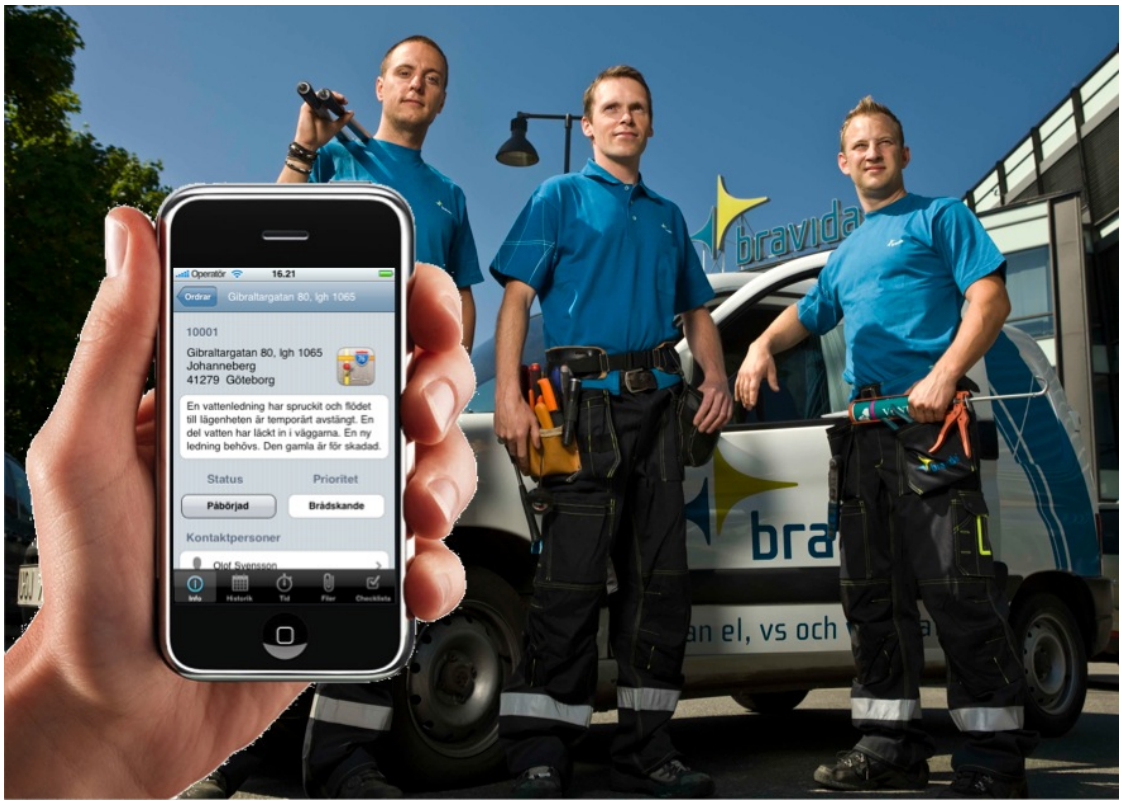
CHALMERS



Usable And Dynamic Interface For Mobile Reporting

Master of Science Thesis in the Master Degree Programme, Interaction Design

ALEXANDER SKOGBERG
OLOF MILLBERG

Department of Computer Science and Information Technology
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden, 2010

Usable And Dynamic Interface For Mobile Reporting

ALEXANDER SKOGBERG
OLOF MILLBERG

Department of Computer Science and Information Technology
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone + 46 (0) 31-772 1000

Cover: Photo of Bravida personnel taken by Petter Karlberg with a superimposed iPhone image from Apple Inc., edited to look like it is running the application prototype developed in the thesis.

Department of Computer Science and Information Technology
Göteborg, Sweden 2010

Foreword

First and foremost we would like to thank the employees at Binea; Andreas Hallingström, Andreas Nilsson, Olof Svensson and Kyrylo Yakymenko for providing us with a great master thesis, superb working conditions and for making us feel like a part of the Binea family.

Warm thanks also go out to Bravida and Göteborg Energi AB, who were friendly and co-operative when approached for user studies.

We would also like to thank our examiner Olof Torgersson for his help and support.

Last but not least a thank goes out to the staff and fellow companies at World Trade Center in Gothenburg for taking care of us and offering awesome cakes, cookies and coffee every Thursday.

Abstract

The purpose of this thesis was to create a design concept for a mobile reporting application and develop a software prototype based on it. The focus of the prototype was great usability and dynamic system architecture for smooth adaption to different users. Insight was also wanted on hostility towards technology. Is it true that older employees in general are more opposed to new technical solutions than their younger colleagues?

Mobile reporting is a software solution for taking care of everyday work tasks such as time reporting, work order handling and equipment handling using mobile devices. Service providers that have field personnel, such as electricians and carpenters, commonly use it. This solution offers a paperless working environment with high efficiency and good communication between employees and the office.

User studies were performed with service providers in Gothenburg using interviews, field studies and paper prototyping. Design solutions were created and then evaluated with cognitive walkthroughs and user studies with these service providers.

The software prototype was implemented on the iOS, which runs on Apple's iPod Touch, iPhone and iPad devices. It was evaluated by interaction design students at Chalmers University of Technology using well-known design patterns and the iPhone Human Interface Guidelines.

The resulting concept is based on user studies and studies of similar applications. The concept offers descriptions and design solutions for must-have features. A design manual for general smartphone GUI design is also provided. Information about technology hostility and ideas about dealing with it based on the user studies are also mentioned. The iPhone application prototype offers features for time reporting and work order handling as well as route directions using Google Maps and other common smartphone functions.

Keywords: Smartphones, Mobile Reporting, Interaction Design, Usability, GUI Design, iOS, Dynamicity, Service Providers.

Glossary

API – Application Programming Interface

Gestures – Movements performed with one or more fingers on multi-touch screens.

GUI – Graphical User Interface. Only focuses on the visuals of a UI.

HCI – Human-Computer Interaction.

IxD – Interaction Design.

Multi-Touch – Interacting with a computer screen using more than one finger.

OS – Operating System.

Platform – Used as a synonym for OS.

Stylus – A pen shaped tool used on touch screens.

SDK – Software Development Kit.

UI – User Interface. Includes all aspects of a user interface.

Xcode – Development platform for creating applications for Mac OS X and iOS.

Table of Content

1 Introduction

In this initial chapter an introduction to the thesis is given. The problem description, deliverables, limitations and purpose are described along with an outline of the report.

1.1 Problem Description

According to Binea, the company providing the master thesis, a revolution took place on the market for mobile devices when Apple's iPhone was released in 2007. Since then the market has grown enormously and thanks to advances in hardware and Internet connectivity it has become feasible to move business systems from the traditional desktop environment to the smartphone platforms.

Mobile reporting is one emerging business solution that is making headway on mobile platforms. Service providers such as electricians and carpenters, who have lots of personnel working in the field, use this form of reporting. Through such a service they handle work orders, time reporting and equipment handling.

Binea has learnt that many of the available mobile reporting applications fall short in the area of usability. Not all features are provided and the applications are often difficult to use for novice users of mobile devices. From a business perspective an application with high usability and a system architecture enabling easy adaptation to different service providers is very interesting.

After having studied previous attempts of moving business systems to the smartphone platforms Binea noticed user hostility towards new technical solutions. Users seem to prefer well tried but inefficient solutions and are hard to persuade. More information regarding this issue is wanted.

1.2 Deliverables

Binea are to be provided with a study of mobile reporting practice among service providers. The study shall include a profile on a typical service provider with needs for or experience of mobile reporting. Must-have features shall be included with advice for overcoming potential technology hostility.

A design manual shall also be included showcasing design solutions for the must-have features of an application of this kind. The manual shall also include general GUI design guidelines for achieving great usability.

Finally, a smartphone application shall also be developed based on the gathered information. For evaluating the design concept and application a final user study shall be performed.

1.3 Delimitations

The software implementation will only be a high-fidelity prototype and not a commercial product due to the short time frame of this master thesis. Developing and evaluating a ready-to-release application in this time frame is not feasible. Especially not without a customer connected to the project.

The design manual will focus on GUI design in general and not on a specific platform. Too much focus on one platform might hinder the design concept from reaching its full usability potential. The manual will be written as an internal tool for Binea to use when developing smartphone applications in the future.

1.4 Purpose

The purpose is to gain knowledge in designing dynamic and easily configurable GUIs with high usability, focused on the business area of mobile reporting.

Knowledge shall also be gained about technology hostility among employees at different service providers. Do elderly employees always have a harder time accepting new technical solutions? If so, how can this be handled? Is IxD and usability the right approach?

Information of the constantly evolving smartphone market shall be gathered. Which platforms are leading the market today? What will the future hold?

For the platform chosen for implementing the application prototype, knowledge shall be gathered about designing a dynamic GUI that is easily adaptable to service providers with different needs.

1.5 Thesis Outline

A brief overview of the structure of the report is presented below.

Background

Covers information about mobile reporting, smartphone platforms, market share and companies involved in the thesis work.

Theory

Presents theoretical areas such as IxD, technology acceptance and how to design for users with different skill levels.

Related Work

Presents similar smartphone applications available on today's market.

Methods

Defines the design process and the methods used in the work process.

Realization

This section explains how the work progressed, what problems were encountered and which design decisions were made.

Result

Here the result of the concept, application prototype and user studies are presented.

Discussion

This chapter discusses the result of the thesis. What went well and what could have been done differently is brought up.

Future Work

Presents decided features, ideas and possibilities for future versions of the application prototype and the design manual. Plans for new user studies are also mentioned.

Conclusion

In this final chapter an overview and short reflection on the thesis is given.

2 Background

This chapter provides information about mobile reporting, smartphones and the companies involved in the thesis.

2.1 Mobile Reporting

Service providers such as electricians, carpenters and contract providers usually have lots of personnel in the field. These employees work independently at different work site and report their actions to their local office every week. This is either done using pen and paper or mobile reporting.

Mobile reporting is a software solution for mobile devices, which makes it easier to handle common tasks. It offers a paperless and time efficient process with good synchronization between employees and their office.

The tasks that must be handled using either pen and paper or mobile reporting are the following:

- Time reporting. How much time and of what type (overtime or regular office hours) was spent at a work site must be specified.
- Work order handling. Information about the work orders an employee is dealing with must be accessible.
- Material handling. Field personnel usually have cars that double as storage units. The available material and what of it that is used for a work site must be presented.

2.1.1 Benefits

Based on case studies (Microsoft, n.d.) mobile reporting offers several improvements:

- Improved productivity and time efficiency. Reporting takes less time and more tasks can be finished.
- Improved data accuracy. Miscommunication and manual calculation errors are reduced
- Increased customer satisfaction. Thanks to increased productivity and saved time employees can spend more time at the work sites.
- Reduced driving time and related expenses. Fuel consumption and driving mileage are reduced thanks to fewer trips taken to the office and work sites.

Hjelm & Kähkönen (2009) also discuss the advantages of a mobile reporting system. They have analyzed the working process for field personnel in companies before and after they started using a system for mobile reporting. Their conclusions were the following:

- Thanks to automatic digital data transfers the need for manual re-validation is decreased.
- Increased efficiency, work task distribution and information access within the organization thanks to frequent synchronization of data.
- Structuring information using mobile devices results in more time for dealing with work tasks.
- More interpretable information thanks to the digital presentation.
- Less communication problems thanks to the set communication channel through the reporting application.
- Better quality of data thanks to predefined choices for data input.
- Increased internal costumer satisfaction thanks to data being more often correct and easier to interpret.
- Increased external costumer satisfaction thanks to the high possibility of information being correct and easily accessible.

2.2 Smartphones

Despite the large market and the term being frequently used there is no proper definition of what a smartphone is. However, it is considered to be a mobile phone with greater computing power than regular cell phones offering features common for desktop computers.

2.2.1 History

In 1993 IBM presented the first mobile device that was considered a smartphone. Its name was Simon and had features such as calendar, address book, world clock, calculator and e-mail client. It had a touch screen that let the user enter text using an on-screen keyboard or by writing using a stylus. Competitors followed and in the mid-90's Nokia entered the smartphone market with their Nokia 9000 series (Sooper Articles, n.d. and Articlesbase, 2009).

In recent years huge advances have been made on the smartphone market. When Apple released the iPhone it had a huge effect on smartphone UI design with its multi-touch technology, innovative design solutions and the AppStore in which professional and amateur developers can get their applications published and purchased.

Figure 1 and 2. Smartphones then and now; IBM's Simon (IBM, 2010) and Apple's iPhone (Apple Inc., 2010).

Today mobile broadband is offering fast connection speeds and small and powerful hardware components enable multitasking and advanced graphics comparable to desktop computers.

2.2.2 Platforms

This section describes the popular smartphone operating systems of today and tomorrow.

2.2.2.1 Android

Android is Google's smartphone platform and has been available since November 2008. The OS is open source and runs on a modified Linux kernel. Development is done in Java using libraries developed by Google (Android, 2010a).

2.2.2.2 Bada

Samsung Electronics' is currently developing Bada. The first phone running the software will be released in April 2010. Development of these applications will be done in C++, but support for Java ME applications will exist (bada, 2010).

2.2.2.3 Blackberry OS

BlackBerry OS is developed by Research In Motion. It runs on all mobile devices by manufactured by BlackBerry (BlackBerry, 2010)

2.2.2.4 iOS

iOS is Apple's contribution to the smartphone platform market. It is run on the iPhone, iPod Touch and iPad. It was first released in conjunction with the iPhone in June 2007. Development is done using C, C++ and Objective-C with the Cocoa framework. Before version 4 iOS was called iPhone OS (Apple Inc., 2010a).

2.2.2.5 MeeGo

MeeGo is an open source platform that is being developed by Intel and Nokia. It will merge Intel's Moblin and Nokia's Maemo project into one project and is slated for a release in the second quarter of 2010. It is hosted by the Linux Foundation. (MeeGo, 2010).

2.2.2.6 Palm WebOS

Palm webOS is Palm's smartphone platform. The first smartphone running the platform was the Palm Pre, which was released in conjunction with the platform on June 6, 2009. The underlying software stack of the OS shares many similarities with Linux running on desktop computers. Development is heavily based around web technologies such as HTML, JavaScript, AJAX as well as C and C++ (palm webOS, 2010).

2.2.2.7 Symbian Platform

Symbian Foundation develops the Symbian Platform. The open source platform was based on Symbian OS, the most used OS for mobile phones. It was integrated with software assets contributed by Nokia, Sony Ericsson and others. The latest version is Symbian^3 (Symbian, 2010).

2.2.2.8 Windows Mobile

Windows Mobile is one of Microsoft's smartphone platforms. The current version is named Windows Mobile 6.5 and is based on the Windows CE 5.2 kernel. The platform is somewhat similar to desktop versions of Windows, both in looks and in functionality. In spite of the future release of the Windows Phone 7 Series platform, Windows Mobile 6.5 will continue to be provided with support (Microsoft, 2009).

2.2.2.9 Windows Phone 7

The Windows Phone 7 platform is Microsoft's successor to Windows Mobile. Unlike Windows Mobile it is developed from scratch and has replaced all of its successor's common interface paradigms. The OS is based on the Windows CE6 kernel. It was first revealed on the World Mobile Congress in February 2010 and is expected to be available for purchase for the holiday shopping season the same year (Engadget, 2010a).

2.2.3 Sales and Market Share

According to studies by Gartner (2010a and 2010b) the smartphone market is alive and well. In the first quarter of 2010 a total amount of 314.7 million mobile devices were sold, an increase with 17 per cent compared to the same period in 2009. Smartphones sales stood for 17.3 per cent of this amount, compared to 13.6 per cent for the same period in 2009. According to their studies in 2009 and 2010 the most popular platforms are Symbian, BlackBerry OS, iOS, Windows Mobile and Android. The development of their market shares from the first quarter in 2009 to the first quarter 2010 is presented below in figure 3 below.

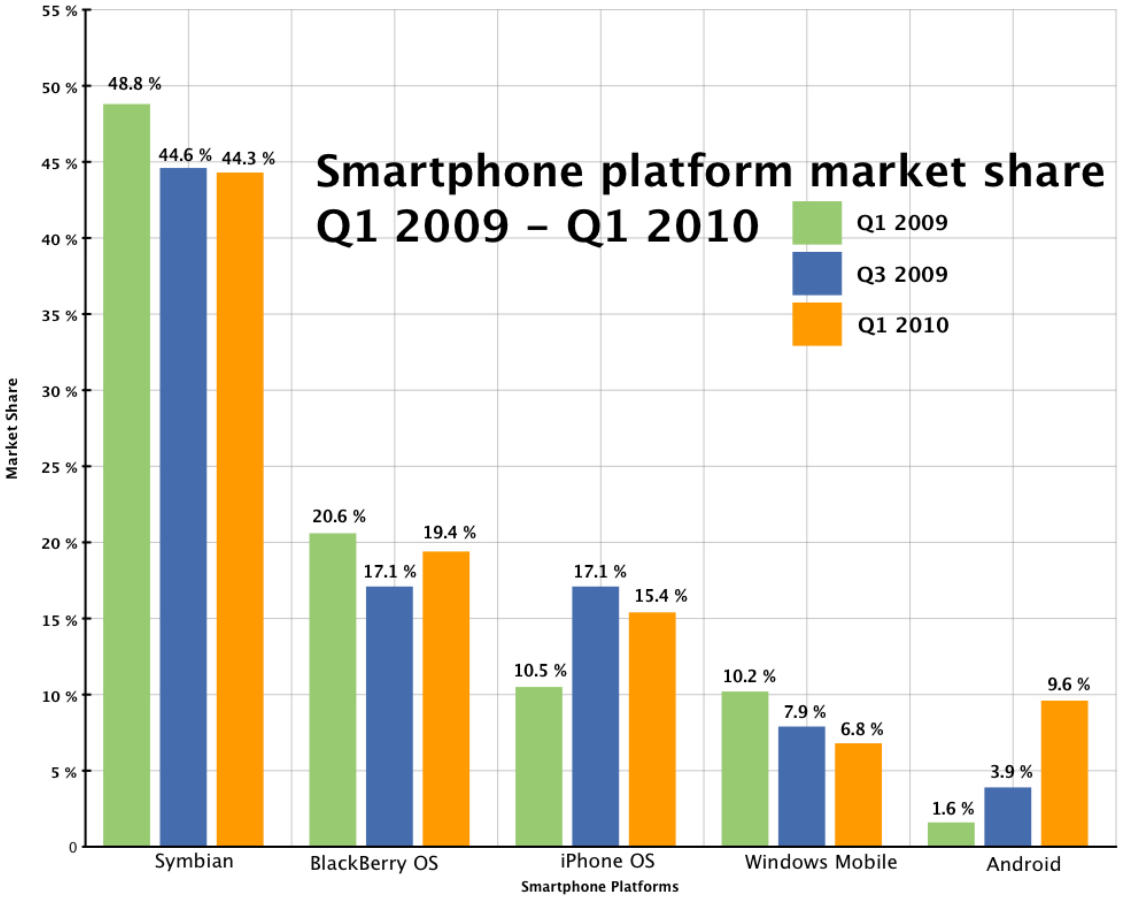


Figure 3. The market shares of the smartphone platforms Symbian, Windows Mobile, Android, iOS and BlackBerry OS in the first and third quarter of 2009 and the first quarter in 2010 (Gartner, 2010a and 2010b).

According to a study by Practical eCommerce (2010) the market shares of the previously mentioned platforms are expected to have changed to the numbers presented below in figure 4 by 2012.

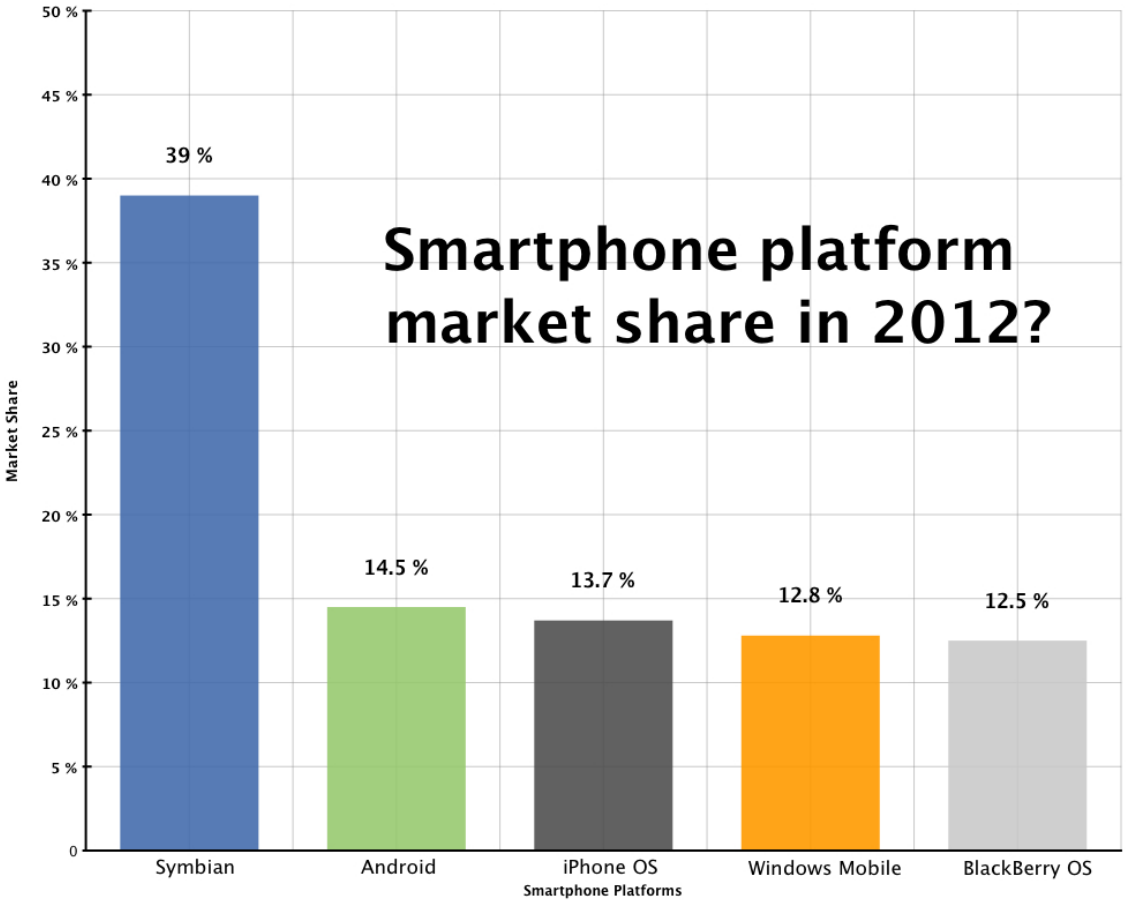


Figure 4. The expected market share of the smartphone platforms Symbian, Windows Mobile, Android, iOS and BlackBerry OS in 2012 according to Practical eCommerce (2010).

2.3 Involved Companies

This section introduces the different companies involved in the master thesis.

2.3.1 Binea

Binea is the company that provided the master thesis. It is a consultancy company based in Gothenburg, Sweden. Their area of business is in development of business and information systems in .NET, web applications and smartphone applications (Binea, 2010).

2.3.2 Bravida

Bravida participated in user studies. The company is the largest provider of technical installations and services for real estates in Scandinavia. Their 9,000 employees handle minor and major service tasks for offices, industries, new-built estates and other facilities including everything from designing and planning to installations and support. They can be found at 175 locations throughout Sweden, Norway and Denmark.

The field personnel in some departments have tested different applications for mobile reporting and are currently using one, while and others are about to start (Bravida, 2010).

2.3.3 Göteborg Energi AB

Göteborg Energi AB participated in user studies. The company is the 4th largest provider of electricity in Sweden. They offer services such as district heating, ready heat, energy services, cooling, gas, optical fibers and electricity supply network. Their field personal has had experience of using smartphones for mobile reporting (Göteborg Energi AB, 2010).

