OViS WIZARD. A USER FRIENDLY MESH NETWORK DEPLOYMENT APP FOR iOS.

Bachelorarbeit
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von
Marcel Stolz
2013

Leiter der Arbeit:
Professor Dr. Torsten Braun
Institut für Informatik und angewandte Mathematik
List of Figures

2.1 The construction site scenario. .............................................. 6
2.2 The OViS scenario. ......................................................... 7
2.3 Hardware components of OViS. .......................................... 7
2.4 Examples of status bars. ................................................... 16
2.5 Example of a navigation bar. ............................................. 17
2.6 Example of a tab bar on the iPhone. ................................... 17
2.7 Examples of table views in grouped style. ............................ 18
2.8 Example of an alert view. ................................................ 18
2.9 Example of a progress view. ............................................. 18
2.10 Example of a split view on the iPad. .................................. 19

3.1 OViS Wizard user-side steps: Setting up a network. ................. 24
3.2 OViS Wizard user-side steps: A finalised network. ................... 24
3.3 Flowchart of the OViS Wizard workflow. .............................. 25
3.4 The home screen. .......................................................... 28
3.5 A regular instruction screen and a dimmed instruction screen. .... 29
3.6 The gateway browser. ..................................................... 30
3.7 A dimmed screen providing status information and an alert view. .. 31
3.8 Some alert views the user is confronted with during the deployment process. 32
3.9 The modified listing of nodes, used to browse for regular nodes. ... 32
3.10 The signal strength screen in various states. ......................... 33
3.11 Screens providing monitoring information. .......................... 34
3.12 The iPad screen in portrait orientation. .............................. 36
3.13 The iPad screen in landscape orientation. ........................... 36
3.14 The iPad showing instructions concerning the signal strength between two nodes. 37
3.15 A dimmed screen on the iPad displaying a warning. ............... 37
3.16 The node bar showing information on the Internet connection status on the iPad screen. .......................... 38

4.1 OViS Wizard model classes .................................................. 42
4.2 ViewController classes of OViS Wizard ............................... 45

5.1 The evaluation questionnaire .............................................. 50
5.2 The components for the user tests ..................................... 51
5.3 Impressions from the user tests.
List of Tables

2.1 Node status flags included in OViS Hello packets: 8
Listings

4.1 Output of the txtinfo plug-in used by OViS. ........................................... 46
4.2 Output of the iwlinks.sh script used by OViS. .......................................... 47
Acknowledgements

First of all, I would like to thank Prof. Dr. Torsten Braun, head of the Communication and Distributed Systems group, University of Bern, for giving me the opportunity to realise the OViS Wizard project and write this thesis. His interest in my work and support over all stages were a great source of motivation and helped me focus on the relevant issues. Furthermore, I would like to express my gratitude to Dr. Thomas Staub, who supervised my work. He invested many resources in providing most useful advice and gave me a glimpse of the ‘light at the end of the tunnel’ every once in a while. He also obtained various hardware and software to support a neat and frictionless development. Last but not least, he spent a great amount of both working and free time to proofread and comment on my work. Moreover, I want to give my thanks to Stefan Ott, who developed the back-end software of OViS. Not only did he support me by introducing me to the concepts and functionalities of OViS, he also provided software scripts that were of great use for the understanding of the functioning of OViS. 

Additional thanks go to my friends Arian Uruqi, Gianluca Greco, Christian Rippstein, Kristina Koščíková, Vojtěch Vasko, Patrick Wyler, Nina and Joscha Frey, Vera and Lukas Böhler, Tobias and David Weibel, Stéphane Vuilleumier, Bruno Rüedi and Markus Portmann, who helped testing and provided relief and distraction in periods of hard and long nightly work. Finally, I want to thank my parents and my sister—Michael, Zuzana and Bernadette Stolz—for their endurance, patience and daily support. The endorsement I felt over all stages of my work—the easy and more demanding ones—was the one only a family can give.

Marcel Stolz, December 2013
Abstract

The On-Site Video Communication System (OViS) [1, 2, 3] is a rapidly deployable temporary wireless mesh network solution. It brings Internet connectivity to places where common wired and wireless technologies are not an option. However, the deployment of a wireless mesh network requires additional knowledge in networking. This poses a major drawback for the success of OViS and other wireless mesh network technologies on the end-user market.

Therefore, we propose the OViS Wizard as a user friendly front-end for deploying temporary wireless mesh networks. This configuration software has been implemented on iOS as a universal application, running on both iPhones and iPads. It aims to provide a very simple approach for novice users and is based on theories and models enhancing user friendliness. It also follows Apple’s guidelines on the visual requirements and design of software—including certain unavoidable technical restrictions.

We have evaluated the OViS Wizard with user tests in order to measure its user friendliness and identify user interface errors or elements that could be improved. We have defined a testing environment and found criteria for measuring user friendliness by means of an adequate usability model. The evaluation results led to the implementation of additional features and enhancements.

The OViS Wizard confronts the potential consumers with a product suiting their needs within the constraints given. We provide an application they can easily use with minor technical knowledge on a widespread platform.
Chapter 1

Introduction

The On-Site Video Communication System (OViS) is a rapidly deployable temporary wireless mesh network solution developed at the University of Bern [1, 2, 3]. It aims to bring up a wireless Internet connection in places where a wired or cellular network is not available. This is done by overcoming the distance to the next wired Internet access point with wireless mesh network technology. OViS Wizard provides a user front-end, which assists rather than constrains an inexperienced user in setting up such a network. In this Chapter, the broader motivation for a user friendly front-end for OViS is introduced and the problems associated with the design of such a front-end are discussed. Subsequently, the contributions of our user front-end are described, followed by an outline of this work.

1.1 Motivation

The availability of an Internet connection is usually taken as granted in the modern world: People use cell phones and tablet computers to read newspapers or check the weather while they are travelling. Even in distant valleys cell phone communication is in most cases possible. Nevertheless, for some scenarios in urban areas no Internet access is available. The OViS solution was motivated by such a scenario—the construction site scenario [1]: An electrician needs to set up an Internet connection in a place with poor wireless reception. A solution to this problem is a temporary wireless mesh network. Technically, OViS [2] provides everything that is needed to set up such a network. However, in practice only a small amount of people is able to set up the network by accessing OViS directly. The procedure needed for the configuration of the OViS network requires advanced knowledge and represents a technical barrier to the success of OViS on the market. Moreover, the electrician in the construction site scenario needs a hand-held device capable of audio- or video-communication as soon as the network is functional. Therefore, creating an easy-to-use interface enabling the configuration of OViS from a hand-held device represents an ideal solution. The OViS Wizard is an application providing such a solution to common users on the widely spread mobile platform of Apple’s iOS, running on iPhones and iPads.
1.2 Problem Formulation

Nowadays, a user friendly (easy-to-use) front-end is a crucial part of good software. The main focus of users lies on the part with which they directly interact: the user interface. Therefore, user interface design is important for a comprehensive solution that is ready for the market. In order to design software in a user friendly way, it is inevitable to get an understanding of what user friendliness is. Furthermore, the potential users and their expectations need to be determined.

OViS is a system that brings network connectivity to places where common wireless, cellular or wired technologies are not available or feasible. It consists of the following components [2]:

- The wireless mesh network nodes that interact with each other and build a wireless “bridge” between a wired Internet access device and the place where users want to use the Internet,
- the software that is installed on these nodes and used to configure and monitor the wireless mesh network,
- and the end-user device that interacts with the OViS nodes and the software installed on them.

OViS Wizard is the software running on such an end-user device. It helps users in setting up the network. Thereafter, users can access the Internet. The knowledge on OViS, user friendliness, the typical users and their expectations have to be combined to ensure an optimal design of OViS Wizard.

OViS should also be ‘ready for the market’. This ensures that it is of use to companies and the end-users, respectively. Being ready for the market is closely connected to user friendliness: A user friendly application enhances the success on the market. Yet, being ‘ready for the market’ goes further: An application may be user friendly, but difficult to obtain or install. It should be made as easy as possible to deploy OViS Wizard to end-users and their companies. Therefore, OViS Wizard has to run on a device broadly available.

Hence, the challenge of developing OViS Wizard consists of combining a new mesh network technology (OViS) with a user friendly interface on a device that is widely used.

1.3 Contributions

Several new features and findings have been brought to OViS by this thesis: First of all, an application—OViS Wizard—has been developed. It enables users with little technical knowledge to access the tasks needed. Second, the application tries to solve the constraints given by generic mobile communication devices in a user friendly way, reducing the steps necessary for configuring a network connection to an absolute minimum. Third, a logical path for the network deployment process has been created and implemented as a workflow. Additionally, the application of OViS has been enhanced by not only applying several models and recommendations on user friendly design, but also by measuring the usability of OViS Wizard. The results have been added to the implementation, as far as possible and reasonable. Finally, several ideas on future
work have been made. They can be used to make the usage of OVıS even easier with a future implementation.

1.4 Outline

Knowledge on OVıS itself as well as user friendliness is inevitable for the development of a user friendly front-end for OVıS. Furthermore, some basics about the devices OVıS Wizard runs on have to be introduced. This is mainly done in Chapter 2. An overview of the technological background of OVıS is outlined and the previously mentioned construction site scenario is introduced. Moreover, the main characteristics of iOS, the operating system of iPhones and iPads, are discussed. This includes some iOS specific recommendations on user friendly design. Additionally, general concepts and best practices of user friendly design are introduced. With this background information, the development of OVıS Wizard is discussed in Chapter 3. First, the goals and typical users of OVıS Wizard are defined. Subsequently, the conceptual design of the application workflow, which describes the main logic of OVıS Wizard, is introduced. Hereafter, the user interface design is discussed. Having described the most important part of our application—the user friendly interface—some interesting implementation approaches of OVıS Wizard are shown in Chapter 4. The use of design recommendations is not the only measure for user friendliness. The impression of the end-user is also of concern. Therefore, OVıS Wizard is evaluated by means of user tests in Chapter 5. Additionally, weak points of the user interface and their solutions are discussed. Finally, considerations on whether OVıS Wizard met the goals defined are made and future work is discussed in Chapter 6.
Chapter 2

Related Work

The OViS Wizard relies on existing work covered in this chapter. First of all, OViS [2, 3, 1], the back-end system of OViS Wizard, is introduced. Second, the motivation for selecting the iOS device family for development is discussed. Third, OViS Wizard aims to provide a good user experience in an ‘easy-to-use’ environment. Therefore, a closer look at some concepts enhancing user friendliness is taken. Last but not least, during GUI design certain guidelines and standards need to be respected, which are defined by the underlying device platform. For this reason, some characteristics and recommendations of Apple’s iOS platform are introduced.

2.1 OViS

OViS Wizard is a front-end for the On-site Video-conferencing System (OViS). The central aspects of OViS are covered in the Master’s thesis of Stefan Ott [2] and the PhD thesis of Dr. Thomas Staub [3]. This section summarises the most important aspects concerning its appliance in OViS Wizard.

2.1.1 The Construction Site Scenario

The main goal of OViS is to provide a rapidly deployable network connectivity solution in places lacking regular Wi-Fi, cellular or wired network infrastructure.

A place where these preconditions occur are construction sites. They provide the central motivation for OViS by the so-called construction site scenario [2, 3, 1] illustrated in Figure 2.1:

The first major phase of a building construction has been completed. In this stage, an electrician has to wire up and configure the switching unit according to plans designed by an electrical engineer. The actual state of the switching unit, however, may differ from the predefined specifications. Hence, the electrician needs to ask the engineer for help. As the switching unit is usually installed in the basement of a building, which forms an area of poor reception for mobile communication, the electrician is obliged to leave the basement in order to phone the engineer. This means that the electrician has to memorise the current state of the switching unit. The engineer, though, may ask a very specific question and the electrician is forced to go back into the basement in order to answer it. In the worst case, this happens several times or the problem cannot be described by means of words. Rather, images or video communication would be helpful.
This construction site problem does not only result in wasted time, it also causes high costs, as two people are bound to a single problem. Often the only solution is a visit of the engineer on the construction site, causing an even higher overhead. Naturally, building companies have been trying to find a possible answer to this problem.

An obvious alternative solution is the use of fixed wires. These would be installed temporarily until the construction is over. However, the installation of fixed wires raises two additional problems: First, the installation of wires on a construction site with many people walking around comprises a severe risk of accidents: Construction workers could stumble over the wires and get seriously injured, which might lead to a liability process. Second, a proper installation of the cables would demand additional work, causing higher costs and a longer construction time. Therefore, the use of fixed wires is not a favourable alternative.

An ad-hoc wireless mesh network provides a feasible solution. However, existing implementations require expert knowledge for deployment. OViS offers a solution without this limitation. Moreover, OViS makes use of on-site infrastructures: On the one hand, a local gateway is used for the Internet connection, which is usually available on construction sites. On the other hand, communication over the mesh network is possible with common wireless end-user devices such as notebooks or smart phones. An illustration of the OViS solution to the construction site problem is shown in Figure 2.2.

