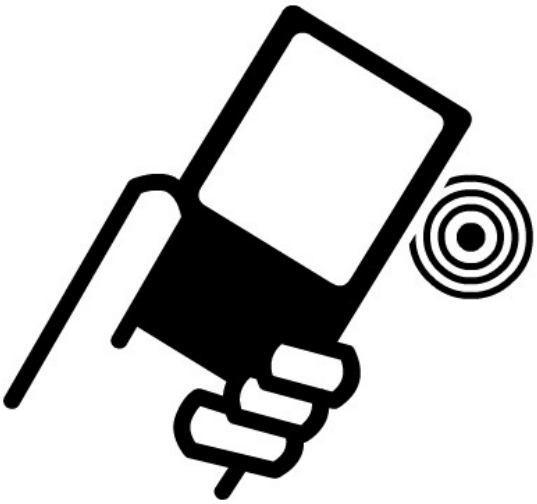
CHALMERS



Usability of Mobile Applications for Near Field Communication

Master of Science Thesis in the Programme Interaction Design

OSCAR LUNDAHL

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Göteborg, Sweden, June 2009

Usability of Mobile Applications for Near Field Communication

OSCAR LUNDAHL

Examiner:
Morten Fjeld
Dr. sc. techn. Associate Professor
Human-Computer Interaction (HCI)
Chalmers University of Technology
Göteborg, Sweden

Supervisors:
Thibaud Cavin
Project Manager at GFI Labs
GFI Informatique
Sophia Antipolis, France

Vincenzo Ciancaglini
Research Engineer on Mobile Technologies
GFI Informatique
Sophia Antipolis, France

Department of Computer Science and Engineering
Göteborg, Sweden June 2009

0.1 Abstract

Todays mobile phones are very complex electronic devices. The fact that they are small to be portable also mean that they are very limited when it comes to presenting and providing easy access to information and features. This is something that makes development of rich mobile applications difficult.

This thesis deals with interaction design and usability of mobile applications with NFC technology. The primary goals are to provide suggestions on how to create usable interfaces for mobile applications and how the NFC technology can be used in such applications to support a good user experience. The approach is to evaluate the use of guidelines and methods for usability in the development process of a prototype for mobile shopping with NFC and QR Code technology. The results are presented as a number of suggestions for interface design and how NFC could be used in future mobile applications.

Keywords: Usability, Mobile Software, Near Field Communication, NFC, QR Codes, Java ME, LWUIT

0.2 Sammanfattning

Dagens mobiltelefoner innehåller en stor mängd komplicerad teknik och avancerade funktioner. Att de är små för att lätt kunna bäras med överallt innebär även att möjligheterna att tillgängligöra information och funktioner blir begränsade. Detta är något som är ett växande problem för utvecklare av mobila applikationer.

Detta examensarbete behandlar interaktionsdesign och användbarhet för mobila applikationer. De främsta målen med arbetet är att kunna bistå med rekommendationer för hur utveckling av användbara mobila gränssnitt bör ske samt hur NFC teknologin kan användas i sådana applikationer för att skapa en bra användarupplevelse. Utgångspunkten är att utvärdera användandet av metoder och riktlinjer för användbarhet i utvecklingsprocessen av en prototyp för shopping med bärbara enheter med NFC och QR Code teknologi. Resultatet presenteras som ett antal rekommendationer för gränssnittsdesign samt hur NFC kan användas i framtida mobila applikationer.

Contents

List of Figures

List of Tables

Wordlist

Abbreviation	**Description**
API	Application Programming Interface - is a set of routines, protocols, and tools for building software applications.
Java ME	Java Micro Edition
JSR	Java Specification Request
JVM	Java Virtual Machine - an engine for executing Java programs (byte code)
LWUIT	The LightWeight User Interface Toolkit - a compact API for creating user interfaces for mobile devices.
MIDlet	A mobile application written for the MIDP
MIDP	Mobile Information Device Profile - a part of the Java ME framework.
NFC	Near Field Communication
OS	Operating System
RFID	Radio Frequency IDentification
Softkey	A button usually located close to the display of mobile devices, it performs a function dependent on the text shown near it on the display.
WTK	Wireless Toolkit from Sun - a PC application with tools for development of mobile software

Chapter 1

Introduction

1.1 Introduction

Since the mobile phone became a mass market product many new technologies have been introduced as a result of manufacturers trying to create more attractive and competitive models. Technologies that were once cutting edge and only available in high-end devices are now standard even in the most basic models, e.g. sms/mms, contact list, media player, camera etc. One property of the mobile phone that has not changed dramatically, at lest not during the last 10 years, is the size which is a main factor for their mobility.

The limited size in combination with the continued integration of new technologies inevitably result in large and complex menu systems. This makes it difficult for users to find and access all features [DAH01]. The advances in mobile technology have also made it possible for third-party developers to create more advanced applications for commercial purposes. To gain competitive advantages it is crucial for such applications to provide interfaces that can achieve a high level of user satisfaction.

Near Field Communication (NFC) is a new technology that most likely will be integrated into mobile phones within a few years. This is a wireless technology derived from RFID (Radio Frequency IDentification). It is not aimed at replacing any of the existing wireless technologies, instead early research projects have shown the possibilities of NFC to open up for new ways to use electronic devices [NFC01, ECM01]. It will enable users to use the phones in physical interaction with the environment to access various services for payment, public transport and data sharing between devices etc. Many companies have seen the commercial potential of this technology. A number of trials with NFC technology in applications for mobile phones are currently in progress all over the world [WIK01].

As this technology is still in an early stage, there has not been much research on how the NFC integration with the mobile phones should be done to suit users needs.

This thesis is based on development of a prototype for mobile shopping with NFC and QR Code technology. It consist of three main parts: a mobile application, a server platform and smart posters. This thesis will primarily focus on the development of the mobile application and interaction with smart posters. The prototype was developed at the innovations lab at GFI Informatique in Sophia Antipolis, a technology park southwest of Nice, France. GFI is an international IT services group employing 10,000 people at the end of 2007 and has positioned itself as one of the leading IT service firms in France and Southern Europe.

1.2 Purpose

As the title reveals this thesis has a focus on usability of mobile applications for Near Field Communication. The main goal is to provide suggestions on how development of such applications can be done in a simple way to ensure that users can achieve their goals with efficiency and satisfaction. The following questions will further define what this thesis will attempt to answer:

- How can usability issues be dealt with in development of interfaces for mobile applications?
- How should the NFC integration with the mobile phones be done to allow users to easily adapt to and use this new technology?
- Are there advantages of NFC over QR Codes, and if yes, what are they?

As a part of the answer to these questions the results from the empirical design process of the prototype will be used to give suggestions on interface design for mobile devices. Additionally, a number of usability guidelines and methods will be evaluated according to how effective they are and how well they can be applied in development of mobile applications.

Although this thesis is aimed at being of help to developers of future mobile applications for NFC there are a few aspects that it may also provide usable feedback for manufacturers of mobile phones and NFC products as several hardware related issues are brought up for discussion.

1.3 Demarcation

The following statements will set some boundaries for interpretation of the suggestions and results presented in this thesis:

- As the Mobile Application of the prototype was developed on the Java platform Micro Edition (Java ME) and specifically with the LWUIT graphical library, the resulting suggestions and conclusions presented here may not be applicable to development in other environments.
- The methods for usability presented in this report are intended to be used by developers with limited resources and with little or no previous experience of usability. Hence, the selection of methods was primarily based on efficiency and ease-of-use. For large scale development projects other methods and more resources would probably be required.

- The late arrival of the NFC-kit used in the project meant that tests of the NFC functionality could not be carried out until the end of the project and were limited in their extent. Ideally, tests should have been carried out in different environments where the system potentially could be deployed.

1.4 Interaction Design and Usability

As technologies are often overly complex for their intended target audience, interaction design aims to minimize the learning curve and to increase accuracy and efficiency of a task [WIK02]. With interaction design frustration over too complex interfaces may be avoided and instead the productivity and satisfaction will increase. As mobile phones are rapidly evolving with new technologies there is a growing need for this kind of research.

Usability is often used as an attribute within interaction design to determine how easy user interfaces are to use. The word 'usability' also refers to methods for improving ease-of-use during the design process [WIK03] . Cooper and Reinmann define usability as:

> "*a process that focuses on measurable characteristics of a user's interaction with a product*" [COO01].

Usability is also defined by the standard ISO 9241-11 [ISO01] and ISO 14598-1 provide relevant information about evaluation of software products.

There are several reasons for why usability is important and should be applied in development processes. It's not only the users who benefit from a product that has been developed with focus on usability. From the developers and managers point of view usability can make the difference between a success or failure of the system being developed.

To ensure a high level of usability it is important to employ an iterative design process that allow changes being made to the design through evaluation of prototypes even in early stages. The evaluation should include actual users and a working system which would provide qualitative feedback to the next design step. However, limited resources may prevent developers from employing adequate usability testing. In these cases there are alternative methods that can be used to ensure that some basic usability criterias are fulfilled.

1.5 Mobile Technology

In the following sections some of the currently existing technologies will be introduced. Focus is then put on technologies that are relevant to the user

interaction: input and output.

1.5.1 Advances in Mobile Technology

The mobile market has during recent years expanded dramatically with high-end models featuring a lot of new technology and are capable of running more advanced applications than the conventional phones. These phones are often grouped into a category commonly known as "smart phones". The smart phones are closing the gap between conventional mobile phones and portable computers as manufacturers are continuously making new models that replace the older ones with new technology and better performance. The following list give some examples of features that can be found in smart phones today:

- Wi-Fi - Enables fast Internet connections via wireless access points.
- Camera(s) - Resolution of up to 10MP, video recording/editing, video calls.
- Push Email - Email are instantly sent to the phone as they arrive to the mail server.
- QWERTY keyboard - A keyboard makes typing easier.
- Touch/Multi-touch screen - For use with pen or finger.
- Accelerometer - Sensor for tilt of the device, can be used to switch between portrait and landscape layout.
- GPS - Can be very useful as a navigational aid when used with an application that provide maps.

A lot is pointing towards a future with phones that have even more features. The smart phones are increasing in popularity, as concluded in a report by Canalys [CAN01]:

> "*These, typically high-end, devices represented around 10% of the global mobile phone market by units in 2007, with annual growth of 60% making them one of the fastest growing segments of the technology industry*".

Despite the depressing economical situation the global shipment of the generally more expensive smart phones increased with 27.9% in Q3 2008 compared to previous year according to a follow-up report from Canalys [CAN02]. This report estimated that the smart phones accounted for about 13% of the market in Q3 2008 with close to 40 million units sold.

1.5.2 Screen Size and Resolution

There is a wide variety of screen sizes on the market. With more computing power and new screen technology todays phones have higher resolution and more colors than their predecessors.

According to an investigation on mobile screen size trends on the market from 2005 to 2008 by Morten Hjerde [COS02], the most common resolution was 240 x 320. This investigation also showed that screens are getting bigger with higher resolution. The increasing use of touch screens is probably a big part of this as they often replace the conventional keys and instead provide a virtual keyboard. In Figure 1.1 a comparison of the resolution of some recent phone models can be seen.

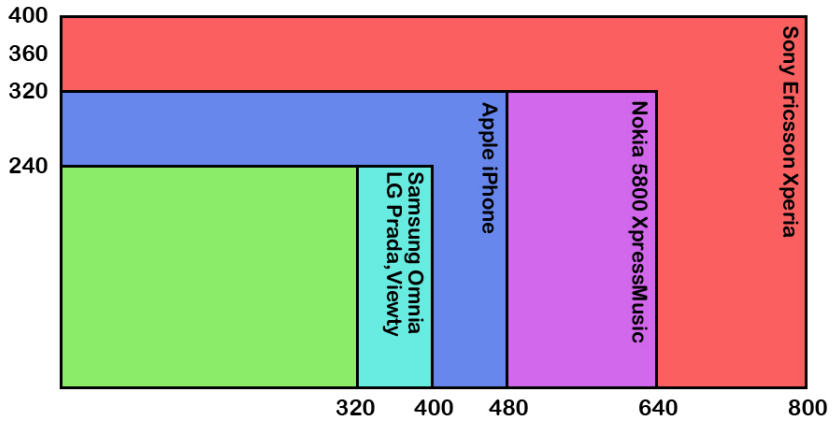


Figure 1.1: Screen Resolution

However, the resolution does not say very much on its own. The physical size of the screen is of course also important, it is commonly measured by its diagonal length. The pixel density[1], the relation between the resolution and the actual screen size, is what determine the sharpness of the display.

1.5.3 Input Methods

There are several different ways of entering data into mobile devices:

- Multi-tap - The conventional way of entering text on mobile phones by - for each letter repeatedly pressing a number key until the wanted letter of the three or four it represents is selected.
- T9 Predictive Text - Is aimed at making it easier to enter text compared to the Multi-tap. By pressing each number key in succession

[1]Often measured in PPI - Pixels per inch.

representing the letters in a word only once, the prediction of the word is made and generally requires fewer number of key presses.

- QWERTY - Devices that have a full QWERTY keyboard provide the user with direct physical access to most characters that then can be entered using a single key press.
- Voice commands - Some devices allow speech recognition for navigation and text input.
- Handwriting recognition - Devices with touch screens may support recognition of letters written on the screen with a pen or finger.
- Virtual Keyboard - A software version of the QWERTY keyboard for touch screen devices that allow users to enter characters by pressing the keyboard displayed on the screen.

The increasing use of messaging services like SMS, email and instant messaging on mobile devices would probably not have had the same impact without the advances made in technology for text input.

1.5.4 Output and Feedback

Most mobile phones have several ways of giving feedback on user interaction and notifying the user of various system states. The screen is continuously updated with information about most activities, but as phones are often tucked away in pockets or bags, more noticeable output methods are required. E.g. sound, vibration or flashing backlight are commonly used for notification of incoming calls/messages, in alarms and games.

1.6 Java ME

The development of the prototype that this thesis is based on involved a great deal of programming in the Java language. For the mobile application Java ME was used.

The Java Platform Micro Edition or Java ME is a Java platform designed for mobile devices and embedded systems [WIK04]. It is aimed at providing a certified collection of Java APIs[2] for the development of software for small devices with limited resources, such as mobile phones and PDAs. Java ME has become a popular option for creating games and applications for mobile phones as they can be emulated on a PC during the development stage and easily uploaded to phones.

[2]Application Programming Interface (API) - a set of routines, protocols, and tools for building software applications.

1.6.1 MIDP and MIDlets

Java ME devices implement a profile. The most common of these are the Mobile Information Device Profile (MIDP) aimed at mobile devices. Applications written for this profile are called MIDlets. Almost all new mobile phones come with a MIDP implementation, and it is now the de facto standard for downloadable mobile phone games [WIK04]. However, many devices can run only MIDlets that have been approved by the carrier.

1.6.2 JSRs

A Java Specification Request (JSR) is a formal document that describe specifications and technologies of the Java platform. Some JSRs that describe technology discussed in this thesis can be found in Table 1.1.

JSR#	Name
68	Java ME Platform Specification
75	File Connection and PIM
82	Bluetooth
118	MIDP 2.x
135	Mobile Media (Camera)
172	Web Services
257	Contactless Communication (NFC/RFID)

Table 1.1: JSRs of interest

1.6.3 LWUIT

The Lightweight User Interface Toolkit (LWUIT) is a compact API for creating user interfaces for mobile devices on the Java ME platform. LWUIT supports visual components and special user interface functions such as theming, transitions and animation. It is freeware and comes with a demo and an application for creating themes.

1.7 NFC

Near Field Communication (NFC) is a short-range wireless connectivity technology that enables exchange of data between devices and tags over a distance of up to about 10 centimeters. It's mainly aimed at making it easier to use services for payment, public transport and data sharing between devices but a number of other uses have been proposed [WIK01, NFC01]:

- Identity documents
- Health monitoring & Identification of medication

- Mobile commerce, guiding of consumers in retail
- Time and attendance applications
- Electronic keys - car keys, house/office keys, hotel room keys, etc.
- Configuration and initiation of other wireless network connections such as Bluetooth, Wi-Fi or Ultra-wideband

1.7.1 Features of NFC

NFC can be used in both read and write mode, operating at 13.56 MHz with data transfer rates at 106, 212 or 424 Kbits/second. Communication between two NFC-compatible devices occurs when they are brought within about 5 centimeters of one another. A swipe or touch can establish an NFC connection. The underlying layers of NFC technology follow universally implemented ISO, ECMA, and ETSI standards [WIK01].

1.7.2 The NFC Tag

The NFC tag is a passive component that usually store data that can be read by a an NFC-enabled device via a temporary wireless connection. The NFC Forum[3] have identified four tag types with different properties regarding data transfer rate, storage capability, security and price [NFC02]. The tags are manufactured in different size and shape depending on the application, in Figure 1.2 some examples are shown.

Figure 1.2: NFC tags

[3]A non-profit industry association advancing the use of NFC technology. www.nfc-forum.org

1.7.3 NFC Devices

The NFC technology may be implemented in a number of everyday devices to make them able to communicate with NFC tags or each other, e.g. phones, digital cameras, vending machines, ATMs or parking meters. NFC enabled devices may change operating mode between the following:

- Reader/Writer - the NFC Device is able to read and write to tags.
- Peer-to-Peer - allows two devices to to exchange data.
- Card Emulation - the device itself can act as an NFC tag.

This thesis will only deal with the use of the Read/Write mode on mobile phones and the interaction with the passive tags on smart posters.

1.7.4 Security

Although NFC is limited to only a few centimeters distance for communication and tags can be configured to be read-only, there are several security issues that need to be addressed. In a presentation by Collin Mulliner at the EUSecWest conference in 2008 [MUL01] the following remarks were made about NFC security:

- Passive tags are primary targets for attacks:
  - Replacing existing with malicious tag.
  - Putting malicious tag on top of existing tag.
  - Hijacking existing tag.
- Bugs in NFC Phones can be exploited.
- NFC phones can be attacked by: Phishing, malware, worms, denial-of-service.

Issues regarding the security of the NFC technology is of great concern to the public, especially when it comes to payment applications. This was confirmed by a study of the acceptance and usability of physical mobile applications by Herting and Broll [HER01]. As recommended by Mulliner, the early users of NFC should always check the exact content of read tags and be very observant before continuing to use the associated service.

1.7.5 NFC Today and in the Future

Today there are only a few phone models with NFC available on the market, mainly acquired by developers for test projects and research, however there are indications that NFC technology will be integrated into an increasing number of phones in a near future. The NFC-forum provides the following predictions on future NFC integration in the mobile market [NFC03]:

- A recent study by ABI Research projects that 450 million mobile phones will be NFC-enabled by 2011, representing nearly 30% of handsets shipped worldwide in that year.
- Strategy Analytics forecasts mobile phone-based contactless payments will facilitate over $36 billion of worldwide consumer spending by 2011.
- According to research firm Frost & Sullivan, one third of all mobile phones will be NFC-equipped in a span of three to five years.

There are currently several trials of the NFC technology with mobile phones in many countries [WIK01]. These are generally concentrated on applications for payment and ticketing.

