

Supporting Communication and Collaboration in the Process Automation Industry

Jonas Brönmark and Mikaela Åkerlind

September 25, 2011

Master's Thesis in Computing Science, 2x30 credits

Supervisor at CS-UmU: Lars Erik Janlert

Examiner: Jerry Eriksson

UMEÅ UNIVERSITY
DEPARTMENT OF COMPUTING SCIENCE
SE-901 87 UMEÅ
SWEDEN

Abstract

This thesis shows new domains for social media applications. More specifically, it explores how communication and collaboration can be supported in the process automation industry.

A concept demonstrator was implemented using the Sencha Touch framework. The prototype is based on several identified use cases, and has been tested and evaluated with end users.

The design and functionality is inspired from social media applications such as Facebook and Stack Overflow. These kinds of popular social media platforms have developed an intuitive way of structuring and grouping information. This report shows that these information structures are indeed applicable in non traditional domains, such as the process automation industry.

The concept answers to identified problem scenarios, e.g., communicating information between shifts and support of handling alarms. It also approaches personalization in order to support users focus and interest.

Contents

1	Introduction	1
1.1	The concept and Social media	1
1.2	The company ABB	2
2	Problem Description	3
2.1	Background	3
2.2	Problem	3
2.3	Goal	4
2.4	Methods	4
2.4.1	The planning phase	4
2.4.2	The design phase	9
2.4.3	The implementation phase	9
2.4.4	The evaluation phase	10
2.5	Definitions	10
2.5.1	Virtual communities	10
2.5.2	Online forum	10
3	Technology study, web applications compared to native development	11
3.1	Why web applications?	12
3.2	HTML5	13
3.3	Differences between native and web applications	13
3.4	Development on the iOS platform	16
3.5	Web applications and frameworks	16
3.6	Overview of identified frameworks	17
3.6.1	Sencha Touch	17
3.6.2	Appcelerator Titanium	18
3.6.3	PhoneGap	18
3.6.4	Adobe AIR	19
3.7	Conclusion	19

4	Guidelines for mobile devices with Natural-User-Interfaces	23
4.1	Introduction	23
4.2	Definition of a Natural-User-Interface	24
4.3	Definition of a mobile device	25
4.4	Method	25
4.4.1	Guidelines for mobile devices according to Zakiah [2]	26
4.4.2	Guidelines for devices with a NUI according to J. Blake [3]	27
4.5	Conclusion	28
5	Accomplishment	31
5.1	The initial planning phase	31
5.1.1	Analysis of ABB's proposed solutions	31
5.2	The Design phase	34
5.2.1	Social media	34
5.2.2	The design	36
5.2.3	Integration of the communication and collaboration application with existing process automation software	38
5.2.4	Scenarios	39
5.2.5	Low level prototyping and validation	40
5.3	The Implementation phase	40
5.3.1	Structure of the application	41
5.4	The Evaluation phase	42
5.4.1	The Think Aloud protocol	43
5.4.2	SUS - the System Usability Scale	44
5.4.3	The mood in here	44
5.4.4	The setup	44
5.4.5	The evaluation	45
6	Results	47
6.1	The implementation	47
6.2	Implementing for multiple platforms	49
6.3	What have not been implemented	50
6.4	The evaluation	50
6.4.1	SUS questionnaire	51
6.4.2	The Think Aloud protocol	51
6.4.3	The mood in here	52
7	Conclusions	55
7.1	Implementation	55
7.1.1	Optimization	55
7.1.2	Working with frameworks	56
7.2	The evaluation	57

7.2.1	The SUS form	58
7.2.2	The mood in here	58
7.2.3	The Think Aloud protocol	58
7.3	Limitations	60
7.4	Discussion	61
7.5	Future work	61
8	Acknowledgements	63
	References	65
A	Screenshots	69
B	SUS questionnaire	77
C	User evaluation scenarios	79
D	Think aloud protocol	81

List of Figures

1.1	Operators working in the process automation industry uses control systems that usually runs on several screens	2
2.1	Tom, Nick and Kumar (from left to right) are typical users of process automation tools. These operators are fictive characters used as personas in the product development	4
3.1	Different platforms have different standard solutions for commonly used features like back buttons. Web applications must embrace these solutions and not create their own standards	15
3.2	JavaScript performance on Windows Phone 7 devices compared to iOS and Android	16
3.3	Sencha Touch utilizes the debugger found in webKit browsers	17
3.4	Illustration of how PhoneGap bridges the gap and allows web applications to use device specific hardware	19
4.1	NUI opens up new interaction possibilities, it is more intuitive to interact directly using your hand as tool in the same manner as you do with your everyday things. This direct interaction is more intuitive than using a mouse or stylus tool	24
4.2	iPhone (smartphone) on top and iPad (tablet) below, both utilizes a NUI	25
5.1	The dark grey bubbles represent phases that have earlier been conducted by ABB and the light grey bubbles represent phases covered in this thesis	31
5.2	Process control software holds references to several thousands different objects. The picture above represents one of these objects, Boiler A1.	34
5.3	The left image is a screenshot from Facebook, the one in the middle is from Stack Overflow and the right one is from Get Satisfaction. These can be found in larger sizes in appendix A	35
5.4	The support community Get Satisfaction; the mood related for a specific topic is represented with 4 smileys	36
5.5	How the application was integrated into the existing workspace	39
5.6	The left screen shows simple sketches, the middle one enables interaction and the final one is not clickable and is only focusing on the design layout, see Appendix A for larger images	41
5.7	The implementation used the MVC pattern	41
5.8	All the Models, Views and Controllers included in the application	42
5.9	UML-sequence diagram of the application flow	43

5.10	The SUS scoring scale	44
5.11	The operators were asked to mark one smily that best represented their overall feeling after each finished task	46
6.1	Screenshot of the application running on a stationary computer using a 27" screen (the full width of the screen was not used)	48
6.2	The application running on an iPad (to the left) and on an iPhone (to the right), showing the difference in available screen size	49
6.3	Illustration of how to better utilize screen space on small devices such as an iPhone	50
6.4	The SUS scores for the three operators participating in the evaluation. On the SUS scale, 0-25 corresponds to "worst imaginable", 25-39 to "poor", 39-52 to "okay", 52-73 to "good", 73-85 to "excellent", and 85-100 to "best imaginable". Operator 1 scored lowest, 55 points, which is mapped to "good", operator 3 scored 80 points, which is mapped to "excellent" and operator 2 scored highest, 96 points, which is mapped to "best imaginable"	51
6.5	The plot shows mean values of how the three operators felt regarding the features of the Communication and Collaboration application	53
7.1	List items in a native implementation vs a web application and the differences of how memory is handled	56
A.1	Low fidelity paper prototype. Profile page for process object PL-220A, the list consists of feeds that belongs to categories like alarms and changes.	70
A.2	Low fidelity prototypes produced in Omnigraffle. The graphical components are clickable, this makes it easier to communicate the functionality of the Communication and Collaboration application	71
A.3	A medium fidelity prototype that is only focusing on layout design	72
A.4	The popular community Facebook, the picture shows how its list of feeds looks like	73
A.5	Stack Overflow is a website for questions and answers about programming. The users can rate questions and/or answers +1 point if they think it is good or rate it down -1 point if they think it is bad	74
A.6	The GetSatisfaction website provides a framework for applications like Facebook and Twitter to support their users when they have different technical or usability probelms.	75

Chapter 1

Introduction

This thesis shows how social media solutions and new mobile technologies can be used to support communication and collaboration in the process automation environment.

Social media has developed rapidly during the last decade, popular communities like Facebook, Twitter and Myspace are used by millions of people on a daily basis to enable communication and information sharing, it has gained in popularity amongst all ages and by people from different nationalities. In 2010 Facebook had over 400 million users world wide and the member rates are steadily increasing [30]. Social medias like Facebook have a well developed, easy to use system that groups and connects information in an intuitive way. How can this powerful design support communication and collaboration in environments like the process automation industry? In these kinds of domains information sharing is important to safely and efficiently run a plant.

The goal of this thesis is to develop an application that enables better digital communication and collaboration possibilities in the process automation industry (see Figure 1.1).

ABB has realized the value of a user centered approach for system development and to constantly work with improving functionality.

In line with this, a user research study had been conducted in order to evaluate users workflow and needs during their work in the process automation industry. This study identified different usability problems, for example regarding communication and collaboration.

1.1 The concept and Social media

Social medias like Facebook, Stack Overflow and Get Satisfaction revolve around users and their social connections. This thesis will show how not only users and their social connections can be supported, but also how machine parts (process objects) in a plant can have the same functionality of triggering feeds as a user. To enable this both the operators and the process objects have a profile page with their associated properties. In Facebook every action and event is logged in a feeds list, the connectivity of context based information and the way to communicate is intuitive and efficient. This design structure has been used in the developed concept demonstrator, the Communication and Collaboration application. The strength of the developed application is that not only users trigger feeds but that also process objects produce feeds. This will be explained in detail in Chapter 5.

This thesis will show how the Communication and Collaboration application tool will support operators to communicate and to collaborate. It will support them to be engaging, social, active, independent and it encourages personal development.



Figure 1.1: Operators working in the process automation industry uses control systems that usually runs on several screens

1.2 The company ABB

ABB is an multinational corporation, leader in power and automation technologies. It is Swiss-Swedish owned and goes back to the late nineteenth century. ABB have approximately 124 000 employees in around 100 countries. Below is some examples of ABB's diverse work around the world [22]:

- "ABB has delivered solutions for the worlds only floating rocket launch platform."
- "ABB has built the world's longest underwater HVDC (High-Voltage Direct Current electric power transmission system) link between Norway and the Netherlands"
- "ABB has built the world's highest substation to power the worlds tallest building Burj Khalifa in Dubai"

This thesis project was conducted in Västerås at Corporate Research, which is ABB's largest research facility with more than 200 employes. Corporate Research has the main responsibility for development of Process Automation, they are also partly involved in development of Power Systems [22].

Even though ABB has a long tradition and a solid experience of power technologies, they have realized the importance of integrating IT solutions in their systems as well as putting the user in centre. Research within areas like usability and user experience has been conducted to support a user centered design.

Chapter 2

Problem Description

The subject of this thesis project will be described; the identified problem and the goal. Also an overview of the phases of the project will be described as well as the applied methods. This chapter ends with a summary of related work that targets communication and information sharing.

2.1 Background

To support a user centered design and development of ABB's process control system, a thorough research study had been conducted by ABB Corporate Research. The study identified several usability problems concerning the communication and collaboration aspect, e.g., difficulties in solving alarms and information loss at shift change.