2.1.2 The Hardware Components and Set-up of OViS

OViS consists of several components that have to work together in order to set up a mesh network [2,3]. In this subsection, the required hardware components and their set-up are presented. As already mentioned in the last subsection, OViS makes use of existing infrastructure: An Internet router with an integrated DHCP server (the gateway) is a precondition for the deployment of OViS. The first mesh node is connected by wire to the gateway and, therefore, called gateway node. It bridges communication from the wired network to the mesh network and is the starting point for the mesh network set-up. One by one, further nodes are added to the mesh network.
in order to overcome the designated distance. Finally, the mesh network is completed and a wireless terminal can communicate with the network over the last node added. Such a network is called finalised. An illustration of the OViS scenario is shown in Figure 2.2. The hardware components needed for OViS are depicted in Figure 2.3.

The gateway nodes and regular nodes in Figure 2.3 are in fact the same nodes. The only difference is the cable plugged into the Ethernet port of the gateway node. The OViS software installed on the nodes is able to recognise whether a network cable is connected and whether a DHCP signal is available on the wired network. If this is the case, the node is defined as a gateway node.

Every node has two wireless interfaces, which enable communication between the nodes. The first wireless interface is always communicating with the previous node (called predecessor) and the second wireless interface communicates with the following node (called successor). As both node types consist of the same hardware and the gateway node has no predecessor, the first wireless interface of the gateway node is not in use.
Table 2.1: Node status flags included in OViS Hello packets [2].

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>There is no network connection available</td>
<td>0</td>
</tr>
<tr>
<td>The network cable is plugged in</td>
<td>1</td>
</tr>
</tbody>
</table>

2.1.3 The OViS Software and HTTPS Web Service

The basic scheme for setting up an OViS network has already been discussed in the preceding subsection. The details about the functionality of the OViS software and the configuration process [2, 3] follow in the upcoming paragraph.

During the set-up of the wireless mesh network, OViS offers a configuration interface. The use of this interface is inevitable, as vital information has to be provided and the deployment of each node needs to be confirmed.

The node’s configuration interface is accessible by an HTTPS web service. Before a node can be configured it has to be discovered: As soon as a node is turned on, OViS establishes a wireless network with the SSID ‘OViS’. This network is an IEEE 802.11b/g wireless ad-hoc network. Using this network, the node broadcasts messages on UDP port 4379, so-called Hello packets. These Hello packets are broadcast repeatedly and contain all information required in order to interact with the node: The node’s IPv6 address can be determined from the UDP broadcast itself. The hostname and node type (regular or gateway) are derived from a string contained in the Hello packet. The string’s content is similar to meshnodeXY:A (e.g. meshnode19:1) where XY is the unique identity number of the node and A is an integer value indicating the connection status of a node. This connection status is used to determine the type (regular or gateway) of a node. The possible integer values and their meanings are shown in Table 2.1.

As soon as a node has been discovered, it is configured. This process is called the deployment of a node. In order to prevent interferences between the wireless signals of the different nodes, OViS provides the possibility to set their frequencies to different channels of the IEEE 802.11a 5 GHz industrial, scientific and medical (ISM) radio band. The terminal device, however, uses an IEEE 802.11b/g connection (and thus the 2.4 GHz ISM radio band). Therefore, the direct connection to a node’s second wireless interface is lost after it is deployed.

The deployment process for an OViS wireless mesh network consists of the following steps: First, the gateway node is discovered by a Hello packet and is thereafter configured by the HTTPS web service. As the gateway node has no predecessor, only the wireless interface connecting to the successor needs to be configured. The connection between the gateway node and the terminal device is, thereafter, lost. Subsequently, its successor is configured. After the discovery of the successor, it first needs to be connected to the gateway node: Its first wireless interface is set to the previously used 5 GHz ISM radio band frequency. As soon as the two nodes are connected, the HTTPS web service provides information on the signal quality, enabling the detection of an optimal distance between the nodes. The regular node is then deployed, i.e. its second wireless interface is set to a frequency in the 5 GHz ISM radio band range differing from the frequency used on the node’s first wireless interface. This deployment process is repeated for as many regular nodes as needed. For the last node, the frequency of the second wireless
interface must not be changed in order to provide compatibility with the terminal device. Yet, it needs to be configured to function as a router for the IPv4 traffic of the terminal device: An appropriate router IP address and subnet mask is set over the HTTPS web service. The network is then finalised and provides an IPv4 Internet connection. Moreover, mesh network monitoring information can be retrieved by the HTTPS web service.

A list of the configuration options for the HTTPS web service is shown below, adapted from [2]. The HTTPS web service is accessed via the URL https://[fc00:6680::2:XY]:443/url_request where url_request represents any item from the list and XY stands for the specific number of the mesh node, e.g. 19 for meshnode19.

- `/netcfg.sh?pass=himbeeri&iface=wlan0&mask=23&ip1=10.241.91.1&mask1=29`
  Suffix for deploying the last node and finalizing the network. The IP address as well as the subnet mask of the second wireless interface are set.

- `/txtinfo.sh`
  Suffix for the retrieval of monitoring information of a deployed network. This request may only be called after a network has been finalised.

2.2 Selection of the End-user Device

In Section 2.1 the functionality of OViS has been described. OViS requires a terminal device to configure the mesh nodes. Technically, any IEEE 802.11b/g capable device can be used. However, there is a great variety of hand-held devices with different hardware characteristics and running different operating systems. A specific platform fitting the goals of OViS and OViS Wizard had to be chosen. Several reasons for the selection of Apple’s iOS platform are discussed in the following paragraphs.

First, the predominance of iOS devices on the smart phone market in Switzerland has been a major criterion [4,5]: Apple’s iPhone disposed a market share of 52% on the Swiss smart phone market during the year 2012. In contrast, the market share of the only considerable alternative, the Android OS, was equal to 42%. Furthermore, in the same year, an amount of 583’000 iPads
was sold in Switzerland, opposed to 228'000 tablet computers from other vendors. Development for the iOS platform, therefore, meets the concern of developing for a widely spread device.

Second, iOS is understood to be a very user friendly platform. As stated in articles on life-hacker.com [6, 7] comparing iOS and Android, the iPhone is the winner concerning user friendliness. Third, Apple controls the hardware design of devices running iOS itself [7]. Therefore, compatibility is ensured. An application developed for a specific iOS version can generally be said to work on all devices that have it installed. On the Android platform not all devices support a function, as some may lack the corresponding hardware [7].

Finally, Apple’s XCode supports the development of a single application for both iPad and iPhone [8]. So-called Universal Apps allow developers to create applications that adapt their user interface and functions according to the device family and still share a lot of code and user interface design elements.

In summary, the selection of the iOS platform for development results from the main goals of OViS Wizard: User friendliness and the use on a broad market. Furthermore, the advantage of developing software for two types of devices—tablet computers and smart phones—gives the iOS platform an extra interest.

2.3 User Friendliness

Nowadays, a very important aspect when designing a piece of software for regular (i.e. non-expert) users is user friendliness, as mentioned in the foreword of Human-Computer Interaction [9]:

‘Designing interactive computer systems to be effective, efficient, easy, and enjoyable to use is important, so that people and society may realise the benefits of computation-based devices.’

This statement complies with the construction site scenario: The benefit of OViS and OViS Wizard would have to be questioned if users first had to get a broader knowledge of mesh networking. This would cost a lot of time and possibly cause opposition to launching the software in a building company. Therefore, it is vital to take a closer look at concepts concerning user friendly design. The concepts discussed in the following subsections heavily rely on Human-Computer Interaction [9] and Usability 101: Introduction to Usability [10].

2.3.1 Principles, Standards and Guidelines

In order to support a developer during the development process, many recommendations and rules enhancing user friendliness may be found. They can be grouped in the categories of principles, standards and guidelines [9]. Their distinction relies on the two dimensions of authority and generality: A rule with high authority should absolutely be followed during the design process, whereas high generality implies that a rule applies to a wide range of applications. Thus, a rule with low authority is rather a recommendation and needs not necessarily to comply with all implementations. A rule with low generality is a very specific rule, applying to a very specific appliance.
The following definition of principles, standards and guidelines is found in *Human-Computer Interaction* [9]:

‘*Principles* are abstract design rules, with high generality and low authority. *Standards* are specific design rules, high in authority and limited in application, whereas *guidelines* tend to be lower in authority and more general in application. [...] We can make another rough distinction between principles, standards and guidelines. Principles are derived from knowledge of the psychological, computational and sociological aspects of the problem domains and are largely independent of the technology; they depend to a much bigger extent on a deeper understanding of the human element in the interaction. They can therefore be applied widely but are not so useful for specific design advice. Guidelines are less abstract and often more technology oriented, but as they are also general, it is important for a designer to know what theoretical evidence there is to support them. A designer will have less of a need to know the underlying theory for applying a standard. However, since standards carry a much higher level of authority it is more important that the theory underlying them be correct or sound.’

Therefore, to enhance user friendliness for OViS Wizard, appropriate principles, standards and guidelines need to be found. As principles have a universal character, they are described in *Human-Computer Interaction* [9] and summarised in the following subsection. For guidelines and standards, advice is taken from Apple’s recommendations and design rules, covered in Subsection 2.4.2 together with some iOS specific design principles.

### 2.3.2 User Interface Design

*Human-Computer Interaction* [9] defines three basic principles for a good design: *learnability*, *flexibility* and *robustness*. These are divided into subcategories that define them more specifically. In the following paragraphs, a selection of the most relevant points is summarised according to their definition in [9].

#### Learnability

Learnability describes the ability of a system to be useful and efficient for novice users. If a new user easily learns to use a system and is able to work efficiently with it after a short time, learnability is considered to be high. The most important subcategories of learnability are *predictability*, *synthesisability* and *consistency*. The subcategories of *generalizability* and *familiarity* are not summarised here, as they are already followed when developing with the iOS SDK or following Apple’s iOS Human Interface Guidelines [11] (see Section 2.4).

The meaning of predictability is quite obvious: A user should be able to predict what his or her action provokes. This may consist of pressing a button and knowing what change in the system it educes.

Synthesisability, on the other hand, characterises the representation of a change of state in the system: For example, a message may be displayed if an action was completed successfully.

Finally, consistency indicates that we should design similar situations in a similar way.
Flexibility

In general, flexibility is understood to be the ability of a system to communicate with the end-user. The most important subcategories here are dialogue initiative, multi-threading and substitutivity.

Dialogue initiative covers the issue of whether the user or the system has the initiative in the communication. With a so-called system pre-emptive dialogue, the system mostly initiates the dialogue and the user only responds to requests of the system, whereas user pre-emptive dialogue consists of the converse. The second dialogue method is considered to be more user friendly. However, the freedom of decision that is granted to the user also brings forth a higher risk for inconsistency and erroneous states within an application. Therefore, it is important to carefully balance the advantages and disadvantages when choosing the dialogue initiative.

Multi-threading describes the ability of a system to accomplish several tasks at the same time. This can, for example, be the distribution of different tasks into different windows.

Substitutivity implies that equivalent values may be substituted for each other. The most important aspect of substitutivity is so-called representation multiplicity, which consists of the redundant representation of information. This may be achieved by showing an attribute as both its absolute value in numbers and by a graphical illustration.

Robustness

Robustness describes the set of features that support the successful accomplishment of a task. We divide this principle into the subcategories observability, recoverability and responsiveness. Observability implies that users should be able to determine the current state of a system by means of an adequate representation in the user interface. This allows users to reflect on what they have already accomplished and what has yet to be done.

Recoverability, on the other hand, is the possibility of the system to answer the needs of users who want to undo a step. This is useful when they have done something they did not intend or when they realise that the result of an action is other than expected.

Responsiveness is the measure of communication between a system and its users. A responsive system should react very quickly to user inputs and indicate state changes as fast as possible. Even if direct communication with the system is not possible at a specific point of time, a responsive system should at least indicate that it is working, for example, by a moving wheel.

2.3.3 Usability Measurement

To achieve user friendliness, not only the development of an application has to be lead by design rules, the application should also be tested by the end-users. The definition of usability helps measuring the user friendliness of an application:

‘Usability is a quality attribute that assesses how easy user interfaces are to use.’

Usability is divided into several subcategories:

‘Learnability: How easy is it for users to accomplish basic tasks the first time they encounter the design?’
Efficiency: Once users have learned the design, how quickly can they perform tasks? Memorability: When users return to the design after a period of not using it, how easily can they re-establish proficiency? Errors: How many errors do users make, how severe are these errors, and how easily can they recover from the errors? Satisfaction: How pleasant is it to use the design?

These subcategories show similarities with the concepts presented in Subsection 2.3.2. The aspect of learnability is even mentioned in both the Human-Computer Interaction design principles and Nielsen’s usability subcategories. Efficiency is closely connected to learnability but still brings in a new aspect: A system that is hard to learn is inefficient. However, a system may even be inefficient, when its usage is obvious, if the number of steps is too high or they are laid out in an unsuitable way. Memorability is already partially covered by learnability. The aspect of errors and error handling is closely related to robustness. A quite new approach is the subcategory of satisfaction: In the last subsection, the categories were laid out to help users in accomplishing a task and in using a system. However, if users like using an application this may be a major advantage for the application on the market. Also, the definition of usability contains a recommendation on the quantity of user tests [10]: ‘To identify a design’s most important usability problems, testing 5 users is typically enough.’

2.4 Rules and Recommendations for iOS Development

In the preceding section, user friendliness has been analysed from a very general point of view. Nevertheless, this Bachelor thesis made development for a certain platform, namely Apple’s iOS. The design always depends on the possibilities of the underlying OS. Some operating systems even impose the use of a limited number of design elements and standards. The architects of these systems believe that similar application design simplifies the use of a new application and improves the user experience. This is also the case for Apple’s iOS. It encourages the usage of certain principles, guidelines and strategies in its iOS Human Interface Guidelines [11].