3 Theory

This chapter presents theoretically relevant areas such as IxD, designing for users with different skill levels and the challenges of designing for mobile devices.

3.1 Interaction Design

IxD is about bringing usability into the design process in order to reach as good a result as possible from a user perspective. It is one of the key factors in the thesis.

The interaction designers Jennifer Preece, Yvonne Rogers and Helen Sharp (2007) define the term as "Designing interactive products to support the way people communicate and interact in their everyday and working lives". They explain that IxD is a broader field and involves more issues, topics and paradigms than HCI. They feel that being a successful interaction designer involves having skills in the fields of psychology, HCI, web design, computer science, information systems, marketing, entertainment, sociology and business.

Winograd (1997 quoted in Huang, 2009, p.3) defines the term as "the design of spaces for human communication and interaction”, and "the construction of the 'interspace' in which people live, rather than an 'interface' with which they interact".

Huang points out two key points from the definition by Preece, Rogers and Sharp. First, design has to be made from the users' perspective by involving them in the design process. The design then has to be viewed as a product that is to be sold on the market. Second, the products should be useful in the users' everyday lives.

3.2 Designing For Different Users

Different users have different needs that must be taken into account when designing an interface. A helpful design solution for a novice user might be counterproductive for an expert user.

3.2.1 Experience Levels

In software design it is important to analyze the skill level of the users. Users of different experience levels have different needs and finding a proper balance might be tricky without understanding how they master new concepts and tasks.

For example, programmers qualify as experts since they need to know all the details of the software they are developing. The marketing department only sees the software from a beginner's point of view. This often results in a design biased towards expert users with cumbersome solutions to meet the needs of the marketing department. Most users fall in neither of these categories but rather in the intermediate category. They have no use for embarrassing UI assistance and no interest in advanced features.

Another aspect is that beginners do not stay beginners for long. Software designers must make sure the transition from beginner to intermediary is smooth and that the product adequately reflect the user's mental model of the task. With regards to helping facilities, the first thing a beginner needs is an overview of the product. A non-obtrusive dialog box communicating scope and goals focusing on beginner issues is often better than an external help document.

In desktop applications beginners often make use of menus to learn what is possible. Therefore it is important that GUI-menus are easy to navigate and not display too much information at once. Experts, however, demand shortcuts to everything and need fast access to frequently used tools. Features put in place for helping new users must not get in the way for experienced users.

Intermediate users already know about scope and purpose and are more interested in specific functionality. Pop-up messages can be a source of information if users do not want to go into reference materials. Like experts they want quick access to frequently used tools but they might not feel comfortable using keyboard shortcuts. Therefore these features need to be placed front and center in the UI (Cooper, 2007).

3.2.2 Technology Acceptance

Technology acceptance differs between ages. Often UIs on mobile devices require a keen eye and steady hands and since spatial visualization skills decrease with age, older people will encounter problems.

InformationWeek (2006) mentions a study by Forrester Research, where different age groups are shown to embrace different aspects of new technology. "[Older consumers] tend to use the new technology to do old things," Forrester analyst Ted Schadler said. "Younger consumers use the new technology to do new things".

According to Ars Technica (2006) young people easily adopt new forms of communication and are more likely to utilize text messaging services and online communities than their older peers. However, online shoppers are more likely to be found among 27 to 40 year-olds.

Much research has gone into what makes companies and their employees accept and adapt to new technology. The most influential theory in this area is the Technology Acceptance Model (TAM). It was developed by Fred Davis and Richard Bagozzi as an adaptation of Ajzen and Fishbein's Theory of Reasoned Action (TRA) to an IT context, adding the two concepts perceived usefulness (PU) and perceived ease of use (PEOU). PU is defined as "the degree to which a person believes that using a particular system would enhance his or her job performance", while PEOU is defined as "the degree to which a person believes that using a particular system would be free from effort" (Davis, 1989).

The model has been expanded to include the effects of social influences and cognitive instrumental processes in the form of TAM2 (Venkatesh and Davis, 2000), making it more accurate by going back to its roots in TRA. There have also been efforts to combine it with competing theories in the same area. This has resulted in the Unified Theory of Acceptance and Use of Technology, which has shown good results (Venkatesh et al., 2003).

It is hard to say how useful TAM is in practice. Its concepts and explanatory qualities with regard to adoption rates and success of IT systems are good to have in mind when designing a system. However, it does not say anything about what actually makes a system useful.

3.3 Designing For Mobile Devices

This section brings up the challenges concerning hardware and software that have to be dealt with when designing for mobile devices. Huang (2009) mentions some of these challenges, while others are based on personal experience.

3.3.1 Hardware Challenges

Due to their small size mobile devices have limited input and output facilities. For input the biggest challenge is equipping the device with a keyboard. Keyboards are placed on the front, slideable or displayed on the screen. Huang mentions studies that have shown that smaller keys do not increase the error rate, but are more cumbersome to use for people with larger fingers or poor manual dexterity.

Today all brand name smartphones are equipped with a touch screen. There are two types of them; capacitive or resistive. Resistive screens are composed of a number of layers. The most important being a metallic and an electrically conductive layer separated by a small gap. When pressing the screen surface will make these two layers connect and cause a change in the electrical current, which detects and registers a touch event. Capacitive screens work by sensing the proximity of a conductive object such as finger.

Capacitive screens allow multi-touch and do not need a stylus for input. They might be less suited for field workers since they work poorly or not at all with gloves on. Using a resistive screen with a stylus enables greater input precision, but losing the stylus (and not finding a suitable replacement) will make GUI navigation troublesome since using a finger does not give the same precision and might occlude other screen content.

The scroll wheel and the trackball can also be used for input. From personal experience the scroll wheel is easy to use but limits the interface to one dimension. The trackball offers more freedom but might take more time to master.

The most important output facility on a mobile device is of course the screen. Finding the optimal screen size and resolution is important. If not achieved, navigating the interface becomes annoying. Good visibility in varying light conditions is another challenge.

Another important output facility is audio. Good audio quality is important for communication but also a good feedback mechanism due to the general lack of tactile feedback, other than vibrations, on mobile devices with touch-screens. Sound effects should not be overdone since they can be perceived as annoying.

Figure 4 and 5. The Motorola Droid with its slideable keyboard (Motorola, 2010) and the Nokia 5800 XpressMusic with its stylus (Nokia, 2010)

3.3.2 Software Challenges

The main software issues on mobile devices are related to navigation. Displays on desktop computers and laptops offer more space for presenting information. When this information is presented on a mobile device it has to be divided into smaller units that are shown on their own. Organizing these units and making navigation smooth is a challenge.

Design of images and icons is another challenge. Downscaling images, drawings and diagrams that are fully viewable on a desktop computer to be just as informative on a restricted display on a mobile device demands a smart solution.

Today a lot of smartphones can be tilted in order to view the content of the screen in landscape format. The GUI elements have to be adapted to the "new" screen ratio if screen tilting is supported.

Development moves fast. Current platforms are often updated and new ones might reach a broad audience fast. For example, the iOS was introduced in 2007 and is now one of the most popular platforms used. Google's OS Android was released in late 2008 and is expected to rapidly increase its market share by 2012 (Practical eCommerce, 2010).

Mobile devices are quickly catching up with desktop computers in terms of performance but they are not quite there yet. Memory management is still very important and CPU heavy applications can drain the battery fast.

3.3.3 Developer Design Guidelines

For several smartphone platforms, applications must follow certain design guidelines to be published legally. This is the case when publishing iOS applications and there will be similar demands for Windows Phone 7 once the platform is released. (Engadget, 2010 and Apple Inc., 2010b). Android, on the other hand, allows users to install applications not published on the Android Market even though they also offer design guidelines (Android, 2010b).

3.4 Developing for iOS

This section provides basic information about developing for iOS, specifically on Apple's iPhone.

3.4.1 Objective-C

The main language used when developing for iOS is Objective-C. Apple defines Objective-C as a set of small but powerful extensions to ANSI C based on the object-oriented language Smalltalk. Any C program can be compiled in an Objective-C compiler and C code can be freely included in an Objective-C class. All object-oriented code is written in Smalltalk syntax.

Objective-C uses dynamic binding and instead of calling methods, like in other languages such as C++, C# or Java, messages are sent to objects. This means that messages can be sent to a set of objects where only some are expected to respond, without the fear of producing runtime errors.

Since there is no garbage collection in iOS memory management needs to be handled manually using reference counting. When an object is created its reference count is set to one. Calls such as *copy* and *retain* increase the counter by one while *release* decreases the counter by one. When the retain count reaches zero the memory is freed.

3.4.2 Cocoa Touch

Cocoa Touch is an API for building applications for Apple's iOS, which runs on their iPhone, iPod Touch and iPad devices. It is based on the Cocoa API for OS X and is written in Objective-C.

3.4.3 Components

iOS applications consist of one or several views. Cocoa Touch primarily provides two ways of moving between views; the navigation controller and the tab bar controller. The former holds a stack of views. The current view is presented with a title bar at the top of the screen with a back button to the previous view. The latter provides access to different views via a bar with clickable icons at the bottom of the screen. Additional GUI elements can be viewed in Appendix B - Components.

3.4.4 Xcode

Xcode is Apple's suite of development tools for application development for OS X and iOS. Its tools include the Interface Builder used for designing the views of an application. It also includes an iPhone simulator in order to run iOS applications on desktop computers. Several analysis tools are also bundled for analyzing aspects such as memory allocation.

3.4.5 App Store

iOS Applications that are to be released have to be approved and published on the App Store. This service is maintained by Apple and allows users of iOS devices to purchase and download applications either directly to their devices or via Apple's digital media player iTunes. Applications can be free or cost money. In the latter case Apple charges 30% of the revenue, the rest goes to the developer.

3.4.6 iPhone Human Interface Guidelines

In order to get an application published on AppStore a developer has to follow the iPhone Human Interface Guidelines. These guidelines cover what kind of applications can be developed for the platform, general design principles and how these should be implemented. Information about the available components is also provided along with how they can be used and modified

4 Related Work

In this chapter smartphone applications for mobile reporting available on the market today are mentioned.

4.1 Handyman

Handyman is available on smartphones and PDAs running Windows Mobile. It is developed by the Norwegian software company ePocket solutions and is used by 50 percent of all electricians in Norway. Its features include job registration, smart checklists, parts ordering, stock management, timesheet completion, synchronization, display customization, secure backup and restore, document management and industry specific glossary (ePocket Solutions AS, 2010).

4.2 RemoteX Mobile Client

The RemoteX Mobile Client is another application for mobile devices running Windows Mobile, developed by the Swedish software company RemoteX Technologies. Its features include assignment management, time registration, checklists, barcode reading, digital signatures, time and resource planning and plug-in functionality (Remotex, 2010).

Figure 6. RemoteX Mobile Client showing a work order (RemoteX, 2010).

4.3 AddMobile Toolbox

AddMobile Toolbox is yet another application for the Windows Mobile platform, developed by the Swedish company AddMobile AB. It offers features such as time reporting, work order management, material ordering and maps and directions (addMobile 2010).

4.4 iPhone Time Reporting Apps

For Apple's devices running iOS there are several applications for time reporting, for example HoursTracker. Since these applications do not fulfill the rest of the needs for mobile reporting for a service provider, none of them are mentioned here in detail (Apple Inc., 2010c).

5 Methods

This chapter explains the methods used throughout the design process.

5.1 Design Process

Initial studies will be done to learn about mobile reporting on smartphones. Then user studies will be performed to gain deeper knowledge in this area. Sketches of design solutions will be drawn in iterations. Once done, another round of user studies will be performed for evaluating these design solutions and potentially discuss some aspects of mobile reporting further. Then more sketch iterations will be made before implementing the smartphone application. During the development some user studies will be performed. An illustration of the process can be seen below in figure 7. More information is also presented in Appendix D - Time Plan.

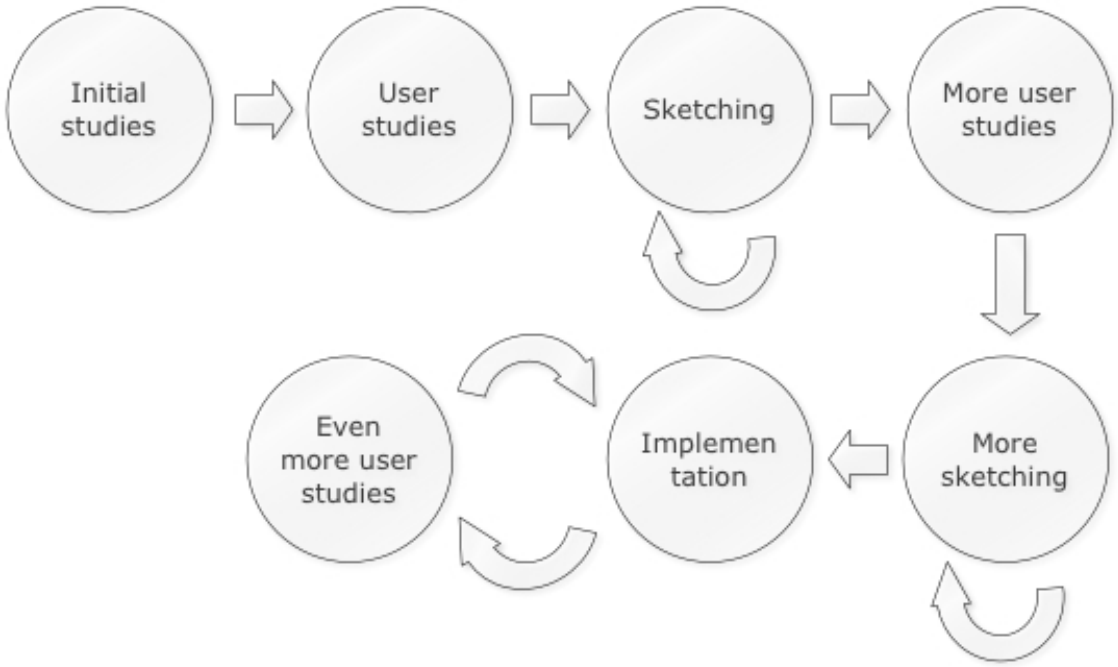


Figure 7. An illustration of the planned design process for the thesis work.

5.2 Secondary Research

Secondary Research means collecting and analyzing already performed research. The output from this method does not come from experiments, interviews or questionnaires. It comes from earlier published research articles, papers and other documents.

5.3 Internet Media Research

The context of smartphones and software development is constantly changing and evolving. Most information is found on news sites, video sites, blogs and message boards. These sources of information and rumors will be followed on a daily basis.

5.4 Cognitive Walkthrough

Cognitive walkthrough is a method that simulates user testing by having a set of experts taking the roles of users. The experts perform certain planned tasks and note down usability related problems. An analysis is then made. An expert could be a usability expert, designer or expert evaluator. The method is suitable for improving web sites and software applications. It can be used on paper prototypes before coding has begun or on a finished prototype (Preece, et al., 2007).

5.5 Prototyping

Prototyping can be done in many different ways and is a popular method for getting an idea of how a product can look like when finished. It is also used for gathering feedback and aid design decisions. The method may or may not involve users.

5.5.1 Paper Prototyping

Paper prototyping is an easy, cheap and fast user-centered usability method for interface design. When performing it participants from the user group interact with paper mockups and perform tasks that the final product will be used for. One organizer will modify the paper version of the interface depending on the user's interaction, but not explain how it works.

A good paper prototype does not look too finished since it may encourage the wrong type of feedback from participants. Good suggestions are to use grayscale colors, no icons and to not be so strict concerning lines and grid. Text should be added for increasing realism.

The method is very useful to perform before implementation begins since it enables major and rapid design changes based on user feedback. Based on personal experience and research by Caroline Snyder (2003 & n.d.) there is always a huge overlap between the results of using paper prototypes compared to high-fidelity prototypes. She explains that she has performed over 100 usability tests and only a few times have high-fidelity prototypes found problems that paper prototypes did not.

5.5.2 High Fidelity Prototyping

Compared to low-fidelity prototypes a high-fidelity prototype looks, feels and behaves like the final product. It is best used late in the development process for providing an accurate image of the product to come. Such a prototype can be used in user tests or just by developers for getting an idea of the look and feel of the final product (Preece, et al., 2007).

5.6 User Studies

Input from former, current and future users of mobile reporting was highly prioritized for making the design concept as usable as possible. Suitable methods were based on information from Preece, Rogers and Sharp (2007) and are presented below.

5.6.1 Interviews

Conducting interviews is a common method for gathering information about users. Interviews may be unstructured, semi-structured or structured. An unstructured interview is exploratory around a certain topic and usually goes into great depth. The questions are open and both the interviewer and the interviewee can control the direction of the conversation. Some topics to cover are planned beforehand.

A structured interview is somewhat similar to a questionnaire. It contains short predetermined questions with closed answer alternatives. This variant is often used when the topics of interest are well covered and understood.

A semi-structured interview is a combination of an unstructured and a structured interview. Preplanned topics are covered in each interview and, when desired, the interviewer probes for more information until nothing new is said.

Interviewing several people at once is called a focus group. A single interviewer assists and documents the discussion among the participants. The underlying idea is that subjects and opinions not mentioned in solo interviews are brought up thanks to discussions and different opinions. Focus groups take more time to set up but usually generate more data. They also have the disadvantage of influential participants fooling the performers into believing the rest are of the same opinion.

5.6.2 Observations

Observations can be performed at any stage during the development cycle; early on for eliciting user requirements, in the middle for evaluating prototypes and design ideas or in the end for evaluating the final product.

Observations may be performed directly or indirectly. When direct, users are observed while performing their activities. When indirect, their activities are recorded and analyzed later. Observations can take place in a controlled environment where the participants perform specific activities under detailed supervision, or in their natural setting where the effect of the natural context can be studied.

Compared to interviews and questionnaires observations can offer more accurate information about the effects of the surrounding context. In interviews participants might leave out important details and not remember information correctly.

5.6.3 Personas

A collection of attributes for a user of a product is called a user profile. To make a user profile more lifelike it can be developed into personas. A persona is a rich description of a typical user, which designers can focus on during product development.

Personas do not describe real people, but are based on real people involved in prior data gathering sessions. Personas are often described in great detail and are separated by their individual goals. Personas also include skills, attitudes, tasks and surrounding environment. This information is presented realistically, in detail and with great imagination.

A persona should also contain information what they enjoy doing in their spare time along with a picture. This information is believable and makes it easier for designers to view personas as real users. It can also make the personas easier to remember.

6 Realization

This chapter gives an overview of how the thesis work progressed from start to finish.

6.1 Initial Studies and Planning

Initially collaboration was planned with one of Binea's contacts. Due to delays and uncertainty this collaboration was put on hold. Instead Göteborg Energi AB and Bravida were approached for user studies. They are both large service providers, have experience with mobile reporting and were co-operative. No other companies needed to be contacted.

In order to identify must-have features for a mobile reporting application available commercial applications were studied. Secondary research was also done on the concept of mobile reporting, smartphone platform market share and technology hostility.

6.2 First Round of User Studies

Once Bravida and Göteborg Energi AB were on board, user studies were performed. At Bravida two employees at their elevator department were visited for a field study and then interviewed. At their electricity department a work leader and smartphone responsible was interviewed. At Göteborg Energi AB a contract director who had been involved with mobile reporting since early 2000's was interviewed. Based on these studies personas were created. These can be viewed in Appendix E – Personas.

Figure 8. Checking out a typical work site for the employees the elevator department at Bravida.

6.3 First Round Of Sketching

Based on the result of the user studies and analysis of available applications requirements for the concept were defined. Most of the features could be related to information or economy. The information related features were divided into general information about a work order and address bound work history. The economy related features were divided into equipment handling and time reporting. Other features included attached files and checklists.

Sketching started on a whiteboard for getting a quick feel and flow of an application. Sketches were drawn of a home screen, a list of work orders and a selected work order. The sketches were then refined on paper and finally refined in vector graphics applications. Design advices by Tidwell (2005) were followed, which are presented in Appendix A – Design Manual.

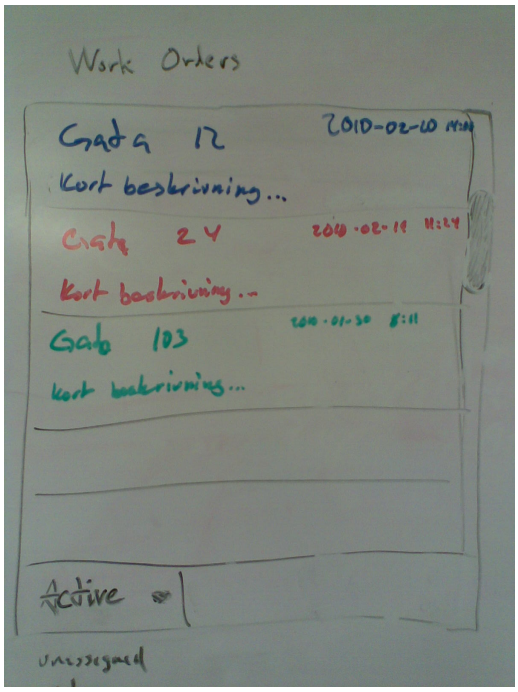


Figure 9. Early sketch of the color-coded work order previews.

Since Windows Mobile 6 was the likely platform to develop for at this stage the initial sketches were designed with stylus input in mind. Later iterations were however made more multi-touch friendly. For example, one time reporting sketch displayed a full week, like in figure 10, and time intervals were set by tapping and dragging. This solution was scrapped due to potential loss of precision when working without a stylus.

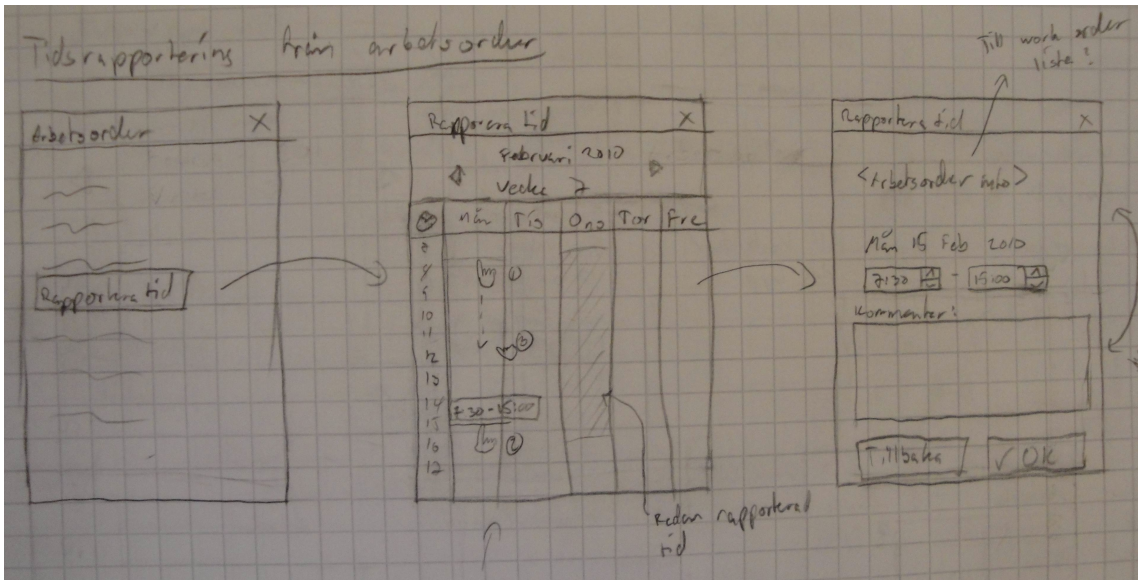


Figure 10. Early sketch for setting time intervals for a time report.

Time reporting was accessible both from the home screen as well as from a selected work order. This was done since sometimes the user might want to report time for multiple work orders on the same day and sometimes for the same work order on multiple days.

Equipment handling was also accessible from the home screen and from a selected work order. When accessed from a work order the user would be able to move items between the car, local office storage and the work order. When accessed from the home screen the available material would be browsed as a catalogue. Access to an online store would also be available.