During the last couple of years the development of smartphones and tablets have revolutionized users possibilities to constantly and in an easy way access information. ABB wanted to explore these new mobile technologies to, if possible, support operators in the process automation industry with better tools for communication and collaboration.

Today the operators use pen and paper to take notes about different objects or behaviors in a plant. These notes are stored in various locations in the plant. This method of communicating information increases the risk of information loss, it is also time consuming to collect this information. In order for the operators to make a valid analysis of plant behavior it is crucial to collect relevant and easy accessible information. Information sharing is important to make the plant run smoothly, efficient and safely.

2.2 Problem

The initial purpose of this thesis was to find out more about how mobile devices could support communication and collaboration in the process automation domain, to let the operators work more efficient using modern tools such as social media. However it was found that the mobile aspect was more of an addition and not the main feature. Communication of events and management of alarms was best supported in conjunction with the existing stationary system. The aim of this thesis was modified to the more general approach: How can communication and collaboration be supported in the process automation domain. The mobility aspect was still approached and had still an added value for the operators.

2.3 Goal

The concept demonstrator should answer to identified scenarios and use cases that are relevant for the communication and collaboration aspect. It should enable users to communicate information like how to solve an alarm, what changes have been made on what process object and what information is relevant for the next shift to take part of. The concepts must be derived from the operator's different responsibilities and needs.

Finally the prototype must be on a fidelity level that enables users to both understand the functionality and to interact with it in a meaningful way. It should be dynamical enough to save inputs from users and to make communication possible. The application must be able to be deployed on both stationary devices as well as mobile devices to fully support the workflow of the targeted users.

2.4 Methods

The work of this thesis is divided into different phases: the planning phase, the design phase, the implementation phase, and finally the evaluation phase.

2.4.1 The planning phase

The first phase of this thesis project was to identify usability problems related to the communication and collaboration aspect and to gain understanding in how process control systems works, who are using them, what will they accomplish and how. This was done by studying and analyzing use cases and scenarios.

Personas

In earlier studies performed by ABB, three personas had been created. The personas were used in the design mockup (Chapter 5) as the operating users, see Figure 2.1. An operator interacts with the process control system; which involves handling different alarms and events in the plant. Personas are an efficient tool to work with because they communicate the typical user and their needs in a simple and clear way.



Figure 2.1: Tom, Nick and Kumar (from left to right) are typical users of process automation tools. These operators are fictive characters used as personas in the product development

1. Tom - senior operator

Tom is 59 years old and has been working at the same plant for 37 years, he enjoys working at the plant. During the years he has gained more responsibility and has advanced from mechanical engineer to senior operator. He is a natural leader with great knowledge about the process industry. He is responsible for planning shifts and making sure that everyone learns all parts of the controlling process. His job is to make sure that the shifts run smoothly and to ensure effective production. He spends a lot of time monitoring, for instance, alarms, trends and events in the plant.

2. Nick - novice operator

Nick is 22 years old and graduated from high school 3 years ago. He still lives with his parents. His father has worked at the plant his entire life and has now retired, Nick then got the opportunity to take his father's place. Nick is social and lives for his spare time, he has a technical interest but do not like studying. He is not fond of his work, he does not find it stimulating to sit still in the control room monitoring alarms. Because of the fact that Nick is inexperienced he must often rely on more experienced operators to solve alarms. His work practice concerns monitoring, *e.g.*, alarms, trends and events in the plant. It also includes regular maintenance as cleaning and changing oils, starting and stopping the process in abnormal situations.

3. Kumar - experienced operator

Kumar is 37 years old and has studied a 2 year technical university program. He has been working as a field engineer at the same plant since he graduated seven years ago. He is passionate about his work and engaged in system development. In stressful and alarming situations he is a reliable key person, which he is contempt with. Kumar also has the responsibility to teach novice operators. He has a genuine technical interest and loves the complexity of the process automation system he uses every day. His work involves monitoring and improving the processes, *e.g.*, observe alarms, trends and events in the plant. He also handles regular maintenance as cleaning and changing oil.

Use cases related to communication and collaboration

ABB's conducted user research had resulted in several use cases (UCs), scenarios and proposed solutions. This material was carefully studied. Among the use cases identified in the earlier conducted user research five use cases were related to the communication and collaboration aspect. All concepts in this thesis will be derived from these documents that are summarized below. Personas (Tom, Nick, Kumar) will be used to explain usability problems and their work flow.

1. UC: Change Shift

– Problem description

When Tom, Nick and Kumar arrives to work they must discuss with the previous shift about their experienced problems or important events that have occurred in the plant. Tom, Nick and Kumar are in this way updated on for example what process objects that needs to be extra monitored or if extra maintenance is needed on certain parts. This information between two shifts can be communicated either orally or in paper shift logs. To enable

this exchange of information Tom, Kumar and Nick have to be at work 20 minutes before they take over the shift.

However if operators are late for the shift change, the probability for information loss is high. With this information loss the operators are not able to get an overview of the current process state.

Sometimes not even paper notes have been taken that are relevant for the following shift to take part of. This may result in safety hazards or increased process stops in the plant

- Proposed solution

A virtual shift documentation system that supports operators to highlight and store important information. This tool could enable operators to add notes, trend information, videos, guidance, information important for the following shift or for information storing. When the next shift arrives they can easily get an overview of all added notes, that are saved in one place, information like time-sensitive operator notes, shift logs, parameter changes and operations. This information can be easily accessed and additional notes can be added for the following shift to take part of.

2. UC: Communicate information

- Problem description

Information and experience needs to be communicated between the operators to be able to run the plant smoothly. No efficient way to share information can result in the same negative consequences as described in UC: Change shift.

- Proposed solution

Suggestion of a virtual operator forum where information can be exchanged with notes that are time sensitive (see UC:1).

3. UC: Mobility support

- Problem description

The operators are occasionally out on the field for example during maintenance. In these situations the operators need to communicate with the control room. The operators would be more effective if they could control the process independently outside of the control room and to retrieve information that today is only available in the control room.

- Proposed solution

It would benefit the operators if information could be accessed through a mobile device when working out on the field. The device could for instance automatically identify which process object the operator is standing next to. It could also enable the operator to get a process overview and to perform simple control operations.

4. UC: Be focused

- Problem description

The operators have to sit in the control room monitoring all day long. This results in bored and tired operators, sometimes they even fall a sleep. If an important alarm is missed because of a sleeping or an unfocused operator

it can have major effect on safety and maintaining the process. Unexperienced operators like Nick often feel that they can not handle for example an alarm situation; this affects his self esteem and motivation to be active in his work. Stimuli and support might help avoiding operators like Nick from being distracted by non work related input or to get tired and unfocused.

– Proposed solution

Bringing in computer games as a reward for certain achievements. To support Nick when for example handling an alarm a simulator could be implemented where he could get credits for handling different situations. A competition could be initiated between the shifts, this would also reinforce their team spirit. The usability and user experience might be improved if game based technologies were considered, like mini maps and a virtual community. These awards and achievements can with this forum be communicated and spread to other users.

5. UC: Plan Shift

– Problem description

Tom has the responsibility to plan shifts. It involves administration and providing needed resources for each shift. Tom does not always get all information he needs to plan the shifts. He needs information about the status of every shift. If he is not provided with relevant information he can not make a valid analysis of the plant and take proper actions.

– Proposed solution

Provide a tool that enables him to retrieve relevant information and to plan shifts.

Scenarios

Three scenarios were related to the communication and collaboration aspect and will be further described. These scenarios cover one or more use cases that were described in the above section.

1. Monitoring - UC: *Communicate Information*, UC: *Plan Shift*

The morning shift consisting of Tom, Nick and Kumar starts at 5.40 AM and has earlier been planned by Tom. When they arrive to the plant they start to discuss with the previous shift about certain events and the overall status of the process. This discussion lasts a couple of minutes, in this way Tom, Kumar and Nick know what process objects that have caused problems and might need further observation. Tom plans the daily activities based on this information because of his overall responsibility. He also looks for about 15 minutes at the trends of the communicated problem areas. Then he heads to the factory floor to do a mechanical round.

Kumar and Nick are mainly responsible for the daily monitoring in which they have already started, they are responsible for one workstation each. Two workstations consist of 13 screens in which the process status is visualized as parameter changes, alarms and trends. More specifically 4 screens are used for live videos, 7 screens for process control, 1 screen for displaying key performance indexes and 1 screen summarizes the most important alarms. Nick and Kumar

switch workstation after a while to stay more focused and to learn how to handle both stations.

The process is mainly controlled automatically but Nick and Kumar still needs to be alert and make proper adjustments on process objects. Kumar retrieves statistical information about the process status to gain information about what to observe extra carefully.

2. Maintenance - UC: *Communicate Information*, UC: *Mobility support*

The clock states 10:00 AM and Kumar and Nick are sitting in front of their workstations monitoring events in the plant. From time to time Kumar also looks at the live video recording of the production process. For a trained eye it is possible to identify differences which he also does, the surface of the coil (produced product) looks a bit too buckled. This must be fixed to avoid unnecessary waste. Kumar suspects that the coil must be changed which is usually done on regular basis. He looks in the operator log in the notepad and reads that the coil was changed 4 hours ago but as they have been running on higher speed than normal it might need to be replaced. Kumar says "Nick, I think it is time to change coil. Can you help me to do that? If you go out on the floor I will support you from the control room. Call in a field engineer to help you". Nick replies "Yes. I will do that". Nick, who has done this before but still needs some assistance from a field engineer, grabs a walkie talkie and goes out on the field. Kumar navigates to the corresponding process object in the process control software. Nick who is now standing in front of the mill stops it and performs the coil change, he notifies Kumar on the walkie talkie. Kumar starts the mill from the control room, and asks Nick if the mill looks better. Kumar can from the display in the control room see that the mill has started. Nick confirms this from the walkie talkie. When Nick is back in the control room they can both see on the live video that the surface already looks better.

3. Handling an abnormal situation - UC: *Be focused*

It is 10:30 AM an ordinary day in the plant. Tom has identified some problematic areas that have shown a reduced effectiveness. Nick is responsible for workstation 1 which controls the section entrance. Kumar is responsible for workstation 2 which handles the outlet to the next section. The produced material in the plant continuously flows through several sections.