In the following part, the most important aspects of Apple’s recommendations are summarised with a main focus on their application in the OViS Wizard. In most cases, it is important to follow Apple’s recommendations, as they help to comply with the principles discussed in Section 2.3.

2.4.1 iOS Basic Design

One of the probably most striking characteristics of any iOS device is the minimalist design of its hardware [11]: Both iPhones and iPads dispose of only one main button called Home button. With this design, Apple follows its principle ‘The display is paramount, regardless of its size’ mentioned in the first Chapter of Apple’s Guidelines [11]: With only one button available, the display can occupy a maximum amount of space. The positive effect is that an application is not
fixed to a predefined amount of buttons but can display various numbers of buttons on the screen. Moreover, these buttons can be hidden, when the screen displays more important information. The iOS platform targets two different device types: One is the small screen device family of the iPhone and iPod touch. The other is the tablet device family of the iPad. Both platforms may be addressed by one single application, called Universal App. These applications need to provide appropriate user interfaces in order to respect the different screen sizes. A further characteristic of the iOS platform is its adaptability to the device’s orientation. When holding a device in landscape or portrait orientation, an application may choose to adapt its display of content correspondingly. This feature is mostly used with the iPad. Even though it is available on the iPhone, many developers choose not to adjust their apps to landscape mode on the iPhone.

2.4.2 The Human Interface Principles

Apple defines a number of key principles in the second Chapter of its iOS Human Interface Guidelines. In the upcoming paragraphs, these key principles are discussed with a focus on their application in OViS Wizard, summarising the most important parts.

Aesthetic Integrity

In order to comply with the principle of aesthetic integrity, the aim of an application has to be considered. In the case of OViS, the aim is to fulfil a well-known, predefined task—to deploy a mesh network. It is important not to overfill an application with unnecessary accessories: If its design is kept plain and concentrates on its main aim, the user is able to accomplish the task more easily.

Consistency

The principle of consistency recommends to display similar contents in a similar way. The standards given by iOS have to be followed. Furthermore, the application should be ‘consistent in itself’. This means that an icon should be used but for one purpose and both the flow of information as well as the instructions have to follow a logical path.

Direct Manipulation

The principle of direct manipulation is of smaller concern for OViS Wizard: It is true that device rotation should be supported on the iPad, however, this is only one issue of direct manipulation. The main aim of direct manipulation is the support for using gestures. This makes sense primarily in applications focusing on some sort of navigation or zoom functionalities. OViS does not provide such functionalities and, therefore, does not support gestures. This would overfill the scope of features and distract the user from the main task.
Feedback

A major concern of OViS Wizard is the principle of feedback [11]: Users obviously experience a lot of status changes when setting up a network environment. Hence, they are interested in the results of their actions. The network has to be scanned for changes, which have to be displayed quickly and in an easy way. This significantly influences user experience and helps the users to understand what they are doing or how to solve occurring problems.

Metaphors

It does not make sense to define metaphors [11] for our wireless mesh network: OViS Wizard configures real-world objects such as network nodes. Metaphors aim to transport a familiar real-world concept into a software system so that users can interact with the software in this well-known way [9]. Therefore, a real-world concept would be used to illustrate the actions on the node. The node, however, is a real-world object itself. Therefore, replacing it by some other real-world object would lead to confusion.

User Control

Finally, user control is a major issue for the design of OViS Wizard. This principle may be described with the words ‘the user is in control’ [11]. Clearly, users cannot be given control over all possible features. This would be counter productive: A wider range of control also augments the required knowledge of the underlying technology. However, users should get the feeling of being in control [11]. This is achieved by providing controls that are predictable, i.e. it should be obvious what actions a button invokes. Moreover, users must be given the opportunity to cancel destructive actions before they begin [11]. Also, as far as possible, the application should leave decisions to the users, which will increase their feeling of being in control.

2.4.3 Relevant Strategies and Guidelines

In this subsection, strategies and guidelines relevant for OViS Wizard are introduced and summarised. The exact formulations and definitions may be looked up in Apple’s iOS Human Interface Guidelines [11] in Chapters 3, 5 and 6.

First of all, Apple defines two important application design strategies [11]. One of these strategies is the application definition statement at the beginning of an application design process. It consists of the following four points [11]:

1. List All the Features You Think Users Might Like
2. Determine Who Your Users Are
3. Filter the Feature List Through the Audience Definition
4. Don’t Stop There

The last step may seem slightly confusing, as it is not very precise in formulation. It implies that the developer should be aware of the foregoing three steps and have the user and the main goal of the application in mind during the whole development process including later iterations.
instance, a new or nicer design should always be driven by the user’s needs. A second strategy proposed by Apple is to design the application for the device [11]. This does not only refer to the usage of the human interface principles discussed in Subsection 2.4.2 or other iOS specific approaches. This strategy encourages the developer to respect the different characteristics of the iPad and the iPhone when developing a Universal App. This particularly involves a different layout as well as adapted sizes of icons and artwork. Yet, also the main functionalities need to be preserved in both device versions of an application.

Apart from these strategies, Apple imposes several guidelines in the Chapter User Experience Guidelines [11]. The most important and relevant guidelines for the implementation of OViS Wizard are Focus on the Primary Task, Give People a Logical Path to Follow, Make Usage Easy and Obvious, Use User-Centric Terminology, Be Succinct, Use UI Elements Consistently, Handle Orientation Changes, Start Instantly, For iPad: Enhance Interactivity (Don’t Just Add Features), For iPad: Reduce Full-Screen Transitions, and For iPad: Restrain Your Information Hierarchy.

The exact meaning of these guidelines is not summarised here, as they have self describing titles and may be looked up in detail [11].

2.4.4 Predefined GUI Elements

This subsection covers some standard GUI elements used by OViS Wizard and provided by Apple’s iOS SDK. The descriptions are derived from [11], where all of the GUI elements are presented in detail.

The Status Bar

The status bar (see Figure 2.4) is always shown in the topmost part of the screen. It displays system specific information to the user, e.g. the current time and battery status. Furthermore, it shows whether the iOS device is connected to a wireless network. It also provides the possibility to display a network activity indicator. Apple recommends to use this element in order to inform the user that the wireless network is being used. However, OViS Wizard does not use it and depicts network activity in a more intuitive way (see Chapter 3).

The Navigation Bar

The navigation bar (see Figure 2.5) is displayed in the top part of a window when navigating through a hierarchy of information. It shows the title of the current view container and allows
navigation within the view hierarchy with an appropriate button. Apple recommends not to overfill a navigation bar with elements or controls.

Figure 2.5: Example of a navigation bar [11].

The Tab Bar

Tab bars (see Figure 2.6) allow switching between different screens. They are displayed at the bottom of a screen and should preferably be used on the topmost level of an application in order to provide a selection of different subtasks. Tab bars should be avoided on an iPad and a split view, presented in Paragraph 2.4.4 should be used instead.

Figure 2.6: Example of a tab bar on the iPhone [11].

Table Views

A table view (see Figure 2.7) provides the possibility to present lists of items, each in a cell. Often, the cells are implemented as tappable, i.e. their tapping invokes a specific action. A table view can be presented in several different styles. The grouped style, used in OViS Wizard, groups the rows in several sections. The sections may feature a header or footer. The content cells can also be presented in different styles. The styles differ in varying positions of the text labels and images.

Alert Views

Some view elements are only displayed temporarily and cover the whole screen. They require a user action to return to the main view. These so-called alert views, as shown in Figure 2.8 indicate a message or ask a question. The users have to choose an option or confirm by pressing a button before they continue. It is important to use short and precise titles, messages and button titles when using alert views.

The Progress View

A further element is the progress view in Figure 2.9. It usually shows the progress of a task and should be used if the remaining amount of time needed can be estimated to support substitutivity. Its appearance is adaptable. For example, the colour and background colour of the bar may be changed.
Figure 2.7: Examples of table views in *grouped* style [11].

Figure 2.8: Example of an alert view [11].

The Split View

The split view shown in Figure 2.10 is a view element only available on the iPad. It is very useful when transferring applications from the small screen of the iPhone to the iPad. The split view is not an element in the manner of the preceding paragraphs. Rather, it is a container for other view elements, splitting the screen into two parts. The bigger part (called *detail view*) is usually situated on the right hand side, while the part on the left hand side (called *root view*) often presents a table view that controls the content of the detail view. This approach is a very convenient option to substitute a tab bar or a regular table view.

Figure 2.9: Example of a progress view [11].
Figure 2.10: Example of a split view on the iPad [11].
Chapter 3

OViS Wizard User Interface Design

In Chapter 2, the basics for the development of OViS Wizard were discussed: First, the fundamentals of the OViS software were introduced. Subsequently, some general considerations on user friendliness were presented. Finally, the basics of our development platform—the iOS—were shown by summarising the most important parts of Apple’s iOS development guidelines. The aim of the current Chapter is to show how these concepts have been used during the development process. The goals of our application and its features are defined and a prediction on possible users is made in Section 3.1. Thereafter, a user friendly workflow and GUI design is introduced in Sections 3.2 and 3.3.

3.1 Goals of OViS Wizard

The main goal of OViS Wizard is user friendliness. In order to ensure a user friendly design, further characteristics need to be defined. Apple’s application definition statement [11] (see Chapter 2.4) builds an ideal foundation for this purpose. This includes a list of features. As OViS Wizard relies on the capabilities of OViS, the features are the same: OViS Wizard provides the possibility to deploy and monitor a network. In order to enable a user friendly design, Apple recommends to define who our users are [11] and to make an application definition statement [11]. This is done in the upcoming subsections.

3.1.1 The OViS Wizard User

To get an idea of the typical users, the construction site scenario from Subsection 2.1.1 is recalled: The person in charge of the switching unit, probably an electrician or craftsman, is the person that has to use OViS Wizard. Therefore, the typical user should not be totally new to using smart phone or tablet devices and a certain knowledge on how to use technical tools may be presumed. This also applies to OViS users in a general sense: A person with the aim to set up a temporary Internet connection has some experience in using computers or hand-held devices. Presumably, the user knows some technical terms such as ‘network cable’ or ‘Internet router’. However, this cannot be taken for granted, as it is not necessary to know these terms when using a smart phone / tablet device or wiring up a switching unit. It is essential to consider this during the development process and to use user-centric terminology [11] or substitutivity [9].
A user may not know anything about Internet routing, IP addresses and so forth. Therefore, it is important to omit these terms. If this is not possible, instructions need to be supported by illustrations. In summary, the nature of the OViS Wizard user is characterised as follows:

**The OViS Wizard User**

- has experience in using hand-held devices,
- knows how to use the Internet via a browser,
- does not necessarily know technical terms,
- and does not have a broad knowledge concerning network technologies.

### 3.1.2 Definition of Application Goals

The knowledge on application features and potential users is now used to define application goals. As stated in the construction site scenario in Subsection 2.1.1, OViS Wizard is primarily used to set up an Internet connection. Monitoring, on the other hand, is only used by advanced users. It provides deeper information on the network structure and requires profound knowledge on networking. Therefore, the monitoring feature is included in OViS Wizard. However, the main focus lies on mesh network deployment. The goals of OViS Wizard are defined as follows:

**OViS Wizard**

- is based on OViS and provides access to the features of OViS with a focus on network deployment,
- and is user friendly.

### 3.2 The OViS Wizard Workflow Design

The OViS Wizard users, features and goals have been defined in Section 3.1. In this Section, the steps necessary for deploying OViS are presented from the user’s point of view. Thereafter, the workflow of OViS Wizard is introduced, relying on what has been defined in Section 3.1 and Chapter 2.

#### 3.2.1 User-Side Steps for Deployment

The technical deployment steps of OViS have already been described in Section 2.1. In order to design OViS Wizard in a user friendly way, these steps need to be discussed from the user’s point of view. Transferring the technical requisites to user focussed steps results in the following set-up:
1. Initially, the users have OViS Wizard installed on their iOS device, as shown in Subfigure 3.1a. Arriving on the construction site, the users should know where to find an Internet gateway. Thereafter, they start OViS Wizard.

2. In order to set up the gateway node, the users are instructed to take a network cable and connect it to the node and the Internet gateway (see Subfigure 3.1b).

3. OViS Wizard instructs the users to connect their iOS device to the wireless mesh network (see Subfigure 3.1c). OViS Wizard configures the gateway node and deploys it. The users should get feedback on the deployment results.

4. The users have to set up an arbitrary number of regular nodes in order to overcome the required distance. The set-up of each node begins with taking a further node and turning it on. OViS Wizard then connects to the node, as shown in Subfigure 3.1d.

5. OViS Wizard connects the node to its predecessor. The signal strength between the nodes is then measured and the users are instructed to place the node within an appropriate distance.

6. The two previous steps have to be repeated for as many nodes as needed to overcome the desired distance. The final node needs to be configured as a router. Therefore, the users are asked for each node, whether they want to deploy a further node. If not, the network is finalised and feedback is provided. The users should then be able to see information concerning Internet connectivity. The final set-up of a network is illustrated in Figure 3.2.

These user-side deployment steps are still a rather abstract definition level of an application. However, they do already allow a first discussion of some principles introduced in Sections 2.3 and 2.4.