1.8 QR Codes

The QR Code was created by the Japanese company Densi-Wave in 1994 as an attempt to solve the growing need for storing more information in bar codes. As regular bar codes only contain data in one dimension, the 2-dimensional QR Code matrix holds a greater volume of information without taking up more space. Figure 1.3 shows a QR Code that contain the message: "*This is a QR Code*".

Figure 1.3: QR Code

Initially QR Codes were used to track parts in vehicle manufacturing and logistics. With the introduction of camera phones and free decoding software the QR Code technology became available to the public. The QR Code is now used for various commercial applications aimed at mobile phone users.

They can be found in magazines, posters, business cards, etc. The content of QR Codes in these applications are often URLs that may link to a website with information about a product or an event. This act of linking from physical world objects is known as a object hyperlinking or physical world hyperlinks. To read a QR Code, a camera phone can be used with software to scan and decode the image and then take appropriate action depending the content, e.g. launch the browser for URLs or show options to call/send sms for phone numbers.

There are several free QR Code generating sites that also provide decoding software [KAY01, SNA01]. This makes it possible for anyone to make their own QR Codes for others to decode.

Chapter 2

The Prototype

In this chapter the process of building the concept will be explained as well as the various parts of the prototype and how they work.

2.1 Defining the Project

A number of suggested projects for development at GFI Labs were presented a few months before any actual work began. Some were about web development and others about mobile software. Both the developers to be assigned to the project and the managers coming up with the project ideas found the NFC and QR Code technologies for mobile phones to be the most interesting and also to have a great market potential. Hence the decision was made to initiate the project based on software development for mobile phones with the NFC and QR Code technologies.

The concept of mobile shopping was also presented in the first description of the project. The initial idea was that QR Codes and NFC tags were put on products in stores. This would allow customers to download information about products to their phones, e.g. reviews or for music and movies samples could be provided. As the user added a product to the cart it would also be added to the virtual cart. The NFC technology would then be used at check out for payment, a simple touch on the cash register would complete the transfer using a virtual payment card [NFC01].

The project was started off by brainstorming, this was done in groups of 2-5 people consisting of developers and managers. This resulted in a wide variety of possible use-cases. As these use-cases were further discussed and defined in the beginning of the project, it was realized that the the concept could be applied in a wider context. There was no need to limit the shopping to take place in stores only. With smart posters representing online products the shopping could take place anywhere. By placing smart posters in strategic places connections could be made between the physical world and the online shopping systems. Today many stores have virtual online shops in parallel with the physical shops. This was the case with the clients initially identified for the first concept that were in the music and home entertainment business.

The expanded context of use meant that the project would also be of interest to companies specializing in online stores. Links (URLs) to their products could be put on smart posters that would then be read into customers phones with NFC or QR Codes in a first step. The wireless technology for accessing Internet which already exist in most mobile phones could then be used to access product information or to place an order.

The main goals of the project were to have a working prototype con-

sisting of a mobile application that could read information from NFC tags and QR Codes for product identification and communicate with a back-end server. The different context of use were not essential when determining these goals. Instead a number of use-cases made up the requirements for the final prototype, these are further specified in section 3.1.2. Some were added, others removed or modified during the development process as obstacles changed the conditions for their implementation.

2.2 Planning

A rough planning of the project was made during the startup phase. Due to several uncertain factors it was hard to create a more detailed plan for the project, e.g. how much time each use-case would take to implement and if it was possible at all.

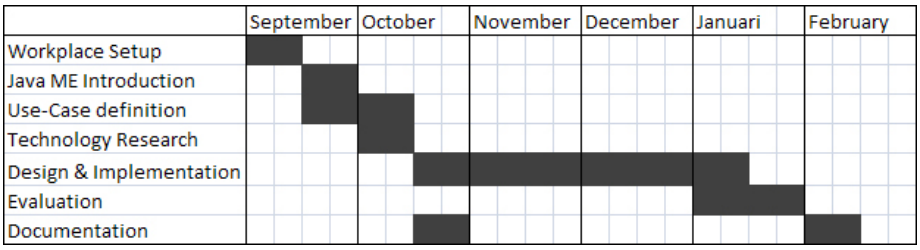


Figure 2.1: Project planning

2.3 Prototype Overview

In this section the different parts of the prototype are explained, more details about the design and development process will be presented in chapter 3 starting on page 32. The prototype consist of the following three main parts:

- The Mobile Application - an application for mobile phones that enables the user to interact with smart posters using NFC and QR Codes.
- The BackOffice - the server side implementation that provide online services for mobile clients.
- Smart Posters - Posters with NFC tags and QR Codes that normally contain information about a featured product.

Figure 2.2 shows the basic use-case of reading a product and the services that then become available. Reading a product from the smart poster can be done either by using an NFC capable phone and touching the tag on

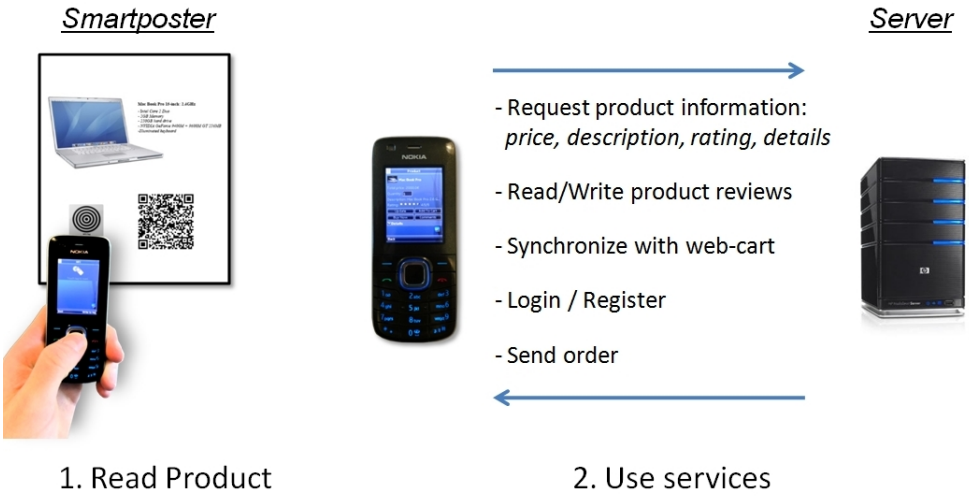


Figure 2.2: Prototype concept

the poster, or by taking a photo of and decode the QR Code. Once the product have been successfully decoded it can be added to a virtual cart in the mobile application.

Products that have been read into the phone can be updated using the update service. Information about the product is then requested from a server and loaded into to product view on the phone. A product update include the following: price, product icon, description, rating, and a number of sections with details if available.

Figure 2.3 shows a technical overview of the prototype. The system provide users with two possible ways of accessing the services: either as PC clients, or as mobile clients. As previously stated this thesis will focus on interaction by the mobile clients, the web site for PC clients was not fully implemented.

2.4 The Mobile Application

The Mobile Application is a MIDlet that was developed on the Java platform Micro Edition (Java ME) and most of its graphical content was built with components from the LWUIT graphical library for Java ME. The application comprises six sub-applications, each represented by an icon in the main menu as seen in Figure 2.4.

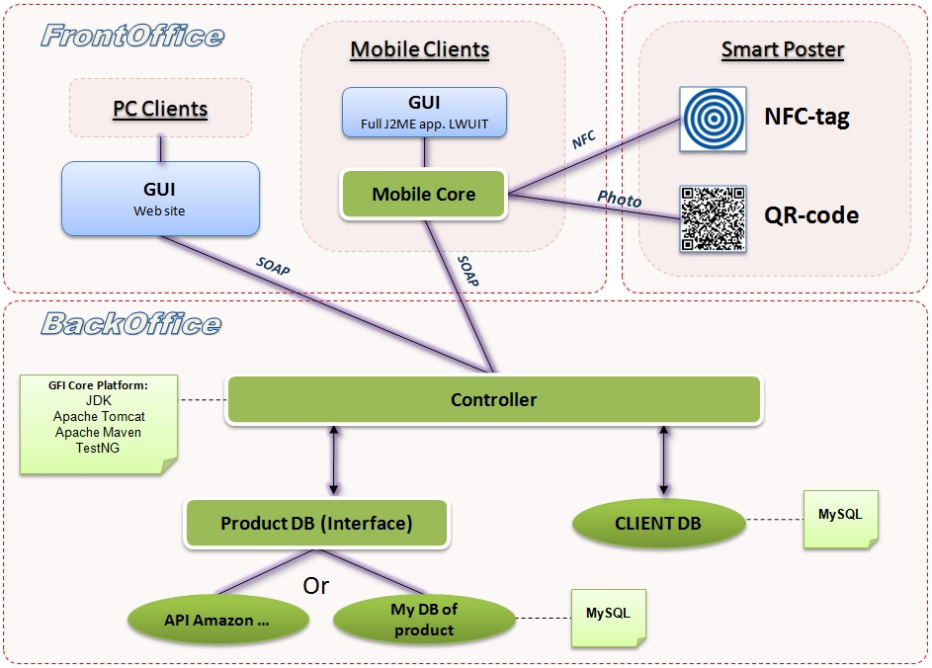


Figure 2.3: Prototype technical overview

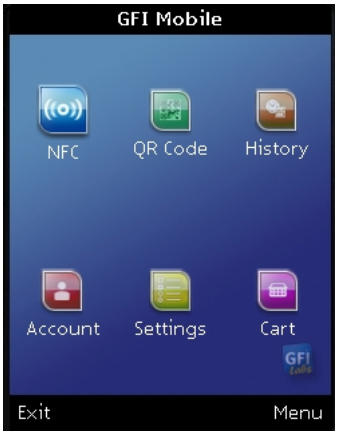


Figure 2.4: Main menu

2.4.1 Views and Transitions

Each sub-application has its own view, which basically is container for various components of the LWUIT library. Additionally there are a few other views that are accessible through some of the sub-applications, e.g. the product view that presents a product and the comments view for product comments/reviews. The menu and view hierarchy for navigation is shown in Figure 2.5.

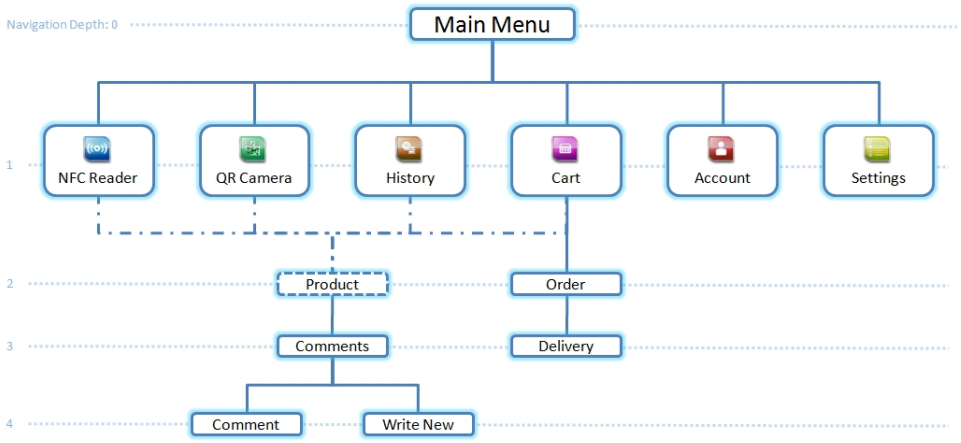


Figure 2.5: Menu and View hierarchy

When navigating from one view to another, transitions are used to help support a mental model of the menu hierarchy. When moving to a deeper level, the view is animated in from the right. Correspondingly, the view of the higher level is animated from the left when going back.

2.4.2 The NFC reader

The NFC sub-application is mainly used to read NFC tags. It requires an NFC capable phone, e.g. the Nokia 6212 that was used for evaluating the prototype. Once the read mode is initiated the user only need to touch a tag to read its content. It is also possible to go into write mode where the user can enter a message, URL, phone number or a product data string[1] to write to a tag.

2.4.3 QR Code

The QR Code camera is used to take photos of QR codes. Depending on the phones operating system and capability with Java ME APIs, the application may do this in two different ways: for devices running Windows Mobile an external application will be used to capture the image, otherwise the application will try to access the camera using the MMAPI[2]. To decode the captured image a free QR Code library was used [QRC01].

[1] A string of data formatted to identify products used in the project, more about this can be found on page 47.

[2] The Mobile Media API extends the functionality of the J2ME platform by providing audio, video and other time-based multimedia support.

2.4.4 History

The messages of recently read NFC tags and QR Codes are stored in the History sub-application. Since the content on the tags does not require very much memory space in comparison to what is usually available on most phones, 10 to 100 NFC and QR messages can be saved to the phones memory without any risk of running out of space.

2.4.5 Cart

Once a product is detected in a message from an NFC tag or a QR Code it is displayed in the product view. Then the user may choose to add the product to the cart. Products are stored in the cart until an order is sent or removed by the user.

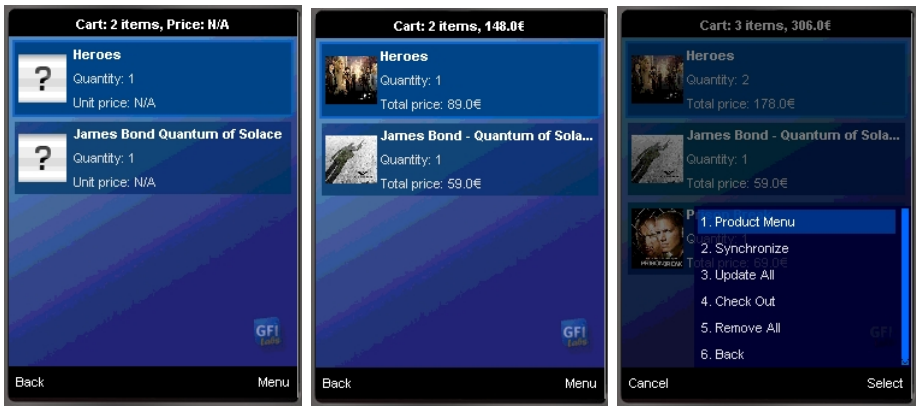


Figure 2.6: Cart view

The picture to the left in Figure 2.6 shows the cart with products that have been added after being read from a smart poster, the picture in the center after both products have been updated with information, the picture to the right shows the menu of the cart.

Check out

From the Cart the user can continue to the check out procedure. This requires an update of all products even if they already have been updated once to make sure that they have latest price. If the price or any information about a product has changed the user will be notified. The last step of the check out procedure is to fill in the payment and delivery information. Once registered this information would be already be available, but a first time user would have to fill it in. For ordinary mobile phones entering information in text fields is a time consuming and rather complicated process. However, with the website previously mentioned it would be possible to fill in this information on a computer once instead.

Synchronization

The Synchronize command in the Cart menu enables synchronization of products in the mobile cart with products in the users online cart. Products that have been added to the online cart using the website would then be downloaded to the mobile cart and the other way around. This was actually a very complex operation as products could be added/removed/modified on both the server side by a PC client[3] and in the cart stored on the phone.

2.4.6 Account

To be able to access the Account the user have to login[4]. This also requires a network connection which for third-party applications in most cases have to be granted by the user, a dialog is usually displayed on the screen that prompts the user to allow network access for the application. Once this is done the account details of the user will be loaded from a server, e.g. name, login information, addresses, credit cards.

2.4.7 Settings

The need for customizing the application was initially not highly prioritized. It was the implementation of support for different languages that lead to the creation of the settings. Then more categories were added gradually:

- Language - English, French
- Camera - Camera resolution
- Themes - Graphical appearance of the application
- Security - How personal data is saved etc.
- History - The number of messages to save
- Reset - Reset the application

More about the settings can be found in the Analysis chapter on page 40.

2.4.8 Product View

After a product have been read from an NFC tag or QR Code, the product view is displayed with the name of the product. An example of this can be seen in the picture to the left in Figure 2.7.

[3]Users should be able to log in to an online store with the same login details as used with the mobile application.

[4]This is only required the first time as login details are saved and loaded between sessions.

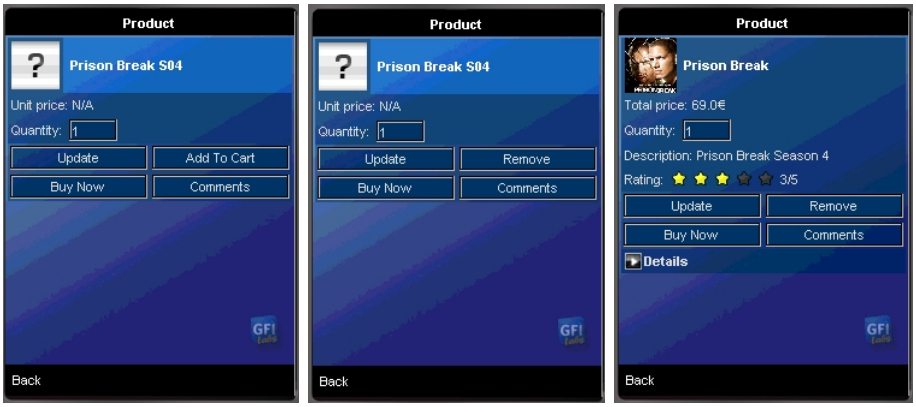


Figure 2.7: Product view; L: Not in Cart, C: Before Update, R: After Update

Depending on the parent view and the product properties, the product view contain four buttons with the following options:

- Update - This option is always available, it updates the product view with information from a server (requires Internet connection).
- Add to Cart / Remove - If the product is not in the cart the button caption is 'Add to Cart', otherwise it is 'Remove'.
- Buy Now - Allows the user to buy a product directly after reading it with NFC or QR Code without having to add it to the cart.
- Comments - Displays comments/reviews posted on this product.

The product view also contain an editable field for quantity. If customers of a product would like to buy more than one they only need to enter the quantity in the field instead of repeatedly reading the product with NFC or decoding its QR Code. For products that have been updated, the product view provide more information as shown in the rightmost picture in Figure 2.7. More information about the product can also be displayed by expanding the details section at the bottom of the view. This product only have one section for details, but some products may have several.

2.4.9 Gestures

For devices with touch screens two simple gestures can be used for navigation, a forward (left to right) and backward (right to left) stroke. The forward stroke is associated with moving to a deeper level in the navigation hierarchy while the backward stroke is consistently used to go back to a higher level.

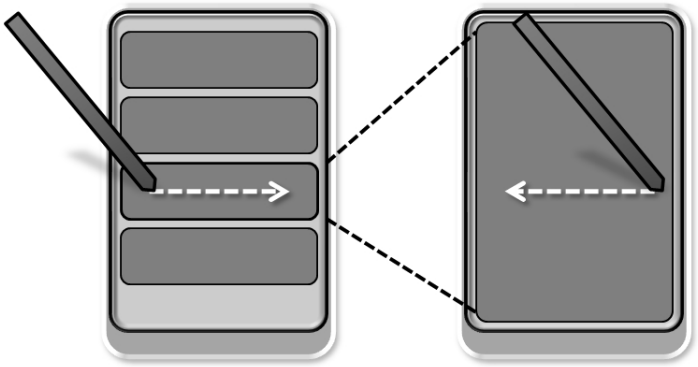


Figure 2.8: Gestures

The forward stroke is used in lists like the one for products in the Cart view. The list element in the position where the gesture is performed is selected and displayed in a new view as illustrated by Figure 2.8.