History was first a simple list of previous work orders but later turned into a feed of time reports, related work orders, material deliveries and general comments. Finally, it was laid out as a list of information of finished work orders with the same address and general comments. The Facebook inspired feed might have presented the user with too much information.

6.4 Second Round of User Studies

After the first round of sketching another round of user studies were scheduled for evaluating and improving the design solutions. An interview in combination with a paper prototyping session was held with our contact at Bravida's electricity department.

Two cognitive walkthrough sessions were also held. One with two colleagues at Binea and one with two interaction design students at Chalmers University of Technology.

Figure 11. Sketch cards used for the cognitive walkthroughs. They were scaled down to the size of a typical smartphone screen.

6.5 Second Round of Sketching

Based on the feedback gained from the second round of user studies the concept sketches were redesigned further. The changes included the ability to tap the navigation header of a selected work order to toggle between showing its address and ID as well as a button for taking a picture and attaching it to a work order.

Time reporting got a large makeover. Due to low interest, the option of reporting time for a certain work order on several days was scrapped. It was also noted that a time report must be able to contain one or more time intervals with individual time types, such as regular office ours and overtime.

This led to a shortage in screen space so something other than sliders had to be used. Two solutions were created; one using drop-down menus and one using slot machine wheels, which can be seen in figure 12 and 13.

Figure 12 and 13. Drop-down menu solution and slot machine solution.

The checklist, previously having shown each item in a separate screen with multiple answer choices, was deemed too complex. It was learnt that a checklist point only has to be checked or unchecked so the design could be simplified into a list of check boxes with single lines of text.

6.6 Delimitations

This part covers the delimitations that were made before the application prototype implementation was started.

6.6.1 Prototype Delimitations

A decision was made to only focus on the handling work orders and time reports for the prototype. According to the performed user studies these are the most important features. They are also more generally applicable than handling and ordering of equipment and material, which have to be tailor-made for different service providers.

6.6.2 Choosing Development Platform

After discussions with Binea, iOS was chosen as implementation platform. The reasons were the following:

- It is attractive from a usability point of view thanks to its smart UI solutions.
- According to Binea it is becoming a strong platform on the business market.
- Binea are optimistic about more iPhone projects in the future.
- A possibility to develop a similar iPhone app for a customer presented itself.
- We were personally very interested in developing for the platform.

The initial choice of Windows Mobile 6 was ignored due to its limited future. Windows Phone 7 will be released by the end of 2010 marking a new path for mobile computing at Microsoft. Despite its promised continuous support and current popularity on the business market it was considered uninteresting.

Windows Phone 7 was briefly discussed. It looks promising and offers an exciting UI. However, it was ignored since Binea expects it to take several years before being a strong competitor on the business market.

Android was also considered. From a usability point of view it is also attractive and it is looking to rapidly increase its market share in the following years (Practical eCommerce, 2010). It was ignored since it is not yet as popular on the business market as iOS according to Binea.

6.7 Implementation

This part covers how the implementation of the application prototype progressed.

6.7.1 Choosing Method For Storing Data

For storing data several options were considered. These are presented below along with the reasons for the final choice.

6.7.1.1 SQLite

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. Unlike most other SQL databases, SQLite does not have a separate server process. SQLite reads and writes directly to ordinary disk files. The code for SQLite is in the public domain and is thereby free to use in both commercial and private applications (SQLite, 2010).

6.7.1.2 XML

XML stands for Extensive Markup Language and is a simple text-based format for representing structured information such as documents, data, configurations and others (World Wide Web Consortium, 2010).

6.7.1.3 Core Data

Core Data is a part of the Cocoa Touch API and is an object-graph management and persistence framework. It organizes the application's model layer into a set of defined in-memory data objects. It then tracks changes made to these objects and reverse changes on demand, for example when the user performs an undo command. When changes in the application's data are to be made, Core Data takes care of archiving the objects to a persistent store. It also saves data into regular files that can be managed with the Finder in Mac OS X (Apple Inc., 2010d).

6.7.1.4 Choosing SQLite

For storing data the choice fell on using a SQLite database. Based on previous experience with Myself and Oracle there was no doubt that it would satisfy our needs and be smooth to use.

Core Data has received positive reception on Internet message boards for iOS development (Stack Overflow, 2009), but was ignored for several reasons. First, it is not as multi-platform friendly as SQLite. Core Data is a part of Cocoa while SQLite is written in pure C. If the application is to be ported to another platform rewriting SQLite related code would be much less time consuming. Second, SQL databases are more common in the industry than Core Data. Less code can be written when working with a company database. Third, at the time the overhead for learning the framework seemed to be too time demanding.

XML was also briefly considered. Data is directly entered in the XML file and the syntax is simple and powerful. However, its disadvantages outweigh its advantages. When selecting certain data the entire content of an XML file has to be read into the iPhone's memory. When updating data the entire content has to be loaded, updated and then re-written. This makes XML a poor choice for storing dynamic data.

6.7.2 Learning iOS

Roughly 3 weeks were spent on learning iOS before implementing the prototype.

A course held by Stanford University proved to be very useful. It was held in the months of January, February and March in 2010 and each lecture was made available, free of charge, in the form of video recordings and notes through iTunes U. Its content was used in combination with books, message boards and reference documentation (Stanford University, 2010).

6.7.3 Developing The Prototype

Roughly 6 weeks was spent on developing the application prototype, excluding the 3 weeks spent on learning the platform.

On a few occasions simple user tests were performed with colleagues at Binea and students at the Interaction Design master's programme at Chalmers University of Technology. Due to time constraints no specific methods were used. The participants were simply allowed to test different iterations of the application prototype and provide feedback, which was then used when adjusting the GUI.

7 Result

This chapter covers the results of the thesis work. More specifically the user studies, mobile reporting concept, application prototype and design manual.

7.1 User Studies

This section mentions what was elicited from the user studies. The full-length conclusions can be found in Appendix C - User Studies.

7.1.1 Bravida - Elevator Department

Two employees were visited at a work site for a field study and interview. A mobile reporting application had not yet been introduced at their department. However, both of the employees had prior experience of mobile reporting at previous employments.

The field study resulted in information about typical working conditions. The interview gave information on improvements of the current time reporting process, must-have features for a reporting application and demands on smartphone hardware.

7.1.2 Bravida - Electricity Department

An interview and a paper prototyping session combined with another interview were held with a supervisor, who was also responsible for the smartphones.

The interview resulted in information about typical work tasks and working conditions, staff member characteristics, current and prior mobile reporting applications and hardware issues. A demonstration of their current application, Handyman, was also given.

The paper prototyping session and interview resulted in lots of feedback and suggestions for improvement on the design solutions. Advice about the challenges and difficulties of implementing equipment handling and ordering were also given.

7.1.3 Göteborg Energi AB

An interview was held with a contract director who had great insight into the company's mobile reporting approach. Resulting information painted a clear picture about the importance of good smartphone hardware. Information was also gathered about typical working conditions, staff member characteristics and how new technical solutions could be accepted among skeptical employees.

7.1.4 Cognitive Walkthroughs

In order to get expert feedback on GUI solutions two cognitive walkthroughs were held.

7.1.4.1 Colleagues at Binea

Both participants had extensive experience of using smartphones and were familiar with the concept of mobile reporting. The walkthrough resulted in some great ideas of GUI improvements. A few potential conflicts with Apple's design guidelines were also mentioned.

7.1.4.2 Interaction Design Students

Both participants had extensive experience of using smartphones, but were not familiar with the concept of mobile reporting. The walkthrough resulted in lots of detailed suggestions for GUI improvements. Apart from the sketches insights were provided on the behavior such as the use flow of the application.

7.1.5 Technology Hostility

Some information about technology hostility was elicited from the interviews and field study. Hardware seems to be the main problem. Field personnel have a lot of experience with smartphones constantly breaking, losing Internet access and running out of battery.

Poor UIs is another problem. Today's popular applications running on Windows Mobile require the use of styluses, which are easily broken or lost. Even with them the GUIs are experienced as difficult to navigate due to the tiny elements and complex structures.

Another reason for this hostility is the ease of traditional field reporting. Pen and paper is easily used and is something that all employees have a lot of experience in.

Finally, older personnel are often quite stubborn. Not specifically concerning new technical solutions but in general. Younger employees are often more interested in technology and accept new technical solutions more easily.

7.1.6 Concept

This chapter briefly explains the parts of a mobile reporting UI concept. More details including design solutions can be viewed in Appendix A - Design Manual.

7.1.7 Structure

A graphical overview of the concept is presented below in figure 14.

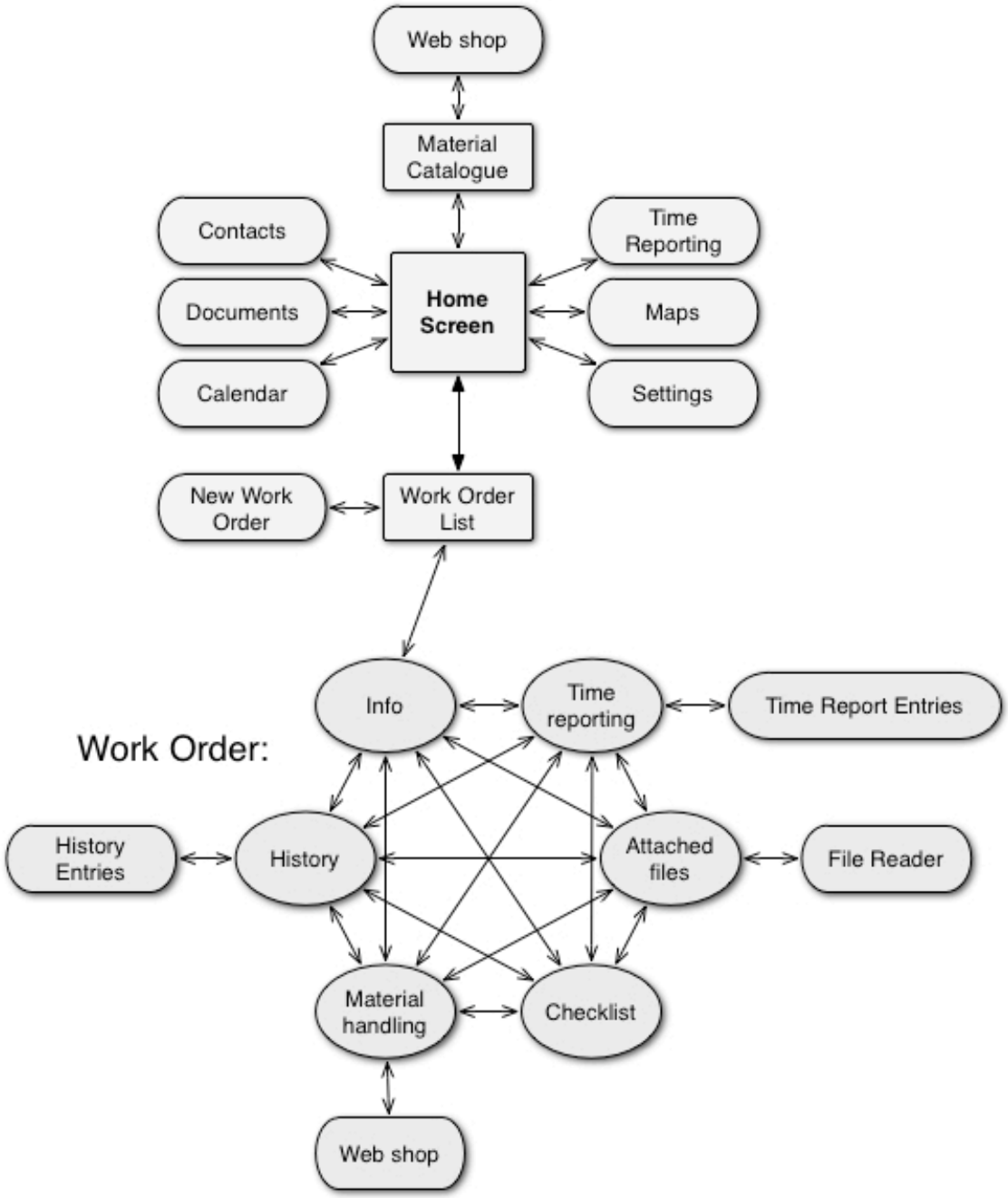


Figure 14. A flow chart of the mobile reporting concept.

7.1.8 Home Screen

The home screen functions like a launch pad for the features of the application. The user is able to reposition icons and add shortcuts to other installed applications. By default features include the work order list, time reporting and equipment catalogue. Apart from features directly related to mobile reporting, shortcuts can also link to standard smartphone features like calendar, address book and mapping service.

It should also be possible to reach the settings menu from the home screen. Here the user can configure typical settings and log out if a colleague needs to use the smartphone with individual settings and data.

7.1.9 Work Order List

The work order list presents a list of previews of all the work orders assigned to the current user along with work orders not assigned to anyone. Each preview is color-coded by priority and contains the address of the work order, a preview of the description and the entry date.

At the top of the list there is a search bar where the user can filter the work orders. For example by entering a part of an address. At the bottom of the list the user can switch between different categories of work orders. In the final iteration these categories are work orders assigned to the current user and work orders not assigned to anyone.

7.1.10 Work Order Sections

When a work order preview is selected in the work order list, the full work order is presented. For structuring the information the concept of a tab bar is used at the bottom of the screen. This is a bar with buttons, where each presents one section.

7.1.10.1 Info

This section provides the user with basic information about the work order including address, problem description, current status and priority as well as a list of associated people. These people are listed either as contact persons, work leaders or employees. Selecting a person will display contact information about that person. If desired, the person can then be given a call or sent a text message or e-mail.

At the bottom of the list of assigned employees there is a button for assigning additional employees. Selecting it launches the smartphone's address book from where the user can select whom to add. The person being added will automatically have the order assigned to him as well. If the current user is not assigned to the work order the button will be replaced with a button for adding himself.

The status can be changed when needed. When doing so the customer will be notified through a text message or an e-mail.

7.1.10.2 History

This section provides a feed of work related information associated to the address of the selected work order. Notes will be posted along with information from old work orders sharing the same address. Notes are written by users at the top of the view in a text field.

7.1.10.3 Time

This section lists all time reports for the selected work order. Each list entry shows how much time was reported for including a comment of what was done in that interval.

When editing a time report or creating a new one the date can be changed using a pop-up calendar which appears when the label displaying the current date is tapped. Back and forward buttons on each side of the label selects the next or previous day. Below the date is a text field for entering a short description of what was done.

Finally, the time intervals are set. Each time interval has a time type and the spent time specified in hours and minutes. An alternative solution using a start time and a stop time would remove the need for setting the time type.

7.1.10.4 Equipment

Here equipment is virtually moved between the selected work order, the user's car and the storage at the local office. The tab bar concept divides these sections and a search field is available for finding desired items.

When needed, a web store can be accessed for ordering new gear, which can be delivered to the local office or to work order address. Items can be added manually through barcode scanning or by typing their article numbers.

7.1.10.5 Files

In this section all files attached to the selected work order are listed. For viewing or editing them the user simply taps them. It is also possible to add existing or newly created files the work order.

7.1.10.6 Checklist

Here a checklist is presented that must be filled out before the work order can be finished. Service must include a checklist for a work order for maintaining ISO standards and keeping a high level of quality. When all points are checked a button at the bottom of the checklist is made clickable for finally finishing the work order.

7.2 Application Prototype

This section describes the structure of the application prototype, which features it has and how they work. How data is stored and which standard and custom components have been used is also presented.

The prototype was named Trawoh, an abbreviation for Time Reporting And Work Order Handling.

7.2.1 Structure

A graphical overview of the prototype structure is presented below in figure 15.

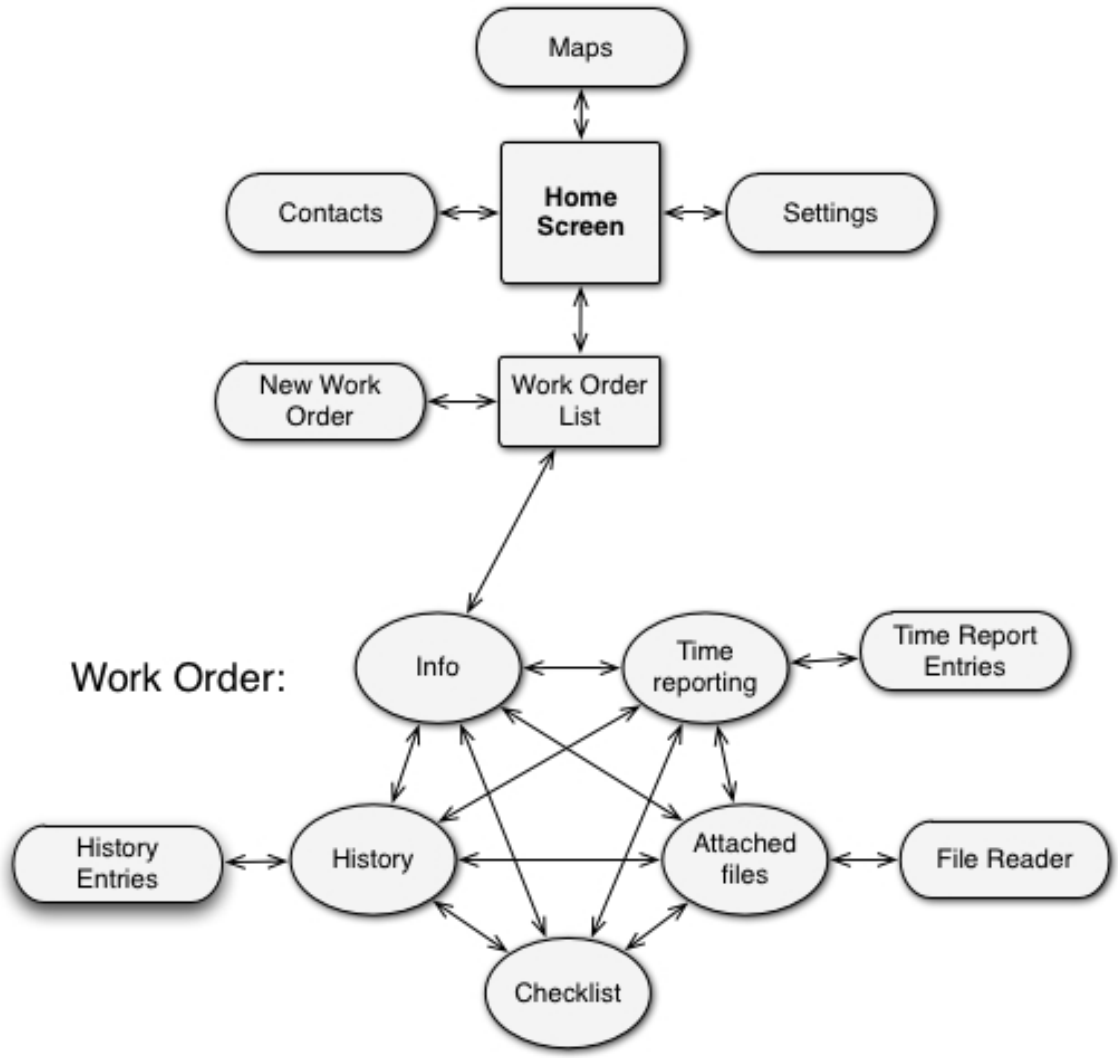


Figure 15. The GUI structure of the prototype. Compared to figure 14 the home screen links to less features and a work order has no section for material handling.

7.2.2 Home Screen

The home screen of the prototype provides access to Google Maps, the work order list, the iOS address book and the settings menu.

In the settings menu the current user can be changed and the default contacts can be re-read from the database if needed to.

Figure 16 and 17. The home screen and the settings screen.

7.2.3 Work Order List

This view lists all work orders. The work orders are either assigned to the current user or not assigned to anyone and can be filtered via the toolbar. An arrow at the right end of each work order indicates that tapping will present the complete work order to the user.

If the user wants to quickly find a specific work order its address (or some parts of it) can be entered in the search bar.

Figure 18 and 19. The list of work orders. The view for a not yet filled out work order.

7.2.4 Creating New Work Orders

When creating new work orders the parameters that can be set are the description, address, priority and contact persons, work leaders and employees.

7.2.5 General Information

This view contains the work order ID, address, access to Google Maps, description, status, priority and assigned contact persons, work leaders and employees. Since iPhone OS 3.1.3 does not offer multitasking, launching Google maps means terminating Trawoh. The user is notified about this and the state of the application is saved for next launch.

Figure 20 and 21. The general information for a work order.

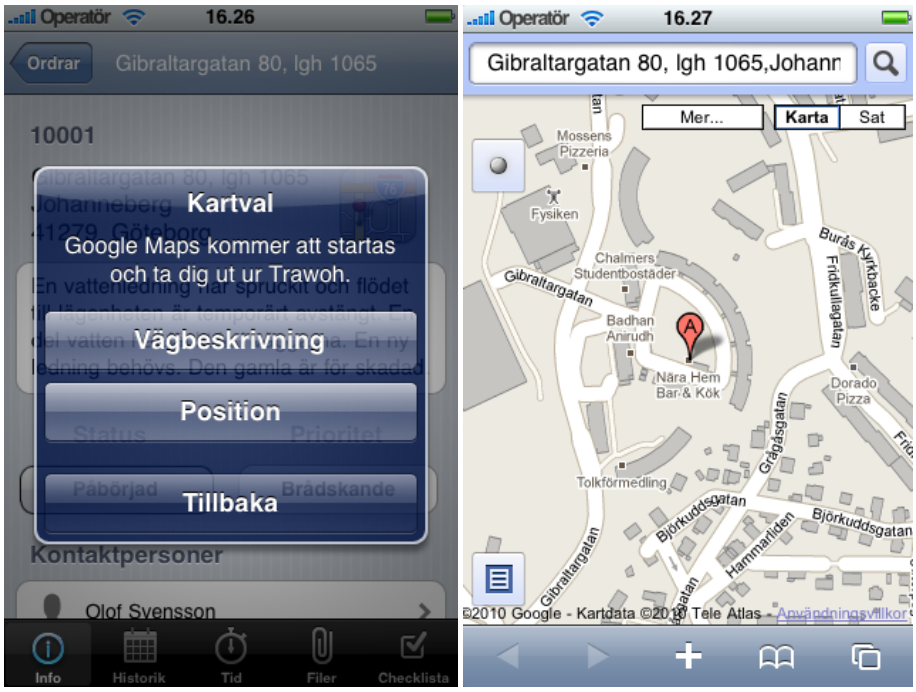


Figure 22 and 23. When Google Maps is launched the user can choose between viewing the location of the work order or a route description from his current position to the address.

7.2.6 Work Order History

This view presents all notes and descriptions of finished work orders associated with the same address as the current work order. Notes are editable but only by their authors. Tapping an author of a note will launch that person's contact profile.

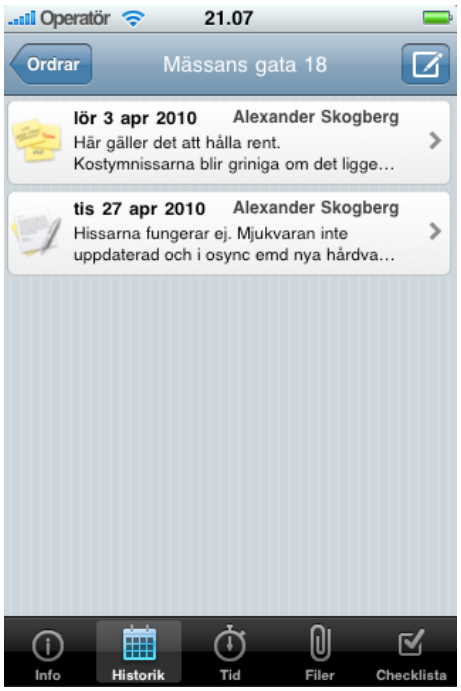


Figure 24. Viewing the list of history entries

Figure 25. Viewing a history entry in detail.