One important aspect is dialogue initiative: The users begin the dialogue with the application by starting it. Thereafter, they have to be guided very closely, as the deployment of the mesh network consists of very specific steps. With straightforward instructions and narrow possibilities, user-side errors can be avoided. Yet, the users should still have the feeling of being in control. Eventually, feedback is of great importance: OViS consists of many components. Hence, state changes within the network have to be forwarded to the users immediately. Also, when communication between components is necessary—which may take some time—an illustration has to be provided, which does not give the users an impression of slow or crashed software.

Finally, in wireless networking a lot of unforeseen states may occur. In order to comply with robustness, these unforeseen states have to be handled in a smooth way and easy solutions have to be provided.

3.2.2 Workflow Design

In defining a workflow, OViS Wizard gives the user a logical path to follow. In Figure 3.3 a flowchart illustrates the workflow of OViS Wizard. The flowchart is divided into three main blocks that represent wireless mesh network monitoring, deployment and error handling. The
Figure 3.1: OViS Wizard user-side steps: Setting up a network.

Figure 3.2: OViS Wizard user-side steps: A finalised network.
Figure 3.3: Flowchart of the OViS Wizard workflow.
most important and detailed of them is the Network Monitoring Box, as this is the main task of OViS Wizard. In the following paragraphs, all references to boxes or forks correspond to the numbered rectangular and diamond boxes in Figure 3.3.

**User Tasks and System Steps**

The deployment workflow distinguishes user tasks, e.g. Box 0.1, and system steps, e.g. Box 0.3. The latter are defined as steps that do not directly involve any action on the user side. Yet, their representation strengthens the feeling of being in control and, thus, supports user control. Boxes 2.1, 2.2 and 2.3 represent three system tasks that could have been combined into a single one. However, representing three distinct tasks to the user supports observability, recoverability and synthesisability. Similar situations come up for Boxes 0.1 and 0.2: It is easier for the user to perceive the tasks as separate instructions one by one, which increases learnability.

**Forks**

Another element in the flowchart are forks: They depict decisions. Fork 3.3, for example, represents a decision implied directly by the architecture of OViS. Forks 1.1 and 2.1, on the other hand, represent situations, which are not part of the OViS specification, yet may come up: It may occur that the user has followed all instructions, yet a node is not spotted by the application. This needs to be considered and solutions have to be found in order to comply with robustness. Fork 0.3 may not be very obvious: Usually, the application is started to set up a new temporary wireless mesh network. However, the cases where users have already set up a mesh network or where they want to set up a new mesh network over an existing one have to be considered. As all settings of an existing network are reset when beginning with network deployment from scratch, this represents a destructive action [9]. Introducing a fork forces the users to consider the consequences of their action. This complies with recoverability and user control.

**The Network Monitoring Box**

From Box 3.3, which is the last step of the deployment process, an arrow leads to the Network Monitoring Box. This is the logical usage of the monitoring function, as it is not possible to monitor a network while it is not deployed. However, there is also a small double arrow connecting the Network Monitoring Box and the Network Deployment Box. The reason for this is simple: It may occur that we start the application in a state where it does not ‘remember’ a previously deployed network. It should be possible to monitor such an existing network even if it was not deployed by OViS Wizard beforehand. Furthermore, the users seize the existence of the monitoring feature even if they have not yet the possibility to use it: They are informed of the possibility to switch to the monitoring feature as soon as the deployment is finished. This gives them the feeling of multithreading and of being in control. The presence of the monitoring feature at all times also complies with consistency: Monitoring is not a feature that ‘pops up’ suddenly in the user interface when it is available. Rather, the feature is shown in a way indicating it is available at a later point of time.
Monitoring consists of a static representation of data. Therefore, the Network Monitoring Box is not further divided into tasks or steps.

The Error Handling Box

It is a fact that errors can occur during application execution. It is inevitable to collect as many erroneous situations as possible and provide easy solutions. This is required by the principles of robustness and feedback. As a problem may come up at virtually any step during deployment, it makes sense to connect the Network Deployment Box directly to the Error Handling Box. This is indicated by the small double arrow in Figure 3.3. Errors may come up during monitoring as well. For aesthetic reasons, and as the application’s main focus lies on deployment, the double arrow connecting monitoring and error handling is left away in Figure 3.3. The Errors and their solutions are of technical kind and require knowledge of the implementation. They are discussed in detail in Chapter 4.

The Starting and End Point

Both the starting point (Box 0.0) and the end point (Box 4.0) of the workflow are connected only to steps in the Network Deployment Box. This illustrates the focus on the main task of OViS Wizard: The application starts instantly with deployment. The workflow ends, as soon as the wireless mesh network is set up. It is still possible to monitor a network or begin with deploying a new wireless mesh network. However, the primary task of network deployment is completed and, therefore, the workflow comes to an end.

3.3 A User Friendly Interface for OViS Wizard

In the previous section, the workflow of OViS Wizard has been defined. The design of the user interface is now laid out according to the steps in the workflow. For this purpose, GUI elements presented in Subsection 2.4.4 are used. As discussed earlier, OViS Wizard is a universal application. Therefore, both an iPhone and an iPad GUI is introduced. Many principles, standards and guidelines from Subsection 2.4.3 apply for each screen presented in the upcoming subsections. We focus only on the most relevant ones.

3.3.1 The iPhone GUI

In a universal application the iPhone GUI is usually constructed first. The reason is the iOS device history: The iPad has been introduced after the iPhone was already established on the market. Many developers transformed their applications into universal applications and constructed the iPad GUI accordingly. The same scheme is followed in this subsection.
As stated in Subsection 3.2.2, OViS Wizard starts instantly with the deployment workflow. This influenced the design of the first screen of OViS Wizard shown in Figure 3.4. The users are confronted right away with network deployment. They can select between setting up a new network or adding a node to an existing network. On the bottom of the screen the possibility to switch to the monitoring feature, as discussed earlier, is available in the tab bar.

Two very similar scenarios were introduced earlier: deployment of a gateway node and deployment of a regular node. These options are not possible at the same time, even though they are both shown on the home screen. This enforces an appropriate mental model of OViS Wizard, similar to the decision of showing users the possibility of monitoring from the beginning. If a user selects to add a node to an existing network, even though a gateway node has not yet been deployed, a text is presented. It indicates that this option is not yet possible and dims the screen that would usually be shown (for an example see Figure 3.5). After the gateway node has been deployed, the application returns to the home screen, as also after the deployment of a regular node.

In summary, the design of the first screen is a simple and straightforward way to introduce the user to the task of setting up a mesh network. Furthermore, a feeling of being in control is triggered by showing options that are possible at a later point of time.
The Instruction Screens

The next step in the workflow is setting up a new network, thus setting up the gateway node. This invokes some user tasks, which are depicted in instruction screens as shown in Figure 3.5. In order to comply with consistency, all instructions are displayed in this kind of screen. The comprehensibility of technical terms is supported by images (as required by substitutivity). The verbal instruction in the cell has to respect a succinct language and user-centric terminology. This also includes the selection of a language: Most of the users of OViS Wizard will know the German language. In case there are people who do not understand the German language, instructions are provided in English. The language is selected on application launch according to operating system standards. The selection does not only influence the instruction text. It includes all labels and texts used in OViS Wizard.

The Gateway Browser

The next step in the OViS Wizard workflow is a fork, as shown in Figure 3.3 Fork f. The user must check, whether an appropriate node is available. A screen providing this possibility is shown in Figure 3.6. When setting up a new network and deploying a gateway node, only this type of node seems to be of concern. However, all other nodes are listed in a second section to achieve observability. This gives the user the opportunity to detect that a node intended as a gateway node is not detected as such. Mechanisms are provided that tell the user what to do in this case. Therefore, the gateway browser consists of one section per node type. The selection of a listed gateway node invokes an appropriate system action. Tapping a regular node, on the
other hand, is not possible.
In case no node is found, the corresponding section displays a GUI element called activity wheel. It shows that the application is looking for devices on the network and thus working. The usage of the activity wheel enhances responsiveness.

Enhancing Feedback with Dimmed Screens

The next step in the workflow is a system step. It is important to display activity information and a message on successful completion in order to reinforce feedback. A simple solution is to dim the last screen and to add text describing the current status of the system step. Additionally, an activity indicator is displayed. As soon as the system step has been finished, the user is informed by an alert view. This ensures that the user realises the accomplishment of the system step. Moreover, it gives the application the opportunity to inform the user if anything went wrong. An example of a dimmed screen containing an activity indicator and an alert view is illustrated in Figure 3.7.

Back to the Home Screen

The deployment of the gateway node has now been finished and the workflow specifies to continue with the deployment of a regular node (see Box 1.1 in Figure 3.3). As stated earlier, the ‘home screen’ shown in Figure 3.4 is used at the beginning of the deployment process. If the user chooses to set up a new network this would destroy the settings of the network that is currently being deployed. Therefore, the user is confronted with an alert view in this case, as...
Deploying a Regular Node

Continuing with the deployment of a regular node, OViS Wizard has to display some instructions. For this purpose the instruction screen, as shown in Figure 3.5, is reused. Hereafter, the network is once again scanned for appropriate nodes. This time, however, there is only one category of nodes listed on the screen: The regular nodes, which may be added to the current network. This decision is based on the following considerations: First of all, no reasonable situation speaks for the event of a regular node showing up as a gateway node. Moreover, in the event of two networks being set up simultaneously, listing an alien gateway node would rather confuse the user. An example of the adapted screen is pictured in Figure 3.9.

After the user selects the appropriate node, OViS Wizard configures it corresponding and establishes a connection to its predecessor. During this event, similar issues play a role as during the deployment and configuration of the gateway node. Again, a dimmed screen including an activity indicator and the current status is displayed.

Signal Strength Screen

After the connection between the nodes has been established, the user has to place the current node within an optimal distance to its predecessor. Hence, signal strength information and an
Figure 3.8: Some alert views the user is confronted with during the deployment process.

Figure 3.9: The modified listing of nodes, used to browse for regular nodes.
Figure 3.10: The signal strength screen in various states.

instruction on what to do is displayed. Figure 3.10 indicates the design chosen. The screen shows certain similarities to the instruction screen, which enhances familiarity and consistency. Furthermore, it depicts the signal strength by means of a coloured progress bar and its physical value. This enhances representation multiplicity and helps to make usage easy and obvious. Derived from the physical value, an appropriate instruction is shown.

An important issue is the freedom of choice the users are given: Even though a recommendation for the optimal placement of the node is provided, the users are free to deploy it at a location of their choice. This ensures user control and may be a convenient feature for more experienced users.

The last node of a network cannot be configured in the same way as a regular node (see Section 2.1). Therefore, the user has to be asked for each node, whether it is the last one. This is done with an alert view, as shown on the right hand side of Figure 3.8. If the user chooses to finalise the network, the common dimmed screen with activity information is displayed. As soon as the network has been finalised, the user is informed by means of an alert view.

The Monitoring Screen

In the OViS Wizard workflow, there is an arrow from Box 3.3 leading to the Network Monitoring Box (see Figure 3.3). Therefore, the OViS Wizard GUI switches directly to monitoring, as soon as the wireless mesh network has been finalised. As stated earlier, the monitoring feature is of little use to the inexperienced user. It is reasonable to include a peace of information that helps
common users. Hence, the Internet connection status is displayed, as illustrated in Figure 3.11. This information is placed prominently on the screen, as shown on the left and middle image. Representation multiplicity is ensured by changing the colour. Further monitoring information, which is of smaller interest to most users, can be accessed by tapping one of the cells in the first section. The details are presented succinctly in a further table view as shown in the right image of Figure 3.11.

3.3.2 The iPad GUI

The iPad GUI is derived from the iPhone GUI presented in Subsection 3.3.1. Special account has to be given to the adaptation of the iPhone GUI to the bigger screen of the iPad. As many iPhone screens are reused and adapted, the derived screens are not discussed in such detail as in Subsection 3.3.1.

The Split Screen and Node Bar

The construction of the iPad screen is best done by first considering the different characteristics of the iPad. The main difference should be the presentation of information on the iPad (as implied by the recommendation Ensure that Universal Applications Run Well on Both iPhone and iPad [11]). We furthermore need to remember the guidelines For iPad: Restrain Your Information Hierarchy, For iPad: Enhance Interactivity (Don’t Just Add Features) and For iPad: Reduce Full-Screen Transitions [11]. This means that useless or far-fetched features have to be avoided. On the other hand, the user
experience can be amplified by displaying more useful information. The bigger screen of the iPad allows to display different and individual content. The screen is, therefore, divided into several parts.

A split view is a convenient method to divide the iPad screen in such a manner. However, the split view has a disadvantage: It only divides the screen into two sections when the iPad is held in landscape orientation. In portrait orientation, the smaller left part of the screen is moved into a popover. This can be a rather confusing behaviour for new users. Furthermore, presenting the grouped table view in the left part of the screen after entering a task gives the users a better overview of what they are currently doing and enhances observability. As the same layout is also found in the iOS settings app, this does not violate consistency or aesthetic integrity.

The iPad screen is divided even more: As the nodes are deployed one after another, a representation of the already deployed nodes seems helpful. A bar at the bottom of the screen displays all nodes and the Internet status. If a node or the Internet item is tapped, a popover provides detailed information (see Figure 3.12). This node bar enhances the observability of OViS Wizard.