2.5 The Backoffice

The server side platform of the prototype was called 'The Backoffice', illustrated in Figure 2.3 on page 25. It consist of the GFI Core Platform and a number of packages that can be added for additional functionality, e.g. email service. This part of the project was intended to be reused and modified to suit different needs of future clients.

The Backoffice was set up to serve requests from both mobile and pc clients. The requests could be for product information, access of an online cart, user registration, etc. The Backoffice is generally not something the user have to know about or even notice, hence it does not play an important role in an evaluation of the system from a users perspective and will not be explained in more detail in this report.

2.6 Smart Posters

The smart posters created for the project were based on NFC-tags from an NFC starter kit [NEX01]. Posters were printed on A3 paper with an image and information about the products. QR codes were directly printed on the posters as well as a target marks to indicate the position of the NFC tags. The tags were then fastened on the back of the posters with glue.

2.7 Project Organization

The team working on the prototype was composed of four people. The work and responsibilities were divided between the team according to Figure 2.9.

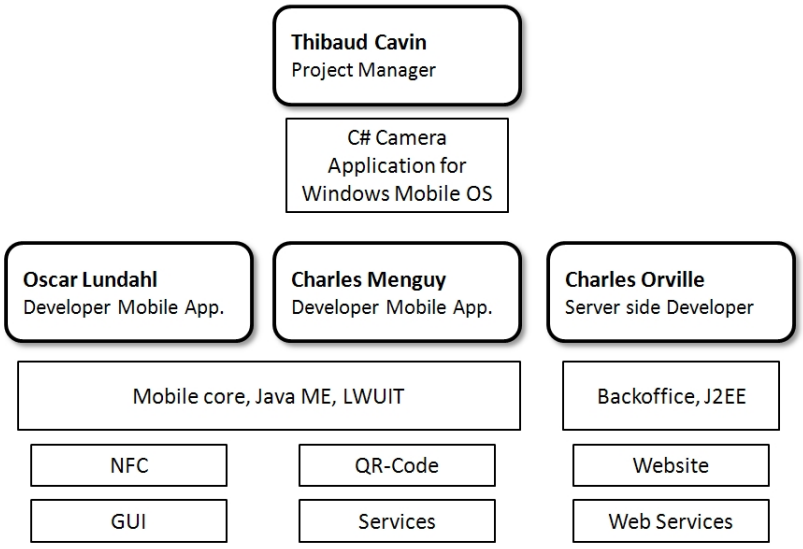


Figure 2.9: Team organization

The project manager had the main responsibility of the project but also for a solution to acquire the camera image for QR Codes on devices running Windows Mobile. Two developers worked together to build the mobile platform, and one with the server side platform (The Backoffice).

In addition to the team members a few technical experts from the company were consulted for advice on development of some parts of the prototype.

Chapter 3

Analysis

3.1 Users

To have an understanding of who the users are is significant when designing an interactive system. Their knowledge and experience of using similar systems should be compared to the tasks they have to perform in the system being developed.

3.1.1 Defining the Users

The following user categories are introduced to make it is possible to identify expectations, needs and what issues can be expected to be more common with user groups with different experience:

- Expert user - wants to use the latest and advanced features.
- Intermediate user - use basic features frequently, sms mms camera.
- Novice user - generally only use the phone for making calls and sms

The products used in the project were aimed at users that were considered to be mainly intermediate or expert users with an interest in technology such as the NFC and QR Codes that were used in the project. Additionally, these users were assumed to be used to buying products over the Internet and be familiar with technologies that are required for this.

In the initial concept the users would be the customers of the multimedia stores which were assumed to be expert or intermediate users. The change of concept to the smart poster would provide stores with greater means of reaching a wider audience and attracting new customers, i.e. new users. This would possibly include in an increasing number of novice users. However, the development of the prototype presupposed that the users would have a higher level of experience.

3.1.2 User Tasks

For the prototype the following use-cases were considered to define the most common and most important tasks the users would perform.

Read a product

To read a product would require the use of either the NFC or QR Code functionality of the application. This is the most fundamental use-case and also something that very few users were expected to have previous experience of. Section 3.3 about smart poster interaction deals with various aspects of this use-case.

Update a product

This is nessecary to retrieve the information about a product that is not stored in the NFC-tags and QR Codes. Although this is done automatically when an order is about to be sent, it is assumed that the user in most cases would like to update the product before this stage. As this requires a working Internet connection it would be beneficial if the user was aware of this.

Remove products from the Cart

The removal of products from the Cart is a basic use-case and most likely something that most users would want to do and be familiar with.

View detailed information about a product

The need for, and importance of product details in the expandable sections vary between different products, e.g. most users would probably be more interested in reading the details about a laptop than a movie. Most users are expected to have knowledge of and recognize this as similar designs are used on web-pages and in PC applications.

Change quantity of a product

In the case where the user would want to buy more than one entity of a product the possibility to change quantity is significant. In most web-based shopping systems this is possible. Again it can be expected that the need for this may vary from product to product.

3.2 User Interface

The following sections describe the interface design process, problems encountered and how they could be solved.

3.2.1 Selecting UI Components

A graphical user interface usually consist of a number of components for data visualization and/or input. When building the interface, designers often have more than one component to choose from for each task. E.g. a component for entering a number could be either an input field or a drop down box. Most graphical components used for the Mobile Application were taken from (or based on) the LWUIT graphical library.

In the beginning of the development process, use-cases were converted into a simple paper prototype where all parts of the application were illustrated with arrows for operations and navigational structure to link them together. Each arrow represented an action that would connect the active view to the next, e.g. the 'Comments' button in the product view would link to the comments view. These paper sketches were used as blueprints for the application, not only for development of the GUI but they also described a lot of how the logic and functionality of the application should work.

To get an understanding of the possibilities with Java ME and the LWUIT library, a first attempt to make an interface for the shopping application was made early in the project. This was based on a single view containing sections with labels that linked to other views. This solution proved to be hard to navigate and did not work well with all screen sizes. The design was intended to provide an overview of all parts of the system, but on smaller screens some parts were pushed outside the view forcing the user to scroll between the sections. Although this was not a good solution, there were several parts that could be used later in the project, e.g. the concept of views.

Main Menu

For the final prototype the 6 sub-applications were created and a grid icon layout was chosen for the main menu as seen in Figure 2.4 on page 25. This layout is similar to what is used on computers and also the main menu for many mobile phones. It provides an overview of the sub-applications and the icons together with descriptive names makes it easier for the user to recognize and remember what functionality each application provide.

Icons

The icons used for the main menu were created in Photoshop and designed to be as close to a final look and attractive as possible. From a marketing point of view this was quite important as the prototype would be used in demonstrations to potential customers and the main menu icons would give a first impression of the look and feel of the application.

In a presentation of how the the iPhone is used the results of user tests confirm the importance of the icon design for applications to the end users by a number of user quotes [IPH01]:

> "- *If it has a poorly designed icon, I'd go right past*"
>
> "- *Nice and crisp icons does have a big impact on my actually going to look at what that is.*"

> "- *But when looking to launch an app, simple names were found more quickly*"

This last quote also help motivate why icons should be used in combination with text.

The icons were given a 3D-look by using shadows and glossy reflections to stand out from the background. To provide feedback on interaction with the icons they become resized: a selected icon becomes larger and the text changes color, when pressed the icon becomes smaller than the default size before launching the associated sub-application.

The application also support repositioning of the main menu icons by drag and drop for touch screen devices. This allows the user to customize the menu and place more frequently used sub-applications higher up in the grid. However, as the icons are nicely animated between positions this functionality was mainly implemented as 'eye-candy' and not really intended to be practical.

View Menus

When more than two commands are added to a view a menu is automatically created for the right soft key. Each item in the menu is associated with a number that allow users to use the number keys as shortcut. The use of shortcuts help speed up navigation and is one of Shneidermans golden rules (A.1.2). The 'Back' command is always located on the left soft key for all views.

It it important to keep the number of menu items to a minimum, especially in mobile applications as a menu with many items easily take up the entire screen on devices with lower resolution. Another reason to minimize the number of menu items is to make it easier for the user to remember what options are available. As a reference it has been proven that the human cognitive memory holds $7(\pm 2)$ items [DIX01]. The menu for the cart view contained 6 menu items which was also the maximum used in any menu of the Mobile Application.

Submenus that are commonly used in PC applications with a lot of menu options are not as suitable for mobile applications. Considering the variety of mobile screen sizes, there is a great risk that submenus would be pushed outside the screen, forcing the use of scrollbars which would make navigation slower.

A better solution would be to try to break the application down into parts according to content (e.g. with views or tabs) that would be more

easily navigated.

For the actions 'show' and 'remove' that can be applied to each product in the cart view, a special menu was created. It is opened either by pressing the fire-key on the selected product or from the view menu.

The first design alternatives suggested for these actions were to either add them to the main menu or to create a submenu. However, neither of these suggestions would have a clear connection to the selected product and they would result in an overpopulated menu. Other advantages of the special menu that was created was that it would not block as much of the underlying view and only require one key press to be displayed whereas the submenu would require at least two.

Text and Input Fields

The text field components from the LWUIT can be set to have various properties. One interesting property is the auto-scroll that enable a selected text field to scroll the text when it does not fit within the field width. This was very useful for displaying text for product details. If the user wanted more information the text filed could be selected and the text would start to scroll.

Another property that is important for security reasons is the password property. This will mask any text that is entered and display in a text field. Although mobile phones have small screens that should make it harder for other people to see what you are typing, they are often used in locations that makes this easy, e.g. in buses. This property was used in the Mobile Application for the password at login.

Combo Boxes

The LWUIT Combo box component is actually a drop-down list that allows only one selection at a time, when a user clicks the combo box button a popup with the full list of elements allows the selection of a single element.

This component is most useful when there is a limited number of options to choose from. If too many elements are added, the combo box become hard to navigate although scrolling is supported. What is advantageous with the combo box is that there is no need to check that the input is valid in the same way as e.g. a numerical input field often requires. Combo boxes were used in the Mobile Application for selecting product ratings (numbers 1 to 5) and selecting existing addresses and credit card numbers at check out.

Buttons

On phones without touch screen technology the placement of buttons is particularly important. If a lot of components are placed around the button

the user would have to to navigate past a lot of these components before the button can be pressed which would be both time consuming and frustrating.

In the Account, buttons were initially placed at the bottom of each tab for saving changes. This was not a very good solution, the button was out of view, forcing the user to scroll down to be able to see and press it. Instead a command was added to the Account view menu for saving account settings. This way, the redundancy created by having buttons for the same action on all tabs was avoided. Additionally, a check for changes was added to make sure no changes are unsaved. When leaving the account, a dialog is displayed asking the user to save or discard changes.

Dialogs

Dialogs from the LWUIT are displayed on top of other components, often as a smaller window with the underlying view still visible but darkened until the dialog is closed. A 'cancel' command is always available on the left softkey to close the dialog while it is displayed. Most other components can be added to a dialog which makes it possible to create a wide variety of modal windows. The login window and progress bar in the Mobile Application are examples of this.

Lists

Lists are suitable for presenting a number of items with similar properties. In the Cart, Comments and History views lists were used to display linking elements to products, comments and NFC/QR history items respectively. The lists automatically use scrollbars when the elements exceeds the list height.

During development attempts were made to use a list component with a fixed height together with other content below in the same view. This caused problems with the navigation as the view sometimes would add its own scrollbar. Two scrollbars would often result in a jumpy and unpredictable movement of the components. To avoid this behavior and having two sets of scrollbars, it was decided that the views with lists should not contain any other components.

Scrollable Containers

To make views scrollable, the content was placed in invisible containers that were configured to automatically enable vertical scrolling when the content exceeded the view height. A scrollbar is then displayed on the right side of the screen. For touch devices the scrolling is done either by dragging a pointer anywhere on the area of the view, or by dragging the scroll bar. On other devices scrolling occurs when a component that is out of view is selected.

Tabbed Panels

Tabbed panels were used in the settings, account and history views. The reason for this was that they all have a lot of content that can be divided into sections in a logical way. The tabbed panels can be configured for vertical or horizontal layout as well as alignment of the tabs.

Figure 3.1: Tabbed Panels in Settings and History

Figure 3.1 shows the vertical, left aligned tabs of the Settings view and the top aligned horizontal layout of the tabs in the History view. As the number of sections in the settings grew, a vertical layout was chosen which proved to work much better than the horizontal that pushed the last tabs out of view on most displays.

Entering Calendar Data

In the personal settings tab in the Account view the user can enter a birth date. The standard calendar component provided by the LWUIT library was initially used for this purpose, it can be seen in the image to the left in Figure 3.2.
It was fine for selecting a day and month, but to select a different year, the user would have to browse through all months in each year. This very impractical solution was replaced by two drop down boxes for selection of day and month, and one numeric input field for entering the year as seen in the image to the right in Figure 3.2.

Expandable Sections

In the product view expandable sections were created to show detailed product information on demand. Instead of adding all details to the vertical layout of the view the expandable section provide an overview of the different details sections of the product. This way the user can select a section placed

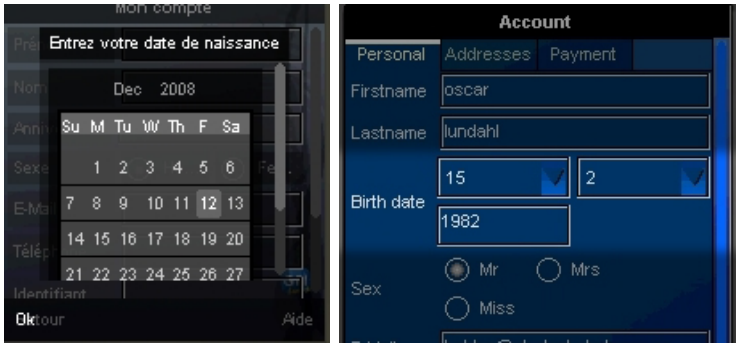


Figure 3.2: Entering Calendar Data

further down to view without having to scroll through all elements in the previous sections. An icon with an arrow is used to indicate the sections state, right-arrow when folded and down-arrow when unfolded. An example of a CD album with three CDs could look like this:

- ▶ Disc 1
- ▼ Disc 2
  - Artist - Song name 1
  - Artist - Song name 2
  - ...
- ▶ Disc 3

Here the section 'Disc 2' is unfolded while the other sections are folded. Every section can be toggled between the folded/unfolded states independently of the others.

Settings

The possibility to change the appearance and behavior of an application makes it more flexible. The settings view contain the following settings, each in a separate tab with radio button groups for making selections:

- Language - English and French were the two languages chosen to be implemented in the prototype as they were identified as most likely to be required by a possible future deployment. Hard coded text for components was avoided to the greatest extent as most text was placed in a single file to simplify adding additional languages. However, a restart of the application is required for language change(although the language is something that is not changed very often it would be better if a restart could be avoided).

- Camera - This would allow the user to change camera mode - a smaller resolution would speed up the decoding of QR-codes but may also be less accurate. As there were problems with accessing the camera on some devices this setting could not be implemented.
- Themes - The possibility to customize the applications appearance could be a great advantage from a marketing point of view as users would be allowed to add an attractive look and feel to the application. This could also include options for improved usability as color schemes, contrasts and text size etc. could be modified to make the interface more easy to use.
- Security - Security is an important issue on mobile phones when it comes to third-party applications. Although most devices prompts the user to allow an application to use e.g. connectivity functionality, camera or contacts it might be necessary for the application to have its own security settings.
- History - The settings for the history would allow the user to change the number of messages stored. The history was implemented mainly to allow the user to recover previously read information, not only products but also URLs, phone numbers and text.
- Reset - The reset option was mainly created for resetting the phones database during development but this would also be necessary if the user would like to remove stored login details.

3.2.2 Fullscreen mode

By using the fullscreen mode of the device, an application can make use of a larger screen area to show its content. This is generally very useful for applications with a lot of graphical content. However, the fullscreen mode may conceal important system information such as time, connectivity and battery level. In Java ME, this is device dependent meaning that some devices may not allow the application to use all of the screen area. The Samsung i900 displays the example application in fullscreen but keep the menu- and system information-bar at the top and bottom which can be seen in Figure 3.11 on page 55. The other phones tested displayed the application in true fullscreen, concealing most system information with the exception of an icon that appeared on top of the application when a network connection was established.

The gravity of problems related to the missing system information in the fullscreen mode may depend on the context in which the application is used. If the application is only used for a minute and then closed, the user will probably not need to know the time or battery level during that time.

When looking at use-cases of the Mobile Application it is reasonable to assume that information about the signal strength and connectivity would provide the most useful information to the user. If the users know that there is a weak or no network connection they might not try to invoke the time consuming operations that involve web services.

3.2.3 Component Focus

For non touch devices the component focus is very important as it shows which component the next input will be applied to, this is also pointed out by the guidelines for mobile websites (Appendix A.1.1).

However, if a view contains a lot of components that are focusable, navigation becomes hard as it takes time to navigate through them all. To avoid this some components that does not need to be focusable, e.g. Labels (that does not autoscroll), Containers and Images should generally have the focus property disabled.

3.2.4 Latency in Input/Output

Slow applications with high latency can be very frustrating for users. To minimize load times for mobile applications is important as they are usually used instantaneously and for shorter periods of time compared to PC applications. This is highlighted by the following quote from the PalmOS UI Guide [OST01].

> "*On a desktop, users don't mind waiting a few seconds while an application loads because they plan to use the application for an extended amount of time. On a hand held, users want to quickly look something up and then go on about their lives, and they do this several times a day.*"

Analyzing and Improving Performance

During tests with different emulators it became clear that some parts of the application took very long time to load. To find out exactly when and why this occurred, the memory monitor from Sun Wireless Toolkit was used to analyze the memory state of the emulator as the application was tested. An example of how the memory of the emulator changes during a session with the Mobile Application can be seen in Figure 3.3.

In the objects view of the Memory Monitor, Figure 3.4, all created java objects can be displayed as the application is run in the emulator. This makes it easy to find which objects take up most device memory[1].

[1] Refers to the random-access memory (RAM) which is generally quite limited on mobile devices.

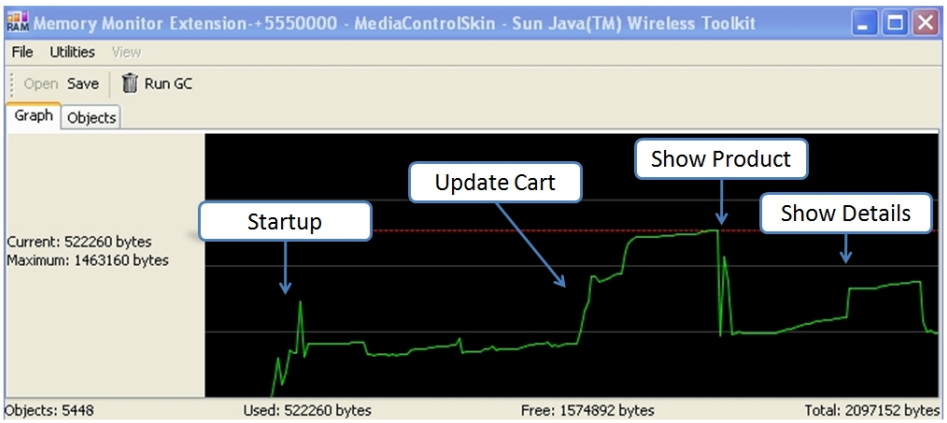


Figure 3.3: WTKs Memory Monitor - Graph view

Keeping objects in memory may result in a faster and smoother interface on devices with a lot of memory while there is a risk for memory overload on devices with less memory dedicated to Java applications. In the Mobile Application the disposal of objects by calling the garbage collection proved to solve problems with the Samsung sometimes running out of memory. But calling the garbage collector is no guarantee that it will actually run and may behave differently depending on device and its Java Virtual Machine (JVM).