It is Friday and Nick is daydreaming about all the fun things he will do this weekend. Suddenly Kumar starts shouting with his eyes wide open "Nick, look what is entering the grind. Stop it!". Nick wakes up from his daydreaming, but too late. They both sees that a large rock is stuck in the grind. Tom rushes out to control the damages at the same time as Kumar shouts at Nick to wake up and close the grind! Several alarms are triggered and the alarm list is quickly filling up. Nick feels very stressed and is feverishly thinking "Where should I start. Is there any help described somewhere? From where do I control the grind?". Kumar sighs and takes the mouse from Nick and asks him to go out to make sure that we have stopped the flow, "I will take over here now". Nick leaves the room feeling like a total failure and his expectations and happy mood about the upcoming weekend is blown away. Nick does not want to work like this, his self esteem in handling abnormal situations has hit rock bottom and for the moment he does not feel that he can manage anything at all.

The literature study

The subjects "Web applications versus native applications" and "Guidelines for mobile devices with Natural-User-Interfaces" were chosen for the literature study. The goal of the first subject "Web applications versus native applications" was to investigate how the prototype should be implemented. The main advantage of implementing a web application is that it can be deployed on several platforms such as a desktop computer, a tablet or a smartphone. This would make it possible to demonstrate features that are adapted to mobile as well as stationary usage. The disadvantage can possibly be limitations in performance. However the area of web applications is relatively new, no scientific valid papers were found so the result is based on product web sites and web articles.

The second subject "Guidelines for mobile devices with Natural-User-Interfaces" was explored because guidance was needed in order to design a user friendly mobile interface, several aspects need to be considered such as context and limited space.

The result of these studies can be found in Chapter 3 and Chapter 4.

2.4.2 The design phase

The design phase revolved around analyzing existing concepts produced by ABB, more of this can be found in Chapter 5.

To enable understandable concepts and at the same time in an easy way modify them the initial design was made by simple sketches. This process was iterative, several discussions with expert users were held. After concluding a satisfying proposal, the fidelity was increased by using the tool OmniGraffle¹. This tool enabled production of an interactive graphic layout where the buttons and other graphical components were clickable; this made it easier to communicate the functionality of the prototype. The final proposal was evaluated by associated stakeholders before it was approved for further development.

2.4.3 The implementation phase

The majority of time available for the project was spent in the implementation phase.

Sencha Touch² was chosen as framework since it provided the possibility to write code only once and then run the very same code across multiple platforms. The identified disadvantages³ with Sencha Touch did not seem to affect the project as the implementation would not utilize any form of camera or other sensor hardware.

The chosen Framework Sencha Touch had to be studied to learn how it could be used and learn more of the different kind of functionality Sencha Touch would be able to provide. As Sencha Touch applications are written in JavaScript time was also spent on learning the JavaScript language. The main source for learning Sencha Touch and JavaScript was online videos, books and tutorials.

Different editors like TextMate, Kod, Aptana and NetBeans were tried out to see which editor had the best support for JavaScript. TextMate had an excellent JavaScript plugin for syntax checking called JSLint⁴ and was for that reason chosen.

¹<http://www.omnigroup.com/products/omnigraffle/>

²Sencha Touch is a framework based on JavaScript, used for implementation of web applications

³No easy way to access the available hardware such as accelerometer and camera

⁴JSLint is a JavaScript code quality tool

2.4.4 The evaluation phase

This phase revolved around planning the evaluation of the application. Important scenarios were tested to validate the functionality of the application, this was made informally with expert users. An additional evaluation with end users was also conducted. This evaluation of the communication and collaboration application was a part of a large project evaluation. The questionnaire had already been made by ABB and was based on the System Usability Scale (more about SUS in Chapter 5). The user tasks were designed to evaluate the features of the prototype. The operators were asked to solve each task while applying the Think Aloud protocol.

2.5 Definitions

The term *online forum* is mentioned in ABB's design suggestions and will be further explained in this section. The term *online community* is also used in this thesis; the finalized design proposal refers to this notion which is defined in the below section.

2.5.1 Virtual communities

The definition of a online community is diverse and depend on area of research. Sociologists defines it as "strong-tie"- and "weak-tie" relationship. A "strong-tie" relationship is about needs and closely knit groups e.g., family relationships. The "weak-tie" is about relationships that are not important for life supporting resources. However all social relationships are built on information sharing [35].

The term "community of practice" is also flourishing and adds to the overall confusion. The term involves people of similar interests, often professionals; the aim is to support each other and share information. All these definitions have made it difficult to make valid comparisons. The term "online community" is therefore often avoided and the broader version "social cyberspace" is therefore frequently used [36].

Rheingold who in an early stage studied communities suggested that an online community can be defined as "a social relationships aggregation, facilitated by Internet technology, in which users communicate and build personal relationships" [26]. However the definitions and what characteristics a online community possess differ.

Researchers agree on the definition of the online community as the "presence of groups of people who interact with different purposes, under the governance of certain policies, and with the facilitation of computer-mediated communication". This definition will be used in this thesis [18].

2.5.2 Online forum

No valid definition is found for the term "online forum". Keeble *et al.* defines it simply as an internet site where users can discuss different topics by posting messages. The messages are written and posted in the forum itself, where they are stored during a period of time [16]. Online Forums often develop to an online community.

Chapter 3

Technology study, web applications compared to native development

Since the introduction of the Apple App Store on July 2008 developers have been submitting applications for the iPhone, iPod and eventually the iPad. The number of available applications for these platforms have now passed 350 000 [1]. The majority of these applications have been implemented in Objective-C because it is the native programming language for devices running the iOS operating system, but also since iOS devices do not support popular programming languages such as Java or C#. In April 2010 when updating the developer license agreement that every developer has to sign, Apple banned the possibility to use any form of cross-compiling, prohibiting techniques such as Adobes flash-to iPhone compiler [12].

3.3.1 - Applications may only use Documented APIs in the manner prescribed by Apple and must not use or call any private APIs. Applications must be originally written in Objective-C, C, C++, or JavaScript as executed by the iPhone OS WebKit engine, and only code written in C, C++, and Objective-C may compile and directly link against the Documented APIs (e.g., Applications that link to Documented APIs through an intermediary translation or compatibility layer or tool are prohibited).

This meant that developers would have to implement their applications natively using Objective-C, C or C++. Another alternative would be to implement a web application, using for example JavaScript, this approach results in an application that must be executed inside a browser. Web applications could be considered second class citizens since they have no way of entering the App Store; instead a separate Web App Store was created that only accepts web applications. This Web App store has not gained the same success as the regular App Store: in the beginning of 2011 there were only about 1700 web applications published.

The restrictive developer agreement was changed in September of 2010. Apple changed the developer license agreement to allow developers more freedom by lifting the restriction with regard to other programming languages [31]. As long as the produced applications would

behave and look just as a native Objective-C application, developers were now free to choose their preferred way of implementation.

This resulted in several ways to implement an application for an iOS device. One way is to use Adobe's Packager for iPhone, which takes ActionScript 3 (AS3) code and compiles it for iOS devices. Another way to create software for iOS devices is by utilizing existing web standards to build a web application. The web application can then be wrapped with Objective-C code, making it possible to publish it on the App Store.

3.1 Why web applications?

As the number of available platforms for touch based devices keeps growing so will the cost of supporting every one of them. As of today iOS and Android are the two major platforms on the market, but many companies are fighting for a bigger market share with their own mobile platforms. Microsoft has the newly released Windows Phone 7, HP is about to release webOS devices, Samsung is developing their own Bada platform and RIM is working with their newly acquired operating system from QNX. Developers who want to support every platform will be busy at work since every platform is different, applications written for a specific platform will not work on other platforms. Not only is this because the supported programming languages differ on most platforms, but every platform comes with a different API for communication and addressing the hardware and basic functionality of the operating system. This makes it both expensive and time consuming for developers who want to reach bigger audiences by developing for every platform available.

Web applications are created with code that does not have to be processed in a compiler, but are instead relying on a browser to interpret the code. When writing web applications, using for example JavaScript, it is important to remember that browsers do not always support the same functionality [33], something that could result in unexpected outcomes in certain browsers.

However, one common denominator of the major touch based mobile platforms is that they ship with a modern web browser. A modern web browser, here means, a browser with support for features like HTML5, CSS3 and JavaScript. iOS and Android devices both ship with modern web browsers, developers are therefore more free to use these new technologies without fear of alienating users of older browsers, a scenario often seen in the desktop environment where browsers like Internet Explorer limit or make development more time consuming as special solutions must be found in order to not exclude the old browsers that are still used by a large percentage of the people surfing the web [34] [29].

Knowing that the most used smartphone platforms ship with browsers capable of handling new standards such as HTML5 and CSS3, it is possible to write applications that reside in the cloud and can be accessed and used from several different mobile platforms. Therefore it is in many cases no longer necessary to develop the very same application several times for all the different platforms; developing once and then accessing it from any device is sure to save both time and money. Developers can also be confident that users are always running the latest version of the software as it is downloaded from the server at launch time.

But are web applications suited for every kind of application, and how well does a web application integrate with the device running it? From a user experience perspective it is important that an application works flawlessly and is equally as fast as a native application. Users also expect an application to look and behave in a certain way, how can this be achieved over several platforms with a web application? This will be further examined in the following sections.

3.2 HTML5

HTML5 is the next major revision of HTML and contains a lot of new functionality. HTML5 was created to support the development of web applications by implementing functionality to make web applications easier to develop and more powerful to use. Some of the new APIs included are

- Offline storage database
- Canvas and SVG¹
- Media playback
- Drag-and-drop

As HTML5 supports offline storage there is no need to constantly be connected to the Internet. If correctly implemented in applications it is possible to use web applications without an active Internet connection, just as a native application.

Canvas support

Before html canvas there was really no way of drawing complex graphics without using proprietary software such as Adobe Flash. With the introduction of the canvas element and SVG, developers can now develop games and applications using javascript and painting their graphics directly onto a html canvas element; there is no longer any need to use proprietary software like Flash to display advanced graphics.

Media playback

Just as with canvas support, up until recently it has not been possible to include a video or audio stream directly in the html code: instead proprietary software such as Adobe Flash or Apple Quicktime had to be used in order to play embedded video and audio. HTML5 includes both video and audio tags, it also supports playback of these without the need to install additional software.

Drag and drop

Drag and drop is something most people are familiar with from their desktop environments, has now been implemented and can be used in a HTML document. This could be used to create functionality recently only offered when developing native applications or using proprietary software such as Adobe Flash.