Thus, the iPad screen is split into three parts. The left part of the screen is called root view and is divided into two sections. One contains the deployment options, the other shows the possibilities for monitoring. On the right hand side task specific information is displayed, such as instructions or signal strength information. This part is called detail view. On the bottom of the screen a bar is displayed, containing all nodes as well as a simplified representation of the Internet connection. This part is called node bar.

**Importing Screens from the iPhone**

In the previous paragraph, the detail view was introduced. It displays instructions and information according to the selection in the root view. Therefore, derived iPhone screens are displayed in the detail view. The modification of the iPhone screen is discussed subsequently.

The iPhone instruction screen encounters the biggest change: On the iPad, two instructions are shown simultaneously (see Figure 3.12 and Figure 3.13). The signal strength view, on the other hand, looks nearly the same as on the iPhone (see Figure 3.14).

A further difference is the design of the warning, which is presented when users choose to set up a new network even though they are already deploying a network or when they just finished doing so: On the iPad, the alert view is no option, as the split view is displayed constantly. Instead, the first deployment instruction in the detail view is dimmed and a button enabling the user to proceed is shown (see Figure 3.15).

As soon as network deployment has been finished, the OViS Wizard switches to the monitoring feature. As the node bar illustrates the network set-up, the Internet status information is displayed there, as shown in Figure 3.16. The Internet symbol changes its colour depending on the connection state.
Figure 3.12: The iPad screen in portrait orientation.

Figure 3.13: The iPad screen in landscape orientation.
Figure 3.14: The iPad showing instructions concerning the signal strength between two nodes.

Figure 3.15: A dimmed screen on the iPad displaying a warning.
Figure 3.16: The node bar showing information on the Internet connection status on the iPad screen.
Chapter 4

OViS Wizard Implementation

Chapter 3 introduced the application workflow and user interface design. The aim of the current chapter is to describe the implementation behind the user interface. Section 4.1 introduces the development platform, its tools and constraints. Thereafter, some external frameworks used in the implementation are introduced in Section 4.2. Finally, the code design of OViS Wizard is discussed in Section 4.3.

4.1 The Development Platform

The operating system versions used for developing, testing and running OViS Wizard were iOS 5 and iOS 6. Software for iOS is usually developed in a programming language called Objective C [12]. It is an object oriented superset of the ANSI C programming language. Therefore, C functions and some C libraries (e.g. Berkeley Sockets [13, 14]) can be used in Objective C [15]. However, Objective C is not the only characteristic of the iOS development platform. There is a set of programming tools available from Apple’s iOS Developer Library [16]. They are introduced in Subsection 4.1.1.

Furthermore, certain restrictions have to be considered when developing for iOS. As they have an impact on the user friendliness of OViS Wizard, they are discussed in Subsection 4.1.2.

4.1.1 Developer Tools

An important tool for iOS programming is Apple’s integrated development environment (IDE) called XCode [8]. It supports coding in a comfortable way and includes tools for the design of the graphical user interface. These tools stimulate the use of the model view controller pattern [17, 18], as stated in Concepts in Objective-C Programming [19].

The Interface Builder included in XCode [8] provides a convenient possibility to lay out GUI elements within the IDE. It is possible to define the set-up of the devices’ screen elements. Furthermore, resizing characteristics can be defined to which these elements adapt when the screen size changes. This feature enhances reusing iPhone view elements on the iPad. Some iPhone screens may even be reused without any changes. Moreover, it is possible to insert generic view elements as place holders for adapted or programmatically defined view elements.
The Interface Builder allows the developer to map GUI elements to instance variables and instance methods. The communication between the instances and these GUI elements is handled during runtime by the programming language and the IDE.

4.1.2 Restrictions in iOS

There are some restrictions in iOS programming that have an effect on the user friendliness of OViS Wizard. Most importantly, the iOS SDK does not allow the programmatic manipulation of network settings. This includes connecting to a wireless network with a specific SSID as well as providing static IP configuration information for a network. This issue cannot be solved by providing a configuration profile when deploying OViS Wizard [20]. OViS, however, requires a connection to an ad-hoc network with a specific SSID and a static IP configuration. This has to be done manually by the user according to instructions in OViS Wizard. The manual configuration of network settings forms a possible source of errors. Its negative consequences need to be kept as small as possible.

4.2 External Frameworks

During the development process, external frameworks were included into the code of OViS Wizard. They are briefly introduced in the upcoming subsections.

4.2.1 TestFlight

In order to ensure convenient testing on different iOS devices the TestFlight platform [21] has been used. Usually, an iOS application that is being developed has to be compiled directly from XCode to a specific iOS device. This requires a Universal Serial Bus connection and has to be repeated for all subsequent application versions. TestFlight offers an easier deployment method over the Internet: New application versions can be downloaded from a webpage and their performance is monitored. In order to keep track of eventual crashes, the TestFlight SDK has been integrated into the OViS Wizard for the time of development.

4.2.2 CocoaAsyncSocket

In order to ensure correct handling of datagram sockets, the CocoaAsyncSocket project [22] has been used. Even though Objective C offers access to the Berkeley Sockets C library (see [13, 14]), it does not support transferring information from the Berkeley Sockets functions to Objective C in a satisfactory way. The CocoaAsyncSocket project solves this issue and provides an Objective C interface to the Berkeley Sockets library.

4.2.3 Extended Split View Functionality

In Subsection 3.3.2 the iPad GUI was introduced. It splits the screen into three parts. As the standard iPad split view divides the screen only into two parts it could not be used for the implementation of OViS Wizard. The MGSplitViewController project [23] provides a programmatic
implementation of an iPad split view. It has been extended in OViS Wizard in order to design the iPad screen as intended.

4.3 Code Design

The model view controller (MVC) pattern [17] [18] was already mentioned in Section 4.1. Its aim is to strictly separate objects of the view, model and controller. In XCode the view—i.e., the user interface—is usually designed in the Interface Builder as an XML file. This XML file is parsed and connected with the appropriate methods and objects of the (view) controllers during runtime. The model, on the other hand, is responsible for the underlying application logic and does not interact with the views directly. The model of OViS Wizard mainly configures the network and communicates with the OViS system. In the following Subsections, the design of the model and view controller classes is discussed.

4.3.1 The Model of OViS Wizard

The model of OViS Wizard consists of the three classes shown in Figure 4.1. The Network Class is the central information handler: It organises the communication between the OViS Wizard and ‘the outside world’. For example, it listens for nodes and observes the network status. The Node Class, on the other hand, is used to represent nodes on the network and enables the interaction with the nodes’ configuration web service. The Reachability Class, finally, supports the Network Class in checking for the availability of specific hosts. These three classes are discussed in the upcoming paragraphs.

The Network Class

The Network Class is the most important class of the model. As only one network is handled simultaneously, this class is set up as a singleton [24], meaning that only one instance of it can be created. This makes information access and retrieval by other objects easier. The network instance is created when calling the Network Class’ static method sharedInstance for the first time. Thereafter, the method always returns a reference to this instance.

The initialisation of a Network Class object consists of the following tasks: First, some instance variables are set describing the network status (e.g. hasGateway, isFinalized). Second, the checkConnectivity method is invoked. It checks whether an Internet connection or a finalised OViS network is available. This functionality is needed in order to monitor a network that has already been set up previously and uses instances of the Reachability Class. Third, the frequencies array is filled with possible frequencies for the OViS node’s wireless interfaces. This array is stored and handled centrally in the Network Class instance in order to avoid frequency interferences: When a frequency is retrieved, it is reused only after all other frequencies of the array have been used.

An important feature of the Network Class is to listen for new nodes on the network: Section 2.1 describes that a node begins broadcasting Hello packets after it is turned on. By invoking the findNodes method, the network class begins to listen for Hello packets. It makes use of the CocoaAsyncSocket framework (see Section 4.2) for this purpose. When a Hello packet is received,
Figure 4.1: OViS Wizard model classes.
the data contained in the packet is passed to the Network Class, where it is handled appropriately: The node information is parsed and a Node Class object created. New messages from nodes already spotted are used to keep the Node Class objects alive, while Node Class objects for which no further Hello packets are received are dropped after a specific time interval. If no Hello packets at all are received, the Network Class instance is informed after a time out of 60 seconds. An appropriate notification is then sent in order to allow error handling by the view controller classes.

An interesting aspect of the node discovery is the handling of the Hello packets. Apart from the node name and IP address, which are extracted, the gateway integer value (see Table 2.1) is used to classify a node as a gateway node or regular node. If a user wants to deploy a gateway node and only regular nodes are being found, the view controller can react to this problem by providing a recommendation.

All nodes discovered are stored as node objects within an array of the Network Class instance called helloNodes. In order to ensure consistency and responsiveness, the Network Class checks periodically for all available nodes and drops those which do not continue sending Hello packets.

The Node Class

The Node Class is an abstraction of a node and the information related to it during the deployment process. Its methods access a node’s HTTPS configuration web service. A Node Class instance cannot communicate with nodes after they have been deployed, as the OViS configuration web service does not supply this function.

As described in the preceding paragraph, each node that announces its presence via a Hello packet is only considered as alive, if it continues to send Hello packets. In the Node Class, the lastOnline property is used to check when the last Hello packet has been received.

The OViS nodes are configured using the HTTPS configuration web service. In order to create and send the HTTPS requests, the Node Class uses the NSURLConnection Class provided in the iOS SDK (see URL Loading System Programming Guide [25]).

As known from Subsection 2.1.3, the URLs needed for accessing the HTTPS web service are similar to each other. This is utilised in the Node Class by defining a base URL. Depending on the desired action, an appropriate suffix is appended.

During the deployment of a regular node, the place method is used to retrieve the signal strength to its predecessor. It repeats sending a request to the HTTPS web service in order to retrieve signal strength information continuously. This ensures consistency and responsiveness. As soon as a node is deployed, the finishPlacing method stops the requests.

In order to receive answers for URL requests sent to the HTTPS web service, the methods defined in the NSURLConnection delegate protocol are implemented. This enables a Node Class instance to react to a proper response or handle errors or time outs. In either case, a notification is sent by the Node Class instance to inform whichever object may be interested.

The Reachability Class

The Reachability Class supports the Network Class singleton in checking whether a specific network address is reachable. The implementation of such a class is required as it is not possible
to check for a working connection to a specific host with the iOS SDK. The Reachability Class is used for two purposes within the Network Class singleton: One instance is created in order to check whether the final node in the OViS network is reachable. A second instance checks for a working Internet connection. For the latter case, it seemed a good choice to check for the reachability of a domain name rather than the corresponding IP address. This ensures that a DNS server is reachable, which dissolves the domain name to the appropriate IP address. For the reachability of the Internet, a server is checked that should ‘always’ be online. The domain name google.ch has been selected for this purpose.

For the implementation of the Reachability Class similar approaches as in the Node Class have been made: The NSURLConnection Class has been used to invoke an URL request and the host address of the reachability instance is checked repeatedly for availability. As soon as the reachability changes, a notification is sent to all instances that might be interested.

4.3.2 The View Controller

In the following paragraphs, the view controller classes are introduced. They are discussed in less details than the model, as many of their functionalities support the representation of the user interface introduced in Chapter 3. We rather point out some interesting facts of the view controller classes. A compilation of the relevant classes is illustrated in Figure 4.2. In this figure, the classes do not contain variables, as this would take too much space.

In Figure 4.2, the classes are grouped according to their usage. As mentioned before, some view elements are shared between the iPhone and iPad GUI. They are contained in the group labelled Shared Classes.

Multiple Language Support

There are some functionalities that cannot be illustrated in the diagram. The handling of multiple languages, as mentioned in Chapter 3, is such a case. Based on the user’s selected language in iOS, the view controller instances choose appropriate string elements. This is done via language specific files that contain all user interface strings of the corresponding language (see Internationalization Programming Topics [26]). A localised string is accessed via a specific key string and retrieved during runtime. This scheme allows the application to easily support a larger amount of languages in a later iteration. Currently, only German and English user interfaces are supported, i.e. English strings are used if the default language is not German.

Parsing Model Data

The view controller classes are responsible for the representation of data passed over from the OViS system. This data is parsed according to the view controller’s needs. An example is the MonitoringOverviewController Class. It has to handle the monitoring data. The data is received by registering for a specific notification of the model. The string provided by the model is shown in Listing 4.1. There are different sections, of which some are displayed in the GUI by means of a table view. As discussed in Chapter 3, the representation is different for the iPad and the iPhone: All of the data is listed right away on the iPad, whereas the iPhone
Figure 4.2: View controller classes of OViS Wizard.
only displays the Section titles inside a first table view. The details are shown in another table view when the appropriate section title is tapped.