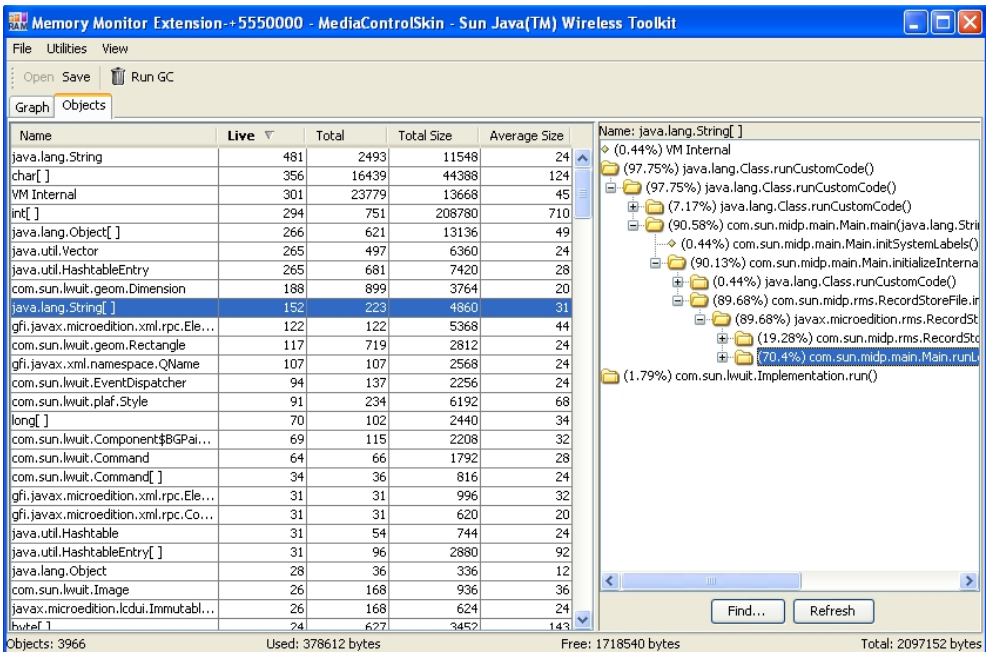


Figure 3.4: WTKs Memory Monitor - Objects view

Progress Bar

A progress bar is displayed for operations that require the user to wait for some time, e.g. when the application communicates with the server. In an early stage of development, the progress bar was just an animation that did not provide any clue on the actual progress. This was later replaced with the one shown to the left in Figure 3.5.

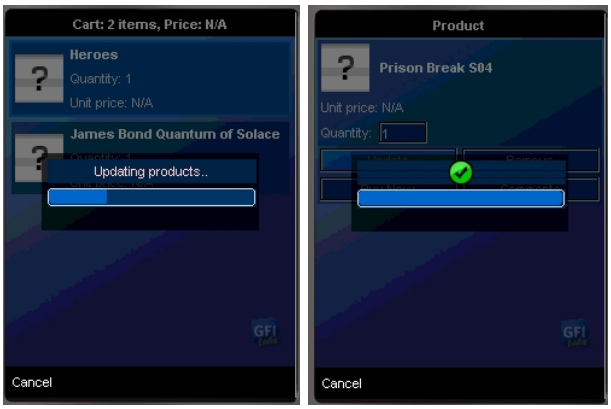


Figure 3.5: Progress bar for product updates

Unlike to the previous version this progress bar reflects the actual progress of the operation. The bar moves forward whenever an important step is complete. Although it is practically impossible to know exactly how long it will take to receive information from the server (due to the unreliability of the network and server delay) the user will continuously receive information about the progress made and an estimation of the remaining time until the operation is complete.

An important, and necessary improvement was to add a cancel command. Before this was done, some operations that could not complete due to network limitations forced the user to close the application, or in worst case restart the phone to abort. The cancel command could be added and with the progress bar running in a separate thread deadlock was prevented. When the operation is complete an icon is displayed to indicate success (the picture to the right in Figure 3.5) or failure.

3.2.5 Application Permissions

Third-party applications are usually not allowed to unconditionally access functionality of devices such as the camera, network connections or contact list. When applications try to do this the device usually show a dialog where the user have to give the application permission to continue. For most devices it is possible to set some of these permissions in the file system. When testing the prototype on the Nokia it was possible to set application

to always have access to NFC communication, preventing interruptions in the process of reading NFC tags. The camera access dialog appearing after a photo have been taken of a QR Code could not be avoided, see Figure 3.6.

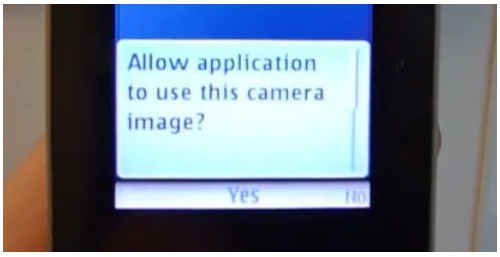


Figure 3.6: Allow camera image access dialog

3.3 Smart Poster Interaction

The use-case of reading a product from a smart poster was essential in the evaluation of the NFC and QR Code technologies. What was found to be the most important aspects around the smart poster interaction will be treated in the following sections.

3.3.1 Feedback

As previously mentioned most mobile devices provide a number of different ways for feedback on user interaction. The general behavior of sound and vibration is controlled by the phones operating system where profiles often can be configured for various situations. Even though the operating system has the overall control, a Java ME application can still use sound and vibration while it's activated.

During evaluation of the Mobile Application a specific use-case was found to benefit greatly from an alternative feedback method, this was the reading of NFC tags. The physical interaction that is enabled by the NFC technology often mean that the phones is further away from the user. This makes the screen hard to see which is why it is more useful to have vibration or sound to notify e.g. when a tag is successfully read.

3.3.2 NFC Target Marks

As the NFC technology is mostly hidden it is necessary to indicate its presence by visual target marks. This have been pointed out in previous research. In a thesis about physical selection in ubiquitous computing by Pasi Välkkynen [VAL01] the RFID technology (from which NFC has evolved) was compared to other technologies for physical interaction between mobile

devices and the environment. It was stated that the RFID technology used for touching:

> "*is suitable for selecting links near the user, and in environments in which the link density is high. When touching a link, the user has to know the exact location of the tag and even the placement of the reader inside the terminal*".

To indicate the position of the NFC tag on smart posters in the project the mark to the left in Figure 3.7 was created. It was based on a generic icon for RFID tags that illustrate the wireless communication as circular waves being transmitted from a single source. A similar pattern was used for the icon of the NFC sub-application.

Figure 3.7: L: Mark used in the project, R: NFC Forums N-Mark

The mark to the right in Figure 3.7 is provided by the NFC Forum and is intended to be used with NFC tags in future NFC applications. This mark is aimed at being of help to consumers of NFC products worldwide. It can be downloaded for free from the NFC Forums website [NFC03] to be used in commercial projects. The NFC Forum will also provide a certification mark for devices with NFC technology which is estimated to be released by the end of 2009.

The NFC phone from Nokia however, did not have any indication of its NFC capability or where the antenna was located. It was noticed during user tests that most first time users that failed to read the tag started to look for something on the phone that could reveal the location of the "NFC part".

From discussion forums on the Internet it was found that more developers face the same problem [NOK01]. Hopefully this issue will be addressed in future models. Nokia has announced a release of a new NFC capable model, the 6216 that is expected to start shipping in the third quarter of 2009 [NFC04].

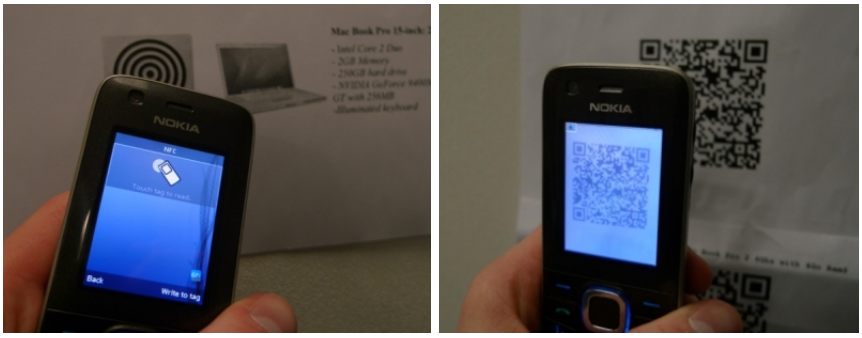


Figure 3.8: L: NFC read, R: QR Code photo

3.3.3 Read Times and Reliability

From analysis of recorded movies of the read process it was concluded that it with NFC took on average 3 seconds to complete the task[2] while it took about 14 seconds with QR codes for an experienced user. The reason for why the QR Code process was so much slower mainly seemed to originate from the fact that the user had to concentrate to hold the camera still to get a good picture. What makes this even harder is that pressing the camera button often cause the device itself to move, resulting in a blurry image that can't be decoded.

Other factors that come into play when taking photos of QR Codes are their size, the lighting conditions, distance and angle of the camera. These factors made the QR Code reading process more error-prone, about one out of three attempts failed. With NFC it was mainly the limited range that sometimes caused a failure. Although it took less than a second to transfer the data from the tag to the phone an interruption caused by the phone not being held close enough would require the process to be repeated.

When it comes to recovery is was also the NFC that had the advantage as the initiation of the camera took a few seconds while the NFC reader could be used again without any noticeable delay.

3.3.4 Size of QR Codes and Storage Capabilities

Unlike NFC tags, the physical appearance of QR Codes depend on the data they contain. The NFC tags and QR Codes of the prototype smart poster both contain the exact same data string of the featured product. An example of a product data string could look like this:

> "@#productid:2#sellerid:1#name:DVD James Bond - Quantum of Solace@"

[2]Time for physical movement and view transitions included, to make the NFC connection and complete the data transfer took less than a second.

In addition to the name of the product this string contain basic information that is needed for the product to be identified when requests are made to the server. The special characters '@' and '#' are used for parsing the string. For most products a total of about 100 characters would be enough for the data string which is about 1/10 of the capability of the NFC tags that were used.

There were discussions of also including an icon of the product to store on the NFC tags but it was concluded that it would most likely take too much space. For QR Codes even an icon with very low resolution would be way too much data to be practical.

Figure 3.9: QR Code product example

The QR Code in Figure 3.9 contain the previously mentioned product string. With more data added, the QR Codes naturally become more complex and usually require higher resolution and more physical space to avoid corruption of the information in the printing and decoding process.

It was assumed that the user would normally be rather close to the poster, about 1 meter to be able to read the information about the product. By holding out the phone in front of the poster the distance would be reduced further while the photo was taken.

The QR Codes finally used for the Smart Posters had sides of about 6cm which was enough to successfully be decoded of a distance of up to about 30cm.

3.3.5 Updating Information

It's not only due to the relatively small data capacity of the NFC tags and QR Codes that they only hold the product name and an ID-number to identify the product. Another reason for only storing this basic information is that some information about a product may change over time, e.g. prices that could change rather often. As changes only take place on the server side the smart poster would soon become out of date. As long as the smart posters consist of passive components like NFC tags and QR Codes, they

would have to be manually updated. This would not be a practical solution in general since there may be many posters of the same product spread out over a wide area. The use of product IDs and web services allow the products that are added to a virtual cart to be updated when needed.

In addition to updates to products made by the user an update is done to every product in the cart of the Mobile Application when the user is about to send an order. This will make sure that product names and prices are correct. If any changes have been made since last update the user will be notified of this before the order is sent.

3.4 Guidelines and Methods

During the development of the prototype a number of guidelines and methods taken from literature and the Internet were used in different stages. These are described in more detail in the Appendix A. In the following sections they will be evaluated with regard to how efficiently they could be used in the project.

3.4.1 Design

User Surveys

Surveys can be used to gather information about users of a system, their knowledge about it and how they use or would use it. As a first step, a number of questions with predefined answers should be formed that are relevant to the design. A way to succeed with this is to focus on what is currently uncertain about the system design. Surveys are today most commonly carried out as forms on the Internet with the advantages of being fast, cheap and easy to analyze results. To make sure the questions and answers make sense and will not be misunderstood it is important to test the survey before it is conducted. A test should be done with test subjects that are well representative of the intended target group to make it as accurate as possible.

A user survey was conducted in the beginning of the project to collect information about users and how they use their mobile phones. For this a simple form was created and handed out to people on the street to fill in. Unfortunately this was not very successful, many did not have time to participate and the answers from about 20 people was not enough to make any specific conclusions. In retrospect, a web based survey would probably have been more efficient and easier to find participants for as well as easier to analyze the answers.

Guidelines

During the design and implementation process several guidelines contributed to improvements of the overall appearance and behavior of the application, e.g. consistency, reversal of actions and more clearly indicating selected items. Despite the fact that the only guidelines that could be found for development of mobile software were for websites, most of them were still relevant as they address issues specific to the context of mobile phones rather than web design.

The second set of guidelines, or 'golden rules', created by Shneiderman are general rules for interface design and are not only used for development of software interfaces. As they are less specific they may require more experience from the developer to be applied to a design.

Both these sets of guidelines summarize good practice collected by experts within the field. However, it might be hard to keep the details in mind and to know when to apply them. As a reminder it could be helpful for developers to print or write some of them down on a post-it note and have at hand during the design process.

3.4.2 Evaluation

In the end of the project an evaluation of the final prototype was done with the following methods.

Heuristic Evaluation

For the Heuristic Evaluation a simplified evaluation form for mobile software was used [COS01]. It was carried out by commenting and comparing how parts of the interface match the list of checkpoints in the form. More details about how this was done can be found in the appendix A.2.2. Some the possible issues found include:

- Size of buttons (for touch screen) cannot be adjusted.
- Transparent backgrounds may compromise the visibility of text.
- Animations (transitions) cannot be disabled.
- Size of icons cannot be adjusted.
- Text font can only be changed with different themes.
- There may be some latency sometimes that may confuse the user, e.g. before a transition.
- Help sections not fully implemented.

Diagnostic Evaluation

Diagnostic Evaluation is a user based evaluation with primary objective of identifying usability problems from observations of the direct use of the system.

It required some time for preparation but was the only method that actually involved users and could provide direct feedback on specific issues with the functionality and appearance of the application. The following steps were identified to be most important to succeed with this method for evaluating the prototype (see A.2.1 for more details):

- Find representative users
- Prepare use-cases that need evaluation
- Keep notes or record the evaluation
- Try not to give the users hints, if they can't figure something out on their own it probably needs to be fixed

The tests were carried out in the office where four smart posters with different products were put on a wall. Due to limited time and resources for this project the users for this test were found within the company, a total of six people which were considered to match the expert and intermediate experience levels of the expected users. An evaluation form was created to be completed by the evaluator while observing the users. Eight use-cases were then selected for the tests, based on the use-cases mentioned in Section 3.1.2:

- Read Product and Add to Cart (Some with NFC and some with QR)
- Buy 3 of some product in the Cart
- Open Account, show addresses
- Update a product
- Find release date of a Movie
- Write Comment about a Product
- Change language
- Remove Products from Cart

Although each user only was assigned to a few use-cases the test revealed several interesting problem areas. Some of these were related to the NFC - smart poster interaction. Other issues were e.g. misconceptions about the naming of functions in the interface. Some of these issues were simple and could be attended to directly while the more complex problems had to be left aside.

Hierarchical Task Analysis

This method is used to analyze in detail what a user is required to do to achieve a specific task. It is based on a common technique for breaking larger activities into smaller activities until a sufficient level of detail is reached. More about how to do this can be fond in the appendix A.1.3.

The Hierarchical Task Analysis (HTA) was quite time consuming and complicated. The following four use-cases were selected for the HTA:

- Read and add product to the cart (NFC)
- Show a product in the cart
- Write a comment
- Change language

The output from this method was a number of diagrams where each step required to be performed by the user to complete the task/use-case can be seen, they are included in appendix B. The main advantage with this method was that it clearly showed differences in use of touch and non-touch screens.

Cognitive Walkthrough

The use-cases and decomposed tasks from the HTA was used as input for the Cognitive Walkthrough (CW). Each action derived from the tasks was tested against the four questions for CW:

- Will the user understand that the action is required?
- Is the action visible?
- Does the action do what the user expect?
- Will the user understand the feedback?

With the CW the following possible issues with the interface were found:

- There is no indication that the products in the list of the cart can be displayed other than the selection of elements. Additionally it was found that the user might expect that the product would be displayed when pressing the selected product in the cart list, but instead the product menu is displayed. As the user then would get the option to display the product this was not considered to be a serious issue.

- The input procedure for text was found to be unnecessarily complicated. One problem with the text fields from the LWUIT was that the T9-mode would not become available until the user started typing or pressed the selection key.

In addition to finding these issues the CW confirmed that most other parts of the interface related to the analyzed use-cases would probably not cause any problems to the user.

3.4.3 Method Comparison

Following the results from the methods provided in Appendix B and how easy and efficiently they could be used in the project a concluding comparison is presented in Table 3.1:

Method	Phase	Difficulty	Efficiency
Use-cases	Requirements	Easy	Medium
User Surveys	Requirements, Post Release	Moderate	Medium
Task Analysis	Requirements	Moderate	Low
Design Guidelines	Design	Easy	Medium
Heuristic Evaluation[1]	Design, Evaluation	Easy	Medium
Diagnostic Evaluation	Evaluation	Easy	High
Cognitive Walkthrough	Evaluation	Medium	High

Table 3.1: Comparison of design methods and guidelines

3.5 Prototype Testing

As the major part of the methods only could be used for evaluation of the final prototype, with exception of the guidelines, the development process relied much on successive tests of earlier prototypes in emulators and three different phones that were available.

3.5.1 Emulators

Emulators were frequently used by the developers to test different solutions during development of the mobile application. With different emulators and settings interface design could be tested on virtual devices with varying screen sizes and resolution. Other functionality tests that could be carried out include the use of touch screen, NFC read and write operations, web services and memory management. Two of the emulators used can be seen

[1] This refers to the simplified evaluation form for Heuristic Evaluation described in the Appendix (A.2.2) on page 76.

in Figure 3.10, the one to the left can simulate interaction with NFC tags while the one to the right could be used to test the touch screen interaction.

Figure 3.10: Emulators used for testing of the Mobile Application, L: Nokia 6131 NFC, R: HTC Touch

3.5.2 Test Phones

For the practical tests of the Mobile Application three phones were used, these are shown in Figure 3.11.

The Samsung i900 and Sony Ericsson w960i phones belong to smart phone category due to their many advanced features. The Nokia 6212 on the other hand is a rather basic phone except for its NFC capability which is why it was used in the project, it was included in the NFC starter kit together with NFC-tags and sample posters and applications [NEX01]. Unfortunately the delivery of this kit was delayed and it did not arrive until the last few weeks of the project leaving little time for proper tests.