7.2.7 Time Reporting

In this view all time reports for the selected work order are shown. When creating a new report, a comment can be entered along with one ore more time intervals. A time interval consists of a time type and the amount of time. The intervals are editable, but not the date of the report.

Figure 26 and 27. Viewing time report entries and creating a new one. Date is set using the right and left arrows or clicking the middle button.

Figure 28 and 29. Setting time type and time for a time interval. These pop-ups are shown when the corresponding buttons (shown in figure 27) are tapped.

7.2.8 Checklists

A work order has a set of checklist points that must be marked before finishing the order. Once finished, the work order is removed from the work order list.

Figure 30. The final point has been checked and the order can be finished.

7.2.9 File Viewing

A work order can have a set of attached files that are viewed when tapped.

Figure 31 and 32. Selecting and viewing an assigned file.

7.2.10 Address Book Access

In addition to the start menu in iOS the address book can also be accessed from the home screen in the application. Contact profiles of people assigned to a work order as employees, work leaders, contact persons or note authors can also be accessed.

Figure 33 and 34. Accessing a contact profile from a work order.

7.2.11 State Saving

When the applications is shut down through the iPhone's home button or by launching Google Maps (no multitasking available on iPhone OS 3.1.3) the state of the application is saved. Next time the application is launched the user will be taken back to the last active screen before the prototype was terminated.

7.2.12 Data Handling

For storing data a SQLite database is used. Its design was based on the information gathered from the user studies. An entity-relationship diagram of the database can be seen below in figure 35.

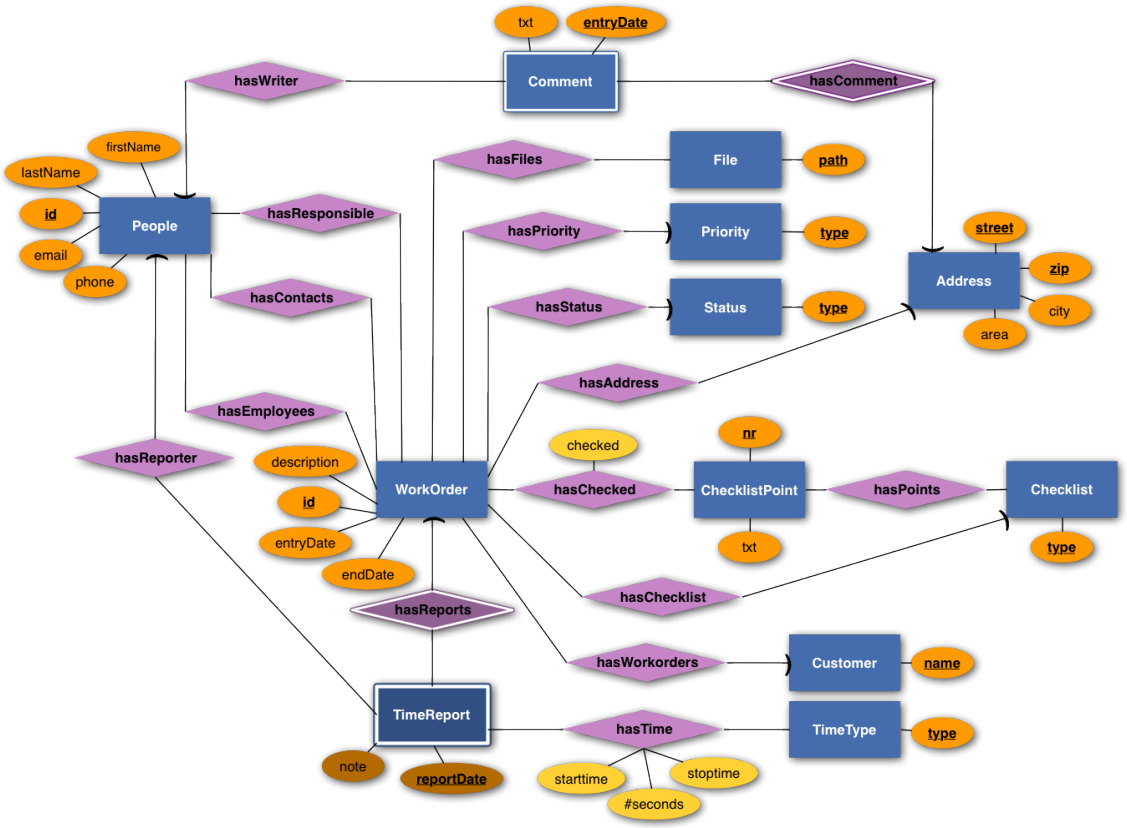


Figure 35. The design of the database used in the application.

A work order has a unique ID number, a problem description, a date for when it was created and an end date for when it was finished. It also has a status and a priority property for explaining how urgent it is. It also has an address, which has a street address, postal code, city area and city.

A work order can contain one or several people who are assigned as employees, work leaders and contact persons. A person has a unique ID number, first name, last name, email address and phone number.

A work order also has a checklist type. Depending on its type a checklist has different checklist points that consist of a unique number and some text.

A note is a comment that is associated with the address of a work order. When handling a work order the user can add a note with useful information. This information will then be associated with every future work orders sharing the same address. Each note has one person assigned as its author.

A work order can have several time reports. A time report has a date for when it was entered, a comment for what was done and at least one time type. Associated with a time report and its time types is the amount of time spent per time type.

Finally, a work order can also have files attached to it. A file entry only has the search path to the file.

7.2.13 GUI Components

The standard and default GUI components used in the application can be found in Appendix B - Components.

7.3 System Architecture

This section describes the architecture of the application prototype.

7.3.1 Overview

The prototype was developed using the Cocoa Touch framework. Cocoa Touch is based on the model-view-controller philosophy, which means that there is three object categories:

- Model objects representing data.
- View objects that can be considered the visual representation of the data.
- Controller objects that handle logic and the connections between the model and the view.

In Cocoa all views are subclasses of UIView and all controllers are subclasses of UIViewController.

The most common way to handle navigation between views is by using a UINavigationController onto which views are pushed. It provides a header with a title and a button for easily navigating back to the previous view. Navigation can also be handled using a UITabBarController. It holds different views, which can be accessed from a tab bar. The prototype puts a UITabBarController inside a UINavigationController.

The majority of the views in the prototype are UITableViews. This view scrolls vertically and consists of separate cells that can be divided into titled sections. The prototype subclasses the standard class UITableViewCell for achieving the desired look-and-feel for each cell.

Since almost all of the views in the prototype use data from a WorkOrder object the class WOTableViewController was written. It is a subclass of UITableViewController containing a pointer to the currently selected work order as well as common logic for loading UITableViewCells for its UITableView. A majority of the view controllers in the prototype inherit from this class.

UITableViewCells are unloaded and cached when scrolled outside of the screen. Since it is impractical to store them in an array each WOTableViewController contains a list of WOTableSection objects, which keep track of the section title, the number of rows in the section and the height for each and finally what kind of custom UITableViewCell a row contains.

7.3.2 Model

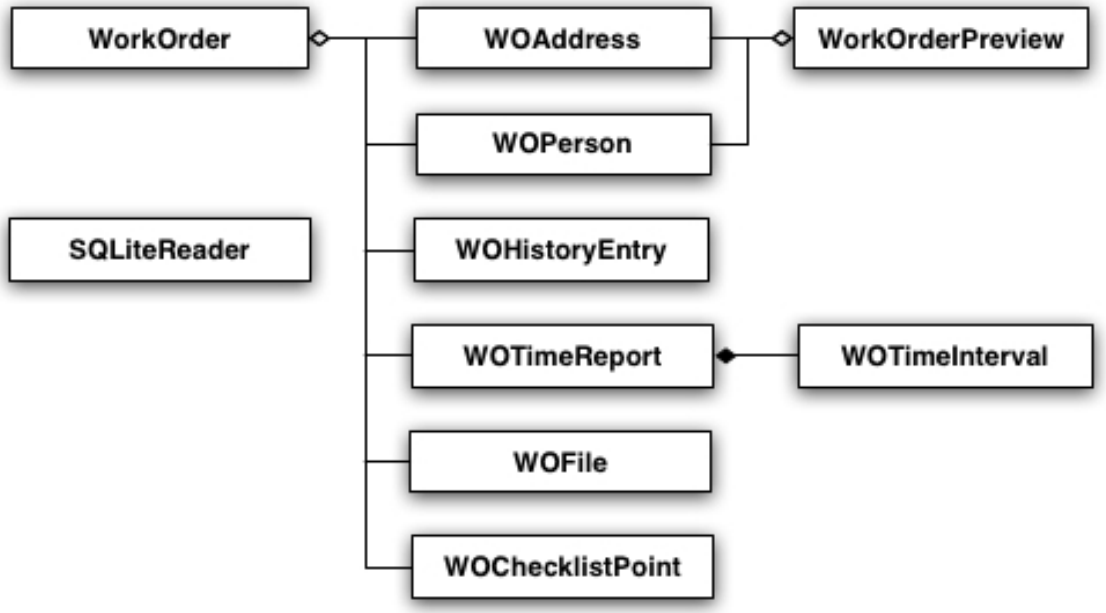


Figure 36. The hierarchy of the utility objects.

The main object in the prototype is the WorkOrder object. It holds all information for a work order. The similar object WorkOrderPreview contains the information displayed in the work order list view. When launching the prototype an array of work order preview objects are instantiated. When one is selected the WorkOrder object is loaded and used in most views of the prototype.

The WOAddress object represents addresses. It holds a street address, city area, postal code and city.

A person is stored in the WOPerson object. It contains an ID, first and last name, phone number and email address.

The WOTimeReport object stores time report and contains the date of the report, a comment and set of time intervals stored in WOTimeInterval objects. A WOTimeInterval object contains a time type and the duration in seconds.

A history entry is stored in the WOHistoryEntry object. It contains the creation date, a comment, an author and a type specifying its content. A history entry can either be a note or a work order summary. Users write notes while work order summaries are derived from finished work orders sharing the same address as the selected work order.

An attached file is stored in the WOFile object. It contains the file's name, URL (all files are stored online) and file type. All information is extracted from the URL.

For keeping track of marked checklist points an array of WOCheckListPoint objects is used. A WOCheckListPoint object contains the index and text for a point along with a boolean specifying whether it is marked or not.

The SQLiteReader object contains a connection to the application database as well as methods for getting and updating work orders and saving created work orders.

7.3.3 View Controllers

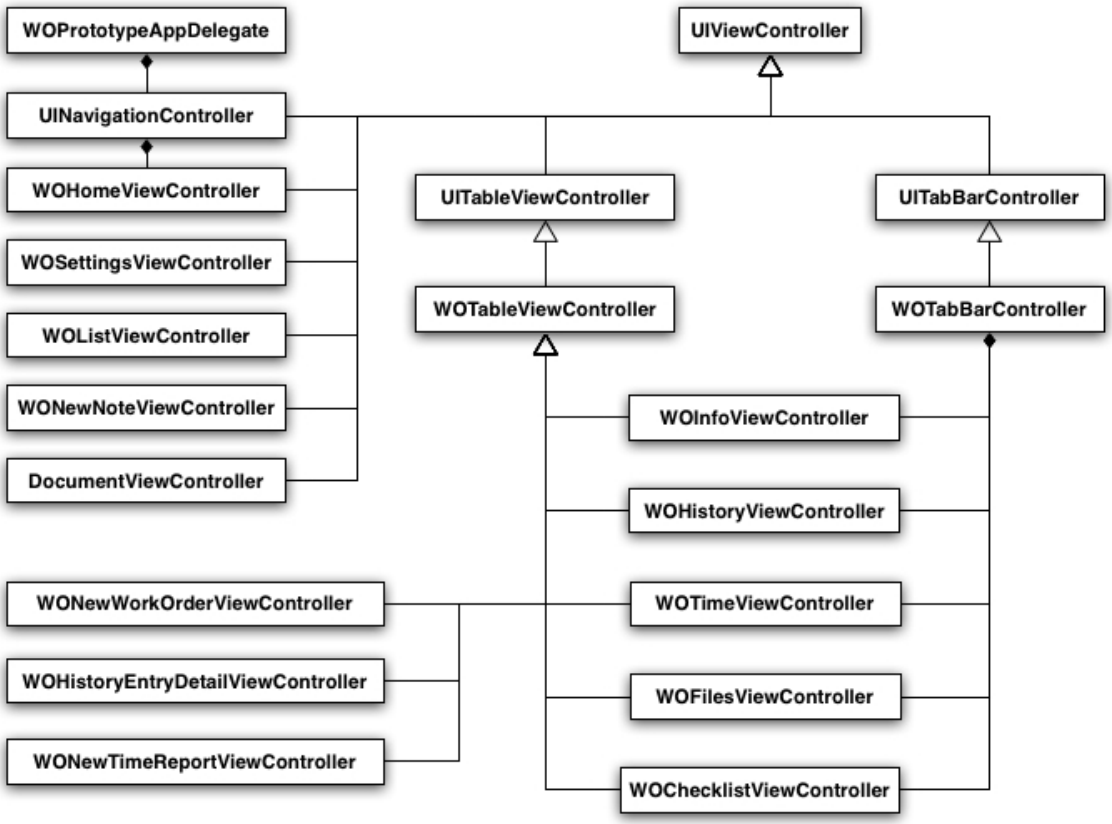


Figure 37. The hierarchy of the view controllers.

The home screen is named WOHomeViewController. Its view has UIButtons for launching the features of the prototype and it is the root view controller of the prototype's UINavigationController.

Settings are handled in WOSettingsMenuController. It contains UIButtons for changing the current user and reloading the default contacts from the database.

The work order list is presented in WOListViewController. It has a custom view containing a UITableView and a UIToolbar. The search feature was implemented using a UISearchDisplayController and a UISearchBar.

A selected work order is presented in WOTabBarController. It inherits UITabBarController and contains code for passing along the selected work order and loading its nested view controllers. It also contains code for custom animations for hiding and showing its UITabBar.

The WOTabBarController object holds view controllers for displaying and handling general information, history entries, time reports, attached files and the checklist. These are all subclasses of WOTableViewController and contain specific code for setting up their respective cells. They are:

- WOInfoViewController
- WOHistoryViewController
- WOTimeViewController
- WOFilesViewController
- WOChecklistViewController

The WONewTimeReportViewController is used for creating a new and viewing existing time reports. It has a custom code for its editable state. This was done because the built-in functionality in a UITableViewController is not flexible enough for editing its UITableView when working with custom UITableViewCells.

For displaying history entries the WOHistoryEntryDetailViewController is used. It contains a section for the text entry and one for the author.

The WONewWorkOrderViewController is used for creating a new work order. The address is entered in several UITextFields and the priority is set by pushing a UIButton and selecting the priority from a UIActionSheet. WOPersons are assigned as contact persons, work leaders and employees by selecting them from the iOS address book.

For entering text entries the WONoteViewController is used. It contains a UITextView, covering the top half of the screen to make room for the keyboard. It is pushed onto the UINavigationController when an editable text cell is tapped.

7.4 Design Manual

The design manual focuses on GUI design for native applications on smartphones where user input is done using a multi-touch enabled touch screen and an on-screen keyboard.

More specifically the manual offers design patterns for interface navigation, user input and graphical design. Each design pattern is defined along with information about when and why it should be used. One example per pattern is also included. Included is also a case study of mobile reporting with design solution for its must-have features.

The design manual can be found in Appendix A – Design Manual.

8 Discussion

This chapter discusses the result of the thesis work as well as the design decisions and the work process.

8.1 Evaluation Against Goals

Binea are satisfied with the result of the thesis work. The studies of available mobile reporting applications along with the user studies of service providers painted them a clear picture of what features such an application must offer and how they should be designed.

Information on technology hostility proved interesting. Older men without much experience of technology are hard to persuade. They are, however, stubborn in general and not just concerning new technology. Poor smartphone quality rather than software issues proved to be the main source of annoyance. Unfortunately very few women worked at the examined service providers so no study was made on the difference between genders.

The design manual offers good basic knowledge of smartphone GUI design and points its readers in the right direction for further studies. The design solutions for the features of a reporting application are well presented.

According to Binea the iPhone application looks great and implements the reporting concept well with smart GUI solutions. The application proved to be much more complete than they expected and will likely be used as a basis for further development.

The only shortcoming of the thesis work was the lack of a final evaluation study of the application prototype. However, a study of this kind was not expected within the time frame.

8.2 Planning And Work Process

The initial time plan proved to be realistic and only had to be modified slightly. No major miscalculations were made and decisions were always based on thorough research.

8.3 User Studies

The user studies went well. We always came prepared and scheduled enough time. However, taking notes for recording data was not always optimal. The participants often had to be asked to repeat themselves and it is likely that some useful bits of information were missed. When interviewing Bravida, video was recorded using our laptops. If possible in future user studies laptops or a small digital camera will be used for recording audio or video for making sure no important information is missed.

8.4 Omitted Methods

Using indirect observations in the form of diaries was considered. In this method the participants take notes of their everyday activities and thoughts. Most of the workload is put on the participants and no special equipment or expertise is needed. However, participants need to be reliable and their memories are easily distorted (Preece, et. al, 2007).

The method was omitted due to time restrictions. Despite our workload being light, the time needed for setting up the method, waiting and gathering and analyzing the data would consume too much time. We were also skeptical whether the method would work well in practice and generate as rich data as interviews, which we had prior experience of.

8.5 Designing For Users With Different Skill Levels

Since a mobile reporting application is going to be used daily the users either already are or will quickly become expert users. The concept and application prototype are therefore designed with expert users in mind. At the same time, in order to overcome technology hostility the application needs to be as easy as possible to learn.

The design advices for different experience levels mentioned in chapter 3. Theory should be considered but are not always suitable for smartphone platforms. For example, keyboard shortcuts for expert users are useless on smartphones without a physical keyboard, and using tooltips as a way to give hints to beginners is challenging since the concept of hovering over something does not exist yet on touch screens.

The limited screen space of current smartphones requires new thinking for GUI design. Labels should not be overdone since they take up much space. It is better to focus on descriptive icons since buttons need to be large anyway in order to be easy to touch. Design patterns like Wizard and One-Window Drilldown (Tidwell, 2005) are more suitable than having everything on a single screen. Although they can be percieved as cumbersome to long-time users, regular cognitive walkthroughs make sure that they are as streamlined and flexible as possible.

One way to please expert users is to offer many ways of doing the same thing. This way the user can select the most efficient one depening on the circumstances. For example, in the design concept developed in this thesis time reporting can be done in two ways. Either by first selecting a work order and then which days to report for, or by selecting a date and then which work orders to report for.

In the end, the limitations of iOS and Apple's design guidelines force a well-balanced design and make the issue of experience levels less important compared to when designing software for desktop computers.

8.6 Application Prototype

In this section the design of the application prototype is discussed.

8.6.1 System Architecture

The goal of the system architecture of the application was to make it as dynamic as possible. It should be easily customized and adapted to needs of different service providers, down to the level of their users.

Different service providers might need to display information in different ways. For example, by rearranging the order of GUI elements and combining views. Ideally it should be possible to specify what information to show and where without recompiling. Dynamicity also forces a heavy object-oriented structure, which makes the code more readable and easily extendable.

The UITableView lends well to this idea with its concept of UITableViewCells. The order of the cells can be rearranged and custom cells can be created for displaying information differently. Thanks to this feature most of the views in the application are UITableViews.

Defining tabs and tables in an XML or database file could be suitable for fulfilling needs of different customers. However, in combination with the automatic caching of UITableViewCells there is no straightforward way of doing this.

By default a UITableView first asks its controller for the number of sections it has, how many rows each of them contains and the height of each row. Finally it asks for the UITableViewCell for each row that is visible on the screen. When a cell is scrolled outside the screen it is released which means that the table view's controller has to set up all connections and enter its data again when it reappears.

As in tutorials and sample code, the logic for doing the above was placed in each UITableViewController. This resulted in unstructured code as the complexity of the cells grew. For abstracting some of the code the WOTableSection class was written. However, the table view controllers still handle the setup and caching of cells along with updating the object containing the current work order. A completely dynamic solution would have some kind of cell controllers handling all of this and there would be no need for subclassing WOTableViewController.

8.6.2 Ignored Features

None of the planned features were completely removed from the concept. The ones not implemented are presented in chapter 9. Future Work.

8.6.3 Deviations From Sketches

Overall, the GUI of the application is very similar to the final concept sketches. The few differences are discussed below.

8.6.3.1 Navigation

During the sketching process there was uncertainty about the placement of the back button in the navigation header and how far back it should take the user. The design was inconsistent. Sometimes the user would be brought back to the previous view and sometimes all the way to the home screen.

According the iPhone Human Interface Guidelines (Apple Inc., 2010b) when using a UINavigationController the back button shall be placed in the top left corner and take the user one step back. In the top right corner a button usually deals with actions such as creating, deleting and saving.

8.6.3.2 Selecting Date

Initially a calendar was to be used for date selection. However, a GUI element of this kind is not provided by default in iPhone OS 3.1.3. Instead a UIDatePicker was used.

A calendar element is superior compared to a UIDatePicker since it demands a lower amount of user actions for selecting a date. When using a UIDatePicker the user has to scroll some of the three columns displaying year, month and day. Using a calendar the date is selected using a single tap assuming the wanted month is displayed. There does exist custom calendar GUI elements for iOS, but since these are not free they were ignored.

Figure 38 and 39. The final sketch and the actual implementation of selecting a date.

8.6.3.3 Selecting From Lists

Initially drop-down menus were planned for actions such as filtering and setting status of work orders. Since iOS does not offer any similar GUI component the UIActionSheet and UIAlertView components were used. They are more suitable thanks to their size, attention grabbing design and extras on demand functionality.

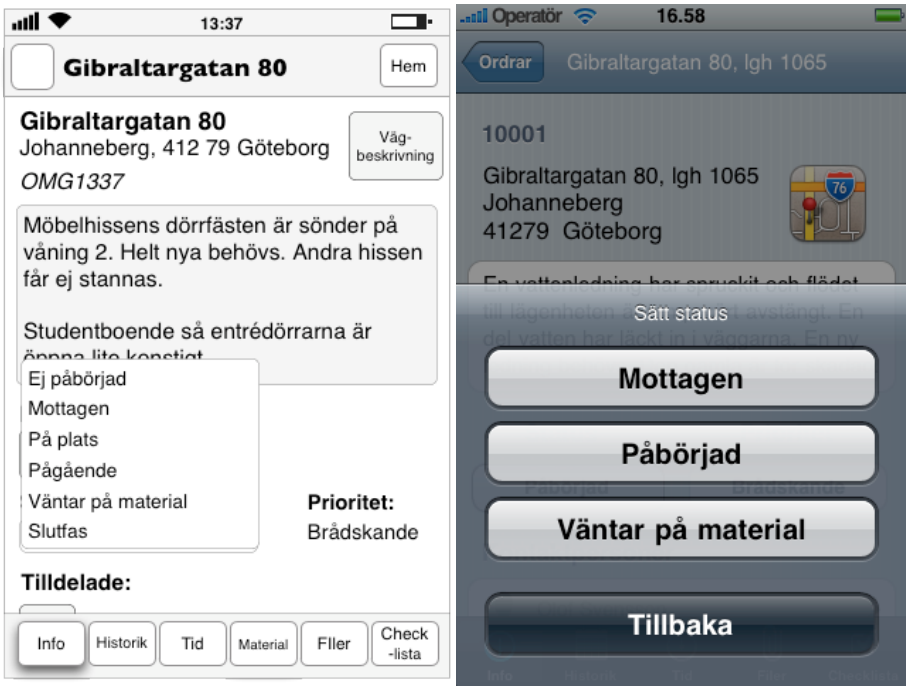


Figure 40 and 41. Setting the status of a work order in the final sketch and in the prototype using a UIActionSheet.