As the parsing of the data is the same for both devices, the iPhone class is inherited by a corresponding iPad class. The section titles of the txtinfo plug-in are provided only in English. Therefore, they are translated manually in the German version of the user interface.

```
HTTP/1.0 200 OK
Content-type: text/plain

Table: Links
Local IP    Remote IP    Hyst.  LQ  NLQ  Cost
10.241.120.18  10.241.121.23  1.000  1.000  1.000
10.241.91.1    10.241.120.23  0.843  0.000  INFINITE

Table: Neighbors
IP address    SYM  MPR  MPRS  Will.  2 Hop Neighbors
10.241.120.23  YES  YES  YES  7      0

Table: Topology
Dest. IP      Last hop IP  LQ  NLQ  Cost
10.241.120.23  10.241.120.18  1.000  1.000  1.000
10.241.120.18  10.241.120.23  1.000  1.000  1.000

Table: HNA
Destination    Gateway
10.241.91.0/29  10.241.120.18
0.0.0.0/0      10.241.120.23

Table: MID
IP address    Aliases
10.241.120.23  10.241.121.23

Table: Routes
Dest. IP      Gateway IP  Metric  ETX  Interface
0.0.0.0/0     10.241.121.23  1.000  wlan0
10.241.120.23/32  10.241.121.23  1      wlan0
10.241.121.23/32  10.241.121.23  1      wlan0
```

Listing 4.1: Output of the txtinfo plug-in used by OViS.

A further class that parses data is the `SignalStrengthViewController` class. It displays the signal quality between two nodes and provides corresponding instructions. The model calls a script called `iwlinks.sh` on the OViS nodes. A possible output of the script is demonstrated in Listing 4.2. The values relevant for placing a node are the signal and the `tx bitrate`. As OViS Wizard needs to wait for the connection between the two node to be established, the view controller first dims the signal strength screen. It is undimmed as soon as the `tx bitrate` reaches a value between 6.0 and 20.0 MBit/s and the signal strength differs from 0.0 dBm. These criteria were chosen as differing values were returned before the nodes had established a proper connection.

The value of the progress bar shown in the user interface is calculated as follows: The limits of
the progress bar are a maximum of $-30.0 \text{ dBm}$ and a minimum of $-100.0 \text{ dBm}$. These values were chosen as they offered an appropriate visualisation of values less and greater than the optimal signal strength \cite{2} of $-60.0 \pm 10 \text{ dBm}$. If the value lies within these limits of the progress bar, the bar is shown with corresponding length in relation to the signal. A signal strength greater than $-30.0 \text{ dBm}$ results in a full length, orange coloured bar. On the other hand, a signal strength of less than $-100.0$ results in an empty bar. An optimal signal quality is known to lie between $-50.0 \text{ dBm}$ and $-70.0 \text{ dBm}$ (see \cite{2}, Subsection 6.1). Therefore, the progress bar has a green colour for these values. If the value is less than the recommendation, the progress bar turns red, for too high values it is coloured orange.

<table>
<thead>
<tr>
<th>Station 00:0b:6b:df:50:6b (on wlan0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>inactive time: 0 ms</td>
</tr>
<tr>
<td>rx bytes: 16373</td>
</tr>
<tr>
<td>tx bytes: 0</td>
</tr>
<tr>
<td>tx packets: 0</td>
</tr>
<tr>
<td>tx retries: 0</td>
</tr>
<tr>
<td>tx failed: 0</td>
</tr>
<tr>
<td>signal: $-31 \text{ dBm}$</td>
</tr>
<tr>
<td>tx bitrate: 6.0 MBit/s</td>
</tr>
</tbody>
</table>

Listing 4.2: Output of the iwlinks.sh script used by OViS.

**Time Outs**

In several situations, time outs need to be handled properly, e.g. in the node or gateway browser or when deploying a node. This is done by registering for notifications. The time outs of the node or gateway browsers are communicated directly via an alert view and instruct the user to retry the preceding steps. In the case of a gateway browser, it is possible that a node is found, which has no network cable connected. In this case, the user is advised to check whether the network cable has been plugged in correctly and to ensure that the Internet access point is turned on.

When deploying a node or finalising a network, OViS Wizard waits for a response to the corresponding HTTPS web service request. After a time out, an error message is displayed by an alert view. The time out of an HTTPS request is defined in the model, while the view controller only has to register for the appropriate error notification.

**The iPad's Node Bar**

The node bar displays the network topology. There are two possibilities how the network topology is retrieved.

First, during the deployment process, the NodeBarController Class registers for notifications distributed when a node has been successfully deployed. The notification contains a pointer to the node instance concerned. The node instance is then appended to the network topology in the NodeBarController Class.
Second, the NodeBarController Class checks whether a finalised wireless mesh network is available when OViS Wizard is started. If an OViS network is present, it builds the node bar according to the monitoring data received. Furthermore, the NodeBarController Class illustrates the status of the Internet connection in the node bar. This is done by retrieving the `internetReachability` property from the Network Class instance.
Chapter 5

OViS Wizard Evaluation

The evaluation of OViS Wizard intends to find out how user friendly OViS Wizard is. It releases aspects that have not been considered during the development process and includes measuring and enhancing user friendliness. In the upcoming sections, first the approach for the user tests is outlined in Section 5.1. Thereafter, Section 5.2 summarises the most important results. Finally, in Section 5.3 the consequences for the application implementation are covered.

5.1 User Test Set-up

In Chapter 2, usability has been introduced as a tool for measuring user friendliness. As stated, it is recommended to do qualitative tests with a small amount of users [10]. This recommendation was followed for OViS Wizard. The focus lay on the users’ feedback and their observation. Subsection 3.1.1 defined who our users are. It is also important to know who our testers are: Their technical abilities and their working environment have an influence on how comfortable they feel in using iOS and the OViS Wizard. Furthermore, it was important to give the testers only the information that is necessary for testing. This ensures that the application test is not influenced by an oral instruction. Finally, a written feedback with the relevant aspects encountered by the testers is an important part in order to make an evaluation possible.

A questionnaire has been designed, which combines these prerequisites. The questionnaire was handed to each user, who tried to use OViS Wizard with no further instructions than those given on the questionnaire. The questionnaire had to provide both uniformity, enabling comparison between the user tests, as well as the freedom to allow comments on all aspects of user friendliness. The questionnaire used for testing is shown in Figure 5.1. It combines the three requirements of personal information, adequate instructions and user comments. The components used during the user tests and some impressions are shown in Figures 5.2 and 5.3.

Obviously, the optimal test user is an electrician working on a construction site. However, in an academic environment, such people are found infrequently. Thus, users were chosen that were as similar as possible to the ‘typical user’. As far as possible, the users should know an environment similar to a construction site.

The testing focussed on the following subcategories of usability [10]: learnability, errors, and satisfaction. The questionnaire asked for statements covering these three aspects.
Step Two: Now, please start the application OViS Wizard on your iPad or iPhone and follow the steps on the screen in order to set up an Internet connection. If you think you are done with the set up, please close the application by pressing the home button and try to open a webpage such as google.com in the Safari application. You may also ask the person who has given you this form to help you with the last step.

Step Three: Evaluation

Please answer the following questions:

1. What type of device did you use for your evaluation? ☐ ☐ iPad ☐ iPhone

2. Did you encounter any problems while using OViS Wizard?

3. Was there anything that did not seem logical while using the app?

4. How did you like or dislike the "look and feel" of the app? Do you think there is a difference compared to other apps for iPhone or iPad?

5. Are there any other comments you would like to make? Any suggestions for improvements?
Figure 5.2: The components for the user tests.

Figure 5.3: Impressions from the user tests. On the left hand side: a screen shot of a Skype session on an iPod touch and its video signal transferred via an OViS network to the iPad. On the right hand side: a tester setting up an OViS network.
5.2 User Test Results

A listing of the completed evaluation questionnaires is provided in Appendix A. A total of ten people tested OViS Wizard: Four users tested the iPad application and another four users tested the iPhone application. Two users tested OViS Wizard with both device types. Even though a number of ten users is too low for statistical significance, it meets the recommended number of five users per device for qualitative testing [10]. Only the most interesting and important results are pointed out in the following paragraphs.

One of the major issues was, as expected, the manual set-up of the initial network connection. Nevertheless, nearly all testers managed to connect to the OViS network. A total of two users needed help. One of them entered a wrong IP address in the iOS settings application, leading to a faulty behaviour of OViS Wizard. Several testers suggested the application itself should configure the network settings. This is, however, not possible due to restrictions in Apple’s API. Another tester complained that it was not possible to copy the IP address and other routing settings from OViS Wizard to the operating system’s clipboard. This would reduce the risk of copy errors.

A further recommendation was that the settings application icon shown in the instruction view should be tappable, i.e. tapping the icon should invoke the settings application. Unfortunately, this is not possible due to restrictions in Apple’s API.

Furthermore, recommendations concerning the illustration of a working Internet connection in the monitoring view came up. It should, additionally to the message stating that it may take up to 3 minutes until an Internet connection is available, provide a ‘waiting’ status instead of a missing Internet connection.

Moreover, some instructions did not seem clear enough. During the signal strength measurement, certain users ran into problems: Instead of taking the node and moving further away or nearer to the preceding node, they left the node where it was and walked around without it. Also, the use of the application in general did only seem clear to some testers right away.

An additional point concerning the GUI design was the perception of the gateway browser screen. One user waited for nodes to appear in the section reserved for regular nodes, even though a node had already appeared in the section for gateway nodes.

An interesting issue brought up by an architecture student was the persistence of an OViS network over longer time. He asked whether it is possible to turn off the mesh nodes and then, at some later point of time, turn them on again and use the same network set-up repeatedly. Unfortunately, this is not possible with the current implementation of the OViS node software.

In summary, it is important to note that the main points which have come up cannot be changed in OViS Wizard; changes in the iOS API or in the OViS node implementation would be necessary. Furthermore, the problems that occurred from the application’s side were minor problems: They were experienced by few users. The application design, on the other hand, seemed good for a majority of the testers. Also, all users liked using OViS Wizard: Nobody complained about the user interface or application logic in general. A high number of users even wrote that they enjoyed using the application and experienced its set-up as very clear, logic and straightforward.
5.3 Consequences

An important part of the evaluation was not only to test the application, but to implement the findings. It was already mentioned in Section 5.2 that a great part of the problems cannot be influenced by the implementation of OViS Wizard. These problems were already considered during the application development process (see Chapter 4). However, the user tests also brought up some issues that can be solved in OViS Wizard.

First, the IP configuration information can be made selectable and copyable. This provides a very convenient way to reduce problems when transferring the data to the settings application. Second, the illustration of an available or missing Internet connection can easily be adapted: An additional idle status ensures that the Internet connection is not displayed as offline right away. Finally, some instructions can be rewritten in order to ensure a clear and unmistakable meaning or enhance a correct mental model of the OViS network set-up. For example, the instructions in the signal strength screen should tell the user to take the current node and move farther away. This prevents the user from moving away without the node. Furthermore, in the gateway browser, a footer can be added, which indicates that the section listing other nodes found may be empty. This ensures that the user is not confused if this section stays empty.

All of these changes were implemented in a second iteration of OViS Wizard and are available in the final version.
Chapter 6

Conclusions and Future Work

In the last chapters, OViS Wizard, its design process and related work was introduced with a major focus on user friendliness. Furthermore, the usability of OViS Wizard was measured, evaluated and some consequences were taken.

In Section 6.1, conclusions are taken of our prototype implementation. Of course, the design of a piece of software is a process that basically never ends. New ideas on possible features and user friendliness can always come up. Therefore, considerations on future work are introduced in Section 6.2.

6.1 Conclusions

OViS Wizard has been shown to make deployment of an OViS network possible for a common user, who does not need to have knowledge on networking technologies. This was achieved by following specific recommendations on user friendliness. Very general principles were used to guide a user through the necessary steps. Furthermore, OS specific guidelines and standards enforced a simple and straightforward user interface design. This design has been recognised as being appealing to the users.

The development of OViS Wizard has not been limited to using recommendations and concepts supporting the aim of user friendliness. The evaluation of OViS Wizard measured whether the goals of OViS Wizard were reached. OViS Wizard has been considered to be user friendly in spite of the constraints given. Even more important, the evaluation of OViS Wizard has brought up some issues that have been enhanced in a second iteration. This makes OViS Wizard even more user friendly and we conclude that OViS Wizard has reached its goals.

With OViS Wizard, the distribution of the OViS system on a broad market has become possible. Even though constraints have been showed concerning user friendliness, this is not a major problem for the economical success: The restrictions faced on iOS are similar to restrictions on other mobile platforms, such as Android. Within the scope of possible terminal devices, the best option has been chosen: OViS Wizard has been developed for a user friendly system running on a device that is broadly used on the Swiss market.

Finally, we conclude that OViS Wizard provides an optimal solution within the constraints given by OViS and the iOS platform. The user friendliness of OViS Wizard was reached by following
appropriate principles, standards and guidelines. The evaluation of OViS Wizard shows that the testers liked using OViS Wizard. Furthermore, it generated additional enhancements in user friendliness.

6.2 Future Work

A piece of software is never finished—new features can always be added and different design approaches expanded, redesigned and enhanced. In this section, some ideas are discussed on what could be implemented differently or additionally within future iterations of OViS Wizard. The design of OViS Wizard heavily depends on the underlying implementation of OViS. One major concern massively enhancing the use of OViS Wizard would be a DHCP server on the OViS nodes: This would make the manual entry of network configuration information dispensable. Furthermore, an easier approach for connecting to a wireless network with a predefined SSID could be developed. Additionally, the usage of technologies reducing the need of IP or DNS configuration could bring enhancements to OViS and OViS Wizard. This could include Bonjour [27], mDNS [28] or other zeroconf [29] technologies.