Combine all these new functions in HTML with CSS for styling of the application and JavaScript for the program logic and it is now possible to write applications using nothing but HTML, CSS and JavaScript.

3.3 Differences between native and web applications

Some of the differences between applications developed using the platforms native language and web based applications have already been briefly addressed, this section will look into

¹Scalable Vector Graphics

Native application	Web application
Lives on the device	Lives on the web and on the device
Built specifically for the device	Built for any device supporting modern web standards
Development for one platform at the time	Development for several platforms at once
Will have a default look and feel	No platform-specific look and feel by default, programmer must provide look and feel
Built-in marketing with the App Store (iOS) and Android Market (Android)	No popular app store
Full hardware accessibility	Limited access to hardware
Low-level code, fast performance	High-level code, performance dependent on browser JavaScript speed

Table 3.1: Native applications roughly compared with web applications

the differences at a deeper level. Listing the differences between a native and web application gives a good overview of how they differ in certain areas, see Table 3.1.

Lives on the device, refers to the case when running a native application for the first time on a mobile device. The application will first have to be downloaded and installed before it can be executed. The user will have to download the application only once unless there are changes to the application and a new version is released. Once the application is installed it resides on the device.

Lives on the web, refers to the case when a web application does not have to be installed on the device in order to run. A web-launched application can either download all of its files to the device and save them locally as it is started, enabling the application to run without access to any network, or be implemented to download parts of the application as the user request them.

Built specifically for the device, refers to the case when an application is implemented in the native language, an iOS applications would typically be implemented in Objective-C using specific iOS API calls. An example of an API call would be to save data to a file, or to access the camera hardware on an iPhone. These API calls are done differently on every platform, requiring large parts of the code to be rewritten for every new platform supported. On the other hand web applications will run on any device with a browser that meets the requirements (requirements could be to understand HTML5 and CSS3, but could also be to support JavaScript or other technologies like Flash). As long as the browser meets the requirements level any device should be able to run the application.

When developing using the native language, applications will automatically inherit a certain platform specific look. Users of iOS devices are used to having a back button in the upper left corner of the screen. This is something Apple is very strict on, and developers of web applications must follow these conventions in order to be allowed to release for the App Store, see Figure 3.1.

A benefit with developing native applications is that the developer does not have to worry about the distribution. Both Android and iOS devices comes with an Application Store, where applications can be bought and downloaded with little or no effort from the end user.



Figure 3.1: Different platforms have different standard solutions for commonly used features like back buttons. Web applications must embrace these solutions and not create their own standards

Web applications on the other hand have the benefit of not having to be approved before release, the application can be used instantaneously on every device. On the other hand, web applications must be wrapped with some kind of third party software in order to be able to be distributed through these App stores. However, the application might as well reside on a server which the users can access by entering a web address. This gives them more flexibility compared to native applications. Native applications for iOS devices can only be downloaded through the App Store.

Hardware is what really sets native and web applications apart. While native applications will give the programmer, close to, full access to the hardware. Web applications are more limited here. Native applications can access most of the available sensors and can use hardware-accelerated graphics to enhance gaming and application experience. Web applications can access some hardware, but not everything. It is not possible to use hardware-accelerated graphics, therefore graphically intense games and applications should probably not be implemented as web applications. Web applications are also dependent on how fast the browser can execute JavaScript code.

As seen in Figure 3.2 JavaScript performance on mobile devices varies greatly depending on the implementation. But as JavaScript has been getting more and more attention, focus

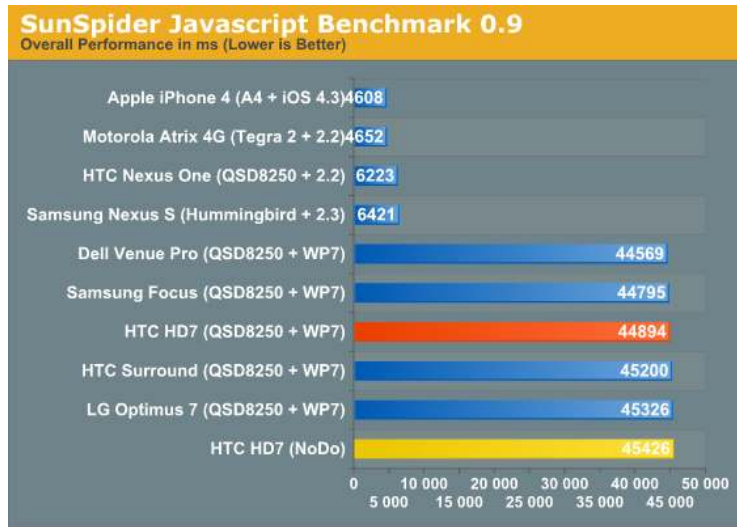


Figure 3.2: JavaScript performance on Windows Phone 7 devices compared to iOS and Android

	App Store	Web based application
Openness	Open to anyone who signs agreement	Completely open
Entry cost	\$99/year	None
Restrictions	Yes	None
Releases	1 to 2 weeks	Instantaneous

Table 3.2: Development for iOS platform, native versus web applications

on JavaScript performance has greatly increased and this is likely to continue as well since this is a way for browsers to improve over their competitors.

3.4 Development on the iOS platform

As this project is focused on the iOS platform a more detailed comparison between native iOS applications and web applications is called for, see Table 3.2.

To develop for the iOS platform it is mandatory to sign up as an Apple developer. This can be done for free, but a free account will grant only the right to write and test applications using the emulator. To be able to download code to a real device the developer must sign up as a paying Apple developer. Web applications can be developed by anyone without any signup process, and the finished products can be tested on real devices without having to pay for a developer account.

3.5 Web applications and frameworks

While it is possible to write web applications from scratch, certain parts of the code will be pretty much the same in every application. This could be graphical elements such as

buttons and sliders, but also code to access commonly used features such as camera, or save data to memory. As web applications are styled using CSS it is possible to provide different CSS files depending on whether the application is run on an Android or an iOS device. This would be time consuming as the developer would have to provide platform-specific behavior and appearance instead of focusing on developing the application. This is what frameworks try to solve. Frameworks are reusable sets of libraries to allow development not to focus on the repetitive parts but instead to focus on writing the application. They also aid the developer by providing the right look and feel for the application.

3.6 Overview of identified frameworks

A number of different frameworks has been identified, developed specifically to accelerate implementation of web applications on mobile devices. As they differ in functionality and level of documentation it is important to know what kind of framework to choose, what are the benefits and the possible drawbacks with them, see Table 3.3 for a quick overview.

3.6.1 Sencha Touch

Sencha touch lets the user create web applications using JavaScript and CSS. It provides the developer with buttons and sliders that replicate the graphical look of native applications on Android, iOS and BlackBerry touch devices. The application logic is written using JavaScript, the finished application will therefore need to be run inside of a web browser. Therefore performance will be limited to what the browser is able to provide.

Sencha Touch is built specifically for browsers running webKit, there is no support for Firefox or Internet Explorer. Therefore Sencha Touch applications need Chrome or Safari to run, should the framework be used in a desktop environment this is important to consider.

Debugging JavaScript code written for Sencha Touch is done through a browser. WebKit browsers come with a very competent debugger, see Figure 3.3, it offers basic logging as well as the possibility to run code one line at a time and to inspect values of variables without using print messages.

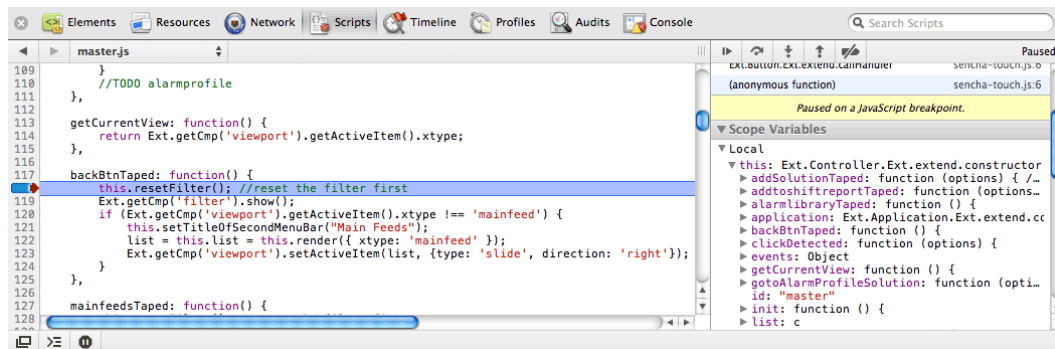


Figure 3.3: Sencha Touch utilizes the debugger found in webKit browsers

Sencha Touch does not provide any integration with the hardware or any platform specific APIs, therefore it is not possible to access the camera or any form of sensor that can be found on a phone or tablet. Since Sencha does not have access to any APIs, functionality like accessing data in the address book is not possible. Sencha can however work together with

PhoneGap (see 3.6.3) in order to provide the developer with OS APIs. A combination of Sencha Touch and PhoneGap would enable developers to write Sencha powered applications that could access data on a device through PhoneGap.

3.6.2 Appcelerator Titanium

Unlike Sencha touch that runs JavaScript code on mobile devices, Appcelerator Titanium allows the programmer to implement applications using JavaScript, and then translates the JavaScript into native code. Titanium supports both Android and iOS devices. This results in native applications as the code will first be translated by Titanium into either Java if deploying to Android or Objective-C if deploying to iOS devices.

A benefit, with using code that translates and compiles into native code, is improved performance. Another benefit is that the produced applications are ready to be sold on the App Store/Android Market just as any other native application. A drawback with using Titanium is that to be able to test the code on real hardware you must pay for an Apple developer account. Applications developed with Titanium are not limited to mobile devices. Titanium also supports translation to code that will run native on desktop computer systems, including Windows, Mac OS X and Linux. Applications for mobile devices developed using Titanium will have access to most hardware and OS API just like any native application would have. Download of the standard Titanium SDK is free of charge. The more advanced version adds support for commercial functionality like barcode readers, PayPal as well as advanced analytics. The standard version offers support through community forum access, where the advanced version offers support from Appcelerator payed staff. No pricing for the advanced version is given through the website.

One major drawback identified with Titanium is the very limited debugging possibilities. Titanium only supports log messages. There is no debugger where code can be executed one line at a time.

```
Titanium.API.debug("This is a debug message");
```

A third party service, Cloudebug [6] was identified which offers cloud based debugging for Titanium. This enables applications to upload log messages to Cloudebug's servers where the developer can browse the data. This however is just a solution for viewing log messages from several devices, it does not act as a debugger where code can be executed one line at the time.