The Sony Ericsson was mainly used for tests of the touch interface. As it is not NFC enabled and there were compatibility issues with its camera it could not be used for any interaction with the smart poster. The Samsung had a good camera but it could not be accesses directly with Java ME, instead a C# application had to be created to acquire the image from the camera that could then be loaded into the Java application. The Nokia as

Figure 3.11: Phones used for testing of the Mobile Application, L: Samsung i900, C: Sony Ericsson w960i, R: Nokia 6212 Classic

the only one with NFC could utilize both ways to read products from smart posters since it also have a basic camera that can be accessed with Java ME.

The resolution of the screens of the three phones is rather similar. What differs more is the physical size, where the Samsung has the largest and the Nokia the smallest screen. In Table 3.2 the screen properties of the test phones are listed:

	Samsung i900	Sony Ericsson W960i	Nokia 6212 Classic
Resolution:	240 x 400	240 x 320	240 x 320
Colors:	65 K	262,144	16 M
Touchscreen:	Yes	Yes	No

Table 3.2: Screen specifications of project test phones

The high ratio of resolution/screen size gives the Nokia a very sharp display but with the drawback that text becomes small and sometimes hard to read.

3.6 Prototype improvements

Both the use of methods and the tests of the prototype contributed to a number of suggestions for improvements. The limited time and resources available for development of the prototype also meant that some parts of the application were never fully implemented. The improvements presented here are intended for a potential further development of the prototype.

3.6.1 Background Process for NFC

It is reasonable to assume that most future users of a fullscreen application like the one presented here will not have it running on their phones all the time. For NFC enabled phones the advantage of being able to launch applications depending on the content a tag contains was demonstrated by the preinstalled software on the Nokia 6212. Tags from different posters automatically launched different applications when they were read.

To have a background process like this would be very useful as users would not have to start the application each time to be able to read a tag, instead a popup could be displayed that would allow the user to add the product to the cart of the application directly. Technical solutions for this were discussed during development but was considered to take too much time to implement.

3.6.2 Prevent Unnecessary Product Updates

Before an order can be sent, all products have to be updated to make sure prices exist and are correct. To prevent unnecessary requests that are time consuming, e.g. from users moving back and forth between the cart and the order views a time stamp could be saved that would allow the application to skip requests during certain intervals as products are not likely to be updated that frequently.

3.6.3 Transitions

During the short period of time it takes for a transition to complete, the user is expected to wait. However, the application still allow input during the transition. In some tests the user accidentally pressed the back command twice (or the back gesture) which then resulted in two transitions corresponding to two steps up in the navigation hierarchy.

To prevent this from happening input could be disabled during transitions. An option could be added to the settings to disable transitions or set the transition time.

3.6.4 Gesture Indication

The use of gestures may speed up navigation on touch screen devices, this does of course require that the user is aware of how and when the gestures can be used. The Mobile Application does not provide any indication of the existence of the gestures which is a big drawback. There were plans to add icons to the interface in the end of the development process to solve this problem. This was not as easy as first expected. The top status bar that was a LWUIT component could only show one icon at a time, to change that would take too much time. In one way it would be better to have the

icons visible on top of the components of the views, at least the icon for the forward gesture that should be associated with each list element. Neither this could be done in a convenient way.

3.6.5 Help Sections

There are several parts of the application that are not totally intuitive. Some menu items and text for buttons does not have a clear meaning. E.g. the purpose the of update and synchronize operations can be hard to understand. The update button used in the product view caused confusion to one user:

"*- I thought I had to press update to save the changed quantity*"

The problem of coming up with text that in short describe operations well enough to be understood by everyone becomes more complex when additional languages are added. The easiest solution for this could be to add help dialogs for each view. These would explain the purpose of each component of the views and how they can be used.

3.7 Hardware Issues

Hardware technology may differ greatly between devices. For developers of mobile applications there may not be much to do about hardware related issues, but to be able to create applications that can be used efficiently on as many devices as possible it is important that developers are aware of the limitations of different devices. To be able to find as many of these issues as possible it is important to test the application thoroughly on a large number of different devices. Then the applications can be adapted to the existing hardware of various phone models. This can be done either by creating different versions of the software or by optimizing for a more general solution.

Some of the hardware related issues that affected the development of the prototype are analyzed here.

3.7.1 Touch Screens

Touch screen technology have become common in high-end devices during recent years, aimed at making it easier to interact with the phones interface. Although manufacturers claim that they can be used with a finger, most devices with touch screen also come with a pen. One major advantage of being able to use a finger instead of a pen is that can be done with only one hand. Pen interaction requires the phone to be held in one hand and the pen in the other. For users on the move, carrying or holding on to something, this

is clearly an important issue.

How well the finger interaction work naturally depend on how big fingers you have but also on the size of the screen and the components of the interface. Of the two touch screen phones used for testing of the prototype the Sony Ericsson with the smaller screen required the use of a pen for a majority of the actions of the Mobile Application. The lower resolution and bigger screen of the Samsung meant that the application components were big enough to allow the use of a finger with adequate accuracy in most cases. To provide better support for finger interaction on other touch screen devices applications could have settings for adjusting the size of components.

The main disadvantages of touch screens are the compromised accuracy and that the interaction itself covers a part of the screen which sometimes makes it hard to see visual feedback. This problem became most obvious during testing when a finger was used as it covers a greater part of the screen. Sometimes the user was not sure whether a button was pressed successfully or not. As a complement to the visual feedback sound is often used for touch interaction and can usually be adjusted in the settings of the phones OS.

3.7.2 Cameras

The compatibility issues with the cameras of two of the phones used for testing mentioned earlier caused a lot of problems. Quite a lot of time was put into creating the workaround that made it possible to access the camera image on the Samsung while no attempt was made for a similar solution for the Sony Ericsson. It seemed to be a general problem with the more advanced cameras of the smart phones that they were too complex and did not provide any backward compatibility to be supported by Java ME. The camera in the Nokia was more simple and could easily be accessed.

3.7.3 Accelerometers

Accelerometers can be used for notification of screen rotation and change layout between portrait and landscape. Of the three test phones, it was only the Samsung that had accelerometers. However, the application was not optimized for the landscape layout, instead most content was adopted to the portrait layout to allow vertical scrolling, hence the landscape did not work very well and was practically impossible to navigate. It was sometimes unintentionally activated while the application was tested.

Chapter 4

Discussion, Conclusions

4.1 Discussion

Here, some of the possible solutions and answers found for the questions that were initially stated for this thesis are discussed as well as general aspects of development of the prototype.

4.1.1 Interface Design

This first question that this thesis was set out to answer was about development of usable interfaces:

- How can usability issues be avoided in development of interfaces for mobile applications?

The development and evaluation of the Mobile Application lead to insights on which components are more suitable in different situations and how they can be combined to create an interface that will make for a high level of user satisfaction. Much of this was presented in the previous chapter, the most important issues will be discussed further in the following sections.

LWUIT

The LWUIT graphical library provided many components and much functionality that made development much easier and saved time. However, it has some limitations, e.g. the top information bar could not display more than one icon at a time and a satisfying solution for placing the much needed icons for gestures could not be found. Since LWUIT is open source[1] it allows developers to modify the source code to suit their specific needs. This would probably have to be done in order to solve the issues with placement of icons.

Navigation

Despite the wide range of components in the LWUIT library to choose from there were some that had to be developed specifically for the project, e.g. the expandable sections. The advantage of the expandable sections is easy to see when comparing with the alternative of displaying a large number of text fields at once in the product view on a phone without touch screen. Navigating between the top components to the ones at the bottom would be very tiresome.

The implementation of gestures was also aimed at making navigation easier and more enjoyable. The forward gesture made it possible to select and display list items in a single action which was much faster than the

[1] Distributed under the GPL [GPL01].

conventional key navigation. In the reverse case of going back to the previous view there was little difference between the gesture and key navigation as the back command is always available on the left softkey.

Input

Although several new technologies have simplified the data input on mobile phones it is still a much more demanding process than typing on desktop computers. To avoid user frustration, mobile applications should minimize the use of long input fields.

One part of the Mobile Application where an extensive amount of input was required was the registration form that users had to fill in (if they were not already registered) before an order could be sent. As an alternative it was proposed the users should be able to register online as a PC client as well. Then the user would only have to enter a username and password on the mobile phone to load all personal information.

Fullscreen

As mentioned in the previous chapter, users will probably not want to have a fullscreen application running for a longer period of time on their mobile phones as they would not be able to see or access anything else.

One solution to this could be to provide the system information within the application, but as this is not entirely possible with current MIDP versions an option would be to allow the user to disable the fullscreen mode.

Widgets

Something that seems to become more and more common on many different platforms, including mobile phones, is the use of widgets. These are small applications that usually share a common space for interaction and can be customized to suit the users needs. A widget could be used to display basic information of the application, e.g. the number of items in the cart, total cost and also be able to launch the full application. This would not only make the system information available but also give users much more flexibility as they could continue to use other applications and functions of the phone. This solution will of course rely on how widespread the support for widgets will be in future phones.

4.1.2 NFC Integration

Despite the limited time available for testing the NFC functionality some issues were found that could be useful in future development of systems with NFC technology. This was central in the second question:

- How should the NFC integration with the mobile phones be done to allow users to easily adapt to and use this new technology?

The following sections deal with issues that are related to this.

Feedback

One problem area that was found in early testing of the NFC interaction was the lack of perceivable feedback for the reading of NFC tags. This was something that most likely would not have been noticed without the tests. To add vibration was a simple and good solution. Sound could also have been used but might be disturbing to other people.

NFC to Launch the Application

To use a widget as previously proposed could be a very good solution to the problem with the application concealing system information and other applications. However it is hard to know whether phones with NFC technology will be able to support widgets and how easy it would be to implement. Another solution that was discussed during development was the possibility to have a background process that could detect tags and launch the application when the tag content was identified as a product for the application. Alternatively, a dialog could be displayed that would prompt the user with options to add the product to the cart and/or launch the application.

Expanding the Space for Interaction

The previous section points out a common ground for interface design and NFC integration which were the two key issues that this thesis was set out to deal with. The NFC technology could actually make the use of other features on the phone more accessible as the user does not have to navigate through the menu system to start an application. It was showed by the demo software of the Nokia test phone, tags with different content could be used to launch applications or initiate services e.g. send booking information for a movie or set an alarm. The user is no longer limited to use the interface on the screen of the phone but could also use the phone to interact with the real world environment to start services.

Range of NFC

The limited range of the NFC of the Nokia test phone caused confusion for both developers and test subjects. The expectations were clearly to be able to read tags from a grater distance. The fact that the phone actually had to touch the paper that the tag was attached to was something most users did not expect. The paper used for the smart posters of the prototype was

regular office paper which raise the question of how well tags could be read that were hidden behind thicker or laminated paper that probably would be required for outdoor use.

The absence of visual indication of the location of the NFC antenna contributed to the confusion. It is likely that by showing the location of the antenna on the device itself would have made it easier to understand how to use it to touch tags.

Since other developers have experienced similar problems with the range of the NFC with this specific model from Nokia, they will hopefully increase this range in future models.

4.1.3 NFC vs QR Codes

The final question was about comparing the two technologies used for reading product information from smart posters:

- What are the advantages of NFC compared to QR Codes?

The compatibility issues with Java ME and the different test phones made the reading of QR Codes problematic from the beginning. On the other hand the much delayed NFC kit meant that there was nothing to compare it to. When the NFC phone was finally received and could be tested the results showed that the process of reading products was much faster using NFC. It could also be concluded that the decoding of QR Codes more frequently failed in comparison to the reading of NFC tags. About one out of three attempts to read a product with QR Codes failed. This could be related to the more sensitive process of taking a photo where the user have to concentrate to hold the phone still at the same time as the angle and lighting conditions have to be right.

When considering deployment of the smart posters there are other aspects to be concerned about. QR Codes would be sensible to various light conditions, weather and wear. Additionally QR Codes are very easy to sabotage, even if there is some degree of error correction it's enough to fill in some of the empty squares with a black marker to alter the data and render the QR Code useless. This makes the QR Codes more suitable for use in magazines or on the web where this kind of sabotage is not as much of a problem. Although several security issues around NFC have been pointed out it is not as easy to sabotage.

The physical interaction for reading NFC tags does require the tags and smart posters to be placed so they can be reached by all intended users. The QR Codes have the advantage of range but as long as all phone cameras does not have zoom this is also rather limited and in proportion to the size of the QR Code.

4.2 Conclusions

The goal of this thesis was to be of help in future projects for development of usable mobile applications with NFC technology. The approach was to use empirically derived results from the design process of a prototype for mobile shopping to give suggestions on interface design for mobile devices. Additionally, the evaluation of a number guidelines and methods for usability would show which could be used most efficiently. The tests and evaluations of the prototype resulted in a number of suggestions for improvement of the prototype which are not included here (see section 3.6 on page 55).

The following points summarize the most important remarks regarding interface design:

- Clearly show which component is focused. This is especially important for non-touch devices as the next input will generally depend on which component is currently focused.
- Limit number of components that can be focusable. Focusable images and labels that cannot be interacted with will only obstruct navigation.
- Do not display too much content in a single view. Break down the application into parts that can be more easily navigated.
- Consistency is important, this makes the user feel familiar with the interface and does not need to spend too much time looking for commands. E.g. to always have the back or cancel commands on the left softkey.
- A fullscreen application might prevent the user from accessing other functionality of the device and conceal system information. This would be a big problem if the application is intended to be used for a long time.
- Gestures can speed up navigation on touch screen devices. However, the availability and use of gestures must be made visible and instructions on how to use them must be provided to the user.

User tests provided insights on both possible usability related issues as well as the advantages that would come with the integration of NFC technology in mobile devices.

- Provide feedback on NFC interaction. As the screen is often further away from the user when it's used to touch a tag it is important to provide an alternative source for feedback, e.g. vibration or sound.

- It is also important to make the points of interaction clearly visible to the user by marking the positions of the tags. For this the N-Mark provided by the NFC Forum should be used.
- A widget and/or background process would be a favorable complement to the fullscreen application for detection of NFC tags.
- The limited range of the NFC in the test phone made it difficult to interact with the tags. As there was no indication of where the antenna was located on the phone it was hard to know how to hold the phone to touch the tag.

The NFC and QR Code technologies were compared against each other in a number of tests to find out how they could best be of use.

- Reading products with NFC proved to be both faster and more accurate than the QR Codes.
- QR Codes are decoded from a photo taken with a camera, hence they have the advantage of a greater range but are also more affected by lighting conditions, angle to camera and ware.
- QR Codes are very easy to sabotage and are better suited for magazines or on the web where sabotage would not be that much of an issue while NFC is more secure to use in smart posters.

The most important method for studying the usability of a system is to test it with actual users, in the development of the prototype described here this was done by conducting a Diagnostic Evaluation (A.2.1). If this is not possible there are a few other methods that can be used. An attempt to investigate other methods that could be used in projects like this with limited resources showed that guidelines (A.1.1) and the simplified heuristic evaluation form (A.2.2) gave the most qualitative results.

Bibliography

[COO01] Alan Cooper, Robert Reinmann, *About Face 2.0, The Essentials of Interaction Design*, (2003)

[DIX01] Alan Dix et al *Human-Computer Interaction*, Pearson Education, (2004) ISBN 0130-461091

[ISO01] International Organization for Standards 1988 ISO 9241 - 11: *Ergonomic requirements for office work with visual display terminals (VDTs)*, Part 2., Guidance on usability.

[CAN01] Canalys research release 2008/021, canalys.com ltd., (2008)

[CAN02] Canalys research release 2008/112, canalys.com ltd., (2008)

[DAH01] Markus Dahm, *Broad study on the usability of mobile phones*, Humans and Computers HuC, (2005)

[ECM01] Ecma - Standardization, Information and Communication Technology *Near Field Communication White paper* Ecma/TC32-TG19/2004/1, (2004)

[SCH01] Schusteritsch et al, *Towards the Perfect Infrastructure for Usability Testing on Mobile Devices*, Google, (2007)

[MUL01] Collin Mulliner, *Attacking NFC Mobile Phones*, Fraunhofer SIT, (2008)

[OST01] Jean Ostrem, *Palm OS®User Interface Guidelines*, Palm Inc., (2002)

[IPH01] Bill Westerman, *How people really use the iPhone*, Create with Context Inc., (2008)

[VAL01] Pasi Vlkkynen, *Physical Selection in Ubiquitous Computing*, VTT Technical Research Centre of Finland, (2007) ISBN 978-951-38-7061-4

[HER01] Tanja Herting & Gregor Broll, *Acceptance and Usability of Physical Mobile Applications*, Media Informatics, Ludwig-Maximilians-Universitt Mnchen (LMU), (2008)

[NFC01] The NFC Forum, *Near Field Communication in the real world - Part I, Turning the NFC promise into profitable, everyday applications*, Innovision Research & Technology plc, Gloucestershire, United Kingdom, www.innovision-group.com

[NFC02] The NFC Forum, *Near Field Communication in the real world - Part II, Using the right NFC tag type for the right NFC application*, Innovision Research & Technology plc, Gloucestershire, United Kingdom, www.innovision-group.com

[NFC03] The NFC Forum
`http://www.nfc-forum.org/`; accessed April 14, 2009.

[NFC04] NFCNews
`http://www.nfcnews.com/2009/04/23/nfc-takes-next-step-with-new-nokia-phone`; accessed May 16, 2009.

[NOK01] Nokia Developer Discussion Boards
`http://discussion.forum.nokia.com/forum/showthread.php?t=149340`; accessed May 16, 2009.

[NEX01] NEXPERTS, Provider of NFC Starter Kits
`http://www.nexperts.com`; accessed April 24, 2009.

[WIK01] Wikipedia, Near Field Communication
`http://en.wikipedia.org/wiki/Near_Field_Communication`; accessed April 17, 2009.

[WIK02] Wikipedia, Interaction design
`http://en.wikipedia.org/wiki/Interaction_design`; accessed April 17, 2009.

[WIK03] Wikipedia, Usability
`http://en.wikipedia.org/wiki/Usability`; accessed April 17, 2009.

[WIK04] Wikipedia, Java Micro Edition
`http://en.wikipedia.org/wiki/Java_Platform,_Micro_Edition`; accessed April 24, 2009.

[KAY01] Kaywa QR Code Generator
`http://qrcode.kaywa.com/`; accessed April 16, 2009.

[SNA01] SnapMaze QR Code Generator, `http://www.snapmaze.com/`; accessed April 16, 2009.

[QRC01] QR Code decoder library
`http://qrcode.sourceforge.jp/`; accessed April 20, 2009.

[WEB01] *7 usability guidelines for websites on mobile devices*, (2007) http://www.webcredible.co.uk/user-friendly-resources/web-usability/mobile-guidelines.shtml; accessed April 23, 2009.

[COS01] Cost 219ter, *Mobile Devices Heuristic Evaluation Instructions* http://www.tiresias.org/cost219ter/toolkit/; accessed April 25, 2009.

[COS02] Morten Hjerde, *Mobile screen size trends* http://sender11.typepad.com/sender11/2008/04/mobile-screen-s.html; accessed April 25, 2009.