8.6.3.4 Creating A History Entry

Inspiration for creating new history entries came from Facebook. On top of the list of history entries the note would be entered in an expanding text field with a button for posting. Due to problems auto-resizing the text field this was ignored. Instead a new view is pushed displaying a large empty text field and the on-screen keyboard. This design solution also follows the common design style for having a prominent button for finishing an action in the upper right corner of the screen.

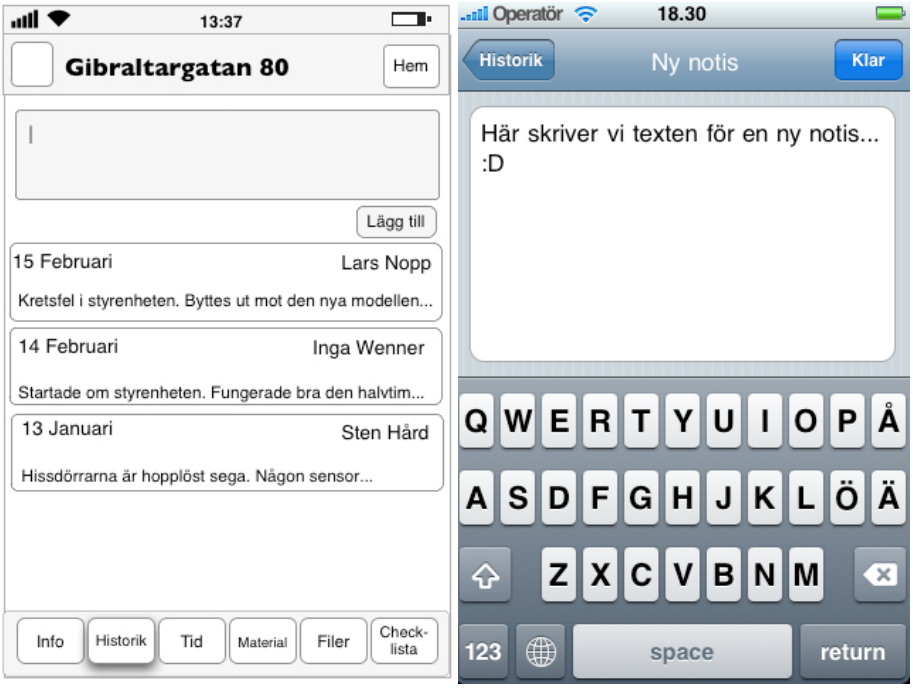


Figure 42 and 43. The final sketch and the actual implementation of creating a new history entry.

8.6.4 Data Storing Method

Using a SQLite database for storing data proved, as expected, to be a good choice. There were only two minor drawbacks. First, the length of the code was longer than expected and took some time to structure. According to prior studies using Core Data would have produced shorter code but this advantage is still outweighed by its earlier mentioned disadvantages. Second, debugging was tedious due to none or hard to interpret error messages often caused by silly syntax faults. Time should have been spent on properly learning the debugging tools bundled with Xcode.

8.6.5 Dealing With New Developments

During the design process lots of exciting developments took place on the smartphone market. Microsoft's new smartphone platform Windows Phone 7 was announced, Hewlett Packard purchased Palm and Samsung announced the release date of their first phones running the new platform bada.

Most relevant for the thesis work was the announcement of iOS 4. The new version, which will be released in the summer of 2010, includes a lot of new features and APIs (Engadget, 2010b).

Two new useful features are multitasking and calendar access. With multitasking the need for state saving when launching Google Maps and thereby exiting the prototype might be redundant. Calendar access would offer new ways of presenting work order and time reporting history for the user.

Despite the likely benefits of the new features the updated platform was ignored for the following reasons:

- It was still in beta.
- Testing the application and hence the platform beta on a real device seemed tricky.
- Support for older platform versions is likely to have been jeopardized.

8.6.6 Developing On iOS

Developing on iOS proved to be a wise choice. The API comes with great tutorials and is easy to understand. Great support is also provided by online on message boards and blogs.

The variety of standard components is somewhat limited. Fortunately, creating custom components and using self-made graphics is just as easy as it is powerful.

The main drawback of iOS development is the lack of a garbage collector in Objective-C. Allocated objects need to be released manually so it is easy for a beginner to end up with memory leaks that are time consuming to take care of. Luckily, Xcode is bundled with a great analysis tool.

8.6.7 Gestures

Interacting with the application prototype is like interacting with most iPhone applications. To click buttons the user taps them once. Flicking one or more fingers on the screen evokes scrolling. The multi-touch zooming gestures are used when viewing files attached to a work order.

Other gestures like touching and dragging and tapping multiple times were not used. It is possible that these gestures would be useful. For example, double tapping a time report entry would take the user directly to editing mode. Ideas for these kind of user actions will be discussed in user studies for further development of the prototype.

8.6.8 Audio Feedback

A decision was made to only use the default system sounds for audio feedback. After some time with the iPhone we noticed how seldom system sounds are used. Audio feedback was only noticed when using the unlock slider, tapping the on-screen keyboard and flicking through the parameters in the UIDatePicker.

Fortunately, iOS offers great support for playing custom sounds. For example, when tapping a UITableViewCell. Ideas on audio feedback are discussed in chapter 9. Future Work.

8.6.9 Tactile Feedback

Tactile feedback is given in the form of heat, pressure and texture. It is used on mobile devices in the form of vibration. For example, when a text message is received or when keys on an on-screen keyboard are tapped.

Tactile feedback was not implemented for two reasons. First, access to phones was limited throughout the design process. The iPhone Simulator was used and sometimes an iPod Touch, which lacks a vibration module. Second, despite the ease of triggering the iPhone's vibration module proper user testing did not fit into the thesis time frame.

How tactile feedback should be added to the application prototype is discussed in chapter 9. Future Work.

8.7 Technology Hostility

According to the studies the following potential improvements should be taken into consideration. First, modern smartphone platforms must be used. Platforms like iPhone and Android offer GUI solutions more suitable for mobile devices compared to Windows Mobile, which according to studies is the most common platform today for service providers. These platforms also remove the need of a stylus, which was mentioned as a source of annoyance.

Another must is protective accessories. Today's smartphones are not suitable for the harsh working conditions for a service provider employee. Introducing accessories such as screen shield, holsters and straps for protection, can reduce this problem.

The higher interest in technology among the younger employees should be used for gently introducing the older employees to the concept. This proved successful according to Göteborg Energi.

User-centered design is an important approach. At Göteborg Energi AB the software was never a problem thanks to the personnel being involved in the design process of the reporting application from the start. The employees and had all of their needs and wishes granted and enjoyed using the application (when the smartphones did work properly).

Finally, an anecdote about being stubborn. When Palm devices were introduced At Göteborg Energi AB in the early 2000's they were met with hostility before being accepted. A few years later iPAQs were introduced and met with just as much hostility despite the popularity of the Palm devices. Today no smartphones are used for mobile reporting. Well, not completely. A few employees aged above 60 still use their 9-year-old Palm devices and will not let go of them.

8.8 Design Manual

Due to time restrictions the manual does not go into any considerable depth on smartphone GUI design. Binea will still have to study the platform specific design guidelines when developing for Android, iOS and Windows Phone 7. Due to this fact using the manual might cause conflicting design advices. The platform specific design guidelines must be given top priority.

The design patterns presented are usable on smartphones, but the majority of them are adapted patterns typically used on desktop applications and web sites. No study on smartphone design patterns was found.

Tactile and audio feedback is not mentioned. This was not asked for by Binea, but should be included in a future version.

9 Future Work

This chapter discusses work to be done concerning user studies, the application prototype and the design manual.

9.1 User Studies

Another round of user studies should be performed for evaluating the application prototype. A suitable approach would be observing field personnel at Bravida and Göteborg Energi AB performing everyday work tasks. Their feedback should then be gathered and further developed using one-on-one or focus group interviews.

Another round of cognitive walkthroughs with the same expert users as earlier would also be suitable for learning what they think have improved and what needs to be worked on further.

9.2 Application Prototype

There are minor features that should be added to the application prototype. If it would be developed into a commercial application there are other major features that must be added. All of these are presented here.

9.2.1 Prototype Improvements

For a time report it is possible to modify the time intervals and the attached comment. However, the entry date of the report cannot be changed and a saved time report cannot be deleted. These features were not implemented due to time restrictions and flaws in the database design. This must be fixed in a future version.

Despite being requested in the user studies, attaching a new file to a work order is not possible, since there is no file manager similar to a desktop OS on iOS. In a future version of the prototype it will be possible to add pictures to a work order using the UIImagePicker controller.

When creating a new work order the address is entered using the on-screen keyboard, regardless if it exists in the database. For reducing text input in a future version two different solutions can be used. One, using auto-completion. Two, presenting all available addresses in a UITableView. Implementing one of these solutions were put on hold due to time restrictions.

For demonstrating the dynamicity of the system architecture a demonstration feature was planned where the "skin" of the GUI would change, presenting alternative solutions for user input. This feature was omitted due to time restrictions but the database is designed for it.

9.2.2 Additional Features

Before implementation started it became obvious that some of the must-have features were too time consuming and dependent the internal systems for a service provider. Since no customer was involved these features were put on hold, however if one were to be found the following are likely to be asked for.

An equipment catalogue must be added along with a web store for ordering new items. Equipment in stock must be virtually movable between the work order and the user's service car, which is used as storage. It should also be possible to manually enter purchased items into the system using the application.

A web service must be implemented for push notifications and synchronization between the personnel and their office. This feature was ignored due to time constraints. It would have been a thesis on its own.

Push notifications offer the possibility of alert users of newly added new work orders and changes to already received orders. On iOS this is done using badges. A badge is a tiny icon that is put on the launch icon of an iPhone app or on the tabs of a UITabBar. It is used in lots of iPhone apps as shown in figure 44 below.

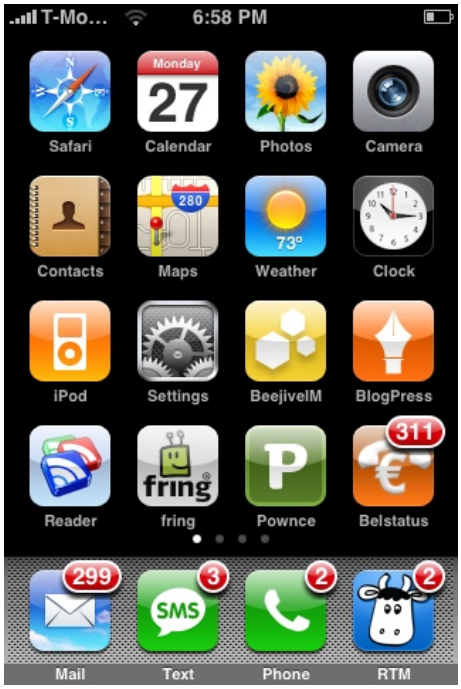


Figure 44. iPhone apps with badges (Apple Inc., 2010).

9.3 System Architecture

In the current version of the prototype it is possible to specify what sections a UITableView should contain. However, code for assigning data to each UITableViewCell in the sections has to be present, thereby limiting the selection. In order to allow any section and cell to be put in any tab this code needs to be abstracted into some kind of cell controller.

Since a cell is automatically released when scrolled outside of the screen its cell controller cannot keep a reference to it. To solve this issue the cell could be passed to a method in the controller, which would assign its data and set itself as its delegate. The delegate methods would then update the work order object based on received user input.

When this is done a method of dynamically loading the cell controllers is needed. The best way would probably be to add the controller functionality to the WOTableSection class and its subclasses, which are loaded when a table is first set up. Once an XML parser is in place the tabs and sections could be specified in an XML file and the desired level of dynamicity would be reached.

For the second "skin" some of the custom components would need additional properties. For example, WOTimePickerCell currently only supports picking a time interval in hours and minutes. It would need a mode property toggling its view to display a start and a stop time. These properties would be set by the cell controller and are easily read from an XML or database file.

9.4 Audio Feedback

Using audio feedback in addition to visual feedback improves the chances of alerting the users that their actions were handled. In a future version of the prototype subtler but attention grabbing custom sounds should be played. For example when tapping a UITableViewCell and receiving a new work order. User studies must be used for evaluating added audio feedback and it must be possible to disable them individually.

9.5 Tactile Feedback

Apart from visual and audio feedback, tactile feedback in the form of vibrations should also be offered. This type of feedback is useful when the smartphone is in the user's pocket and when the audio feedback cannot be heard. It can also be used for simulating the physical feedback when tapping a touch screen button. As with custom sounds this feedback would be possible to disable.

9.6 Design Manual

The content of the design manual focuses solely on GUI design for native smartphone applications. An area that could be covered in a future version is web design for smartphone screens, which is radically different from web design on a desktop computer.

The manual must also deal with GUI design for smartphones with different hardware setups. UIs using slideable keyboards and styluses are still common and should be taken into consideration.

For getting applications published on the different web stores for Android, iOS and Windows Phone 7 different design guidelines must be for each platform. A future manual must include detailed GUI design information based on these guidelines for the most popular platforms.

Finally, a GUI is only a part of the total UI. Design advice for audio feedback and tactile feedback must also be added in a future version.

10 Conclusion

The aim of this thesis was to create concept for a mobile reporting application for mobile devices. Based on this concept a software prototype should have been developed. The goal of the application prototype was to have high usability and have dynamic system architecture for smooth adaptation to different customers. Insight was also wanted on hostility towards technology among field personnel at service providers.

The goals were reached. Knowledge of mobile reporting and the working process of service providers were gathered by field observation, paper prototyping sessions and multiple interviews with Bravida and Göteborg Energi. Information was also gathered from studies of reporting applications available on today's market.

Based on these studies the must-have features for a mobile reporting application were identified as the following:

- *Work order handling.* An overview of a users active work orders should be presented. When viewed in detail a work order should contain information such as problem description, address, contact person and involved colleagues. New orders should be creatable.
- *Work history.* When viewing work order old information related to the same address should be presented.
- *Time reporting.* Users must be able to specify the time type and time in hours and minutes spent on a work order.
- *Equipment ordering.* Through a web store equipment and material should be presented and be able to be delivered to the current work site or local office.
- *Equipment handling.* Equipment in stock must be presented and virtually movable between the work order bill, the user's service car and the local office storage.
- *Mapping service.* The user must be provided with locations and route descriptions to addresses of work orders.

Choosing iOS for implementing a prototype of the design concept proved to be wise. The platform is known for its smart and usable GUI solutions and appealing aesthetics. All features listed above are implemented apart from equipment ordering and handling. They proved to be too time consuming and customer specific and were slightly out of scope due to the back-end functionality that would have to be developed.

To make the system architecture as dynamic and adaptable as possible table components with interchangeable cells are used for most views in the prototype. Work is still needed in order to allow arbitrary cells but the system is despite this fact already quite flexible and easy to extend.

A simple and usable GUI was developed along with several ideas for improvements. This was the result of staying true to the well-evaluated GUI sketches along with feedback by the cognitive walkthrough participants.

Concerning technology hostility, the myth of older employees being more hostile is partly true. The work force at Bravida and Göteborg Energi can roughly be divided into two groups; men in their 20's or early 30's and men in their 50's and early 60's. The younger group has a larger interest and experience in technology and is easily persuaded into using new technical solutions. The older group is stubborn and skeptical of change in general. They need sturdy smartphones with larger screens, a simple GUI and the possibility of using a full-size keyboard. If hardware like this is available the younger group can be used as a stepping-stone before introducing the solution to their older colleagues.

The design manual offers a basic introduction to smartphone GUI design with design patterns and general advice. Design solutions for the must-have features of a reporting application are also included. However, an improved design manual would summarize the platform specific design guidelines for Android, iOS and Windows Phone 7 and use them as a basis for further design advice.

Concerning the work process, the time planning was successful and we were always well prepared for the next step. No major miscalculations were made and the methods used for gathering information and evaluating design solutions proved to be successful.

References

addMobile, 2010. *Dags att mobilisera!* [Online] (Updated n.d.)
Available at: http://www.addmobile.se/index.htm
[Accessed 25 January 2010]

Android, 2010a. *Android Developers.* [Online] (Updated n.d.)
Available at: http://developer.android.com/index.html
[Accessed 5 May 2010]

Android, 2010b. *User Interface Guidelines.* [Online] (Updated n.d.)
Available at: http://developer.android.com/guide/practices/ui_guidelines/index.html
[Accessed 6 May 2010]

Apple Inc., 2010b. *iPhone Human Interface Guidelines.* [Online] (Updated n.d.)
Available at:
http://developer.apple.com/iphone/library/documentation/UserExperience/Conceptual/MobileHIG/Introduction/Introduction.html
[Accessed 2 March 2010]

Apple Inc., 2010b. *iPhone OS Technology Overview* [Online] (Updated n.d.)
Available at: http://developer.apple.com/technologies/iphone/
[Accessed 7 May 2010]

Apple Inc., 2010c. *HoursTracker - Time Sheet.* [Online] (Updated 22 April 2010)
Available at: http://itunes.apple.com/us/app/hourstracker-time-sheet/id321923934?mt=8
[Accessed 14 May 2010]

Apple Inc., 2010d. *Developing with Core Data.* [Online] (Updated n.d.)
Available at: http://developer.apple.com/macosx/coredata.html
[Accessed 10 May 2010]

Apple Inc., 2010e. *iPhone Dev Center.* [Online] (Updated n.d.)
Available at: http://developer.apple.com/iphone/index.action
[Accessed 10 June 2010]

Ars Technica, 2006. *Study shows youth embracing technology even more than before.* [Online] (Updated 1 August 2006)
Available at: http://arstechnica.com/old/content/2006/08/7401.ars
[Accessed 10 February 2010]

Articlesbase, 2009. *A Short History of the Smartphone.* [Online] (Updated 2 February 2010)

Available at: http://www.articlesbase.com/cell-phones-articles/a-short-history-of-the-smartphone-847439.html
[Accessed 12 May 2010]

bada, 2010. *What is bada.* [Online] (Updated n.d.)
Available at: http://www.bada.com/whatisbada/
[Accessed 5 May 2010]

Binea AB, 2010. *Om Oss*. [Online] (Updated n.d.)
Available at: http://www.binea.se/index.php?link=about
[Accessed 2 March 2010]

BlackBerry, 2010. *BlackBerry Device Software.* [Online] (Updated n.d.)
Available at: http://na.blackberry.com/eng/services/devices/#tab_tab_features
[Accessed 12 May 2010]

Bravida, 2010. *Om Bravida.* [Online] (Updated n.d.)
Available at: http://www.bravida.se/Om-Bravida/
[Accessed 2 March 2010]

Cooper, A., Reimann, R. & Cronin, D., 2007. *About Face 3: The Essentials of Interaction Design.* Wiley Publishing, Inc.

Davis, F. D., 1989. Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Quarterly, 13(3), 319-340.

Engadget, 2010a. *Windows Phone 7: the complete guide.* [Online] (Updated 18 March 2010)
Available at: http://www.engadget.com/2010/03/18/windows-phone-7-series-the-complete-guide/
[Accessed 18 March 2010]

Engadget, 2010b. *iPhone OS 4 unveiled, adds multitasking, shipping this summer.* [Online] (Updated)
http://www.engadget.com/2010/04/08/iphone-os-4-0-unveiled-shipping-this-summer/
[Accessed 1 June 2010]

ePocket Solutions AS, 2010. *Welcome to ePocket Handyman.* [Online]
(Updated n.d.)
Available at: http://www.epocket.se/omepocket
[Accessed 24 January 2010]

Gartner, 2010a. *Gartner Says Worldwide Mobile Phone Sales Grew 17 Per Cent in First Quarter 2010.* [Online] (Updated 19 May 2010)

Available at: http://www.gartner.com/it/page.jsp?id=1372013
[Accessed 20 May 2010]

Gartner, 2010b. *Gartner Says Worldwide Mobile Phone Sales Grew 8 Per Cent in Fourth Quarter; Market Remained Flat in 2009.* [Online] (Updated 23 February 2010)
Available at: http://www.gartner.com/it/page.jsp?id=1306513
[Accessed 13 February 2010]

Göteborg Energi AB, 2010. *Om Oss*. [Online] (Updated n.d.)
Available at: http://www.goteborgenergi.se/Om_oss_DXNI-5165_.aspx
[Accessed 2 March 2010]

Hjelm & Kähkönen, 2009. *Verksamhetsnytta vid användande av handdatorer: En studie med fokus på fältarbetare i tjänsteföretag.* Linköping Universitet, Institutionen för ekonomisk och industriell utveckling.

InformationWeek, 2006. *Generation Y Taking Technology To New Level.* [Online] (Updated 31 July 2006)
Available at: http://www.informationweek.com/news/global-cio/showArticle.jhtml?articleID=191600959
[Accessed 31 May 2010]

Kuo-Ying Huang, 2009. *Challenges in Human-Computer Interaction Design for Mobile Devices.*
Proceedings of the World Congress on Engineering and Computer Science 2009 Vol I WCECS 2009, October 20-22, 2009, San Francisco, USA.

MeeGo, 2010. *FAQ*. [Online] (Updated n.d.)
Available at: http://meego.com/about/faq
[Accessed 1 June 2010]

Microsoft, n.d. *Windows Mobile AppStars*. [Online] (Updated n.d.)
Available at: http://www.microsoft.com/emea/windowsmobileapps/default.mspx
[Accessed 28 January 2010]

Microsoft, 2009. *Vanliga frågor om den nya versionen av Windows Mobile.* [Online] (Updated 6 October 2009)
Available at: http://www.microsoft.com/windowsmobile/sv-se/meet/wm65-faq.mspx
[Accessed 31 May 2010]

Microsoft, 2010. *UI Design And Interaction Guide For Windows Phone 7.* [Online] (Updated n.d.)
Available at:
http://download.microsoft.com/download/D/8/6/D869941E-455D-4882-A6B8-0DBCAA6AF2D4/UI%20Design%20and%20Interaction%20Guide%20for%20Windows%20Phone%207%20Series.pdf

palm webOS, 2010. *About Us.* [Online] (Updated n.d.)
Available at: http://palmwebos.org/
[Accessed 5 May 2010]

Practical eCommerce, 2010. *Chart of the Week: Google's Android Mobile OS Will Outpace the iPhone, Others.* [Online] (Updated 19 January 2010)
Available at: http://www.practicalecommerce.com/articles/1575-Chart-of-the-Week-Google-s-Android-Mobile-OS-Will-Outpace-the-iPhone-Others
[Accessed 13 February 2010]

Preece, J., Rogers, Y. & Sharp. H., 2007. *Interaction Design Beyond Human-Computer Interaction*. 2nd ed. John Willey & Sons.

Remotex, 2010. *Produkter.* [Online] (Updated n.d.)
Available at: http://www.remotex.se/Svenska/Produkter/RemoteXMobilklient.aspx
[Accessed 25 January 2010]

Snyder Consulting, n.d. *paper protoyping.* [Online] (Updated n.d.)
Available at: http://www.snyderconsulting.net/article_paperprototyping.htm
[Accessed 11 February 2010]

Snyder, C., 2003. *Paper Prototyping: the fast and easy way to design and refine user interfaces*. [e-book] San Francisco: Morgan Kaufmann
Available at: Google Book
Search http://books.google.com/books?id=5OhE7dyGtmgC&hl=sv
[Accessed 11 February 2010]

Sooper Articles, n.d. *The History of The Smartphone.* [Online] (Updated n.d.)
Available at: http://www.sooperarticles.com/communications-articles/mobile-cell-phones-articles/history-smartphone-24602.html
[Accessed 12 May 2010]

SQLite, 2010. *About SQLite.* [Online] (Updated n.d.)
Available at: http://www.sqlite.org/about.html
[Accessed 10 May 2010]

Stack Overflow, 2009. *Use CoreData or SQLite on iPhone?* [Online] (Updated 9 March 2010)
Available at:
http://stackoverflow.com/questions/1318467/use-coredata-or-sqlite-on-iphone
[Accessed 30 March 2010]

Stanford University, 2010. *CS iPhone Application Development.* [Online] (Updated 26 May 2010)
Available at: http://www.stanford.edu/class/cs193p/cgi-bin/drupal/
[Accessed 31 May 2010]

Symbian, 2010. *Symbian FAQs.* [Online] (Updated n.d.)
Available at: http://www.symbian.org/about-us/symbian-faqs
[Accessed 31 May 2010]

Tidwell, J., 2005. *Designing Interfaces: Patterns for Effective Interaction Design.* O'Reilly Media, Inc.