3.6.3 PhoneGap

PhoneGap acts as a wrapper for web applications that are placed on the App Store. It provides a shell of Objective-C code around the Javascript and thus allows web applications to be treated just as a native application, including the possibility to upload to the App Store. PhoneGap also acts as a bridge between web applications and the OS API and hardware, thus enabling web applications to access phone features such as the address book, camera hardware etc. This will of course give the web application extra complexity as an extra layer is introduced in order to access phone hardware and data, see Figure 3.4.

PhoneGap does not come with any theming to make PhoneGap applications look and feel like native applications.

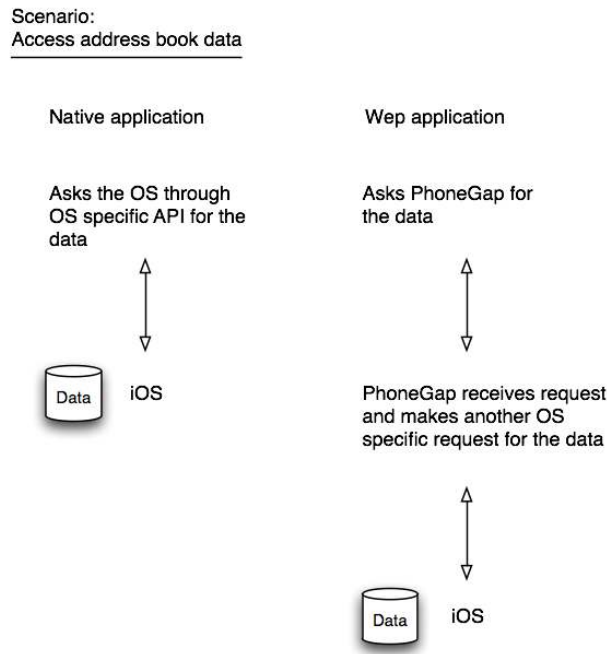


Figure 3.4: Illustration of how PhoneGap bridges the gap and allows web applications to use device specific hardware

3.6.4 Adobe AIR

Adobe Packager for iPhone is another way to develop applications for multiple devices. Applications are written using ActionScript 3 (AS3) and the Adobe AIR (Rich Internet Applications) framework which supports both the iOS and Android platforms. AIR applications have good support for accessing the hardware. Since applications written with AIR are compiled, performance should also be good.

AS3 is frequently used in web programming, because of this the available documentation for AS3 is extensive with many books and tutorials available. However, a majority of the AS3 documentation is related to web programming and not AS3 for mobile devices

In September 2010, Apple changed their developer license and banned applications that utilized cross compiling [12]. This had the effect that development of the AIR runtime for iOS devices was put on hold. Apple later, once again, changed the license to allow cross compiling, making AIR applications on iOS devices possible once more [31].

Since the 2010 Apple ban, it seems as the development and the hype around the AIR packager for iOS has declined. Even though AIR applications are once again accepted into the App Store, information on how to build applications for iOS devices using AS3 is very sparse.

3.7 Conclusion

As the market for web applications is probably just in its beginning web applications is an area that will probably see a fast development, making an overview like this outdated probably well within a year. But what can be said is that the arrival of the mentioned

frameworks opens up new possibilities for developers.

Developers unfamiliar with Objective-C have no need to first study and learn the language before they can be productive. If the frameworks can deliver what they promise, developers should now have the ability to achieve the same result using a language they already know together with one of the mentioned frameworks.

	Sencha	Titanium	PhoneGap	AIR
Access to hardware and OS API	No ^a	Yes	Yes	Yes
Can be placed on App Store	No ^b	Yes	Yes	Yes
Debugging tools	webKit debugger	Console only	Unknown	Unknown
Native looking GUI components	Yes	Yes	No	Unknown

Table 3.3: Framework specific features

^aYes, by utilizing PhoneGap^bYes, by utilizing PhoneGap

Chapter 4

Guidelines for mobile devices with Natural-User-Interfaces

4.1 Introduction

This chapter investigates how user interface guidelines for mobile devices with Graphical User Interfaces (GUI) can be used for mobile devices with Natural-User-Interfaces (NUI) in terms of logical information structures, relevant feedback and how to design to facilitate interaction.

An interface that is hard to manage and confuses the user has little value to the user, even if the software is powerful. An interface design must for that reason be easy to use and give relevant information about the features of the device. When it comes to designing interfaces for mobile devices a number of challenges arises like, limited memory, small screen size, a small keyboard and limited space for information. These devices also raise issues about context, where the user are and what she or he is doing. How can this context-based information be used to facilitate interaction? When designing for a mobile device one must consider that users occasionally, get interrupted from for instance an incoming call, or must walk and keep track of the surrounding environment when at the same time reading the GPS [14]

A successful design for a mobile device is when the content is communicated with ease and simplicity and only requires a limited number of taps to perform different tasks. The displayed information should be fast loading, concise and follow a logical structure [5].

There is a need for a valid and comprehensive framework for designing mobile interfaces with NUIs. Yesterday's mobile devices were built on traditional GUIs which applied an additional layer of interaction controllers in terms of mechanical buttons. The GUI based mobile phone will most likely eventually to a large extent be replaced by the new generation mobile phones called smartphones. Today's smartphone is built on a NUI which in comparison with GUI phones has significantly reduced the button layer; for example no mechanical buttons can be found for numbers, these have been replaced by a virtual representation presented on the display. NUIs are natural and direct which makes it more easy to understand. For example in our everyday life we are used to interact directly with objects; this natural and direct approach is applied for NUIs. Mobile GUI guidelines are for that reason not sufficient for this new generation mobile phones which is based on a totally "new" way of interaction.

4.2 Definition of a Natural-User-Interface

Today's smartphones and tablets open up new User Interface (UI) interaction possibilities. The main difference from traditional mobile phones and Personal Digital Assistants (PDAs) is that NUI (sometimes also called Gestural Interface) uses a touchscreen. A touchscreen increases the number of interaction possibilities because of the increased UI space, and the touchscreen in itself enables more diverse ways to interact. These interfaces are categorized as NUIs and aim to be a more natural and intuitive interface approach, compared to GUIs and Command Line Interfaces (CLI). With NUIs users have the possibility to use different input modalities such as multi-touch, motion tracking and positioning. J. Blake puts it "A natural user interface is a user interface designed to reuse existing skills for interacting directly with content" [3]. Existing skills is referring to human skills such as different kinds of communication, verbal and non verbal; it makes more sense to use your hands as a tool for information input instead of an external tool like a mouse [3]. These new interaction possibilities make it possible to refer to real-world metaphors that makes it easier for the user to understand how to interact with a device, *e.g.*, dragging your thumb and finger towards each other on the screen to zoom in a picture is more intuitive than clicking a small plus sign with a stylus.

Common patterns for touch screens and interactive surfaces are listed below [27].

1. Tap to open/stop/activate/select
2. Pinch to shrink and Spread to enlarge
3. Drag to Move Object
4. Slide or Spin to scroll
5. Slide and Hold for continuous scroll
6. Two Fingers to scroll



Figure 4.1: NUI opens up new interaction possibilities, it is more intuitive to interact directly using your hand as tool in the same manner as you do with your everyday things. This direct interaction is more intuitive than using a mouse or stylus tool

4.3 Definition of a mobile device

A mobile device is here limited to include devices with a NUI, *e.g.*, a smartphone or a tablet computer. A smartphone [19] is a combination of a PDA and a mobile phone, *e.g.* the iPhone. An example of a tablet computer is the iPad which is a mobile computer with a touchscreen see Figure 4.2. There are a lot of different smartphone manufacturers on the market, *e.g.*, HTC, Apple and Samsung, and they all have their own interface guidelines to reinforce their brand. Some similarities can possibly be identified between Apples iPhone and its competitors, most likely because Apple was first out on the market with a smartphone that the users considered to be good looking and most of all easy to use.



Figure 4.2: iPhone (smartphone) on top and iPad (tablet) below, both utilizes a NUI

4.4 Method

Literature has been reviewed that targets UI guidelines for mobile devices and devices with a NUI. The guidelines for the traditional mobile phones that are built on GUIs will be expanded and reinforced by guidelines for NUIs.

Zakiah, *et al.*, proposed a framework for mobile application development [2]. In that framework he proposed guidelines for mobile devices built on GUIs that are based on a

modified version of Shneiderman, *et al.*, Seven Usability Guideline for Mobile Device, and W3C Mobile Web Best Practices¹. They approach the mobility aspect. In addition, J. Blakes guidelines for NUI mobile application design will also be revised and compared [3].

4.4.1 Guidelines for mobile devices according to Zakiah [2]

These guidelines target design of mobile devices with a GUI.

1. Enable frequent users to use shortcuts

A large number of people are using their mobile device on a daily basis which raise a desire for a efficient way to interact. Efficient here means for example a reduced number of steps for completing a given task [21].

2. Offer informative feedback

Feedback is crucial for a good interaction design, for example when users are pressing a key or has sent a message, an appropriate feedback after these operations are of great importance. If no feedback is given the probability that the user gets frustrated about unwanted behavior increases [20].

3. Consistency

User inputs in a mobile device are often needed in other mediums, for example when the user creates a new event in the calendar, he or she might want to retrieve this information when sitting by the desktop computer, this is usually done by synchronizing the two calendars. Supporting these issues is recommended.

4. Reversal of actions

A design of a mobile device should make a reversal of operations or inputs possible. This is often hard to implement because of the constrained amount of data storage and computing power [20].

5. Error prevention and simple error handling

The context for the usage of mobile devices makes the probability of errors larger than when interacting with a laptop computer [15].

6. Reduce short-term memory load

Because of our limited short memory capacity the UI design of mobile devices should support this issue. Users often experience noise from the surrounding environment which could reduce their attention span and memory load. An approach to this is to design for recognition of functions rather than memorization of commands [15, 11].

7. Design for multiple and dynamic contexts

The context of mobile users are more diverse in comparison with stationary computer users. The surrounding environment affects users behavior and interaction, for example in the presence of other people the feature of speaking commands to the mobile might feel uncomfortable as well as trying to cycle while searching the calendar. One way to approach this is to implement context-aware applications and self-adapting functionalities [10, 15].

¹W3C is an international community that develops standards to ensure the long-term growth of the Web [7].

8. Design for small devices

Because of the limited size of the display in mobile devices, data input can sometimes be problematic. In some environments speech input might be a good solution [10].