There are additional features that do not depend on the changes of other components. First, in our application we focussed mainly on network deployment. This makes sense due to the usage scenario of OViS Wizard. However, in a different scenario with more experienced users we could lay the focus on monitoring. In this case, the presentation of all sorts of available monitoring data could be extended and optimised. Second, direct access to a video conferencing application such as Skype could be provided. Meanwhile, Skype offers an API for direct access in iOS. This would enhance the out-of-the-box experience of OViS Wizard: The user could start with video conferencing right after deploying an OViS network. Furthermore, a feature could be added, which allows transfer of photographs in case that the network connection quality on the construction site is insufficient for video communication.

There are scenarios different to the construction site scenario for which OViS could be used. First, a radio station could make use of temporary wireless mesh networks such as OViS for broadcasting from difficult places. Such places may include cellars in the historic centre of a city, as cultural locations are typically situated in such places. If the radio wants to make a live broadcast from such a place, this may be difficult to accomplish using the Internet by means of regular network technologies. OViS Wizard and possibly also the OViS system would have to be tested and adapted in order to be used in this scenario.

A second scenario for OViS Wizard may be the military use of a rapidly deployable temporary mesh network infrastructure in bunkers. Typically, Internet access is available somewhere in the bunker. This is, however, usually not the location where the Internet connection is required and the massive walls damp wireless signals too much over longer distances. The military commonly deploys a temporary wired network connection in this case. However, deployment of such a connection takes some time. Under time-pressure, a solution such as OViS could be used until the wired connection is set up. This would massively enhance military efficiency in time critical operations. However, security issues would have to be met and OViS Wizard would have to be optimised for very fast deployment.
Appendices
Appendix A

Evaluation Questionnaires

This appendix lists the filled out user evaluation questionnaires.
Evaluation of OVIS Wizard for iOS

Bachelor's Project Marcel Stolz
CDS Group, IAM, University of Bern

Thank you for participating in this short survey. All data specified will be used for academic and educational use only.

Personal Information

First Name: James
Last Name: Hudson
Achieved educational graduation(s):
A-levels (Klebenthal)
Your personal technical abilities:

What technical instruments do you use in your every day life?
(e.g. mobile phone, PC, laptop, ticket machine, …)

Evaluation of the OVIS Wizard for iOS

Note: You are going to evaluate OVIS Wizard for iOS. It is important that you write down all problems you encounter. There may be problems which you put down to your personal skills. However, we think that these problems are also software issues. Therefore, we kindly ask you to write them down as well.

Your goal: In the following part, you have to set up an ad-hoc network for an internet connection using the OVIS Wizard. You will do this using the OVIS Wizard application either on an iPhone or an iPad. There will be only some very short instructions in the following, as we think that most steps should be evident from within the application. Please write down a remark on this evaluation sheet if this is not the case.

Step one: Take your iPhone or iPad and, if not yet done so, turn it on. Please make sure that you have got at least three mesh nodes ready (they do not need to be turned on yet) and that you know where the next Internet router is located.

Step Two: Now, please start the application OVIS Wizard on your iPhone or iPad and follow the steps on the screen in order to set up an Internet connection. If you think you are done with the setup, please close the application by pressing the home button and try to open a webpage such as google.com in the Safari application. You may also ask the person who has given you this form to help you with the last step.

Step three: Evaluation
Please answer the following questions:

1. What type of device did you use for your evaluation?
   - [ ] iPad
   - [ ] iPhone

2. Did you encounter any problems while using OVIS Wizard?
   Not really problems. But it takes a few minutes to get access to it and to realize what you can use it for.

3. Was there anything that did not seem logic while using the app?
   When it said: "run on a mesh node didn’t work at first not sure if it had to turn on another one. Furthermore, I didn’t realize that I had to take the mesh node with me to get a better signal.

4. How did you like or dislike the "look and feel" of the app? Do you think there is a difference compared to other apps for iPhone or iPad?
   It looks similar to other apps, which makes it (at least for Apple users) easier to work with. However, it is a bit more "complicated" compared to other apps, but the parts which are not evident for non-technical users are pretty well explained (with pictures) which is really good.

5. Are there any other comments you would like to make? Any suggestions for improvements?
   Maybe it would be helpful to get some information about the use of the app in daily life.

Figure 1: Mesh Node

Figure 2: Icon of OVIS Wizard
Evaluation of OVIS Wizard for iOS

Bachelor's Project Marcel Stolz
CDS Group, IAM, University of Bern

Thank you for participating in this short survey. All data specified will be used for academic and educational use only.

Personal Information
First Name: Lia
Last Name: Batschelet
Achieved educational graduation(s): Nativva

Your personal technical abilities: [ ] Very good [ ] good [ ] average [ ] poor [ ] none

What technical instruments do you use in your everyday life? (e.g. mobile phone + iOS, laptop, ticket machine, ...)

Laptop, iPhone, (windows)

Evaluation of the OVIS Wizard for iOS

Note: You are going to evaluate OVIS Wizard for iOS. It is important, that you write down all problems you encounter. There may be problems which you put down to your personal skills. However, we think that these problems are also software issues. Therefore, we kindly ask you to write them down as well.

Your goal: In the following part, you have to set up an ad-hoc network using the OVIS Wizard. You will do this using the OVIS Wizard application either on an iPhone or an iPad. There will be only some very short instructions in the following, as we think, that most steps should be evident from within the application. Please write down a remark on this evaluation sheet if this is not the case.

Step one: Take your iPhone or iPad and, if not yet done so, turn it on. Please make sure that you have got at least three mesh nodes ready (they do not need to be turned on yet) and that you know where the next Internet router is located.

Step Two: Now, please start the application OVIS Wizard on your iPad or iPhone and follow the steps on the screen in order to set up an Internet connection. If you think you are done with the set up, please close the application by pressing the home button and try to open a webpage such as google.com in the Safari application. You may also ask the person who has given you this form to help you with the last step.

Step three: Evaluation
Please answer the following questions:

1. What type of device did you use for your evaluation? [ ] iPad [ ] iPhone

2. Did you encounter any problems while using OVIS Wizard?
   It's not possible to copy paste the IP-Address and following needed information.

3. Was there anything that did not seem logic while using the app?
   The wifi connection was cut after trying to add another mesh node...

4. How did you like or dislike the “look and feel” of the app? Do you think there is a difference compared to other apps for iPhone or iPad?
   Beside the design of the Icon, the "look and feel" of the app is very similar to other apps.

5. Are there any other comments you would like to make? Any suggestions for improvements?
Evaluation of OVIS Wizard for iOS

Bachelor's Project Marcel Stolz
CDS Group, IAM, University of Bern

Thank you for participating in this short survey. All data specified will be used for academic and educational use only.

Personal Information

First Name: David
Last Name: Weibel
Achieved educational graduation(s): Handelsmatur

Your personal technical abilities: [ ] Very good [ ] Good [ ] Average [ ] Poor [ ] None

What technical instruments do you use in your everyday life?

(E.g. mobile phone • iOS • laptop • ticket machine...)

Sony with Android OS • Laptop (Windows)

Evaluation of the OVIS Wizard for iOS

Note: You are going to evaluate OVIS Wizard for iOS. It is important, that you write down all problems you encounter. There may be problems which you put down to your personal skills. However, we think that these problems are also software issues. Therefore, we kindly ask you to write them down as well.

Your goal: In the following part, you have to set up an ad-hoc network for an Internet connection using the OVIS Wizard. You will do this using the OVIS Wizard application either on an iPhone or an iPad. There will be only some very short instructions in the following, as we think, that most steps should be evident from within the application. Please write down a remark on this evaluation sheet if this is not the case.

Step one: Take your iPhone or iPad and, if not yet done so, turn it on. Please make sure that you have got at least three mesh nodes ready (they do not need to be turned on yet) and that you know where the next Internet router is located.

Step two: Now, please start the application OVIS Wizard on your iPad or iPhone and follow the steps on the screen in order to set up an Internet connection. If you think you are done with the setup, please close the application by pressing the home button and try to open a webpage such as google.com in the Safari application. You may also ask the person who has given you this form to help you with the last step.

Step three: Evaluation
Please answer the following questions:

1. What type of device did you use for your evaluation?
   - [ ] iPad
   - [ ] iPhone

2. Did you encounter any problems while using OVIS Wizard?
   - [ ] Gateway - Browser:
     - Es war mir nicht ganz klar was der Kriterium auszulassen war.
   - [ ] Kriterium Plausibilität:
     - Es war mir nicht klar ob ich den Kriterium wilkhinmon dann

3. Was there anything that did not seem logical while using the app?

4. How did you like or dislike the "look and feel" of the app? Do you think there is a difference compared to other apps for iPhone or iPad?
   - [ ] Das "Look and Feel" hat mir gut gefallen, das App war klar und logisch strukturiert.
   - [ ] Die Darstellung ist kein iPhone oder iPad konzipiert, kann ich also nicht mit anderen vergleichen.

5. Are there any other comments you would like to make? Any suggestions for improvements?
Evaluation of OVIS Wizard for iOS

Personal Information
First Name: Rino
Last Name: Rusinski
Achieved educational graduation(s): Natural (CH), currently studying mathematics
Your personal technical abilities: [ ] Very good [X] Good [X] Average [ ] Poor [ ] None
What technical instruments do you use in your everyday life? (e.g. mobile phone, tablet, laptop, ticket machine, ...) Mobile phone (Samsung Android), MacBook

Evaluation of the OVIS Wizard for iOS
Note: You are going to evaluate OVIS Wizard for iOS. It is important, that you write down all problems you encounter. There may be problems which you put down to your personal skills. However, we think that these problems are also software issues. Therefore, we kindly ask you to write them down as well.
Your goal: In the following part, you have to set up an ad-hoc network using the OVIS Wizard. You will do this using the OVIS Wizard application either on an iPhone or an iPad. There will be only some very short instructions in the following, as we think, that most steps should be evident from within the application. Please write down a remark on this evaluation sheet if this is not the case.

Step one: Take your iPhone or iPad and, if not yet done so, turn it on. Please make sure that you have got at least three mesh nodes ready (they do not need to be turned on yet) and that you know where the next Internet router is located.

Step two: Now, please start the application OVIS Wizard on your iPad or iPhone and follow the steps on the screen in order to set up an Internet connection. If you think you are done with the set up, please close the application by pressing the home button and try to open a webpage such as google.com in the Safari application. You may also ask the person who has given you this form to help you with the last step.

Step three: Evaluation
Please answer the following questions:
1. What type of device did you use for your evaluation?
   [ ] iPad [ ] iPhone
2. Did you encounter any problems while using OVIS Wizard?
   No, not really. The only thing I needed to do was to connect additional node.
3. Was there anything that did not seem logic while using the app?
   No.
4. How did you like or dislike the "look and feel" of the app? Do you think there is a difference compared to other apps for iPhone or iPad?
   I didn't know other apps for iOS. I can answer this second question. As for the first one, I can only think of "Siri" as an app accessible directly through OVIS.
5. Are there any other comments you would like to make? Any suggestions for improvements?
   No, not really.
Evaluation of OVIS Wizard for iOS

Bachelor's Project Marcel Stolz
CDS Group, IAM, University of Bern

Thank you for participating in this short survey. All data specified will be used for academic and educational use only.

Personal Information
First Name: Claire
Last Name: Girle
Achieved educational graduation(s):
Master of Science in

What personal technical abilities do you have:

What technical instruments do you use in your everyday life?
(e.g., mobile phone + G3, laptop, ticket machine, ...)

Handy, MacBook

Evaluation of the OVIS Wizard for iOS

Note: You are going to evaluate OVIS Wizard for iOS. It is important, that you write down all problems you encounter. There may be problems which you put down to your personal skills. However, we think that these problems are also software issues. Therefore, we kindly ask you to write them down as well.

Your goal: In the following part, you have to set up an ad-hoc network using the OVIS Wizard. You will do this using the OVIS Wizard application either on an iPhone or an iPad. There will be only some very short instructions in the following, as we think, that most steps should be evident from within the application. Please write down a remark on this evaluation sheet if this is not the case.

Step one: Take your iPhone or iPad and, if not yet done so, turn it on. Please make sure that you have got at least three mesh nodes ready (they do not need to be turned on yet) and that you know where the next Internet router is located.

Step Two: Now, please start the application OVIS Wizard on your iPhone or iPad and follow the steps on the screen in order to set up an Internet connection. If you think you are done with the set up, please close the application by pressing the home button and try to open a webpage such as google.com in the Safari application. You may also ask the person who has given you this form to help you with the last step.

Step three: Evaluation
Please answer the following questions:

1. What type of device did you use for your evaluation? [ ] iPad [x] iPhone

2. Did you encounter any problems while using OVIS Wizard?
   [x] No

3. Was there anything that did not seem logical while using the app?
   Not really, there were even pictures of everything.

4. How did you like or dislike the "look and feel" of the app? Do you think there is a difference compared to other apps for iPhone or iPad?
   It is functional and very simple. I don't know. I have never used an iPhone before.

5. Are there any other comments you would like to make? Any suggestions for improvements?
Evaluation of OVIS Wizard for iOS

Bachelor's Project Marcel Stoiz
CDS Group, IAM, University of Bern

Thank you for participating in this short survey. All data specified will be used for academic and educational use only.

Personal Information

First Name: Gianluca
Last Name: Greco
Achieved educational graduation(s): Automobiliker
Your personal technical abilities:

What technical instruments do you use in your everyday life?
(e.g. mobile phone + OS, laptop, ticket machine, ...)