[GPL01] GNU General Public License http://en.wikipedia.org/wiki/GPL_linking_exception; accessed April 28, 2009.

Appendix A

Methods

A.1 Design

A.1.1 7 Usability Guidelines for websites on mobile devices

The following Guidelines provided by Webcredible [WEB01], derived from user research tests, are specifically related to web design for mobile devices.

1. Meet users needs quickly

   Mobile and PC users can have different reasons for visiting the same site. Mobile users are more likely to want information to help them at that location or time, such as finding directions or finding out what's going on nearby. Also, they might want quick entertainment to pass away a short period of time, like something to read on the bus or while waiting to meet a friend. For your site, predicts users' needs and fulfill these as quickly as possible. Exceptions to this are items people download to keep on their phones (e.g. buying ringtones).

2. Don't repeat the navigation on every page

   Usable websites designed for PCs usually repeat the navigation on every page. However, screen real estate is precious on a mobile screen and navigation can push content off screen.

3. Clearly distinguish selected items

   Mobile phone users tend to have poor cursor control. This is because moving the pointing device down (with the joystick or direction buttons) simultaneously scrolls the page and highlights links, buttons and form fields. Due to this lack of control it's important to clearly feedback to users what item is in focus. This can be done by changing the appearance of an item to make it stand out from everything else. For example, you can change the font and background colour of links and buttons.

4. Make user input as simple as possible

   Allow users to input information by making selections instead of entering free text (or at least provide this as an alternative method). Entering text on a mobile phone can be painfully slow and error-prone on the typical 12 button mobile keypad. Mobile users are more likely to make mistakes (due to misspelling or mistyping) or take shortcuts. Sets of well thought out links on quick loading pages can be very usable.

5. Only show essential information

   Mobile phone screens are of course tiny and have only a fraction of the area or pixels on most PC monitors. Be sure to identify page

requests coming from mobiles and only send down the most essential of information. Otherwise, important content might be pushed down or difficult to find amongst everything else on the page.

Also, most mobile phone users aren't on flat rate data packages so the larger the page the more users have to pay. Users become frustrated if they have to pay to download page content they don't want.

6. Place basic browsing controls on the page

   To save screen space, mobile browsers often don't display basic controls such as 'Back' or they display the web page in full screen mode. As such, always include a 'Back' button on every page other than the homepage.

7. Design mobile-friendly page layouts

   On your website, make sure you design the page to present content in the right order and render well on mobile screens. Website layouts for large landscape PC screens usually don't work well on small portrait mobile phone screens. Furthermore, mobile browsers and page transcoders usually vertically stack pages suitable for portrait display.

   It's often best to have completely different page designs to meet mobile users' needs. If mobile phone users are a big part of your business then you should consider creating a site just for mobiles. Sites that are designed for mobiles perform significantly better with users than those that aren't.

A.1.2 Shneiderman's Eight Golden Rules of Interface Design

These golden rules are intended to be used during system design and evaluation. They focus on the software interface and may provide direct answers to design related issues.

1. Strive for consistency. Consistent sequences of actions should be required in similar situations; identical terminology should be used in prompts, menus, and help screens; and consistent commands should be employed throughout.

2. Enable frequent users to use shortcuts. As the frequency of use increases, so do the user's desires to reduce the number of interactions and to increase the pace of interaction. Abbreviations, function keys, hidden commands, and macro facilities are very helpful to an expert user.

3. Offer informative feedback. For every operator action, there should be some system feedback. For frequent and minor actions, the response

can be modest, while for infrequent and major actions, the response should be more substantial.

4. Design dialog to yield closure. Sequences of actions should be organized into groups with a beginning, middle, and end. The informative feedback at the completion of a group of actions gives the operators the satisfaction of accomplishment, a sense of relief, the signal to drop contingency plans and options from their minds, and an indication that the way is clear to prepare for the next group of actions.

5. Offer simple error handling. As much as possible, design the system so the user cannot make a serious error. If an error is made, the system should be able to detect the error and offer simple, comprehensible mechanisms for handling the error.

6. Permit easy reversal of actions. This feature relieves anxiety, since the user knows that errors can be undone; it thus encourages exploration of unfamiliar options. The units of reversibility may be a single action, a data entry, or a complete group of actions.

7. Support internal locus of control. Experienced operators strongly desire the sense that they are in charge of the system and that the system responds to their actions. Design the system to make users the initiators of actions rather than the responders.

8. Reduce short-term memory load. The limitation of human information processing in short-term memory requires that displays be kept simple, multiple page displays be consolidated, window-motion frequency be reduced, and sufficient training time be allotted for codes, mnemonics, and sequences of actions.

A.1.3 Hierachial Task Analysis

This is a process of representing how large tasks can be decomposed into smaller components, and the logical relationship between these. A common technique used is called hierarchical decomposition, which means breaking larger activities into smaller activities until a sufficient level of detail is reached.

Figure A.1 shows an example of a scheme of a HTA.

Decomposing Tasks

To break a task into sub-tasks, a good method is to ask the question “how?” in each step. It can also be useful to repeatedly ask the question “why?” in order to assist in this process, with activities becoming increasingly more abstract.

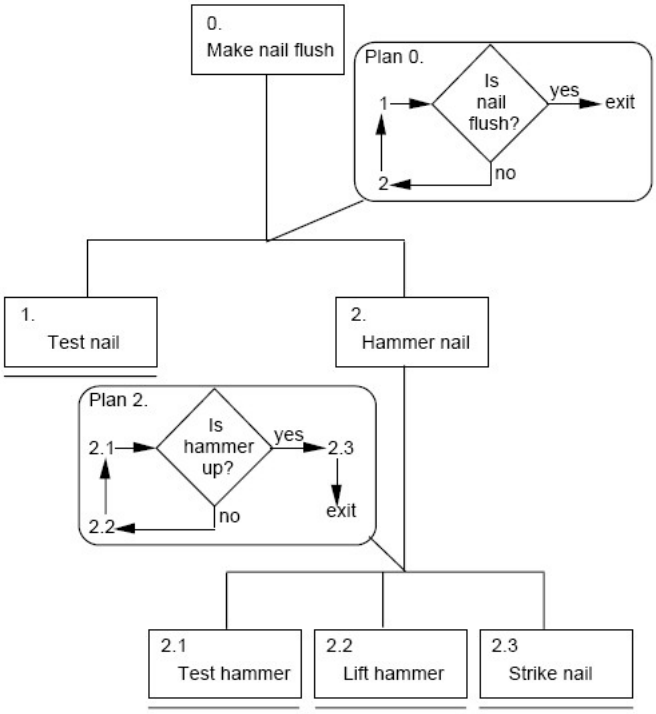


Figure A.1: Example of HTA with goal of 'Make nail flush'

It should also be noted, that even if not used in any formal way the principles of task analysis can be of considerable value in focusing attention on relevant things to consider when designing products for a person. The approach can help a developer think about the wider issues of a products use, and it is often valuable to take time to consider how a product will be used, and how it is likely to fit into the wider environment that the person operates in. Even just taking time to consider what a product will be used for can provide insights as to possible requirements that users may have.

A.2 Evaluation

Evaluation of prototypes may provide clues on things that needs to be addressed in the next step of development. It may also confirm solutions that work well. It's always important to gather information from evaluations and save the results in some way. If the evaluation method does not itself include documentation an observer could keep notes or make video or sound recordings. Schusteritsch et al describes various setups for usability studies on mobile phones [SCH01].

A.2.1 Diagnostic Evaluation

A diagnostic evaluation is a user based evaluation of a working system. The primary objective of this method is to identify usability problems from observations of the direct use of the system. It may also give a deeper understanding of why the user have difficulties with the system.

Planning

1. It is important that the users, tasks and environment used for the test are representative of the intended context of use.
2. Select the most important tasks and user group(s) to be tested (e.g. the most frequent or the most critical).
3. Select users who are representative of the user group(s). 3-5 users are sufficient to identify the main issues. 8 or more users of each type are required for reliable measures. For complex systems such as an e-commerce web site, larger numbers may be require to explore all aspects of the system.
4. Consider using user-defined tasks, where users are asked to define their own goals prior to the evaluation session.
5. Produce task scenarios and input data and write instructions for the user (tell the user what to achieve, not how to do it).
6. Plan sessions allowing time for giving instructions, running the test, answering a questionnaire, and a post-test interview.
7. Invite developers to observe the sessions if possible. An alternative is to videotape the sessions, and show developers edited clips of the main issues.
8. Two administrators are normally required to share the activities of instructing and interviewing the user, operating video equipment (if used), noting problems, and speaking to any observers.
9. If possible use one room for testing, linked by video to another room for observation.
10. If usability measures are required, observe the user without making any comments.
11. If measures are not required, prompt the user to explain their interpretation of the contents of each screen and their reason for making choices.

Running Sessions

1. Welcome the user, and give the task instructions.
2. Do not give any hints or assistance unless the user is unable to complete the task.
3. Observe the interaction and note any problems encountered.
4. If required time each task.
5. Either ask the user to think aloud, or prompt the user to explain their interpretation of the contents of each screen and their reason for making choices.
6. At the end of the session, ask the user to complete a satisfaction questionnaire.
7. Interview the user to confirm they are representative of the intended user group, to gain general opinions, and to ask about specific problems encountered.
8. Assess the results of the task for accuracy and completeness.

Output

1. Produce a list of usability problems, categorised by importance (use sticky notes to sort the problems), and an overview of the types of problems encountered.
2. Arrange a meeting with the project manager and developer to discuss whether and how each problem can be fixed.
3. If measures have been taken, summarise the results of the satisfaction questionnaire, task time and effectiveness (accuracy and completeness) measures.
4. If a full report is required, the offsite Common Industry Format provides a good structure.

A.2.2 Heuristic Evaluation

Expert Evaluation

The general idea behind heuristic evaluation is that several evaluators independently critique a system to come up with potential usability problems. To find these usability problems the evaluators use a set of heuristics created by Jakob Nielsen [DIX01] p.325:

1. Visibility of system status
2. Match between system and the real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Help users recognize, diagnose, and recover from errors
10. Help and documentation

The severity of each usability problem is then based on four factors:

1. How common is the problem?
2. How easy is it for the user to overcome?
3. Will it be a one-off problem or a persistent one?
4. How serious the problem will be perceived?

Simplified Evaluation Form for Mobile Software

Heuristic evaluation is usually carried out by several usability experts. The Heuristic evaluation form for mobile phones created by Cost 219ter is a simplified method that only require a single evaluator. It can be downloaded from their website *http://www.tiresias.org/cost219ter/toolkit/*

The form was designed to be completed electronically and the evaluation should therefore be done in a spreadsheet application.

- If the Evaluator is unfamiliar with the Heuristic Evaluation for Mobile phones (Hardware and/or Software) he/she should read them thoroughly before commencing with the evaluation.
- The Evaluator must have sufficient time to analyse the mobile device before starting to complete the Heuristic Evaluation.
- To complete the form accurately the Evaluator should go through each point one at a time.

- Any point that is not applicable to the phone (e.g. if a phone does not have a joystick), it should be marked 'N/A' in the Pass/Fail column. Completing the Heuristic Principles Using the heuristic evaluation for mobile phones (Hardware and Software) as a reference guide, the Evaluator needs to complete the comments section for each point.

3.1.4	Words must be mixed case.	Yes, mixed case is used	High	P
3.1.5	Single letters should be in capitals	No single letters	High	N/A
3.1.6	Colour and/or text attributes (bold/underline/italics) must not be the sole means to convey important information	No, icons and background color is also used to notify importance	High	P
3.1.7	The characters on a visual display should contrast (in colour and tone) against the background (ideally characters will be very dark against a light coloured background) so that they are easy to read.	Depends on theme, but generally light text against dark background give good contrast.	High	P
3.1.8	A solid background colour must always be used behind written information.	Also theme-dependent, the default have semi-transparent background for product details and dialogs	High	F
3.1.9	A visual focus indicator (highlighter) must be used on a visual display when a user moves between menu items.	Yes, focus indicator is used for menus	High	P
3.1.10	The display must have a backlight.		High	N/A
3.1.11	Information displayed on the screen must be static (e.g. it must not scroll, blink or move)	Scroll is used for some textfields that have larger content than screen width, but only when selected.	High	F
3.1.12	The user must be able to turn animations on and off	No option for this available	High	F
3,2	**Audible Output**	**N/A**	**N/A**	**N/A**
3.2.1	All meaningful information must be spoken		High	**N/A**
3.2.2	Words must be pronounced correctly.		High	**N/A**

Figure A.2: Heuristic Evaluation

Once the comments section for a point is complete the Evaluator should cross-reference their answer with the checkpoint. If the answer doesnt match the statement then the Evaluator should fail that point and put an 'F' in the 'Pass/Fail' column. If the answer does relate to the statement the Evaluator should pass the point by putting a 'P' in the 'Pass/Fail' column. Functional Requirements for Heuristic Evaluation for Mobile Phones (Software) Mobile phones contain a wide range of functionality, which may vary from one device to another. Therefore the top functions that users frequently use in their mobile phones should be evaluated when completing the Heuristic Evaluation for Mobile Phones (Software). It is vital that these frequently used functions are evaluated when performing the Heuristic Evaluation. The Evaluator should explore each function prior to completing the Heuristic Evaluation. If the Evaluator finds any problems they should be noted under the appropriate Heuristic principle when they complete the form. If a problem does not fit into a particular Heuristic principle, it should be noted under the additional comments section at the end of the document.

A.2.3 Cognitive Walkthrough

A Cognitive Walkthrough is a task-oriented walkthrough of an interface, where an evaluator imagine novice users thoughts and actions for a number of key functions. The focus of this method is explicitly on learnability. The design to be evaluated may be mock-up or working prototype. Analogous to structured walkthrough in software engineering. Based on cognitive model of human exploratory learning.

Preparation

1. Identify user population.
2. Define suite of representative tasks.
3. Describe or implement interface or prototype.
4. Specify correct action sequence(s) for each task.

Cognitive Walkthrough Steps

For each action in solution path, construct credible "success" or "failure" story about why user would or would not select correct action. Critique the story to make sure it is believable, according to four criteria:

1. Will the user be trying to achieve the right effect? What is users goal - will they want to select this action?
2. Will the user know that the correct action is available? Is control (button, menu, switch, triple-click, etc.) for action apparent (visible)?
3. Will the user know that the correct action will achieve the desired effect? Once users find control, will they recognise that it is the correct control to produce the desired effect?
4. If the correct action is taken, will the user see that things are going ok? After correct action, will users realise progress has been made towards the goal (feedback)?

Appendix B

Results

B.1 Heuristic Evaluation

Checkpoint Number	Checkpoint	Comment	Priority Rating	Pass/ Fail
1	**Allow choice over the input and output methods**			
2.2.4	Buttons on touchscreens must be adjustable in size	No, buttons cannot be adjustable	High	F
2.2.5	The colour of the buttons on touchscreens must be adjustable.	No, color of buttons cannot be adjustable	High	F
2.2.6	Buttons on touchscreens should be grouped by function	Not that many buttons except for buttongroups which by default are grouped by function	Low	P
3	**Optimise the output method**			
3,1	**Visual Output (including touchscreens)**			
3.1.2	The software should support a display of 256 colours or above	Yes, Software supports a least 256 colours	Low	P
3.1.3	Sans serif fonts must be used.	Yes, only Sans Serif text used throughout the entire application	High	P
3.1.4	Words must be mixed case.	Yes, mixed case is used	High	P
3.1.6	Colour and/or text attributes (bold/underline/italics) must not be the sole means to convey important information	No, icons and background color is also used to notify importance	High	P

3.1.7	The characters on a visual display should contrast (in colour and tone) against the background (ideally characters will be very dark against a light coloured background) so that they are easy to read.	Depends on theme, but generally light text against dark background give good contrast.	High	P
3.1.8	A solid background colour must always be used behind written information.	Also theme-dependent, the default have semi-transparent background for product details and dialogs	High	F
3.1.9	A visual focus indicator (highlighter) must be used on a visual display when a user moves between menu items.	Yes, focus indicator is used for menus	High	P
3.1.11	Information displayed on the screen must be static (e.g. it must not scroll, blink or move)	Scroll is used for some textfields that have larger content than screen width, but only when selected.	High	F
3.1.12	The user must be able to turn animations on and off	No option for this available	High	F
5	**Allow personalisation of the output**			
5,1	**Visual Output**			
5.1.1	The user must be able to select different colour schemes which provide good contrast (colour and tonal) between the information and the background of the display.	Yes, the user can select different themes	High	P
5.1.2	The user must be able to alter the size of icons	No option for icon size available	High	F

5.1.3	It must be possible to change the font size/type/weight of text information	No, not possible to change font settings except for when changing themes.	High	F
7	**Promote flexibility**			
7,3	The software must allow the user to turn predictive text on and off for any area that requires text based input.	Yes, when entering text the user may turn this on or off	High	P
8	**Promote consistency**			
8,1	The software must remain or restart in the last chosen user preference (e.g. silent mode) until the user changes it.	Yes, user settings are saved	High	P
8,8	The icon used to identify the application in the main menu of the operating software must be relevant to what the application does (e.g. a time related application must have a clock or some other time related image as the icon)	Yes, icon is provided.	High	P
9	**Provide easy to use menus**			
9,1	The menu must use meaningful symbols and/or words for each option.	Yes, menu consist of descriptive icons and text	High	P
9,3	The main menu should allow users to change the order of the options.	Yes, icons can be moved	Low	P
9,5	It should be easy to go forward and back through the menu.	Yes, a back command is always available	Low	P
9,6	The user should be able to display the menu items in a list view or a grid view	Menu is visibie in a grid view	High	P

9,7	When menu items are displayed in a grid view, all menu items should be displayed on the same screen.	Yes, all icons appear on the same screen, although it is scrollable when needed	High	P
9,8	Simple orientation information of the menu software should be provided (e.g. a down arrow to indicate there are more options available below or a summary of where a user is in relation to the number of menu items available – 1 of 5)	A scrollbar is visible when icons take more space than screen size	High	P
11	**Promote Easy to Use Softkeys**			
11,1	Softkey labels must be positioned on the display next to the actual button they are connected too.	Yes, softkey labels are always displayed just above the represented key	High	P
11,2	Softkey labels must always appear at the same position on the display.	Yes, back command always on left key, menu on the right	High	P
11,3	Softkey labels must be consistent throughout the software (e.g. the back button is always the same).	Yes, back command always on left key, menu on the right	High	P
11,4	Softkey labels must not be abbreviated except for user defined softkey labels.	No abbreviations used, except for T9-mode	High	P
11,5	Softkey labels must use plain language	Only plain language used	High	P
12	**Aid recovery and prevent errors**			

12,4	Technical jargon or codes must not be used to express an error message.	Error messages are mostly understandable text, but since it is still a prototype there is some debug information too	High	F
12,5	The software should be designed to prevent latency causing frustration for the user.	There are latency sometimes, e.g. between transitions	Low	F
12,6	The software must provide constructive feedback on how to recover from the error.	Expected errors like fail to decode QR-code image provides relevant help, but some unexpected errors like out of memory exceptions may not be recoverable.	High	F
12,7	The software must help users recover from errors, by providing functionality so that users can move forwards or backwards (to undo or go back to a previous state) to recover from the error.	Undo is not possible	High	F
14	**Aid Task Completion**			
14,1	The software must display only relevant information for the user to complete a task successfully.	Yes, only relevant information displayed on most screens	High	P
14,5	A help function should be available.	Help exist in code but not fully implemented	High	F