9. Design for speed and recovery

Because of users dynamic contextual environments, recovery of input data is desirable. For example if spending a few seconds writing notes in an application and something unexpected happens, this will shift your attention which increases the risk of input data loss. A good design would save the data enabling the user to proceed from where she was operating [10].

10. Design for "top-down" interaction

Only a limited amount of information can be presented on a mobile display. Different contexts enable different possibilities for focus and interaction. For example a worker who is busy might only want to know how important a received message is. To reinforce this constraint the designer ought to implement hierarchical information structures [10, 15].

11. Allow for personalization

Mobile devices are often personal. This opens up the possibility to personalize it. Users differ in terms of behavioral patterns, skills and preferences. For example if the application adjusts the font size depending on the surrounding light, some users might always want the font to be large [10].

12. Don't repeat the navigation on every page

When the space for displaying navigation options are limited the navigation items must be reduced; some applications show all possible navigation options on all pages which forces the user to scroll down to the desired information. This issue can be approached by breadcrumbs [25].

13. Clearly distinguish selected items

It is important to give the user feedback on selected items, the page could for instance be loading and if the user gets confused about whether or not he or she has pressed a button, it can result in multiple pushes which can further delay the page loading [10].

4.4.2 Guidelines for devices with a NUI according to J. Blake [3]

J. Blake suggestion of guidelines for design of a devices with a NUI.

1. Instant expertise

A skill is an ability to perform a operation that often has required practice, such as bicycling. A natural skill here refers to a skill that is significant for a human being, for example communicating or using fingers and hands in a number of ways. When designing for today's mobile devices and their corresponding NUI the designer ought to take advantage of user's natural skills to increase the chance of reducing the time spent on solving a task. If this matter is considered when designing an application, the user will quickly learn how to handle the device. Instant experts can be created by reusing domain-specific skills and

reuse common human skills. Domain-specific skills can be applied; for example if developing an application for medicine doctors, they surely know terms like cardio and metaphors that are specific for their domain.

2. Cognitive load

This guideline states that the developer should design so the user uses innate abilities and simple skills for the more common interactions. This will increase the probability that the user will suffer from less cognitive load and also experience the interface to be easy to use. This will result in a decrease of time spent on learning how to handle the interface. However, it is in the long run better to learn skills if reusing simple skills are not possible; for example learning how to navigate with a touch gesture is better rather than to navigate with the mouse, because the touch gesture will cause less cognitive load.

3. Progressive learning

The designer ought to design both for the novice and the advanced user. The user should be able to learn step by step how to manage the interface. As with gaming design, the user gets familiar with the features in the first levels before moving on to the more challenging tasks.

4. Direct interaction

It is important to design interfaces that are high frequency, direct and relevant to the user's context. High frequent means that it is preferable to have many small interactions with a small amount of feedback rather than as in most GUIs where the user has to navigate through a deep tree and not until the end state receive feedback in large chunks. Direct and high frequency interactions are present in the real world, for instance when you are cooking a stew, you stir and add ingredients one after another, and you get immediately feedback on consistency and texture.

Contextual interaction design means that the interface is mapped to the context of the user; this approach enables reduction of choices in the interface and minimizes the cognitive load. In GUIs all choices are often presented to the user at once, this makes fast navigation possible but it also increases the risk of choice overload.

The interface should make it possible for the user to directly interact with an object. Blake talks about three kinds of directness, Spatial proximity, Temporal proximity and Parallel action. Spatial proximity is when for example touching an object icon, the physical action of the finger is physically close to the object icon. Temporal proximity is the immediate feedback produced after a user input. Parallel action is when the user slides his or hers finger on a slider, the slider moves in the same direction as the sliding finger. When implementing direct interaction, the designer can minimize interactive objects in the interface. For example in the iPad the user can zoom in and out with the slide of his or hers thumb and index finger instead of having to press a zoom button. This direct interaction is faster and more intuitive and mapped to our interactions in the real world.

4.5 Conclusion

Little research has been conducted to establish mobile guidelines that are relevant for mobile users of today. In 2009, Nielsen showed in a study that 21 percent of the American popu-

lation were using a smartphone [9]. 45 percent of those who did not have a smartphone stated that their next device was going to be a smartphone, and as the prices are falling, and the technology is constantly improving smartphones will keep increasing in market share [9]. Designing mobile phones utilizing a GUI is not sufficient, traditional mobile guidelines needs to be expanded to also approach the NUI aspect.

J. Blake's material can possibly reinforce and expand Zakiah, *et al's*. GUI based framework. The list below is additional guidelines derived from J.Blake that is relevant to add to Zakiah, *et al's*. guidelines.

1. Design for context of use

Take advantage of user's domain knowledge and context. For example if designing for a medical team, communicate functions by using medical metaphors and information with medical terms. This makes reduction of interaction choices possible which minimizes the users cognitive load. However if fast navigation is more prioritized the option of present many options at the time is more suitable.

2. Reuse innate skills

Users have innate skills like communicating and to use their fingers in certain ways for certain tasks. For example using your index finger to move a lightweight object.

Enable the users to use their innate abilities and skills for the more simple and common actions. The user will in this way experience the interface to be easy and fast to learn and it will minimize their cognitive load.

3. Support both the novice and the professional user

The learning curve should be progressive, as with gaming, the first levels are easier and enables the user to overview the system and learning how to use all features before moving on to the more advanced levels.

4. Reduce large chunks of feedback

Design an interface that is high frequency and direct. Support continuous feedback in small chunks from user's input. The users should not have to make several interactions and only get feedback after finished the task.

5. Support direct interaction

Direct interaction minimizes the amount of interactive widgets, for example a list in which the user can choose an item for inspection, in the interface, for example pinching directly on a map to enlarge it instead of pressing a plus and minus icon. This operation is fast and intuitive and resembles interactions in the real world.

Chapter 5

Accomplishment

The following sections will describe the work process, what challenges arose and how they were approached. The project was divided in four phases; the planning, the design, the implementation and finally the evaluation phase.

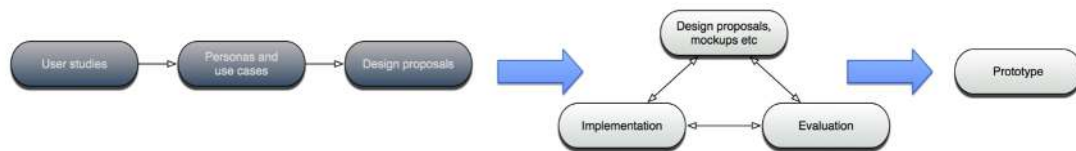


Figure 5.1: The dark grey bubbles represent phases that have earlier been conducted by ABB and the light grey bubbles represent phases covered in this thesis

5.1 The initial planning phase

This phase revolved around getting an overview of the project. Since this thesis project, which targeted the communication and collaboration aspect, was part of a big project that had been ongoing for almost a year, it was crucial to first study what had already been done in the project so far (see Figure 5.1). It was also important to get a good understanding in how the process control system works, how it is used, by whom and for what purposes.

5.1.1 Analysis of ABB's proposed solutions

Five use cases and three scenarios were related to the communication and collaboration aspect and was for that reason chosen as foundation for the developed design. Moreover, ABB's initial concepts that targeted these scenarios and use cases were analyzed in order to conclude what to use for further development.

The use cases

1. UC: Change Shift

The operators in a process automation plant are divided in different shifts. There was a need for a tool for enabling communication of important and relevant information to the following shift.

ABB's idea of a operator forum where time sensitive notes could be posted to a feed list was a good idea as it enabled the operators to use the same kind of tool to document events. It also enabled the operator to store the information in one place, instead of as it is today where notepads can be found in various places in the plant and sometimes notes are not documented at all.

The idea with a common virtual forum tool where information can be stored and easily accessed was further developed.

2. UC: Communicate information

The lack of an easy to use communication tool influences user's efficiency to run the plant smoothly. The suggested online forum enables a operator to tap on a process object in the process control software and send a message about its status. The operators could also add photos and videos. This idea will be taken in consideration because it is context aware, the online forum does not need to be started manually, the operators can add notes directly where they are operating, that is, around the object in question. It also connects the operators to the chosen object, this was further explored.

3. UC: Mobility support

There is a need for the operators out in the field to be able to control the process and also to gain relevant information. Today the operators in the control room must communicate through a walkie-talkie with the operators out on the field to, for instance, get feedback whether a machine part is running or not. A need for a mobile communication and control device was identified. However, the control aspect is beyond the scope of this thesis and will be ignored. The communication and information sharing part will however be approached.

ABB's initial concept of an operator forum that runs on a mobile device can possibly support the mobility aspect. However this concept does not approach the described mobility needs in this use case. But if information were available for operators on the field, the mobility aspect can still have an added value for the operators, this will be further explored.

4. UC: Be focused

Young inexperienced operators often experience stress and lack of confidence when they repeatedly are unable to handle alarms without consulting a senior operator. The user study identified difficulties for the operators to manage the control system and a lack of support for improving performance. The drastic change from calmly observing the monitors to the stressful situation when an alarm is triggered is too dramatic and it does not stimulate learning. This adds up to their lack of motivation and increased depression and stress. A need for an educational tool and an easier way to handle alarms must be developed. In order to increase focus some sort of positive stimulus might also be needed.

The suggestion to offer the operators to play computer games as a reward for certain achievements was a good idea in the sense that it supported an active operator. However simpler rewards can possibly have the same effect. When studying communities like Stack Overflow the users are motivated to be active

and to regularly contribute with information by getting positive visual feedback in terms of credits. These credits can be interpreted as some sort of representation of social status which has showed to be important for supporting users to contribute in virtual communities [17]. This increase the user's freedom to contribute and change information. In Stack Overflow the concept of awarding with credits has proven to be successful: contribution of information and active users are correlated to online community success [23]. Some sort of support must be explored that encourages operators to be active in a virtual community. Today operators report that they are feeling unstimulated and unable to affect their situation.

5. UC: Plan Shift

The more experienced operators are responsible for administration of shifts and its required resources. Information about the status of the different shifts is needed in order to inform the operators about certain problem areas and to effectively plan the shift. A need to retrieve relevant information is identified. The above suggested online forum could answer also to this aspect but needs further development.

Analysis of scenarios

1. Monitoring - UC: *Communicate Information*, UC: *Plan Shift*, UC: *Be focused*

Before Tom, Nick and Kumar start their shift they must talk to the previous shift to gain important information about the status of the plant. This information helps Tom to plan their daily activities. Kumar and Nick are responsible for different workstations that each holds several screens that display information and the status of the plant. To get some variation and to learn how to handle both stations they rotate workstations.

