. Available at: <http://www.washington.edu/accesscomputing/webd2/student/unit6/module2/lesson1.html>
- [53] SEGAN, Sascha. *Apple iOS 7 Beta 3: What's the Big Deal?*. In PCMAG.COM. [online]. [cit. 2013-08-05]. Available at: <http://www.pcmag.com/article2/0,2817,2421491,00.asp>

- [54] BOSTINNO. iOS 7 Release Date & Reviews. Everything We Know About iOS 7. [online]. [cit. 2013-08-05]. Available at: <http://bostinno.streetwise.co/2013/07/15/ios-7-release-date-reviews-everything-we-know-about-ios-7-photos/>

- [55] Apple Inc. 2013. iOS 7. [online]. [cit. 2013-08-05]. Available at: <http://www.apple.com/ios/ios7/features/>

- [56] Apple Inc. 2014. iOS 8 Preview. [online]. [cit. 2014-06-21]. Available at: <https://www.apple.com/ios/ios8/developer/>

11 Appendix: Smart Home Point for iOS on CD

Included CD contains source code of the Smart Home Point project with the following structure:

- Third party libraries
- Resources needed for starting the applications
- Project file (.xcodproj)
- „smarthomepoint“ folder with the source code and application resources

To open the project it is necessary to use the Xcode development environment.