Venkatesh, V., & Davis, F. D., 2000. A theoretical extension of the technology acceptance model: Four longitudinal field studies. Management Science, (46:2), 186-204.

Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D, 2003. User acceptance of information technology. Toward a unified view. MIS Quarterly, (27:3), 425-478.

World Wide Web Consortium, 2010. *XML Essentials*. [Online] (Updated n.d.)
Available at: http://www.w3.org/standards/xml/core
[Accessed 31 May 2010]

Appendix A – Design Manual

1 Introduction

The aim of this manual is to provide guidelines for creating a great looking and highly usable graphical user interface on smartphones.

This version of the design manual focuses on GUIs for native applications on smartphones that are equipped with multi-touch displays and on screen keyboards. In future versions focus will also be put on smartphone interfaces with support for a slideable physical keyboard and stylus pens for navigation.

The main focus of the manual is on information architecture (structuring and organizing information) and information design (visualizing and presenting information). Design patterns will be presented with in-depth descriptions, examples and their advantages and disadvantages.

Included is also a case study of how to design the must-have-features for mobile reporting as mentioned in our user studies. These features include time reporting, work order handling, equipment handling and equipment ordering.

2 Design Goals

If our design concept has been followed for the development of a smartphone application we want the following goals to be fulfilled:

- High usability. There is no room for suboptimal interface solutions in the 21st century.
- Great aesthetics. A good-looking application will always be more attractive to use than a less good looking one.
- Maximizing multi-touch possibilities. Modern smartphones offer several multi-touch gestures that enable smart new action.
- Minimal user text input. Using a keyboard on a smartphone is far from as effective from a desktop keyboard.

2.1 Developer Guidelines

Important to note when developing a native application for Android, iOS and Windows Phone 7 is that certain design guidelines are offered for aiding developers to create a GUI with high usability. These design guidelines have similarities, but also differences from each other. They might also differ from the design patters presented below.

2.2 Adaption to Hardware

On today's smartphone market there exists lots of different types of smartphones. The equipped hardware matters just as much as the software when it comes to GUI design. This section lists typical hardware setups and mentions how these affect GUI design.

2.2.1 Multi-touch

A multi-touch screen enables having less button and thereby a less cluttered GUI. For example, when zooming in on a map pinch zooming can be used instead of having two buttons for zooming in and out.

Multi-touch will also speed up interaction since the input area is larger than the area of a specific button. For example, two-finger-scrolling lets you scroll anywhere on the screen and not on a certain section.

Multi-touch gestures can also be linked to actions that normally demand the user to enter a menu system. For example, double tapping on an email for deleting it.

The drawbacks of multi-touch are that some action can be troublesome to perform using a single hand, when wearing gloves or when having really cold hands. Everyone who has owned an iPhone or iPod Touch has experienced this. Novice users might also have trouble understanding what to do if no indication of the result of a gesture is shown.

2.2.2 Keyboard

The advantage of using a physical keyboard is that screen space can be used for something else. A physical keyboard also provides with the physical feedback when clicking a button, which so far only can be simulated using vibration.

On many smartphone models the user is forced to tilt the phone when using the slideable keyboard. This means the content on the screen must be adapted to a landscape view. In some cases the new screen ratio will suit the content better, but it could also be the other way around.

2.2.3 Stylus

The main advantage of a stylus is more precise input. For example when drawing and writing signatures.

The big disadvantage is an unusable GUI if the stylus is lost and no temporary substitute is available. Using a finger is usually not an option since stylus oriented interfaced tend to demand a too fine detailed input precision.

Stylus dependent interfaces also tend to include too many and too tiny elements. This can be tiring for a user constantly getting over stimulated by GUI elements.

2.3 Design Patterns

This section lists suitable design patterns for organizing content, navigating an interface and getting input from users. The majority of the patterns has been provided by Tidwell (2005) and has been modified when needed for suiting mobile devices.

2.3.1 Interface Navigation

These patterns deal with designing a suitable navigation pattern.

2.3.1.1 Global Navigation

What
Put a consistent set of icons on the same area of each page enabling the user to access the different sections with just one click.

When
The information to be presented can be categorized and the user is likely to want to go back and forth between the divided parts frequently.

Why
Reduces the amount of clicking for going back and forth between the different sections.

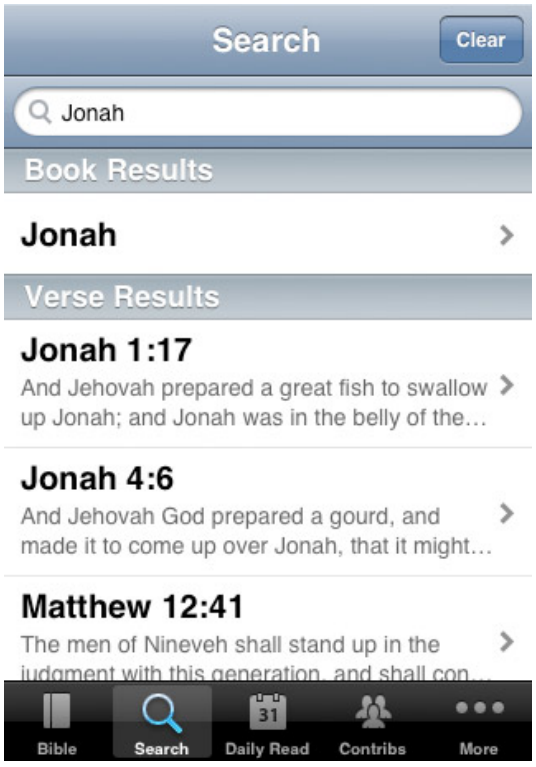


Figure 1. A tab bar on iOS (© YouVersion).

2.3.1.2 Hub and Spoke

What
Divide the application into mini-applications and let the user access them through a main menu using one tap. From within the selected mini-application the main menu should be revisited by one tap.

When
The features are very different from one another and the users are unlikely to want to go back and forth between them while performing a task.

Why
Hub and Spoke is a navigation pattern found in almost all mobile phones and smartphones on the market today. The interface can be kept cleaner. User errors are less likely to occur since the user has less "possibilities" to put the software in a bad state.

Figure 2. The home screen in iOS (© Apple Inc.).

2.3.1.3 Pyramid

What
Link pages to each other using elements on each page to access its neighbors. Each page should also contain an element linking to an overview page where the subpages can be accessed directly. An element could be a link, a button or a gesture.

When
If a set if similar or otherwise connected objects are to be viewed.

Why
The user interface will be clearer and more understandable if the users can decide for themselves what information should be presented or not.

Figure 3 and 4. The image library on iOS.

2.3.1.4 Modal Panel

What
Show a page with a set of choices the user has to make before being able to continue with anything else.

When
The application is in such a state that it cannot continue without an action from the user. Must only be used in important situations. Note, overuse will likely cause users frustration.

Why
Suitable when wanting important information since the user's full attention will be available and because the wanted input can not be ignored.

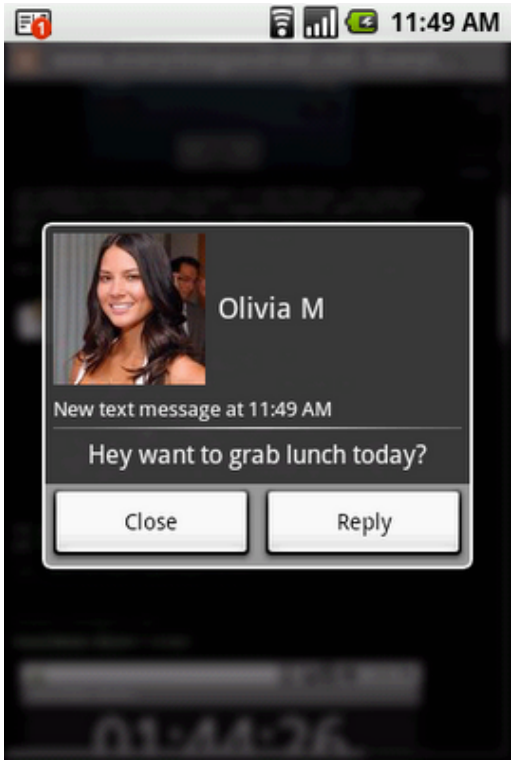


Figure 5. Receiving an SMS text message on the Android platform (© Google).

2.3.1.5 Wizard

What
The user is guided through the interface one step at a time in a prescribed order.

When
The task to perform is complicated or contains many steps where the answer to the former affects the choices of the latter.

Why
By splitting a task into different sections you simplify it as a whole by allowing the user to fully focus on each step separately.

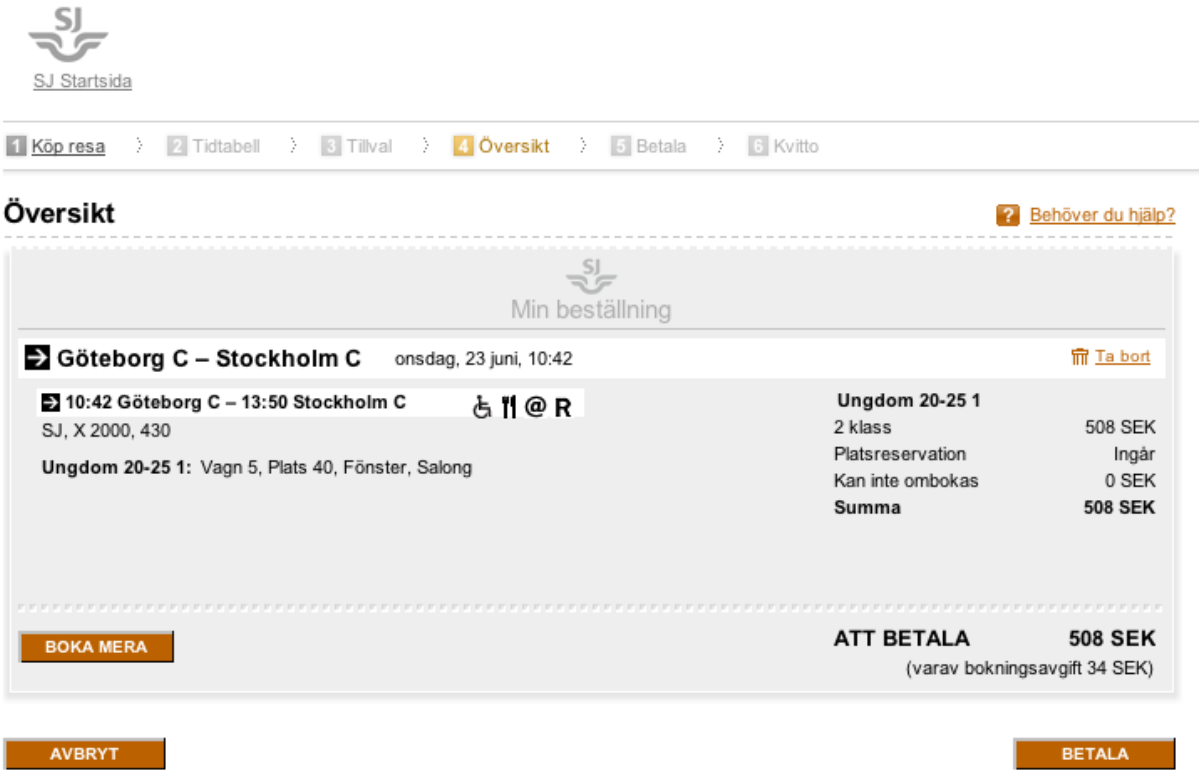


Figure 6. Ordering tickets from SJ.

2.3.1.6 Animated Transition

What
When the interface will make an obvious change from state 1 to state 2, animate the transition.

When
The interface is heavy on obvious changes that will take the users full attention. For example, zooming on a map, pop-up windows, new pages being opened, current pages being closed and so on.

Why
Sudden drastic changes of the GUI might confuse the user. By having an animation showing the change from state 1 to state 2 they are more likely to understand where they currently are and not be dislocated by a change coming "out of nowhere".

Example
When launching apps and going back and forth between views in them on an iPhone. Animations are usually provided by default on smartphone platforms, like the iOS.

2.3.1.7 Extras On Demand

What
The most important information is shown up front. The rest is hidden, but easily reached using a simple action.

When
There is too much stuff to show on the screen and not all of it is relevant to show up front at the same time.

Why
Save space and reduce clutter. The UI will be clearer and more understandable if the users can decide for themselves what information should be presented.

Figure 7 and 8. Keyboard setup on iOS.

2.3.1.8 Card Stack

What
A combination of the Global Navigation and the Extras On Demand patterns. Content on a page is split up and put under different tabs.

When
There is too much content to fit on a single page and the user is likely to want to go back and forth between the different sections often.

Why
A well-used pattern on smartphones for smart usage of screen space. Each chunk of information becomes easy to handle from a cognitive point of view.

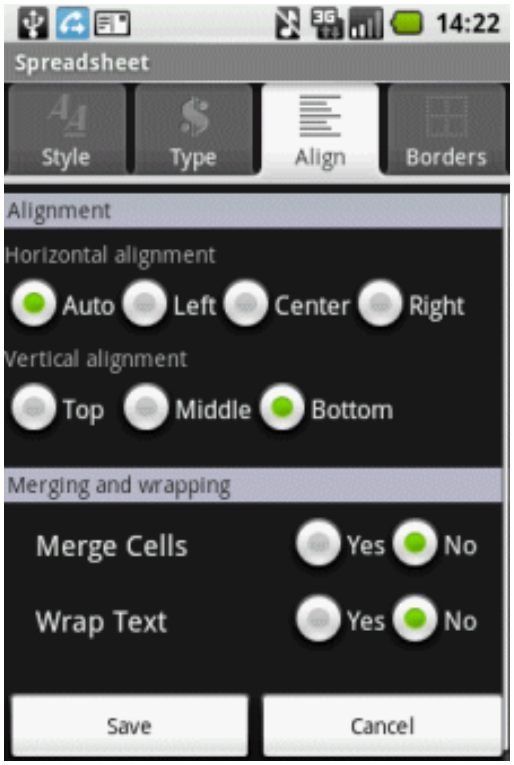


Figure 9. Spreadsheet properties on Android (© Google).

2.3.2 User Input

These patterns should be used when wanting input from the users.

2.3.2.1 Auto completion

What
When the user enters text into a field an educated guess is made by the system, which then completes the entry.

When
The users enter text that is predictable. For example, addresses to web pages, dates, name of cities and so on.

Why
Text input on smartphones is tedious. This pattern saves time and reduces the risk of user errors when entering long text segments.

Figure 10. Auto-completion on Android. Here the user can also select one of the guesses (© Google).

2.3.2.2 Forgiving Format

What
Allows the user to enter input in different formats.

When
The user is able to enter information with any valid amount of whitespaces, capitalizations, abbreviations and so on.

Why
The user should not have to think about remembering how the syntax should be when entering some types of information. The system can take care of this.

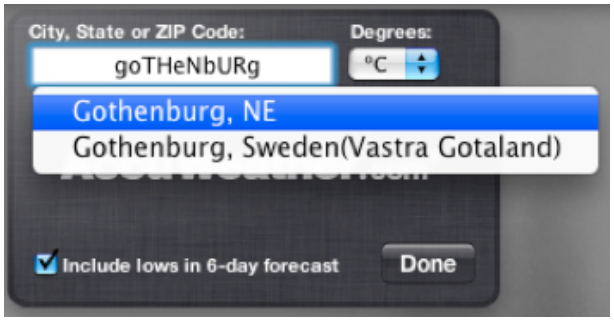


Figure 11. Setting city on the weather widget on Mac OS X.

2.3.2.3 Input Prompt

What
Near an editable text field, place a comment of what the user should enter.

When
The input is not obvious and more words in a label before the text field are not suitable.

Why
Decreases the chance of a misunderstanding of what should be entered. Since the hint is separated from the label advanced can easier ignore it.

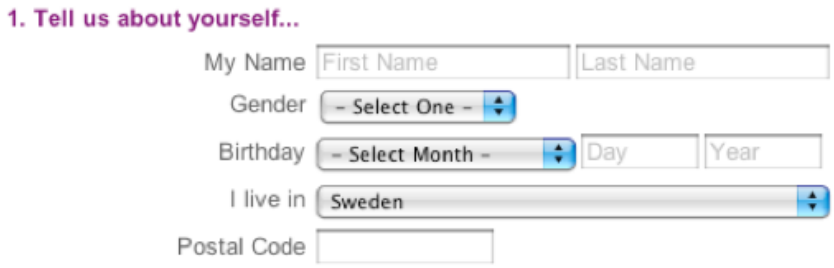


Figure 12. Creating a profile for Yahoo Mail.

2.3.2.4 Input Hints

What
Fill a text field for input with instructions of what the user should enter.

When
No good default values are available and the user could need a reminder.

Why
Helps make the user interface self-explanatory. Removes the need of explanatory labels before an input text field. Provides instructions that cannot be missed or ignored (compared to the Input Hints pattern).

Figure 13. Entering postal code on LinkedIn.

2.3.2.5 Good Defaults

What
Where text input is needed, fill the text field with a calculated guess.

When
Users have filled in enough information about their selves to make a good guess. The information being entered on guess should not be sensitive or politically charged.

Why
Save time by reducing text input and thereby the risk of errors. However, caution must be used. Sensitive and important information must not be entered by default since it can be accepted by mistake.

2.3.2.6 Illustrated Choices

What
When making a selection use pictures in addition to or instead of words.

When
The differences between objects to choose from are best explained using small images of them.

Why
The cognitive load on the users will be reduced. Instead having to think of how something looks like based on a description it can be seen direct. Images also look good and spark up an interface.

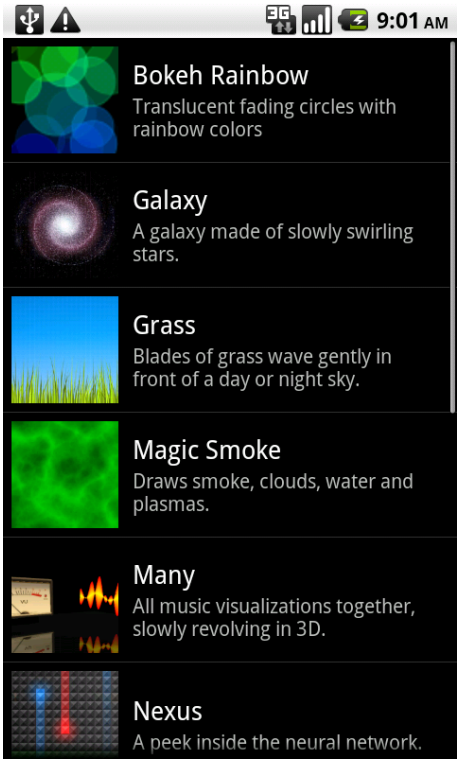


Figure 14. Changing the wallpaper on Android (© Google).

2.3.2.7 Prominent "Done" Button

What
Place a big and well-labeled button finishing a set of actions at the end of the visual flow

When
All the time when there is need for a button like "Send", "Ok", "Order" and so on.

Why
Giving the users a feeling of closure, that they know everything is done once the button is clicked.

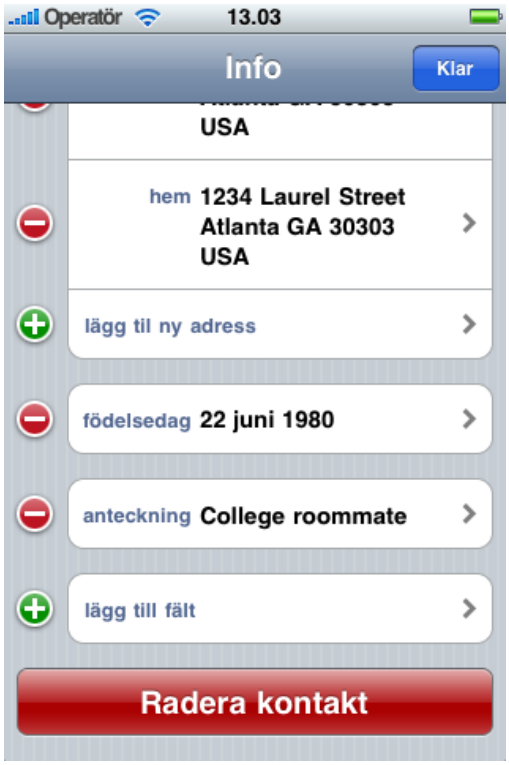


Figure 15. Deleting an iPhone contact profile.

2.3.2.8 Right/Left Alignment

What
When designing two columns next to each other, right-align the elements on the left and left-align the elements on the right.

When
You put elements in you GUI, which will have a label to their left.

Why
When a label is put right next to the element it explains a strong visual connection is made. Compared to labels with lots of whitespace between them and the elements they label, the users are less likely to get lost and link the incorrect label to the wrong element.

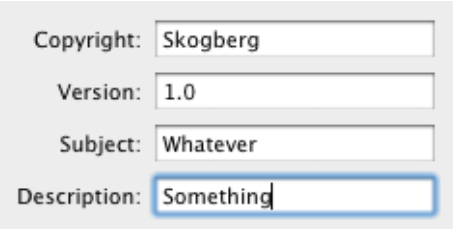


Figure 16. Entering properties in OmniGraffle.

2.3.2.9 Row Striping

What
Use two shades of the same color when listing several rows with information about objects.

When
You are dealing with a table where the rows have a hard time standing out from each other. This is usually the case when there are many columns or lines per row.

Why
When viewing a table with a single background color it is common to view the columns as unities on their own. The Row Striping pattern helps users to read the actual rows by making them be the focus of attention.

Figure 17. Rows in different shades of gray on iOS (© Roambi).

2.3.3 Aesthetics

Just as important as information architecture is also information design. It does not matter how well structured and organized your application is, it also needs to be aesthetically pleasing.

Tidwell (2005, p. 269) cites interaction design guru Donald Norman saying "positive affect enhances creative, breadth-first thinking whereas negative affect focuses cognition, enhancing depth-first processing and minimizing distractions". He also says "Positive affect makes people more tolerant of minor difficulties and more flexible and creative in finding solutions". Good looks equal improved usability.

When developing a smartphone application much of this is taken care for you by the standard components and the platform specific design guidelines. However, some things are worth pointing out.

2.3.3.1 Icons

Make sure to put time on designing good icons. In recent years graphics on mobile devices have been through major improvements and offer quality and detail comparable to desktop computers. For example, the icons on the Apple iPhone look just as great as in Mac OS X.

2.3.3.2 Color

Along with shape and form, color is the immediate thing a user notices when viewing your application. Therefore, the choice of color must be carefully chosen.

Never use red and green too distinct important information. This will lead to problems for the 10 per cent of men and 1 per cent of women who suffer from color blindness.

To make text readable it must be clearly distinguishable from the background. Dark text on bright backgrounds and vice versa. It is also important not to combine complementary colors (opposite colors on the color-wheel) with each other.

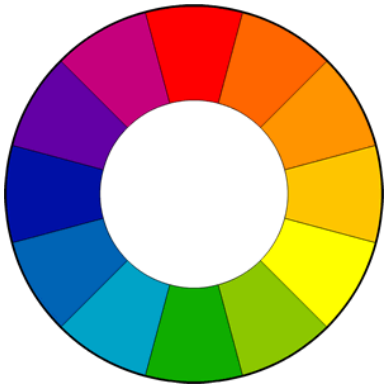


Figure 18. The color-wheel.