The previous mentioned operator forum can possibly be developed to engage the operators in a meaningful way. Although the process is automatically controlled, Nick and Kumar need to stay focused and make proper adjustments on process objects to avoid process stops. The forum might be a tool for retrieving information about plant events, as earlier described in UC: *Change shift*. This would make it easier to get an overview of the system.

2. Maintenance - UC: *Communicate Information*, UC: *Mobility support*

The operators do regular maintenance out on the field which requires information that is only available in the control room. In this scenario Nick needed assistance from an field engineer to solve a problem.

The suggestion of only making the operator forum mobile does not answer to this scenario. However a mobile device that displays instructions might help Nick, this would make him more efficient and independent in his work.

3. Handling an abnormal situation - UC: *Be focused*

In this scenario Nick was daydreaming and missed to prevent a process stop. He was also not able to handle the stressful situation because of his lack of knowledge. Some sort of alarm guidance might help operators like Nick. When an alarm is triggered a corresponding description and solution can be displayed.

5.2 The Design phase

After analyzing the use cases and the scenarios it was concluded that there was a need for something more than an operator forum where only notes could be displayed and produced. The use cases and the scenarios were of great support in understanding user needs and behavior.

The current way of connecting objects and grouping information resembles the architecture of online communities like Facebook with its users that belong to different groups. More of this will be described in the Social media section with examples. Could this architecture be used in the design of the communication and collaboration application? Facebook is interesting to study because it supports communication and information sharing in an intuitive and successful way. Social media have features that are relevant for the communication and collaboration aspect. Some interesting user behavior has also been identified in social media applications that is relevant to analyze.

The concept design will be presented in relation to the described five use cases and three scenarios.

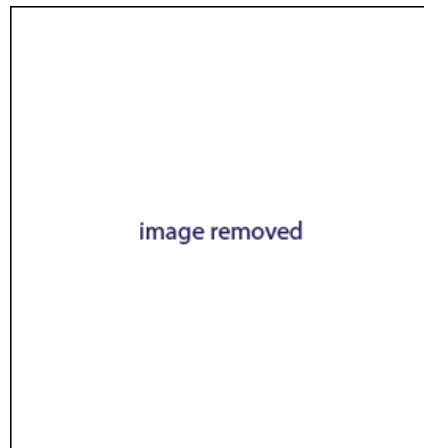


Figure 5.2: Process control software holds references to several thousands different objects. The picture above represents one of these objects, Boiler A1.

5.2.1 Social media

Social media like Facebook, Stack Overflow and Get Satisfaction will be described in relation to current process automation software and to the communication and collaboration aspect.

Facebook

Facebook is a community that originally was a platform for students and is today the largest social media network. The community has members from all over the world with different backgrounds and interests. Facebook enables people to share information by posting comments on their friend's pages, and to create profiles with personal information and pictures. It also has the functionality of creating virtual groups based on politics, interests or hobbies. The average member spends about 20 minutes per day on Facebook and two thirds of the community accesses it at least once a day [8]

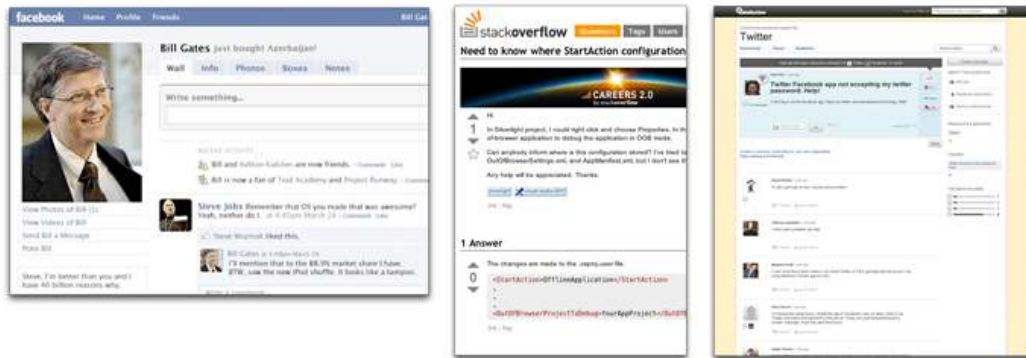


Figure 5.3: The left image is a screenshot from Facebook, the one in the middle is from Stack Overflow and the right one is from Get Satisfaction. These can be found in larger sizes in appendix A

Facebook is built on automatically generated feeds, users and their profile pages (see Figure 5.3). Everything that happens in the community that is related to the user is visualized in a feed list, it can be their friend's birthday or that the user has received an invitation, every event is simply listed.

The strength of Facebook is how its users are connected at different levels and how information is filtered in an intuitive way. The application is built on algorithms that for instance calculates your friend's friends and how they are connected with you. When the user searches for another user, Facebook initially starts to traverse added friends and then members that somehow are related in terms of location, groups, occupation and interests. This hierarchy of objects and information has similarities with the object-oriented structure of the existing software.

These similarities between Facebook and current software developed by ABB have been identified;

- Facebook and current process automation software have objects/users with associated properties
- Facebook users are connected at different levels; friends are connected with each other and can then be further connected by certain groups of interest. Current existing software have process objects that can similarly be connected in different levels. For example see Figure 5.2, where the process object A1 holds several alarms.
- Connectivity: Facebook is a network where everything is connected, one user can influence a large part of the system. A similar model applies for the current software, with its process objects, if one process object malfunctions it could affect the whole plant.

How can this successful information design of Facebook be applied in the Communication and collaboration application?

Stack Overflow

Stack Overflow is a social media application that is built on users and their posted questions and answers about programming related issues. Both the questions and the related answers can be rated by users in terms of relevance or how good it was formulated. This is an efficient

way to communicate quality of information which is desirable in the Communication and Collaboration application. As users gain credits from contributing to the community they get more freedom to edit information and to vote. They receive batches that represent's the earned credits. This encourages users to contribute with information. Stack Overflows way of encouraging users to gain credits and how they are rewarded with more freedom to modify material was found interesting. This is apparently enough to support user's willingness to be active. Without user participation, in these kinds of applications, they have no meaning as the content is provided by the users.

Get Satisfaction

Get Satisfaction is a support service, encouraging dialog between companies and their customers. It is a community where messages can be posted and replied to both by users and company staff members. Indicators show's users current mood related to the post, when users post their messages they are prompted to inform how content they are. The result is visualized in the side bar. Can this feature be used to visualize status of certain areas in the plant or the quality of information?

Get Satisfaction has realized the value of user generated information, users show's an altruistic behavior by making an effort to share information with each other. The application enables solutions to spread and to avoid having to be repeated. This is relevant for development of the alarm support feature. The web site claims that over 48 000 companies are using the service, amongst them are famous companies like Facebook and Twitter.

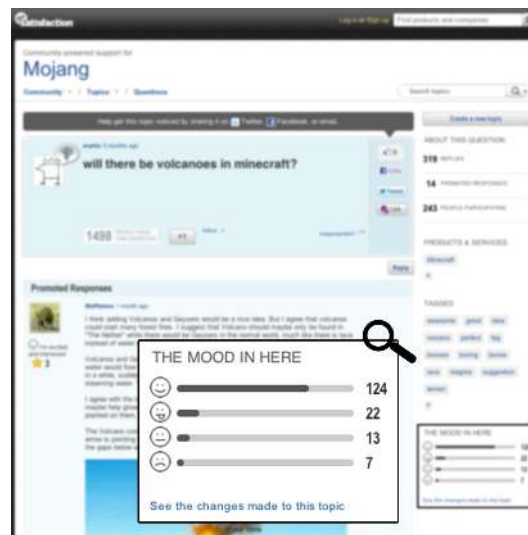


Figure 5.4: The support community Get Satisfaction; the mood related for a specific topic is represented with 4 smileys

5.2.2 The design

The operator's domain consists of several computer screens that displays changes or alarms in the plant. The operators are constantly exposed to a lot of information, for that reason it was important to design an unobtrusive application.

The finalized idea is to offer an additional layer in the current software solution that can be either manually, or automatically visualized. For example when an operator is sitting in front of the control system, entering an operator note on a on a visual representation of a process object. The developed application is launched and displayed as an additional layer on top of the current process automation software interface. The design also enables the application to be started by the operator by browsing the menu.

The suggestion of an operator forum was developed to a virtual community similar to Facebook.

The functionality of automatically generated feeds that can be seen in popular new media is a feature that would most likely improve the usability of the existing process automation software since it displays information in an efficient way, this architecture will be used in the design.

The idea with this design is: when an alarm is triggered in the plant a corresponding alarm feed will be generated and visualized. The same functionality applies to different kind of settings on process objects or user generated events.

Moreover, the operators can in the same manner as in Facebook and Stack Overflow make comments on feeds. This operation connects the operator to that specific process object and alarm. This is possible because every operator has a simple profile page which enables the operators to browse their coworker's page to further look into what they have been doing during their shift.

The communication and collaboration design compared to Facebook is that also process objects and alarms have a profile page. This enables the operators to retrieve information in an intuitive and easily accessible way.

The concept design derived from the use cases will here be further explained.

1. Change Shift

The application supports information archiving relevant for shift change. The operators can add important identified feeds to a feature called Shift report. The following shift can access this function to retrieve important information about alarms or process objects that have been problematic.

2. Communicate information

Communicate information is interpreted as not only user to user communication but process object communication as well, *e.g.*, when an alarm has been triggered this is visualized in the feeds list in the same manner as when an operator has made a change on a process object. The automatically and manually generated feeds are neatly sorted in the feeds list by time. The feeds belongs to four different categories; Alarms, Changes, Notes and Proposed alarm solutions. This enables the application to have a feature for filtering by category, which makes it easy to overview information. The operation of entering profile pages for alarms, process objects and operators, can be seen as a sort of filtering operation as well. When, for example, an alarm profile page is entered, information related to that alarm is listed.

A manual feed can also be posted by the operator, however the feed is then not connected to an alarm or a process object. Names of users, process objects and alarms are clickable. The strength of this concept is that user events and information are connected and can be easily accessed, in the same manner as Facebook

The operators can also post a message from the process graphic display, the message is sent to the feeds list. Movies and a pictures can also be added. This

operation connects the operator to that specific object which enables users to filter by operator or object.