14,10	The software should allow information that has been received to be automatically added to the phone without the user entering any additional information (e.g. receiving business cards that can be saved into the contacts menu)	Yes, very essential to this application is it's ability to read information with NFC and QR Code	Low	P

B.2 Cognitive Walkthrough

Add Product NFC

Action	Understand Action Required	Visibility of Action	Wanted Action Effect	Understand Action Feedback
1.1 a	*Probably, although there's a risk that some users might expect to be able to use the NFC-reader without going into the Read-mode*	*Yes, icon is visible in main menu. Concept of clicking icons to open a program should be familiar to the user.*	*Yes, the NFC icon separates the NFC-reader from other parts of the application.*	*Yes, the icon is resized when pressed and the NFC-view is animated with a transition.*
1.1 b	*Yes, they are familiar with the icons in the main menu they know that an icon have to be selected before it can be pressed.*	*Yes, the selected icon is larger than the others. If the NFC icon is not already selected the user will see that.*	*Probably, to complete this action it might be neccessary to step through more icons until the NFC icon is selected.*	*Yes, the selected icon becomes larger.*
1.2 b	*See 1.1 a*	*See 1.1 a*	*See 1.1 a*	*See 1.1 a*
2.1	*Probably, since the display shows a message and icon to touch the tag, the users know they have to locate the tag first.*	*Yes, the display asks the user to tuch a tag, hence it have to be found first.*	*Yes, the user wants to know where the tag is to be able to complete action 2.2*	*N/A*
2.2	*Yes, the display asks the user to touch the tag.*	*Yes, the display asks the user to touch the tag.*	*Probably, depending on the content of the tag it will displayed either as a message or a product.*	*Yes, if the tag contained a product it will be displayed, otherwize a message with the content.*
3.1 a	*Probably, however it is possible that users believe that whenever a product is read from a tag it's automatically added to the cart.*	*Yes, the button is visible.*	*Yes, the cation on the button is a precise description of the action.*	*Yes, the cart is displayed with the added product.*
3.1 b	*Yes, similar to 1.1 b the button have to be selected before it can be pressed.*	*Probably, all components that can be selected showthat by changing color or size.*	*Probably, to complete this action it might be neccessary to step through more components until the button is selected.*	*Yes, the button text changes color.*
3.2 b	*See 3.1 a*	*See 3.1 a*	*See 3.1 a*	*See 3.1 a*

Show CartProduct

Action	Understand Action Required	Visibility of Action	Wanted Action Effect	Understand Action Feedback
1.1 a	*Yes, the concept of icons to access different parts of the application like the Cart should be familiar to the user.*	*Yes, icon is visible in main menu.*	*Yes, the Cart icon separates the Cart from other parts of the application.*	*Yes, the icon is resized when pressed and the Cart-view is animated with a transition.*
1.1 b	*Yes, they are familiar with the icons in the main menu they know that an icon have to be selected before it can be pressed.*	*Yes, the selected icon is larger than the others. If the Cart icon is not already selected the user will see that.*	*Probably, to complete this action it might be neccessary to step through more icons until the Cart icon is selected.*	*Yes, the selected icon becomes larger.*
1.2 b	*See 1.1 a*	*See 1.1 a*	*See 1.1 a*	*See 1.1 a*
2.1 a	*No, the user might no be aware of this action and use action 2.1 b instead.*	*No, there is no way of knowing that this action exist.*	*Probably, but the user might not understand that the vertical position of the gesture determins which product is displayed.*	*Yes, the product view is animated with a transition from right to left to enhance the logical connection to the gesture.*
2.1 b	*Maybe, the user might expect to be able to show the product by pressing it in the list. Instead the product menu is displayed.*	*No, the Product Menu option is hidden in the Softbutton menu or by pressing a product in the list.*	*Yes, once the users are aware of the product menu they will know how to use it.*	*Yes, the product menu is displayed on top of the cart view.*
2.1.1 b	*Probably, the other option is to click the product in the list which displays the product menu which is the goal.*	*Yes, the Menu Softbutton is visible.*	*Maybe, the user might expect to find the Product Menu directly.*	*Yes, the Menu is diplayed.*
2.1.2 b	*Yes, since the menu is displayed and the product menu entry is selected by default.*	*Yes, the user can see that the product menu entry is selected.*	*Yes, to press fire or the softbutton again to execute the menu entry is consistent for all menus.*	*Yes, the product menu is displayed on top of the cart view.*

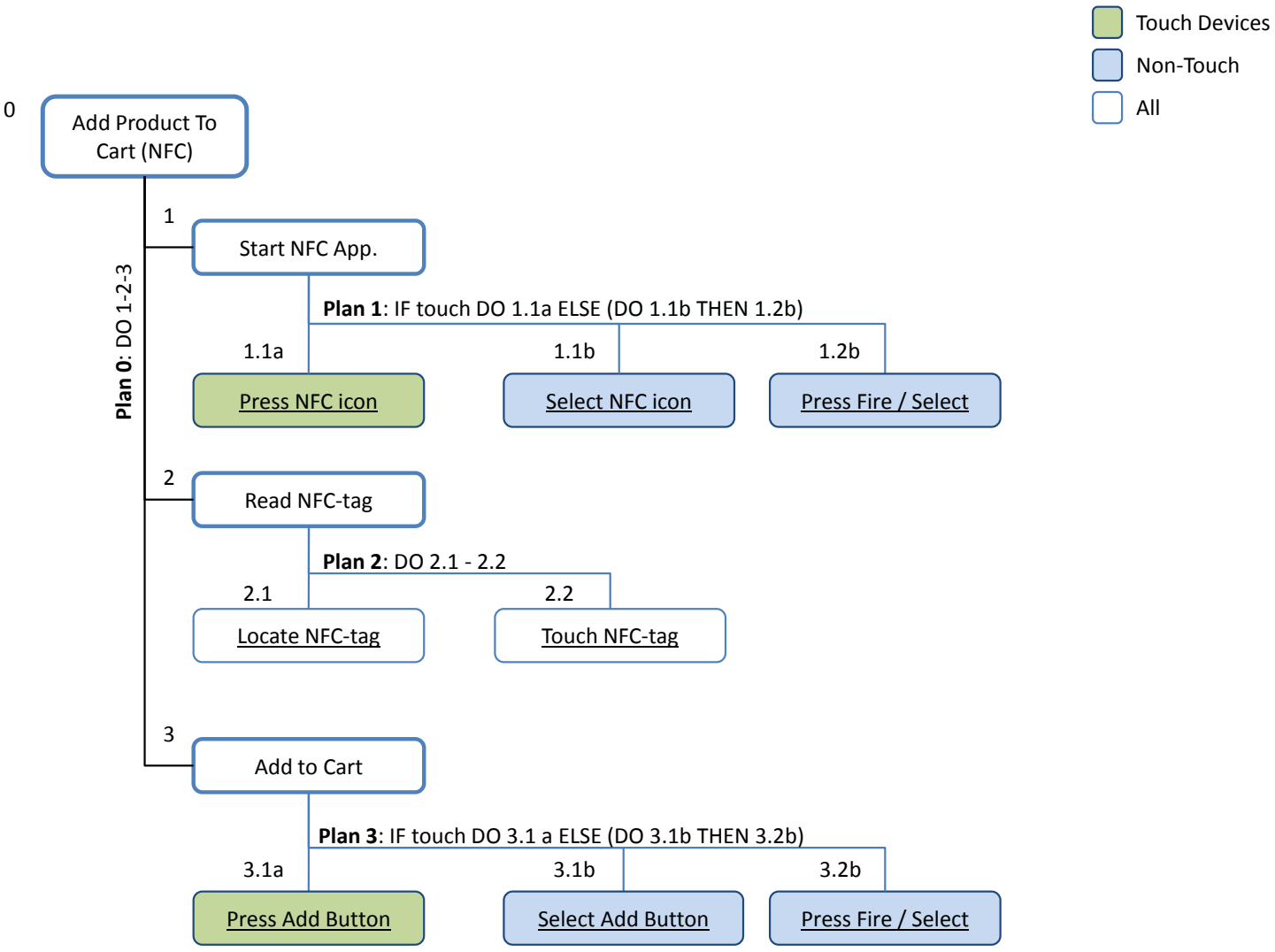
Action	Understand Action Required	Visibility of Action	Wanted Action Effect	Understand Action Feedback
1.1 a	*As long as the user is in the comments view, the menu is familiar and the natural place to look for this action. There might be a risk that the user expects to be able to use the forward gesture here, but that is use to display existing comments.*	*Yes, the menu is visible on the right softkey.*	*Yes, the menu is familiar to the user.*	*Yes, the menu is displayed.*
1.2 a	*Yes, the entry for Write Comment is what the user is looking for.*	*Yes, visible in the menu.*	*Yes, the user probably expects to go to a view to write a comment.*	*Yes, the write comment view is animated.*
1.1 b	*As long as the user is in the comments view, the menu is familiar and the natural place to look for this action.*	*See 1.1 a*	*See 1.1 a*	*See 1.1 a*
1.2 b	*See 1.2 a*	*See 1.2 a*	*See 1.2 a*	*See 1.2 a*
1.3 b	*Yes, the user knows how to use the menu.*	*Yes, the select command visible on the right softkey.*	*Yes, the user probably expects to go to a view to write a comment.*	*Yes, the write comment view is animated.*
2.1	*No, there is no way to know that this action is required. However it is possible to use the standard input.*	*No, the T9-mode is not available until the user press fire or starts to type.*	*Only if the user already knows about the T9-mode.*	*Maybe, the right softkey says T9. Not all users can be expected to know what this is.*
2.2	*Maybe, it is possible to use the non-T9 mode, but for users that know and would like to use it, it should be clear.*	*Yes, the T9 mode is visible on the right softkey.*	*Maybe, expected by users familiar with the T9 mode.*	*Maybe, the default T9 input window is displayed. However, this might be confusing as it is likely to have a completely different look.*
2.3	*Probably, depending on device. A text input area should be displayed and something the user understands.*	*Probably, the textarea should invite the user to start writing.*	*Probably, the users are most likely to expect the text to be display in the text area as they start to type.*	*Yes, the text is displayed in the text area as the user types.*
2.4	*Probably, depending on device. There should be some command like OK to finish typing.*	*Probably, after typing it would be natural to go back to the previous view somehow.*	*Yes, the user would probably expect to return to the previous screen. It might not be clear that the text entered appear in the original textfield.*	*Yes, the text appear in the original textbox.*
3.1 a	*Yes, users with non-touch devices are used to select components before they can be activated.*	*No, not really visible.*	*Yes, the users would expect to be able to select the textarea.*	*Yes, the there is a cursor indication that text can be entered.*
3.1	*See 2.1*	*See 2.1*	*See 2.1*	*See 2.1*
3.2	*See 2.2*	*See 2.2*	*See 2.2*	*See 2.2*
3.3	*See 2.3*	*See 2.3*	*See 2.3*	*See 2.3*
3.4	*See 2.4*	*See 2.4*	*See 2.4*	*See 2.4*
4.1 a	*See 3.1 a*	*See 3.1 a*	*See 3.1 a*	*See 3.1 a*
4.1	*Probably, the user should be familiar with drop down boxes and hence know that this action is required.*	*Maybe, there is an arrow to indicate that the drop down box can be expanded.*	*Yes, the user expect to be able to see the content of this component.*	*Yes, the content is displayed.*

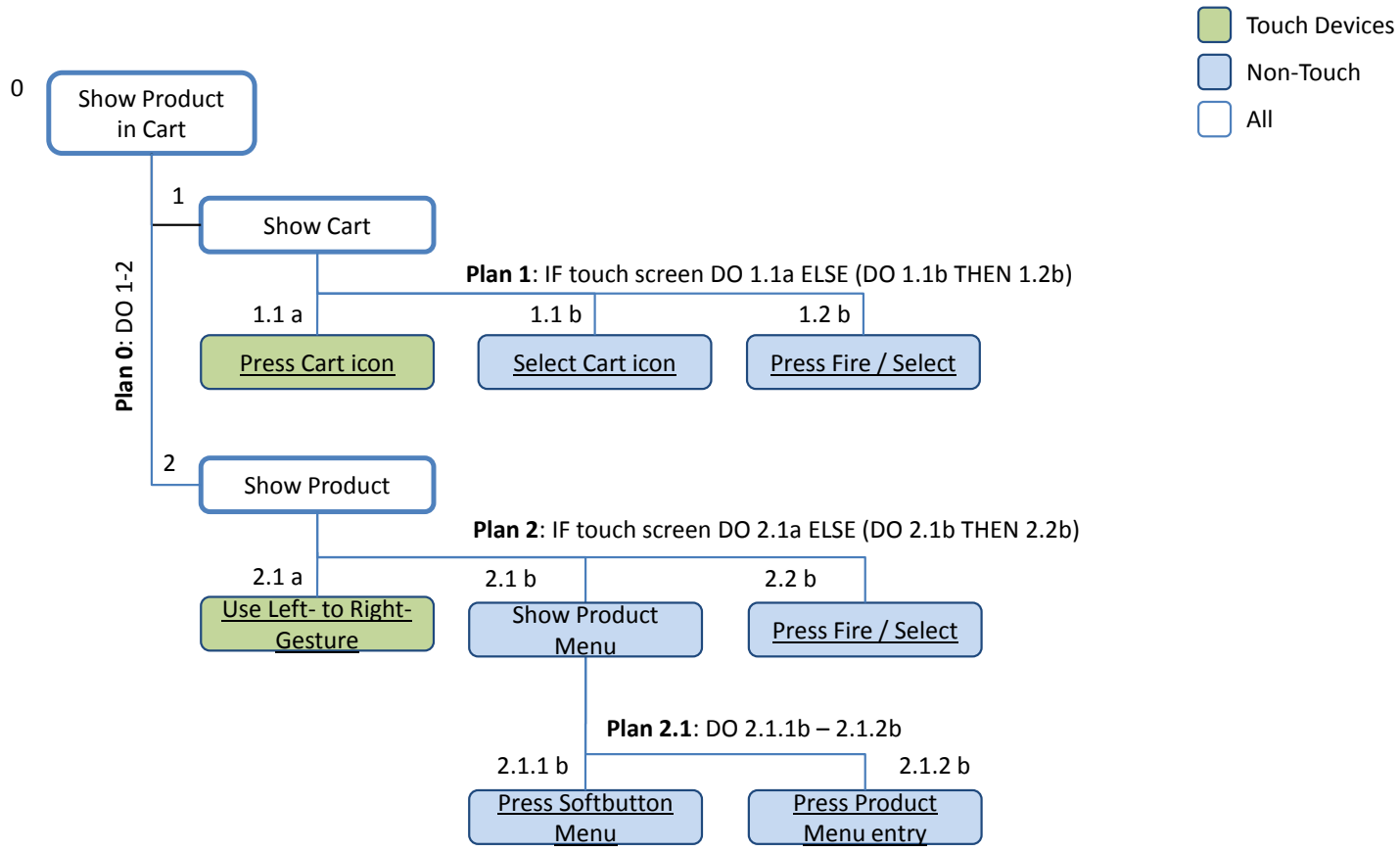
4.2	*Yes, it should be clear that a number have to be selected for the rating. However, it could be expected that a number can be selected by pressing the number keys.*	*Yes, the selected entry can be distinguished, hence the user understands that other entries can be selected.*	*Yes, the user would expect to be able to select other entries.*	*Yes, a new entry is selected. Different background- and foregroundcolor.*
4.3	*Probably, the user would expect to be able to close the drop down box. The fire button is the consistent way to do this.*	*No, there is no indication that this has to be done.*	*Yes, the user would expect the drop down box to contract.*	*Yes, the drop down box is contracted and only the selected entry is displayed.*
5.1 a	*See 3.1 a*	*See 3.1 a*	*See 3.1 a*	*See 3.1 a*
5.1	*Yes, the user would expect to submit or save the comment at some point.*	*Yes, the submit button is clearly visible at the bottom, consistent with the workflow from top to bottom. Could be added as a command to provide easier acces for non-touch devices.*	*Yes, otherwise the user would use the back command.*	*Yes, a progressbar should appear (not implemented).*