When picking a color palette for an application, focus on 2 or 3 different color hues and choose different variations from those hues based on brightness. This way the GUI is given character without looking like the rainbow. Less is more. To many different colors will clutter the GUI.

2.3.3.3 Typography

The type, style and size of font set the tone of how text is experienced for a user. Small differences can have huge effects on how it is perceived.

Do not use an italicized or cursive font style. It is often unreadable at small sizes. The same goes for highly geometric fonts with circular letters.

Fonts that are all-caps can be suitable for labels and headlines, but not for longer text segments. Sans-serif fonts tend to be more readable on computer displays.

3 Case Study: Service Providers

Service providers include people in the professions of electricians, estate service providers and carpenters. This case study mentions their relationship with technology and what must-have features a smartphone application for mobile reporting must include. These features are explained in detail and for each of them several design solutions are presented.

3.1 User Studies

For gathering information, several interviews and field studies were conducted with field personnel and work leaders at Göteborg Energi AB and at the electricity and elevator department at Bravida.

Once this information was gathered and analyzed several iterations of sketches of GUI solutions were done. These sketches were then reviewed in cognitive walkthroughs and improved before a final paper prototyping with Bravida. Based on the feedback from the session the sketches were finalized.

Based on the studies we learnt that there are roughly two groups of field employees when it comes to interest in technology. One, younger men in their mid-twenties to their early thirties. These employees are used to smartphones, play video games, use Spotify, surf the web and have a Facebook account. In short they have some interest and knowledge in what is happening on the technology scene.

The second group is men in their early fifties to mid-sixties. They are not that interested in technology, mainly use the Internet to read newspapers and have a negative attitude towards smartphones. However, they are just difficult to persuade not impossible.

3.1.1 Concept Features

Based on the user studies we learnt what features a mobile reporting application must provide to its users. These features include work order handling, time reporting, material ordering, material handling and a mapping service. These features are presented below in greater detail.

3.1.1.1 Work Order Handling

A work order contains lots of data. Based on the information from the user studies it has been divided into the sections Information, History, Reported Time, Attached Files and Checklist.

Information

This section should include general and most important information of the work order. This includes the following:

- The unique ID number of the work order.
- The full address.
- Access to mapping service.
- A problem description.
- The current status.
- The time priority or deadline.
- Access to contact information for assigned contact persons, work leaders and employees.

History

A wanted feature when viewing a work order is to be able to view the work history for its address. This includes problem descriptions from finished work orders with the same address along with general comments.

Checklist

Before finishing a work order a checklist has to be checked. What the checklist points should say is branch specific. From our user studies we got the impression that its main use is to be there for maintaining ISO standards

Attached Files

If needed files should be able to be attach to a work order. For example, circuit schemas, manuals and datasheets.

3.1.1.2 Mapping Service

Companies such as Bravida have offices all over Scandinavia and provide services in large regions. Even for senior employees with locations to an address is sometimes useful to know. A mapping service providing map location and route description is mandatory. The third party services Google Maps or Bing Maps are ideal choices.

3.1.1.3 Time Reporting

Time reporting is most likely the most important feature. Customers are charged for and employees paid for the exact amount of hours and minutes reported. Time is reported in hours and in minutes. Depending on the weekday and during which hours different types of overtime are to be reported for.

3.1.1.4 Material Handling

Just as with time, reporting what material and equipment used on a work order is a must when charging the customer. Not all companies have an equipment storage at their offices. Field employees instead have a service car, which also acts as their mobile storage unit. It must be possible to assign equipment back and forth between the car and a work order.

3.1.1.5 Material Ordering

Wherever the field employees are they must be able to order needed material and equipment. This requires an online web store included in the smartphone application.

3.1.2 Concept Design Solutions

This chapter provides an in-depth look into different design solutions for the previously mentioned features.

3.1.2.1 General Elements and Components

During the sketching process several ideas for useful GUI components were thought of. Some of them proved to be abstract enough to use for different purposes. These components are presented below.

The Multi-Functional Slider

This component looks and is meant to be used as a slider, but also offer other ways of input. First, its buttons for incrementing and decrementing serve two purposes. One, they provide guidance by saying that sliding the marker to the left will decrement the current number and sliding it to the right will do the opposite. Second, they work as ordinary buttons for incrementing and decrementing the value.

It is also possible to tap the field displaying the number and enter the wanted amount using a pop-up on-screen num-pad.

This component has the advantage of enabling multiple ways of input and demands low screen space. However, setting proper maximum and minimum values might be tricky along with the increment/decrement value.

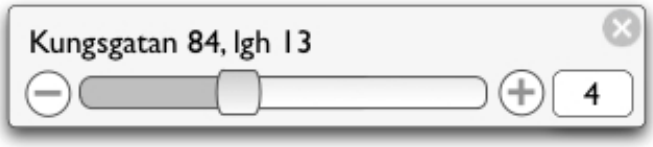


Figure 19. Sketch of the slider.

The Slot Machine Picker

Heavily inspired by the Picker component used on the iOS this component is suitable for entering amounts and quantities. The user simply rolls the wheels up or down

It has the advantage of having different formats and parameters on different wheels and is simple to use. However, it has the disadvantage of demanding lots of screen space.

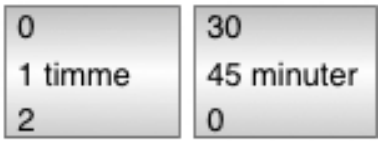


Figure 20. Sketch the slot machine picker.

Information

The general information for a work order can be presented in a view similar to figure 21 and 22 below.

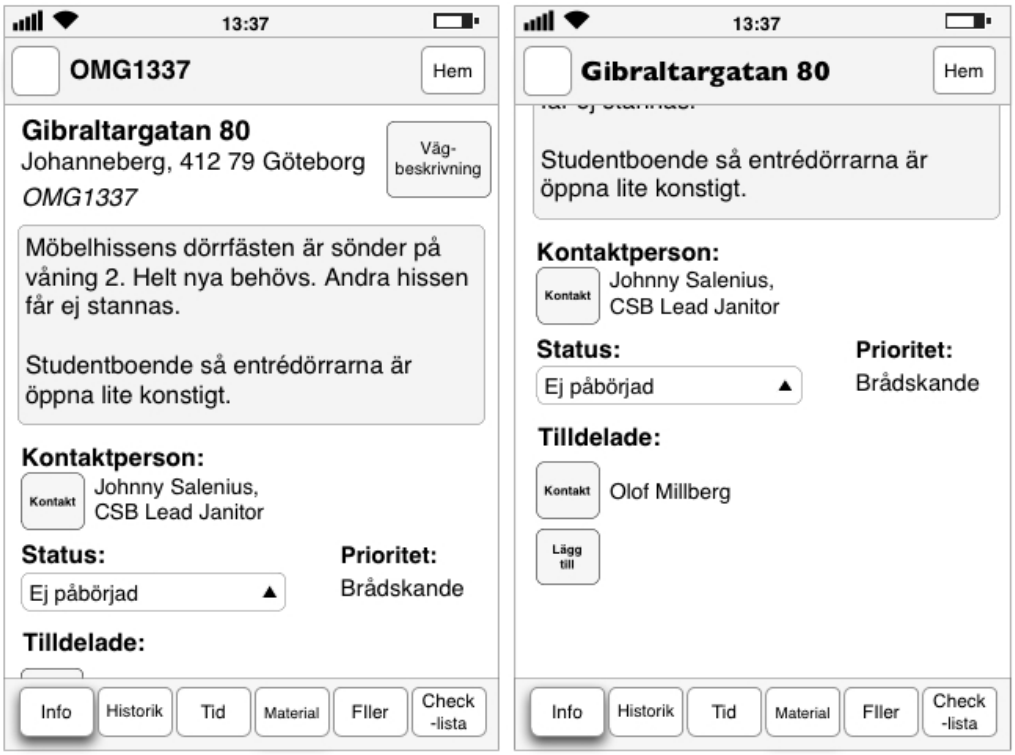


Figure 21 and 22. A presentation of the general information of a work order.

Time Reporting

Time reporting is performed differently depending on the company. Based on this fact from the user studies the following different design solutions were created. The main goal is to get away from using an on-screen num-pad for input.

The user must be allowed to do the following:

- Report time in hours and minutes.
- Report time for different time types (there are different types of overtime).
- Add a comment to each time reporting entry.
- View one's time reporting history.

Selecting Date

When handling a selected work order the date for a time reporting is selected using a calendar.

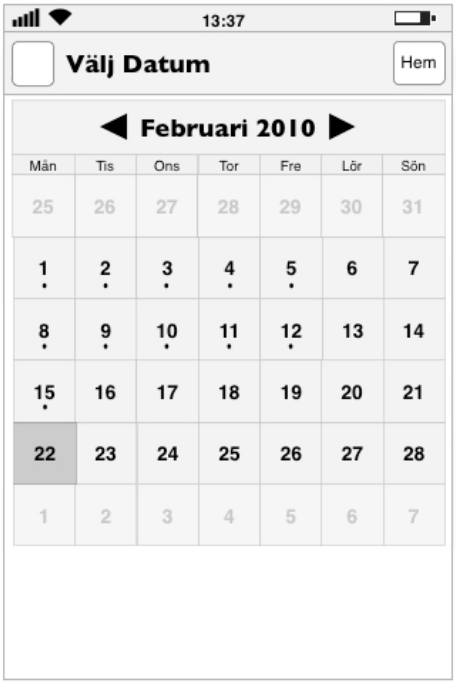


Figure 23. Selecting date for time reporting.

Depending on the length of average work tasks at a company selecting work tasks and reporting time should be done differently. If a user often works with the same order for several days it is more suitable to first select the work order and then the days. If many orders usually are handled per day it will be the easiest to first select the day and then the work orders.

Dealing With Overtime

Common for all service providers is the importance of overtime. For field employees the hours spent per day will vary. If overtime is spent there is different types of overtime depending on the time. Users must be able to quickly add overtime and get a good overview

Solution 1 - Drop Down Menus

This solution uses three drop down menus. One for time type, one for hours and one for minutes. This works since the amounts of parameters in each drop-down menu are limited enough.

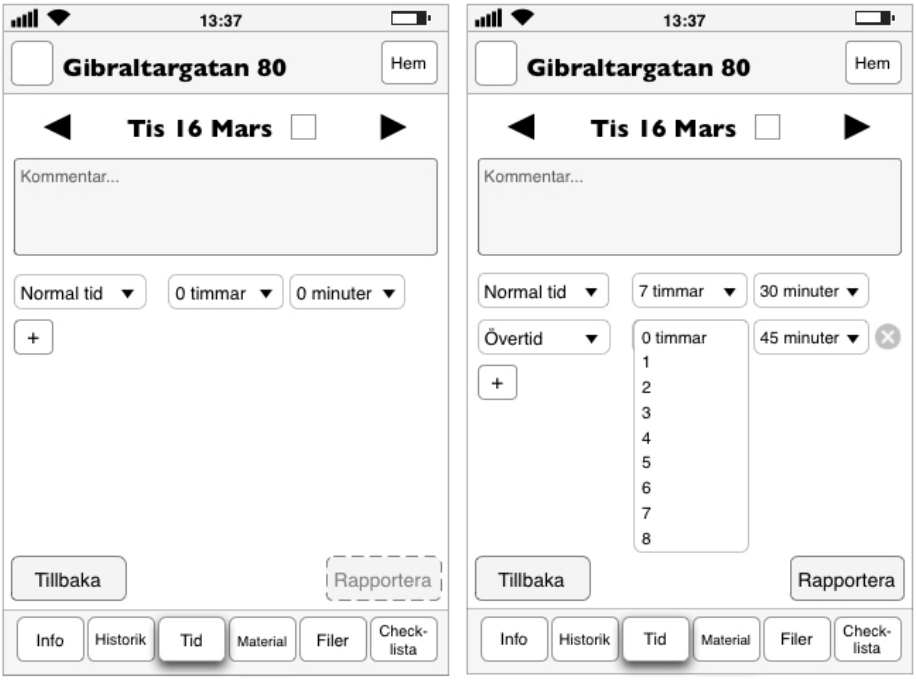


Figure 24 and 25. Setting time using drop down menus.

Solution 2 - Slider

This solution uses the previously mentioned multi-functional slider.

Solution 3 - Slot Machine Picker

This solution uses the previously mentioned slot machine picker.

Work Order History

The history entry previews will be presented in a list. Each preview should include the author name, post date and snippet of the description. When an entry is selected the same information is presented in a new view with a full description.

The user should also be able to add a note to the work order (and thereby to that address) if wanting to notify future employees of something useful to know about that address.

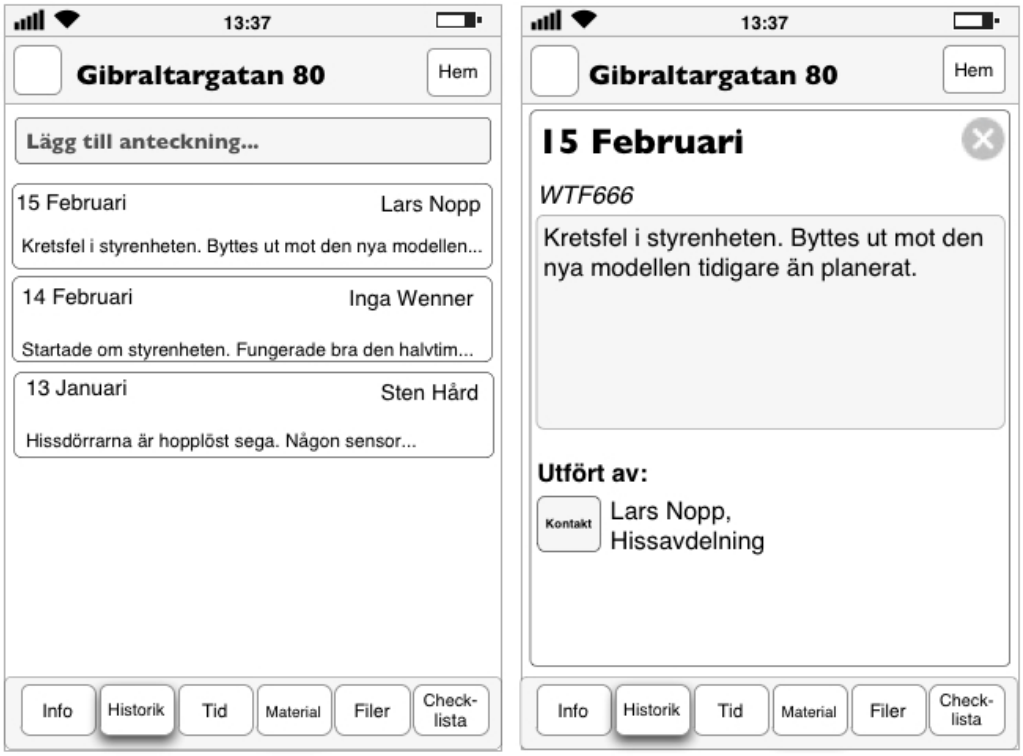


Figure 26 and 27. The list of entries. Viewing one note in detail.

Attached Files

The attached files should be presented as clickable icons with different icon images depending on their file types. Presenting them in a list and showing information like file type, date of latest modification, size and so on.

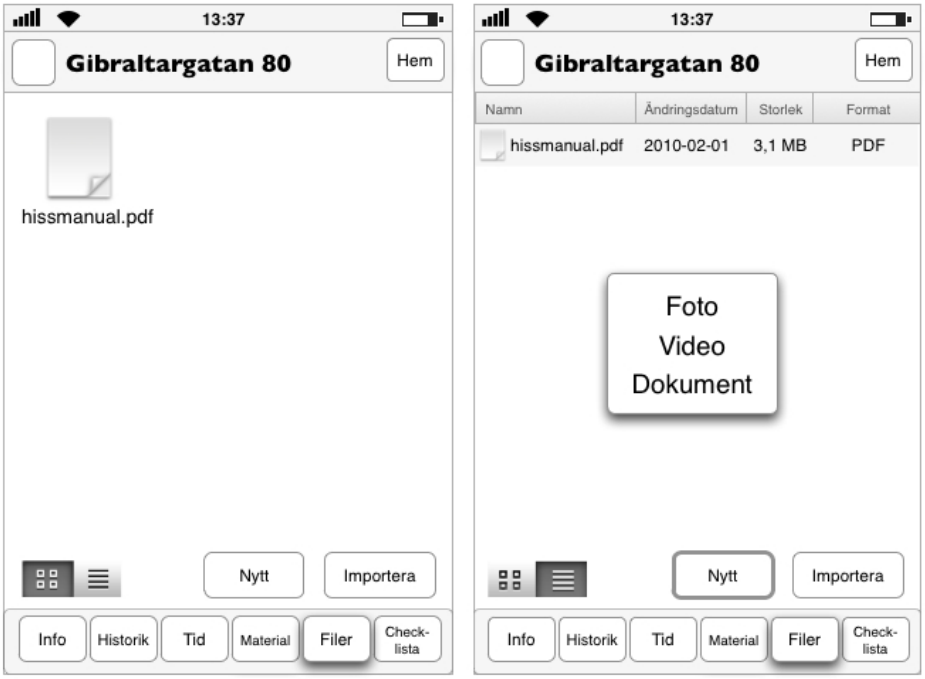


Figure 28 and 29. Viewing attached files. About to create a new file to attach.

Checklist

Checklist points do not go into any detail. Each point only needs a box to check and in some cases there are at most multiple alternatives to choose from. Once all points are checked a button for finishing the work order should be made clickable.

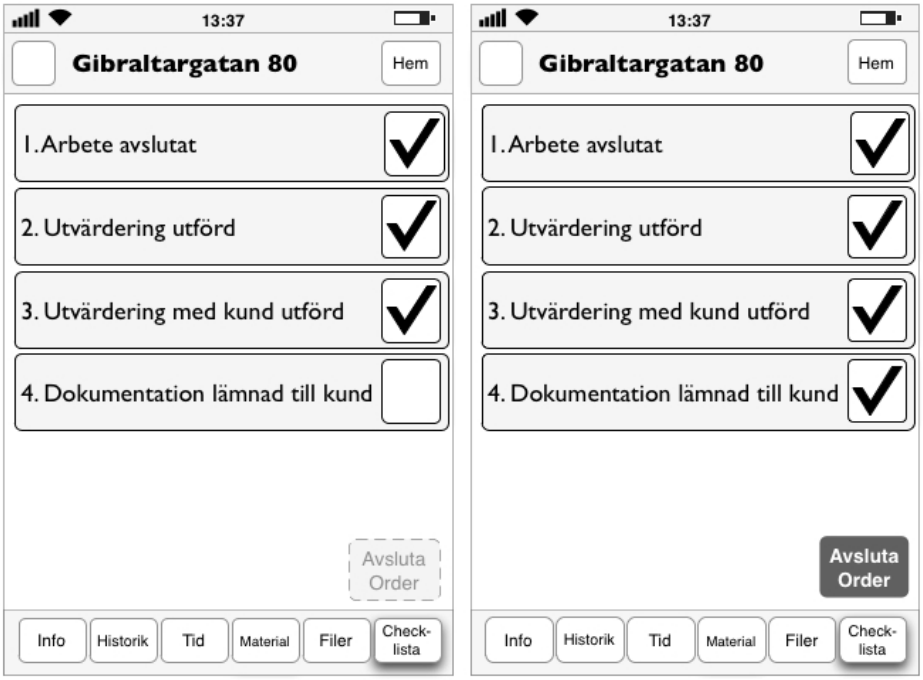


Figure 30 and 31. About to check the final box. All checked and the order can be finished.

Material And Equipment Handling

Designing a good abstract material handling solution is challenging since the needs for different companies differ a lot.

For each material section a list of material previews should be shown. The information given in the preview of an item entry should include:

- A photo thumbnail.
- Number of available units.
- Its article name and number.

For quickly moving a single unit of an item back and forth between the work order, the user's car and the office storage it should be possible to drag-and-drop the item from a list to the wanted tab.

For moving more units of an item the mentioned multi-functional slider components would be suitable due to the low cost in screen space.

When selecting an item, more in-depth information should be displayed in addition to the information given in the preview. This information should be:

- A photo.
- The manufacturer.
- A detailed description.
- Article name and number.
- Number of available units.

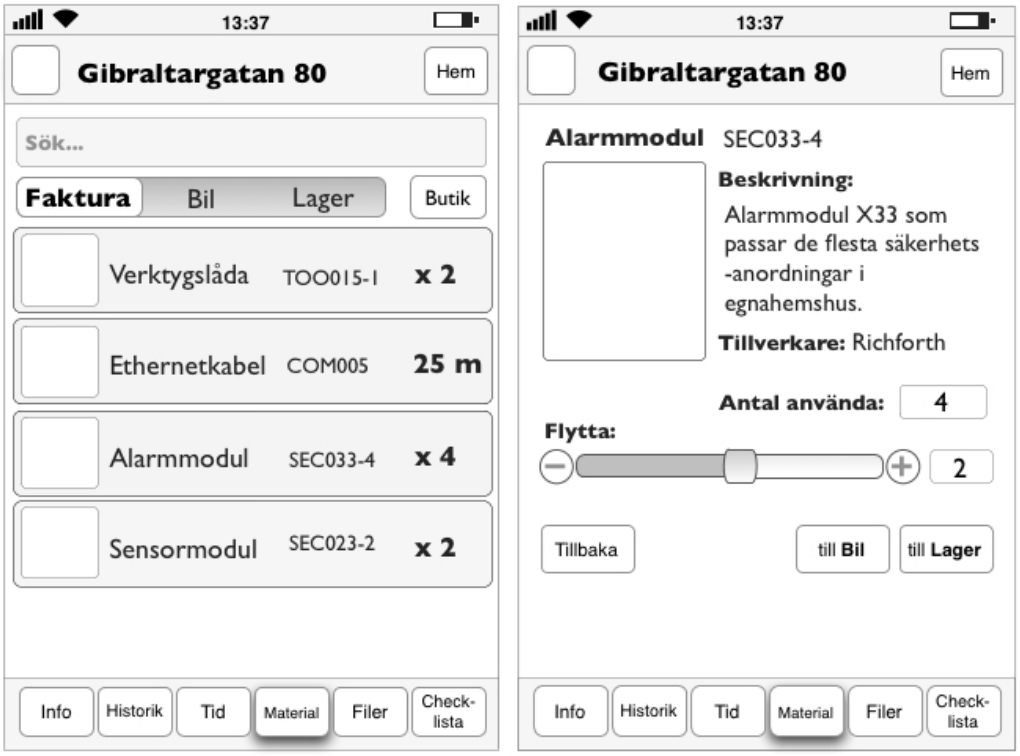


Figure 32 and 33. Viewing the items assigned to the bill of the work order. Viewing an item in detail.

Material And Equipment Ordering

Designing a good solution for equipment ordering is a huge challenge since the possibilities are very dependent on the ordering system the customer company is using and what data their equipment providers are offering through it.

In our design solution, the user is able to search for the wanted item, view its information and specify the amount of it to add to the shopping cart. When finishing the equipment order it can be specified to where the items should be delivered. For example, the current work site or the local office.

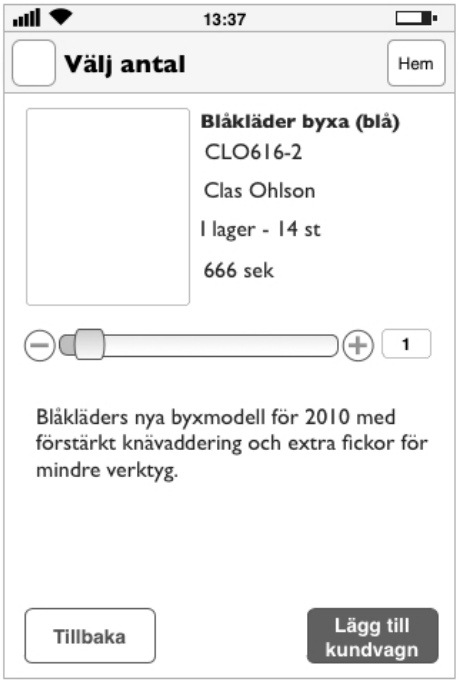


Figure 34 and 35. Searching for the wanted item. Selecting amount to add to shopping cart.