Evaluation of the OVIS Wizard for iOS

Note: You are going to evaluate OVIS Wizard for iOS. It is important, that you write down all problems you encounter. There may be problems which you put to your personal skills. However, we think that these problems are also software issues. Therefore, we kindly ask you to write them down as well.

Your goal: In the following part, you have to set up an ad-hoc network using the OVIS Wizard. You will do this using the OVIS Wizard application either on an iPhone or an iPad. There will be only some very short instructions in the following, as we think, that most steps should be evident from within the application. Please write down a remark on this evaluation sheet if this is not the case.

Step one: Take your iPhone or iPad and, if not yet done so, turn it on. Please make sure that you have got at least three mesh nodes ready (they do not need to be turned on yet) and that you know where the next Internet router is located.

Step two: Now, please start the application OVIS Wizard on your iPad or iPhone and follow the steps on the screen in order to set up an Internet connection. If you think you are done with the set up, please close the application by pressing the home button and try to open a webpage such as google.com in the Safari application. You may also ask the person who has given you this form to help you with the last step.

Step three: Evaluation

Please answer the following questions:

1. What type of device did you use for your evaluation?
   - [X] iPad
   - [ ] iPhone

2. Did you encounter any problems while using OVIS Wizard?
   - Die Internetverbindung konnte nicht richtig hergestellt werden.

3. Was there anything that did not seem logical while using the app?
   - Bedingungen sollten im Anhang für einen Zweck formuliert werden. ❌
   - Servo Schalt Text besser formulieren

4. How did you like or dislike the "look and feel" of the app? Do you think there is a difference compared to other apps for iPhone or iPad?

5. Are there any other comments you would like to make? Any suggestions for improvements?

Figure 1: Mesh Node

Figure 2: Icon of OVIS Wizard
Evaluation of OVIS Wizard for iOS

Bachelor's Project Marcel Stolz
CDS Group, IAM, University of Bern

Thank you for participating in this short survey. All data specified will be used for academic and educational use only.

Personal Information
First Name: Arion
Last Name: Uragi
Achieved educational graduation(s): Matura (Hauptschulabschluss)

Your personal technical abilities: [ ] very good [ ] good [ ] average [ ] poor [ ] none

What technical instruments do you use in your everyday life?
(e.g. mobile phone + G3, laptop, ticket machine, …)

Evaluation of the OVIS Wizard for iOS

Note: You are going to evaluate OVIS Wizard for iOS. It is important, that you write down all problems you encounter. There may be problems which you put down to your personal skills. However, we think that these problems are also software issues. Therefore, we kindly ask you to write them down as well.

Your goal: In the following part, you have to set up an ad-hoc network for an Internet connection using the OVIS Wizard. You will do this using the OVIS Wizard application either on an iPhone or an iPad. There will be only some very short instructions in the following, as we think, that most steps should be evident from within the application. Please write down a remark on this evaluation sheet if this is not the case.

Step one: Take your iPhone or iPad and, if not yet done so, turn it on. Please make sure that you have got at least three mesh nodes ready (they do not need to be turned on yet) and that you know where the next Internet router is located.

Step Two: Now, please start the application OVIS Wizard on your iPad or iPhone and follow the steps on the screen in order to set up an Internet connection. If you think you are done with the set up, please close the application by pressing the home button and try to open a webpage such as google.com in the Safari application. You may also ask the person who has given you this form to help you with the last step.

Step three: Evaluation
Please answer the following questions:

1. What type of device did you use for your evaluation?
   - [ ] iPad
   - [x] iPhone

2. Did you encounter any problems while using OVIS Wizard?
   - [ ] Yes
   - [ ] No
   - [ ] Other: OVIS Network we couldn’t reach...and unidentified...

3. Was there anything that did not seem logical while using the app?
   - [ ] Yes
   - [ ] No

4. How did you like or dislike the “look and feel” of the app? Do you think there is a difference compared to other apps for iPhone or iPad?
   - [ ] Like:
   - [ ] Dislike:
   - [ ] Other: Just a Nick

5. Are there any other comments you would like to make? Any suggestions for improvements?
   - English: Look harder...in the case and check signs straight, their Internet-Input “Network”...
Evaluation of OVIS Wizard for iOS

Bachelor's Project Marcel Stolz
CDS Group, IAM, University of Bern

Thank you for participating in this short survey. All data specified will be used for academic and educational use only.

Personal Information
First Name: Lukas
Last Name: Schler
Achieved educational graduation(s):
Landschaftsbaumeister, Berufsmaturaushule
Your personal technical abilities:
Very good good average poor none
What technical instruments do you use in your everyday life?
Mobile phone, MP3, laptop, tablet, scanner

Evaluation of the OVIS Wizard for iOS

Note: You are going to evaluate OVIS Wizard for iOS. It is important, that you write down all problems you encounter. There may be problems which you put down to your personal skills. However, we think that these problems are also software issues. Therefore, we kindly ask you to write them down as well.

Your goal: In the following part, you have to set up an ad-hoc network using the OVIS Wizard. You will do this using the OVIS Wizard application either on an iPhone or an iPad. There will only be some very short instructions in the following, as we think, that most steps should be evident from within the application. Please write down a remark on this evaluation sheet if this is not the case.

Step one: Take your iPhone or iPad and, if not yet done so, turn it on. Please make sure that you have got at least three mesh nodes ready (they do not need to be turned on yet) and that you know where the next Internet router is located.

Step Two: Now, please start the application OVIS Wizard on your iPad or iPhone and follow the steps on the screen in order to set up an Internet connection. If you think you are done with the set up, please close the application by pressing the home button and try to open a webpage such as google.com in the Safari application. You may also ask the person who has given you this form to help you with the last steps.

Step three: Evaluation
Please answer the following questions:

1. What type of device did you use for your evaluation?

   - [ ] iPad
   - [x] Phone

2. Did you encounter any problems while using OVIS Wizard?
   Beim wählen des WLAN-Netzes hat die Verbindung einige Male nicht funktioniert.
   "Bess Solgal die Verbindung ist zu stark gehen sie weiter weg"
   Nicht klar, dass man den Messknopf minnehmen soll.

3. Was there anything that did not seem logic while using the app?

4. How did you like or dislike the "look and feel" of the app? Do you think there is a difference compared to other apps for iPhone or iPad?
   Benutzerfreundlich alle Schritte sauber erklärt.

5. Are there any other comments you would like to make? Any suggestions for improvements?
   Anstatter: "Gehe 'Sie' weiter weg'
   "Gehe 'Sie' mit dem Messknopf weiter weg"
   Tablet-Comments: Mehr Infos. Evt. für erste Nutzung
   Bisschen viel Infos.
Evaluation of OVIS Wizard for iOS

Bachelor's Project Marcel Stolz
CDS Group, IAM, University of Bern

Thank you for participating in this short survey. All data specified will be used for academic and educational use only.

Personal Information

First Name: Wahl
Last Name: Wyler

Achieved educational graduation(s):
Bachelor of Science

Your personal technical abilities:
Very good ☣ good ☣ average ☣ poor ☣ none ☣

What technical instruments do you use in your everyday life?
(laptop, switch, mobile phone, (Android), radio,)

Evaluation of the OVIS Wizard for iOS

Note: You are going to evaluate OVIS Wizard for iOS. It is important, that you write down all problems you encounter. There may be problems which you put down to your personal skills. However, we think that these problems are also software issues. Therefore, we kindly ask you to write them down as well.

Your goal: In the following part, you have to set up an ad-hoc network for an Internet connection using the OVIS Wizard. You will do this using the OVIS Wizard application either on an iPhone or an iPad. There will be only some very short instructions in the following, as we think, that most steps should be evident from within the application. Please write down a remark on this evaluation sheet if this is not the case.

Step one: Take your iPhone or iPad and, if not yet done so, turn it on. Please make sure that you have at least three mesh nodes ready (they do not need to be turned on yet) and that you know where the next Internet router is located.

Step Two: Now, please start the application OVIS Wizard on your iPad or iPhone and follow the steps on the screen in order to set up an Internet connection. If you think you are done with the set up, please close the application by pressing the home button and try to open a webpage such as google.com in the Safari application. You may also ask the person who has given you this form to help you with the last step.

Step three: Evaluation

Please answer the following questions:

1. What type of device did you use for your evaluation?
   ☣ Pad ☐ Phone

2. Did you encounter any problems while using OVIS Wizard?
   ☐

3. Was there anything that did not seem logical while using the app?
   ☐

4. How did you like or dislike the "look and feel" of the app? Do you think there is a difference compared to other apps for iPhone or iPad?
   ☐ Einheit Handhabung ☐ Erfüllt meine Erwartungen nicht in Bezug auf Layout und Anleitung

5. Are there any other comments you would like to make? Any suggestions for improvements?
   Copy & Paste were on Work!
**Evaluation of OVIS Wizard for iOS**

**Bachelor's Project Marcel Stolz**
CDS Group, IAM, University of Bern

---

**Personal Information**

**First Name:** Patrick

**Last Name:** Nybr

**Achieved educational graduation(s):**

---

**Evaluation of the OVIS Wizard for iOS**

**Note:** You are going to evaluate OVIS Wizard for iOS. It is important, that you write down all problems you encounter. There may be problems which you put down to your personal skills. However, we think that these problems are also software issues. Therefore, we kindly ask you to write them down as well.

**Your goal:** In the following part, you have to set up an ad-hoc network for an Internet connection using the OVIS Wizard. You will do this using the OVIS Wizard application either on an iPhone or an iPad. There will be only some very short instructions in the following, as we think, that most steps should be evident from within the application. Please write down a remark on this evaluation sheet if this is not the case.

**Step one:** Take your iPhone or iPad and, if not yet done so, turn it on. Please make sure that you have got at least three mesh nodes ready (they do not need to be turned on yet) and that you know where the next Internet router is located.

---

**Step Two:** Now, please start the application OVIS Wizard on your iPad or iPhone and follow the steps on the screen in order to set up an Internet connection. If you think you are done with the set up, please close the application by pressing the home button and try to open a webpage such as google.com in the Safari application. You may also ask the person who has given you this form to help you with the last step.

**Step three: Evaluation**

Please answer the following questions:

1. **What type of device did you use for your evaluation?**
   - [ ] iPad
   - [x] iPhone

2. **Did you encounter any problems while using OVIS Wizard?**
   - [ ] Yes
   - [ ] No

3. **Was there anything that did not seem logic while using the app?**
   - [ ] Yes
   - [ ] No

4. **How did you like or dislike the “look and feel” of the app? Do you think there is a difference compared to other apps for iPhone or iPad?**
   - [ ] Yes
   - [ ] No

5. **Are there any other comments you would like to make? Any suggestions for improvements?**
   - [ ] Yes
   - [ ] No
   - **Comment:**
     - [ ] More instructions Gebrauchsanleitung
     - [ ] More illustrations

---

**Figure 1: Mesh Node**

**Figure 2: Icon of OVIS Wizard**
Evaluation of OVIS Wizard for iOS

Bachelor's Project Marcel Stolz
CDS Group, IAM, University of Bern

Thank you for participating in this short survey. All data specified will be used for academic and educational use only.

Personal Information

First Name: Christian
Last Name: Rippstein

Achieved educational graduation(s):
- Bachelor of Arts in Economics
- Master of Science in Economics

Your personal technical abilities:
- Very good
- Good
- Average
- Poor
- None

What technical instruments do you use in your everyday life?
- Mobile phone
- Tablet
- Laptop
- Desktop computer
- Flat screen monitor
- Virtual reality glasses
- Virtual reality headset

Evaluation of the OVIS Wizard for iOS

Note: You are going to evaluate OVIS Wizard for iOS. It is important that you write down all problems you encounter. There may be problems which you put down to your personal skills. However, we think that these problems are also software issues. Therefore, we kindly ask you to write them down as well.

Your goal: In the following part, you have to set up an ad-hoc network for an Internet connection using the OVIS Wizard. You will do this using the OVIS Wizard application either on an iPhone or an iPad. There will be some very short instructions in the following, as we think that most steps should be evident from within the application. Please write down a remark on this evaluation sheet if this is not the case.

Step one: Take your iPhone or iPad and, if not yet done so, turn it on. Please make sure that you have got at least three mesh nodes ready (they do not need to be turned on yet) and that you know where the next Internet router is located.

Step Two: Now, please start the application OVIS Wizard on your iPad or iPhone and follow the steps on the screen in order to set up an Internet connection. If you think you are done with the set up, please close the application by pressing the home button and try to open a webpage such as google.com in the Safari application. You may also ask the person who has given you this form to help you with the last step.

Step three: Evaluation

Please answer the following questions:

1. What type of device did you use for your evaluation?
   - [ ] iPhone
   - [x] iPad

2. Did you encounter any problems while using OVIS Wizard?
   - [ ] Yes
   - [ ] No

3. Was there anything that did not seem logic while using the app?
   - [ ] Yes
   - [ ] No

4. How did you like or dislike the “look and feel” of the app? Do you think there is a difference compared to other apps for iPhone or iPad?
   - [ ] Like
   - [ ] Dislike

5. Are there any other comments you would like to make? Any suggestions for improvements?
   - [ ] Yes
   - [ ] No

Figure 1: Mesh Node

Figure 2: Icon of OVIS Wizard


[24] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1994.