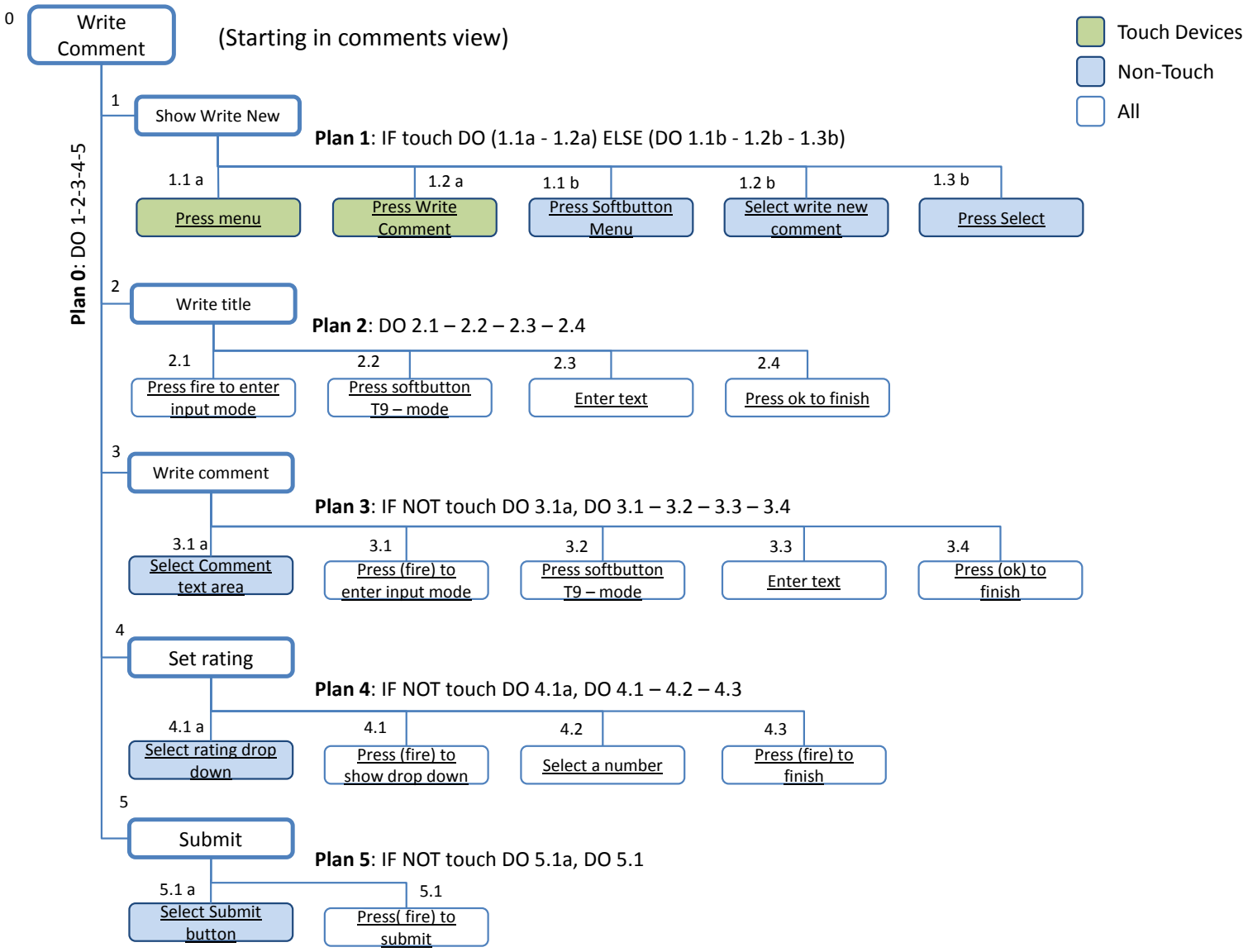
Change Language

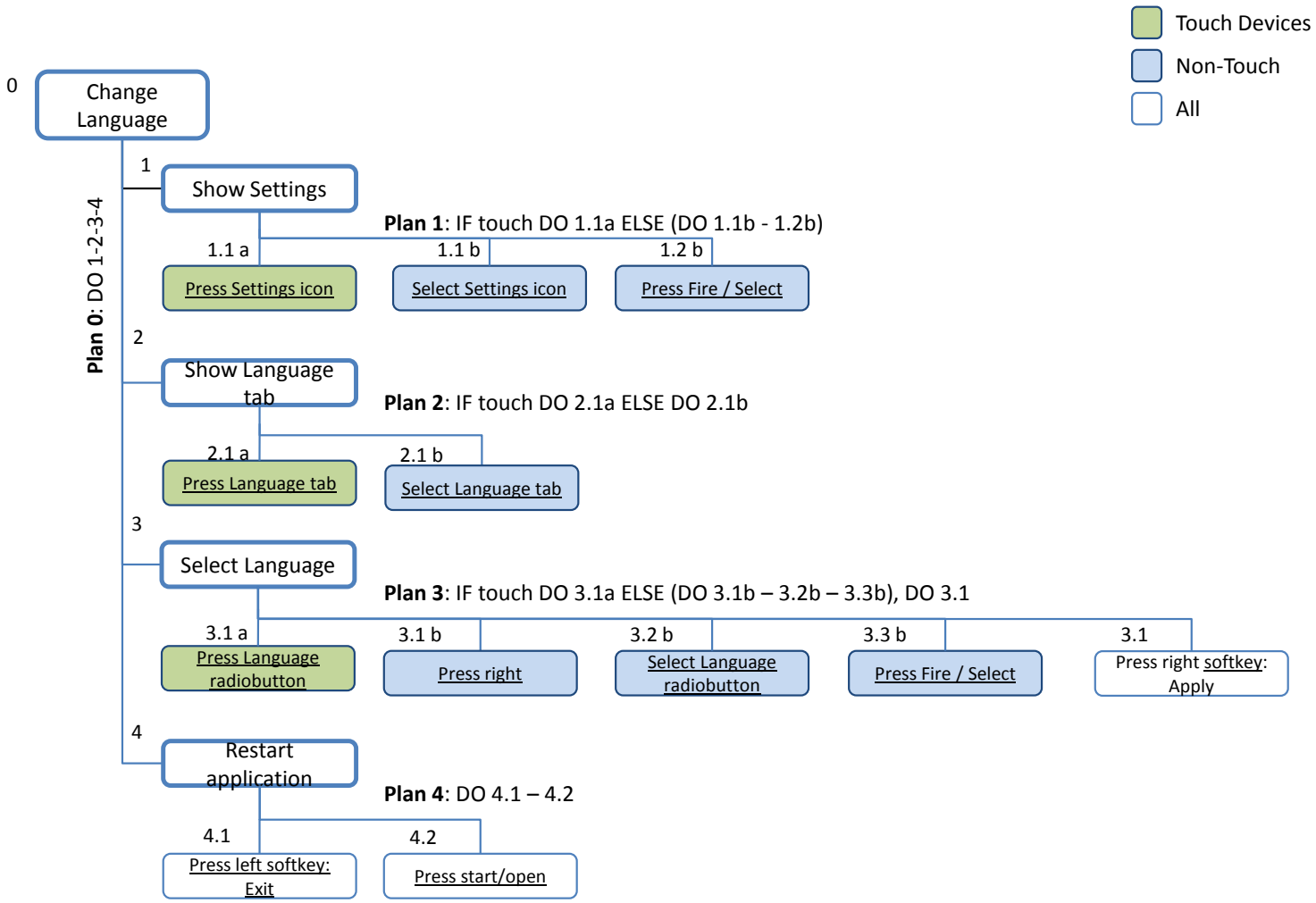
Action	Understand Action Required	Visibility of Action	Wanted Action Effect	Understand Action Feedback
1.1 a	*Yes, the user would probably look for themes in settings and know about the icons in the main menu.*	*Yes, icon is visible in main menu.*	*Yes, the user is looking for themes.*	*Yes, the icon is resized when pressed and the Settings-view is animated with a transition.*
1.1 b	*Yes, they are familiar with the icons in the main menu they know that an icon have to be selected before it can be pressed.*	*Yes, the selected icon is larger than the others. If the Settings icon is not already selected the user will see that.*	*Probably, to complete this action it might be neccessary to step through more icons until the Settings icon is selected.*	*Yes, the selected icon becomes larger.*
1.2 b	*See 1.1 a*	*See 1.1 a*	*See 1.1 a*	*See 1.1 a*
2.1 a	*Probably, the user is expected to have some experience with tabs. Since the themes tab is visible it would be natural to press it.*	*Yes, the tab is visible*	*Yes, the user is looking for settings for themes.*	*Yes, the themes tabs content is displayed.*
2.1 b	*Probably, like all other components the tabs have to be selected to allow interaction for non-touch devices.*	*Indirectly, the top tab is selected.*	*Yes, the user wants to view the themes.*	*Yes, the themes tabs content is displayed.*
3.1 a	*Yes, the user should know about the radiobutton toggle-modes.*	*Yes, the selected radiobutton is distinguished from the others.*	*Yes, the user selects the new theme.*	*Yes, pressed radiobuttom becomes selected.*
3.1 b	*Probably, perhaps not totally intuitive that it is required to press right to set focus to the buttongroup.*	*Only visible by the fact that the tab to the left has focus.*	*Yes, the user would probably not expected any other action by this.*	*Yes, the first radiobutton gains focus.*
3.2 b	*Yes, as with selection in menus and drop down boxes this action should be known to the user.*	*See 3.1 a*	*See 3.1 a*	*See 3.1 a*
3.3 b	*See 3.1 a*	*See 3.1 a*	*See 3.1 a*	*See 3.1 a*
3.1	*Probably, however if back is selected and changes have been made a confirm dialog is displayed.*	*Yes, visible on the right softkey*	*Yes, the user wants to save the changes.*	*Probably, a dialog is displayed to inform that the application have to be restarted for changes to apply.*
4.1	*Yes, the user has been prompted to restart application.*	*Yes, exit command visible on the left softkey.*	*Yes, the user would expect the application to close.*	*Yes, the application is closed.*
4.2	*Yes, if the user wants to continue to use the application it has to be started again.*	*Yes, depends on device but the user already know how to start the application.*	*Yes, the user probably wants to start tha application again.*	*Yes, loadbar displayed as application is loaded.*

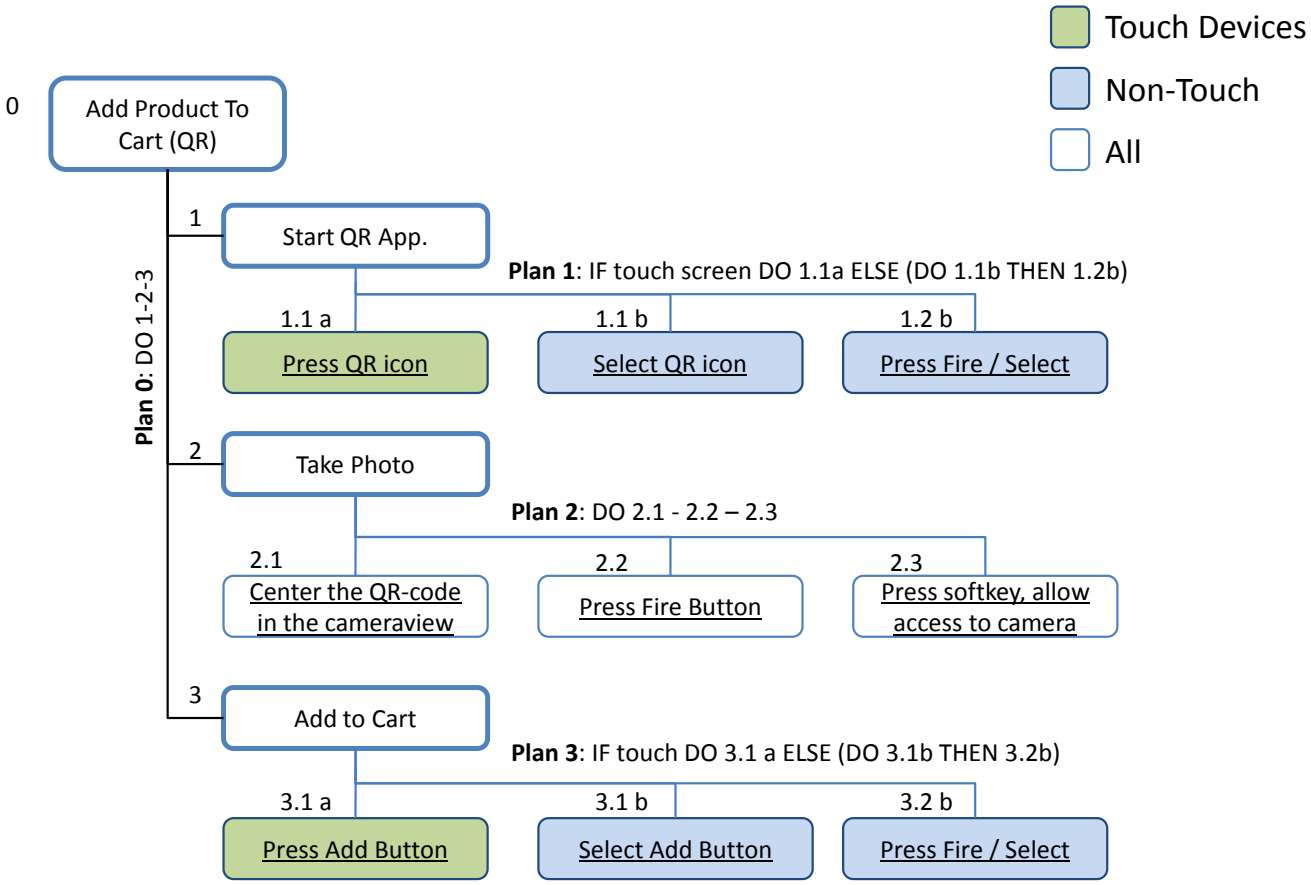
B.3 Hierarchial Task Analysis

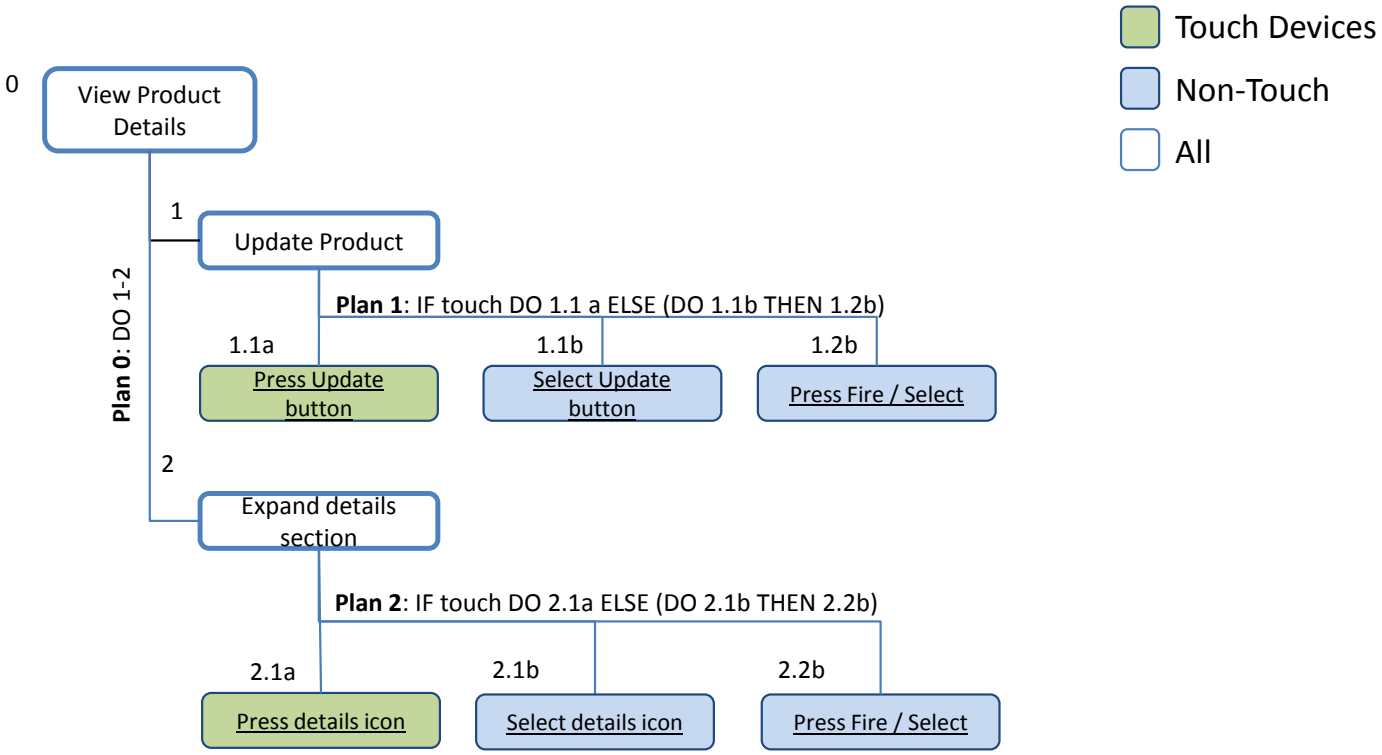

Touch Devices
Non-Touch
All
0
Add Product To Cart (NFC)
Plan 0: DO 1-2-3
1
Start NFC App.
Plan 1: IF touch DO 1.1a ELSE (DO 1.1b THEN 1.2b)
1.1a
Press NFC icon
1.1b
Select NFC icon
1.2b
Press Fire / Select
2
Read NFC-tag
Plan 2: DO 2.1 - 2.2
2.1
Locate NFC-tag
2.2
Touch NFC-tag
3
Add to Cart
Plan 3: IF touch DO 3.1 a ELSE (DO 3.1b THEN 3.2b)
3.1a
Press Add Button
3.1b
Select Add Button
3.2b
Press Fire / Select


Touch Devices
Non-Touch
All
0
Show Product in Cart
Plan 0: DO 1-2
1
Show Cart
Plan 1: IF touch screen DO 1.1a ELSE (DO 1.1b THEN 1.2b)
1.1 a
1.1 b
1.2 b
Press Cart icon
Select Cart icon
Press Fire / Select
2
Show Product
Plan 2: IF touch screen DO 2.1a ELSE (DO 2.1b THEN 2.2b)
2.1 a
2.1 b
2.2 b
Use Left- to Right-Gesture
Show Product Menu
Press Fire / Select
Plan 2.1: DO 2.1.1b – 2.1.2b
2.1.1 b
2.1.2 b
Press Softbutton Menu
Press Product Menu entry


0
Write Comment
(Starting in comments view)
Touch Devices
Non-Touch
All
Plan 0: DO 1-2-3-4-5
1
Show Write New
Plan 1: IF touch DO (1.1a - 1.2a) ELSE (DO 1.1b - 1.2b - 1.3b)
1.1 a
Press menu
1.2 a
Press Write Comment
1.1 b
Press Softbutton Menu
1.2 b
Select write new comment
1.3 b
Press Select
2
Write title
Plan 2: DO 2.1 – 2.2 – 2.3 – 2.4
2.1
Press fire to enter input mode
2.2
Press softbutton T9 – mode
2.3
Enter text
2.4
Press ok to finish
3
Write comment
Plan 3: IF NOT touch DO 3.1a, DO 3.1 – 3.2 – 3.3 – 3.4
3.1 a
Select Comment text area
3.1
Press (fire) to enter input mode
3.2
Press softbutton T9 – mode
3.3
Enter text
3.4
Press (ok) to finish
4
Set rating
Plan 4: IF NOT touch DO 4.1a, DO 4.1 – 4.2 – 4.3
4.1 a
Select rating drop down
4.1
Press (fire) to show drop down
4.2
Select a number
4.3
Press (fire) to finish
5
Submit
Plan 5: IF NOT touch DO 5.1a, DO 5.1
5.1 a
Select Submit button
5.1
Press(fire) to submit


Touch Devices
Non-Touch
All
0
Change Language
Plan 0: DO 1-2-3-4
1
Show Settings
Plan 1: IF touch DO 1.1a ELSE (DO 1.1b - 1.2b)
1.1 a
Press Settings icon
1.1 b
Select Settings icon
1.2 b
Press Fire / Select
2
Show Language tab
Plan 2: IF touch DO 2.1a ELSE DO 2.1b
2.1 a
Press Language tab
2.1 b
Select Language tab
3
Select Language
Plan 3: IF touch DO 3.1a ELSE (DO 3.1b – 3.2b – 3.3b), DO 3.1
3.1 a
Press Language radiobutton
3.1 b
Press right
3.2 b
Select Language radiobutton
3.3 b
Press Fire / Select
3.1
Press right softkey: Apply
4
Restart application
Plan 4: DO 4.1 – 4.2
4.1
Press left softkey: Exit
4.2
Press start/open


Touch Devices
Non-Touch
All
0
Add Product To Cart (QR)
Plan 0: DO 1-2-3
1
Start QR App.
Plan 1: IF touch screen DO 1.1a ELSE (DO 1.1b THEN 1.2b)
1.1 a
Press QR icon
1.1 b
Select QR icon
1.2 b
Press Fire / Select
2
Take Photo
Plan 2: DO 2.1 - 2.2 – 2.3
2.1
Center the QR-code in the cameraview
2.2
Press Fire Button
2.3
Press softkey, allow access to camera
3
Add to Cart
Plan 3: IF touch DO 3.1 a ELSE (DO 3.1b THEN 3.2b)
3.1 a
Press Add Button
3.1 b
Select Add Button
3.2 b
Press Fire / Select


Touch Devices
Non-Touch
All
0
View Product Details
Plan 0: DO 1-2
1
Update Product
Plan 1: IF touch DO 1.1 a ELSE (DO 1.1b THEN 1.2b)
1.1a
Press Update button
1.1b
Select Update button
1.2b
Press Fire / Select
2
Expand details section
Plan 2: IF touch DO 2.1a ELSE (DO 2.1b THEN 2.2b)
2.1a
Press details icon
2.1b
Select details icon
2.2b
Press Fire / Select