Figure 36. Viewing the shopping cart.

3.1.2.2 Icons

It is important to note that icons that usually are associated with certain actions can be interpreted differently depending on the context the user is working in. For example, an icon of some tools is usually associated with settings. For a service provider it likely associated with material handling or ordering. Different shapes of icons will make them more memorable and easier to find as well.

References

Android, 2010. *User Interface Guidelines.* [Online] (Updated n.d.)
Available at: http://developer.android.com/guide/practices/ui_guidelines/index.html
[Accessed 6 May 2010]

Apple Inc., 2010. *iPhone Human Interface Guidelines.* [Online] (Updated n.d.)
Available
at: http://developer.apple.com/iphone/library/documentation/UserExperience/Conceptual/MobileHIG/Introduction/Introduction.html
[Accessed 2 March 2010]

Cooper, A., Reimann, R. & Cronin, D., 2007. *About Face 3: The Essentials of Interaction Design.* Wiley Publishing, Inc.
Microsoft, 2010. *UI Design And Interaction Guide For Windows Phone 7.* [Online]
(Updated n.d.)
Available at: http://download.microsoft.com/download/D/8/6/D869941E-455D-4882-A6B80DBCAA6AF2D4/UI%20Design%20and%20Interaction%20Guide%20for%20Windows%20Phone%207%20Series.pdf

Preece, J., Rogers, Y. & Sharp. H., 2007. *Interaction Design Beyond Human-Computer Interaction*. 2nd ed. John Willey & Sons.

Tidwell, J., 2005. *Designing Interfaces: Patterns for Effective Interaction Design.* O'Reilly Media, Inc.

Appendix B - Components

1 Default Components

This section describes the custom GUI elements. How they work and how they are constructed is mentioned along with their purposes and notable advantages.

1.1 UIDatePicker

The UIDatePicker uses multiple rotating wheels to allow users to select dates and times. It is used in the prototype for selecting dates and setting time in hour and minutes.

1.2 UIActionSheet

The UIActionSheet displays a slide-up message to the user. It is usually used for forcing the user to make a selection of for confirming a decision. It used in the prototype for selecting time type when reporting time and setting the status of a work order.

1.3 UIAlertView

The UIAlertView displays a pop-up alert message to the user. It can contain buttons for forcing the user to make a selection before disappearing. Overall it is just a variation of a UIActionSheet. It is used in the prototype before Google Maps is launched.

1.4 UISearchBar

The UISearchBar displays a search bar for text input along with a button for cancellation and starting the search. It is used in the prototype for filtering work orders.

1.5 UITabBar

The UITabBar is used for dividing information into different sections reachable via the UITabBarItems in the UITabBar. If a UITabBar contains more than 5 tab bar items, they will be listed under the same tab. It is used in the prototype for dividing up the different sections of a work order.

1.6 UIToolBar

The UIToolbar is used for the same purpose as a UITabBar. The difference is that a selected item is only momentarily highlighted or not changed at all. It is used in the prototype for dividing assigned and unassigned work orders.

2 Custom Components

This section describes the custom GUI elements. How they work and how they are constructed is mentioned along with their purposes and notable advantages.

2.1 Custom Icons

These icons are used on the home screen of the prototype and are designed to resemble the icons on the iOS SpringBoard (the main menu in iOS). They consist of a UITextLabel and a custom UIButton. Apart for the icon for work order (top left) they rest of them are identical to other icons with the same purpose on iOS. For example, just as from the SpringBoard tapping the same icon shows the address book. These icons offer the advantage of being familiar. With the label the user is likely to know where they will be brought when tapping them.

Figure 1. The custom buttons at the home screen.

2.2 Work Order Preview Cells

A work order preview cell is a custom made UITableViewCell that displays the most important information about a work order; address, problem description, system entry date and priority. There are three colors for representing the three different priority levels; blue for low, orange for normal and red for high. There is currently no way of seeing which work orders are new. In the UITableView they are ordered first by priority and then by system entry date. They offer the advantage of displaying needed information in a better structured way than the default UITableViewCells.

Figure 2. Preview cells for a work order.

2.3 History Entry Cells

A history entry cell is another custom made UITableViewCell that displays a preview of a history entry. There are two types of this custom cell, notes and old work orders, which are separated by an icon. Apart from that they contain the entry date, author and some text. in the current version of the prototype the work leader for the work order is set as the author. In retrospect it should perhaps be the employee who finished. They also offer the advantage of displaying needed information in a better-structured way than the default UITableViewCells.

Figure 3. Cells for work order history entries.

2.4 Time Report Entry Cells

A time report entry cell is yet another custom made UITableViewCell that displays a preview of a time report entry. A cell displays the report date, the total hours and minutes reported for and a snippet of the optional text note. These cells also offer the advantage of displaying needed information in a better structured way than the default UITableViewCells.

Figure 4. The cell for a time report entry.

2.5 Text Field

A text field is actually a custom made UITableViewCell that is either empty and uses the Input Prompt design pattern including an arrow for telling the user what should be entered (as in the image above) or is filled with entered text.

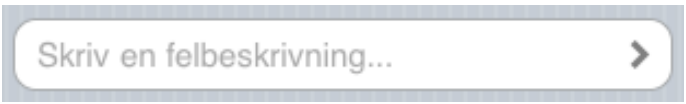


Figure 5. And empty and editable text field.

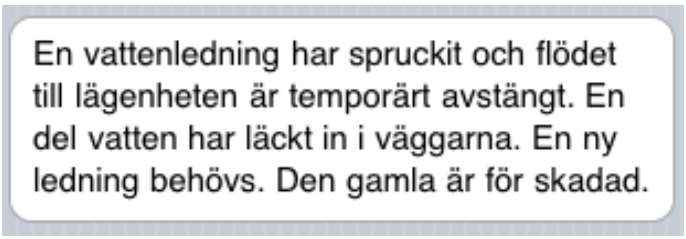


Figure 6. A filled and not editable text field.

When tapped a new view is pushed along with the on-screen keyboard. Once text is entered and the finishing button in the top right corner is tapped the previous view is pushed. The text field at the previous view then contains the entered text.

Figure 7. Entering text.

2.6 Address Input Field

This component is a custom UITableViewCell that consists of 4 regular UITextFields and is used for entering an address when creating a new work order. It offers the advantage of saving screen space and putting all relevant input field in the same area.

Figure 8. The cell for address input.

2.7 Address Information

This component is a custom UITableViewCell and consists of 4 UILabels and a custom UIButton The UIButton lunches Google Maps for viewing either the route description or map location for the work order address. The title of the cell is set as the ID of the work order. The address is presented in the UILabels.

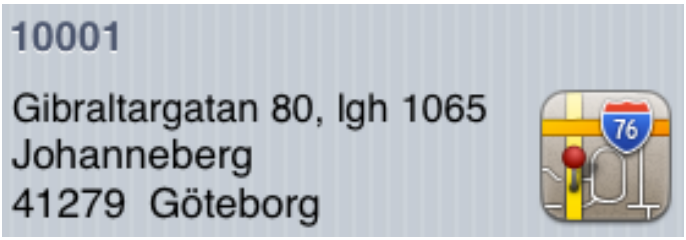


Figure 9. The cell for address information.

2.8 Contact Section

A contact section is a section of a UITableView where each UITableViewCell leads to the address book profile of that contact. The section might offer the user to add another contact by displaying an add button in at the bottom. If tapped the iOS address book is shown. Tapping a user adds them to the contact section. If the contact section is editable, the disclosure indicators are replaced with delete buttons.

Figure 10. The contact section component.

2.9 Date Selector

This component consists of 3 custom UIButtons. One big in the middle for displaying the selected date and pushing a UIDatePicker when tapped and 2 small button at each side for selecting next or previous day. When date is not editable the button are replaced with a standard section header of a UITableView for displaying the date.

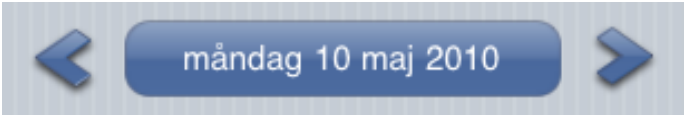


Figure 11. The component for setting the date.

2.10 Time Interval Selector

This component shows all reported time intervals in a time report. It consists of a section of a UITableView with 2 types of UITableViewCells. One type contains 3 UIButton. The first button pushes a UIActionSheet where the time type is selected. The other button displays a UIDatePicker where the time in hours and minutes is set. The third button deletes the entry if there is more than one time interval left.

The other cell type contains a UIButton for adding another cell of the previous type.

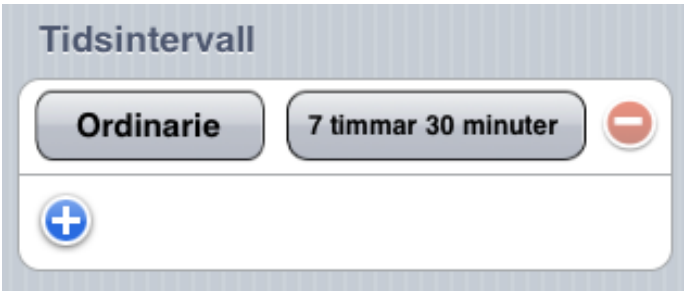


Figure 12. One time interval, hence the disabled delete button.

Figure 13. Two time intervals, hence enabled delete buttons.

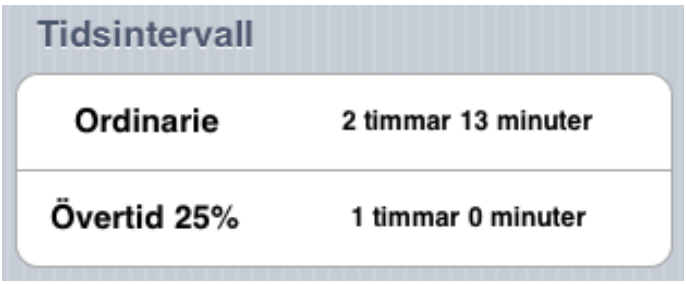


Figure 14. When not in editing mode the information displayed in the UIButtons are displayed in UILabels.

2.11 Checklist

The checklist section consists of UITableViewCells, one for each checklist point. Each cell has a custom image assigned stating if the point is checked or not. In the current configuration the finish button is enabled when all points are checked.

Figure 15. The checklist component.

Appendix C - User Studies

This appendix explains the user studies performed throughout the thesis. How these studies were performed is mentioned, but the focus is on the resulting data.

1 Göteborg Energi

Göteborg Energi was a perfect fit for user studies. They provide wide services and have lots of employees working in the field.

1.1 Interview

An interview was scheduled with contract director Robert Schwartze who has great insight in mobile reporting at the company. The interview went according to plan and resulted in lots of useful information. This was the last of 3 interviews performed in the first round of user studies, which led to a more focused and better documentation.

1.2 Result

1.2.1 Work tasks and Environment

Göteborg Energi AB does all sorts of installation work concerning electricity, ventilation and heating. This work is mainly done on existing buildings, but also at constructions sites. The employees spend as much time indoors as outdoors. Heights and noise are very common elements.

1.2.2 Field Personnel

The field personnel consist of 56 employees of which a large majority is men. Only a few women work in the field. Most of the men are between the ages of 55 and 65 and a smaller group between the ages of 25 and 30. In between there are only a few. The younger group is a bit more interested in technology than the older group, but there is never any chat about tech stuff.

1.2.3 Mobile Reporting Experience

Before using mobile devices the field personnel took paper notes. Office personnel then entered this information in the company's computer system.

Around 2000 an Excel based application was introduced and was a great success for the field personnel. The application had a great GUI and was easy to use. However, the entered information still had to be manually transferred to their computer system just as before, since the Excel system on the devices was not compatible with the economy system.

Because of this a new system was developed. It worked on both the Plam devices and later introduced iPAQs. This was great since the iPAQs were ignored due to their inferior quality.

The mobile client for this system was used until 2008 when the iPAQs were abandoned. Today the desktop version of the system is used, but just as before notes in the field are done using pen and paper. Every Friday they spend 1 to 1.5 hours transferring the information system. The employees, who usually work all alone during the weeks, appreciate this time.

1.2.4 Software Issues

Important to note is that the software never was a reason for the mobile client being removed. Göteborg Energi always maintained a user-centered approach. Meetings were held with the developers and the field personnel for eliciting needs and presentation of information. The slow performance was due to the hardware.

1.2.5 Hardware Issues

When introduced the iPAQs were not met with any hostility, but when put into practice they became hated due to their poor quality. Their memory and settings were erased at random, the screens broke easily and they were too slow for processing the large amounts of data for work orders.

1.2.6 Technology Acceptance

An interesting thing is that some of the oldest employees still use the mobile client with the same Palm devices as they did 8 years ago. At first these employees were hostile against using them and now they will not let go. When the iPAQs were introduced they were familiar with the Palm devices, but were just as hostile to them as they first were to the Palm devices.

For getting the older employees interested in starting to use the Palm devices they were first handed to the younger employees, which were keener on using them. Once the older employees saw how much easier and more time efficient their work got they decided to follow.

Another likely reason for the acceptance was the user-centered approach Göteborg Energi took when developing the client. Since the developers were located in the same office, getting the software modified usually took a few hours or at worst a few days.

1.2.7 Future Plans

A new system for mobile reporting will be brought to use in the not too distant future. This time it will be a purchasable application. Just as before the plan is to involve users in the development process for improving usability.

1.3 Bravida - Electricity Department

Being the largest service provider in Scandinavia, Bravida was another perfect fit for the user studies. Their electricity department had just switched from one application for mobile reporting to another.

1.3.1 Interview

An interview was scheduled with work leader and smartphone responsible Niklas Karlsson. The study went according to plan and lots of useful information was gathered. However, some problems occurred when wanting to record video when their current smartphone application was demonstrated.

1.3.2 Result

1.3.2.1 Work Environment and Tasks

The field personnel work both outdoors and indoors in buildings where additional electrical installations are made. Examples of typical work tasks are repairing lampposts, installing entry phones and mounting lamp switches in apartment houses. Internet access and mobile reception is usually great. When not, accepted work orders cans still be read and modified since they are loaded on to the smartphones when possible.

1.3.2.2 Employees

All employees at the electricity department are male. Their age vary from around 30 to well over 50. A few of the guys around 30 are interested in technology and chat about it once in a while. The older guys are not that interested.

The staff around the age of 55 and above does not use the current application on their smartphones. Instead they connect them to a desktop computer using the program Pocket Controller Pro. They experience the content on the screen small and frustrating and easily get lost when navigating deep in the application.

1.3.2.3 Hardware Issues

About once a week a smartphone needs service. The displays get cracked easily, styluses are not in sync and sometimes the microphones break as well. Styluses also often get lost and are replaced by regular pens or tools like screwdrivers (!).

1.3.2.4 Before Mobile Reporting

Before a mobile application was used work orders were handed out on paper and stored in folders in their company cars. The personnel filled out time sheets once a week where they specified the date, time spent and the work order id. This approach was something all employees were used to.

The drawback was that once the papers were filled out and handed in to the local office the same information had to be entered again into the computer system.

1.3.2.5 AddMobile Toolbox and Handyman

The electricity section of Bravida has been using Handyman for mobile reporting for about a year. Before that AddMobile Toolbox was used for about a year and a half.

AddMobile Toolbox was more adaptable. The developing company, AddMobile AB, was small and modifications were fast when needed. It also had thee features Handyman lacks. One, color-coding for classifying work orders. For example, new orders were marked red and ongoing were marked green.

Two, the personnel could change the status of a work order. For example, "Accepted" and "Ordering Equipment". When changed the customer received an update by SMS.

Three, the office could send new work orders to the personnel whenever needed to. With Handyman the personnel have to manually synchronize with the office.

Handyman is a more complex application compared to AddMobile Toolbox and offers greater functionality. However, since the company behind Handyman, ePocket Solutions, is a much larger company compared to AddMobile AB updates takes much longer time.

1.3.2.6 Handyman Features and GUI

When viewing listed work order too much GUI focus is on their ID numbers. The focus should be on the address and some information about the work task instead of the ID number.

Attach a post-it note to a work order is very usable for quickly writing and viewing important information. For viewing more information about a specific work order without leaving the current screen expander arrows are used for displaying this information if wanted.

Another great feature is the possibility of viewing the work history of a work order when additional work needs to be done. Who did what and when can be seen.

Other useful features include viewing unassigned work orders and accepting the ones you want, creating new work orders and getting directions to a work site using the built-in GPS and map services.

Once a work order is selected different information about it is viewed using tabs. At the tab for handling material the navigation structure complicated. This is where the older employees often get lost.

Handyman users can both view and create all common document types, which is needed when synchronizing information with the computer system. The overview tab shows who is working at the site. Here additional employees can be assigned to the site.

The time tab is used for reporting who you are and how long time was spent on the site. Information about additional costs such as parking tickets and notes about other stuff can be added too.

The tab for additional notes for the office shows notes not to be seen by costumers. For example if an unexpected event would hinder the work for some time or other sensitive information.

1.3.2.7 Complementary Information - Time Reporting

From a later email conversation it was learnt that an employee has to report exactly how many hours were spent each work order. The field personnel are supposed to report the hours spent by the end of the day. However, this varies and they often report for the entire week on Friday or early the following Monday.

1.3.2.8 Paper Prototyping Session

Niklas Karlsson was the only participant in this session too. The performance went well and resulted in lots of great feedback. Due to Nikolas' extensive feedback and meeting time only around an hour some of the scenarios had to be slightly rushed through.

We did not specifically ask what Niklas Karlsson thought was missing, but since this was our second meeting and since he had previously mentioned changes he wanted to be made it feels safe to assume nothing major was missed.

1.3.3 Result

This is Niklas' comment on the GUI sketches.

1.3.3.1 Main Menu

Great overview. All good.

1.3.3.2 Selecting a Work Order

At first sight it is obvious what to do. The color-coding is excellent. There should be different colors for active, new, assigned and finished work orders.

There should be different tabs for viewing your assigned work orders and unassigned work orders available for everyone. The date should be the date when they were entered in the system.

1.3.3.3 Work Order Information

The ID number of the work order should be displayed in the header. Focus is put on the description, which is great.

The connection with built-in address book is great. Field personnel must be able to call the work order contact person.

Status with email or SMS notifications to the customer is well designed. Available status messages should be "mottagen", "påbörjad" and "väntar på material".

It should be possible to assign colleagues to a work order. Adding yourself is not necessary for Bravida since they 9 times out of 10 will assign an employee to a work order directly when creating it.

1.3.3.4 Creating New Work Orders

This varies depending on the system of the company. For Bravida it must be possible to assign the order to a work leader and employee and then categorize it.

1.3.3.5 History

The idea is great. The challenge is to make the associations work. It is important that the addresses are as specific as possible. If not, too many history entries per address will be presented. The type of information in an entry is perfect. "Ansvarig" should be changed into "Utfört av". Seeing the text field for entering a note was not obvious at first hand. Separating general notes and citing specifications is a good idea.

1.3.3.6 Attached Files

Great. Exactly the same thing exists and is often used in Handyman.

1.3.3.7 Checklists

This varies a lot depending on the organization. In Bravida's case it should focus on simple checkbox options, which can be shown without having to click on the entries. No text input should be demanded from the users.

Must-have checklist points for Bravida should be "Arbete avslutat", "Utvärdering utförd", "Utvärdering med kund utförd", "Dokumentation lämnad till kund".

1.3.3.8 Material Handling

The interface solutions are really great and easy to understand. Modifying the bill is smoother compared to Handyman. Office storage could be removed. Several companies like Bravida do not use office storage anymore. The suppliers are many and can be found almost everywhere.

In Bravida's case purchased material is registered to the system, but only the costs. If the staff wants to add the ordered items to their own car they have to do it manually. This is never done in practice.

1.3.3.9 Material Ordering

The interface solutions are good, but the problem is getting it to work on the underlying system layers. The order system has to already exist and then be adapted to the mobile platform.

1.3.3.10 Time Reporting

Time reporting is always done separately for work orders. Accessing this feature from the main menu is not needed.

The factor of overtime must be handled. Depending on the weekday and during what hours there are different types. It must be possible to report ordinary work time and, if needed, different types of overtime for a work order.

Finally, previous time reports should be presented in a list. When creating a new report it should be possible to add comment with information of what was done.

Appendix D – Time Plan

This is the latest revision of the time plan created in the initial phase of the thesis work.

Week 3 (18 - 24 January): Planning, literature studies
Week 4 (25 - 31 January): Literature studies, analysis of smartphone applications, and analysis of Windows Mobile
Week 5 (1 - 7 February): Company contact, planning of user studies
Week 6 (8 - 14 February): Users studies, gather specifications
Week 7 (15 - 21 February): Analyze gathered data
Week 8 (22- 28 February): Design, prototyping
Week 9 (1 - 7 March): Design, prototyping
Week 10 (8 - 14 March): User studies, analyze feedback
Week 11 (15 - 21 March): Analyze concept, implementation preparations
Week 12 (22 - 28 March): Analyze concept, implementation preparations
Week 13 (29 March - 4 April): Report and design concept manual writing
Week 14 (5 - 11 April): Implementation preparations
Week 15 (12 - 18 April): Implementation preparations
Week 16 (19 - 25 April): Implementation of prototype
Week 17 (26 April - 2 May): Implementation of prototype
Week 18 (3 - 9 May): Implementation of prototype
Week 19 (10 - 16 May): Implementation of prototype
Week 20 (17 - 23 May): Analysis of prototype
Week 21 (24 - 30 May): Report writing, preparation for oral presentation.
Week 22 (31 May - 6 June): Submit report and perform oral presentation.

Note, writing the report and design concept manual will be done throughout the entire project.

Appendix E - Personas

William - The Veteran

William is a 58-year-old man that has been working as an electrician for almost 40 years. Most of the time he has spent in Sweden, but he has also lived and worked a few years in Norway, Denmark and Germany, where he met his wife Milla 30 years ago. Now he lives in Gothenburg with his wife and two kids Sven and Erica and works as a work leader and field employee for his on electrical company he started 10 years ago with some old colleagues. His everyday work tasks consist of dealing with fellow colleagues concerning work tasks and other paper work. He also spends a lot of time in the field doing everyday service tasks. He enjoys working alone, going to new sites and meet new people. He's very social and friendly.

William is, however, not a big fan of technology. He feels that paper is the best choice for him to keep track of all his work information, even if it can be a bit cluttered and chaotic from time to time. He has tried several smartphones and Pads and he has hated them all. He feels the displays are too tiny, the stylus pens are difficult to master with his big hands and poor, which are often covered by gloves. He also thinks they break too easily and that the software is too complex and detailed. He is aware of other colleagues using smartphones and liking them. He likes some features but mainly sees only the problems.

Jimmie - The Youngster

Jimmie is an elevator repairman out of school since a few years back. He lives in Stockholm in a small apartment, but often stays at his girlfriend Maja's place. Once she is done with her studies they are planning to move to the suburbs together. The firm Jimmie works at has about 30 employees with often work in pairs when dealing with elevators. Most of the staff does all kinds of service work, but Jimmie is only focused on elevators. He usually works with Anton, a 5 years older electrician who started at the firm when he did.

Jimmie is quite interested in technology. He is active on social medias such as Twitter and Facebook. He uses Spotify on a regular basis and plays games on his Playstation 3. When he was a kid he was even more interested, but more focused on hardware. He built radio-controlled cars with his dad who also works as an electrician.

On work he uses a smartphone for ordering equipment, reporting time and viewing, selecting and finishing work orders. He thinks the software it decent, but could be smarter and have more features. He thinks some procedures are complicated and tedious to perform. Thanks to his personal interest in technology he solves a lot of problems on his own and helps out. He is always looking forward to new models and come sup with suggestions for the board of the firm.