3. Mobility support

Operators at plants that have frequently occurring alarms, could benefit from this mobility support, for example, an operator can take a coffee break and still monitor triggered alarms, and if an alarm has low priority the operator can finish her coffee. This design facilitates shift planners to be mobile instead of having to collect information on a stationary computer. The workflow is in this way not constrained by how information can be accessed.

4. Be focused

A tool for support of handling alarms was designed. The idea was that when an alarm is triggered a corresponding alarm description and solution appear on the screen (see scenario no.2 in the following section).

There are a lot of different alarms that can be triggered in a plant, and one specific alarm can have many solutions. The solutions are also plant specific. To address this, the alarm support supports the operators to contribute with their own solutions, similar to the community Stack Overflow. In this design, when the operators posts a solution they earn credits which will hopefully encourage contribution and a more active approach. If an alarm description has many added solutions these will be listed by their ranking. If for example an operator has experienced an alarm, read the corresponding solution and found it useful, the solution can be voted up one credit. The solution logs who has posted it and when. It is also possible to add screenshots to these solutions.

In this way the unexperienced operators can be more independent and be more confident in their work. Personal development and a sense of control of their own situation should increase their sense of content and reduce their stress levels. This support is likely to motivate and stimulate the operators to stay focused. The alarm assistance is of course of value to all operators.

An additional support was also designed to support young inexperienced operators, the Alarm library feature. This feature is as the name suggests a library of alarms that can be entered and studied for learning. The feature consists of two category lists, one list hold alarms that is often triggered in the plant and the other list hold alarms that have gained most votes.

5. Plan Shift

The operators that have responsibility to plan a shift could, using this tool, easily access information through the logged feeds. If a shift has experienced a lot of problems or if a process object has been more problematic than usual. The communication and collaboration application enables intuitive and easily retrievable information relevant for planning shifts.

5.2.3 Integration of the communication and collaboration application with existing process automation software

One important challenge with designing a communication and collaboration support system was to integrate it well in the existing software to support the operator's work flow. An additional program to add to the complex process automation software increases the

operator's cognitive load. It was for that reason important to make the application context aware. This was solved as mentioned in the previous section by making the application appear when an alarm is triggered.

The Communication and Collaboration application was integrated into the existing workspace by adding functionalities into the existing context menu (see Figure 5.5). By right clicking any object a note on that particular object can be entered, this data will then be found in the process feed and can also be retrieved by entering the profile of the mentioned process object and the operator.

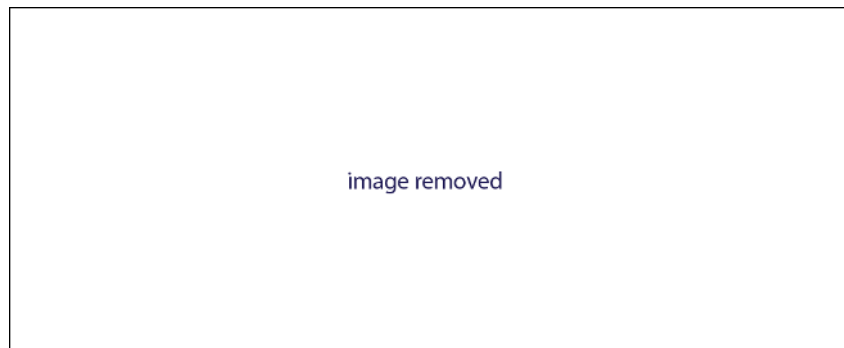


Figure 5.5: How the application was integrated into the existing workspace

Another important integration with the existing software is the Alarm assistance feature. When an alarm is triggered in the plant, buttons for muting and inspecting the alarm appears on the screen. When an operator presses the corresponding button the Communication and Collaboration application starts and visualizes both the alarm description and the proposed solutions.

An operator can also click on a process object in the existing software to display its corresponding profile page. This way of integrating features of the Communication and Collaboration application reinforces the impression of a diversified and coherent software.

5.2.4 Scenarios

1. Kumar is entering an operator note

Kumar is sitting in front of his four screens in the operator room. For the past few hours he has worked with a large number of alarms that have been triggered in the process object Boiler A2. He suspects that the pressure is too low even though the system indicates normal levels. He wants to monitor its behavior before he takes his idea further. Kumar right-clicks on Boiler A2, a text input window appears on the screen. He starts to type his concerns and that he strongly suggests that the following shift should be aware of the problems caused by Boiler A2. He opens the Communication and Collaboration application in the menu bar. In the main feeds list he can see that his message on Boiler A2 has been added, he chooses the "send to" and the "shift report" button. The feed has now been added in the Shift report. Kumar's concerns about Boiler A2 can now be read by the following shift. Tom who is working on the same shift as Kumar discovers Kumar's note when browsing the feeds list on his iPad in the coffee room. He clicks on the name Boiler A2 in the message and navigates to its profile page where he can inspect the status of Boiler A2. He agrees with Kumar, Boiler

A2 is behaving really strange, this object has had a lot of alarms the past few days. He adds a comment in response to Kumar's entry: *"Let me know if you guys experience as much problems as we have, then I must take this to higher management."*

2. Nick is handling an alarm

Nick is sitting in front of the operator workstation, monitoring the plant process. Suddenly the familiar sound of an activated alarm is cutting through the silence in the control room, his heart rate frequency increases. He is alone in the control room, Tom and Kumar are out on the field. *"This is my chance to master the situation"*, Nick thinks to himself. He stares at the warning signal that has appeared on the screen, he mutes the alarm sound and with a slightly shivering hand he clicks on the signal. The Communication and Collaboration application appears on the screen, visualizing the alarm description. *"Temperature too low"*, Nick does not recognize this type of problem. It has two proposed solutions, the top solution has a rating of "6" and the other is rated "2". The chance that the top solution will help Nick solve this alarm is higher than then the bottom one. Nick starts to read the solution with a rating of "6": *"..decrease g6 levels to 30 percent..."*, a picture has also been added, which gives Nick additional information of where to make the necessary adjustments.

Nick manages to solve the alarm, the alarm signal disappears from the screen. When he enters the feeds list he proudly can see a feed has been added that states Nick solved alarm in Tank A5. He goes back to the alarm assistance feature and rates the solution: *"this solution did indeed help me to solve the alarm"*.

5.2.5 Low level prototyping and validation

Low level paper prototypes were produced (see the left screen in Figure 5.6). Discussions were held with expert users about the design. Their inputs were valuable to fully understand the targeted user's workflow and needs.

After several iterations the prototype was ready to advance one fidelity level. Digital sketches were made using the software tool Omnigraffle (see the middle screen in Figure 5.6). Omnigraffle enabled creation of an interactive low-fi prototype. Therefore it was possible to on an early stage test the application flow and different layouts. This made it easier to understand the proposed functionality and how the application was supposed to be used. This also made discussions easier and it was also found helpful when entering the implementation phase; it was surprising how fast important details were forgotten. This design was presented to project stake holders. An additional prototype was also constructed that only targeted the graphical design of the application (see the right screen in Figure 5.6).

5.3 The Implementation phase

The majority of time available for the project was spent in the implementation phase. The chosen Framework Sencha Touch had to be studied to learn how it could be used and what kind of functionality Sencha Touch would be able to provide. The main sources for learning Sencha Touch and JavaScript were online videos, books, live examples and online tutorials.

As work progressed beyond simple examples there was a need to structure the project according to the Model View Controller (MVC) pattern, see Figure 5.7.

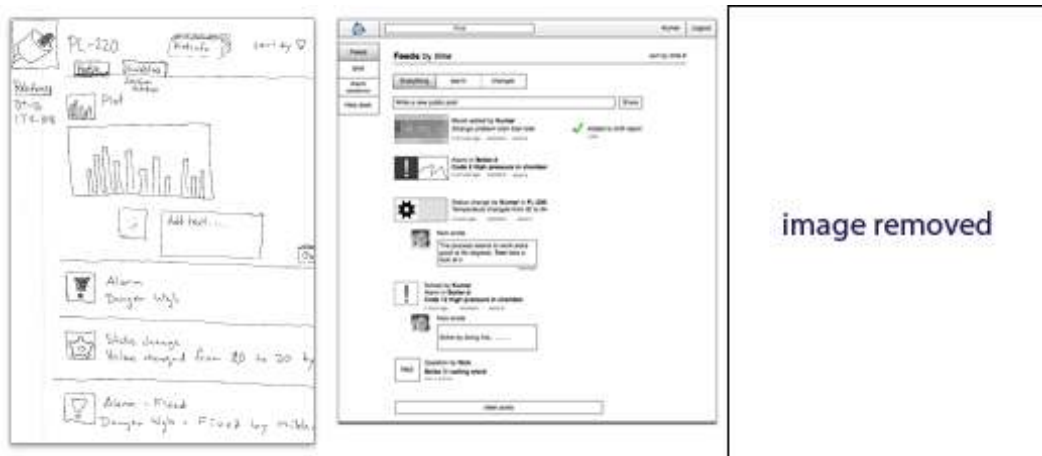


Figure 5.6: The left screen shows simple sketches, the middle one enables interaction and the final one is not clickable and is only focusing on the design layout, see Appendix A for larger images

5.3.1 Structure of the application

Most of the Sencha Touch examples used simple code which was packed into one big file. It was quickly decided that some kind of programming pattern would have to be utilized as the code otherwise would become too complicated and hard to expand. Sencha Touch strongly encourages developers to write code based on the Model View Controller pattern, see Figure 5.7.

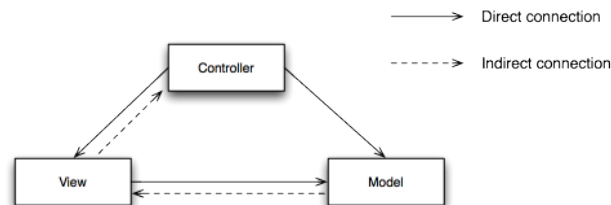


Figure 5.7: The implementation used the MVC pattern

The MVC pattern enables separation of logic, user interface and the application data which aids the development as it allows developers to work on separate parts of the project without interfering with each other.

The application was implemented to have a master view, called viewport. As different views would become active the viewport would load the new view and replace the old view with the new. The viewport would also make sure to destroy old views as they became inactive in order to conserve system resources. The controller that implemented most of the application logic was the master. The other controllers were mainly used to collect similar code, e.g all database requests were handled by the connection controller, see Figure 5.8 for a detailed list of classes.

When launching the application the application would first start by asking the server for an event list. This event list contains everything that has happened in the system, but does not hold the actual data of what has happened, only a unique id of an event and the

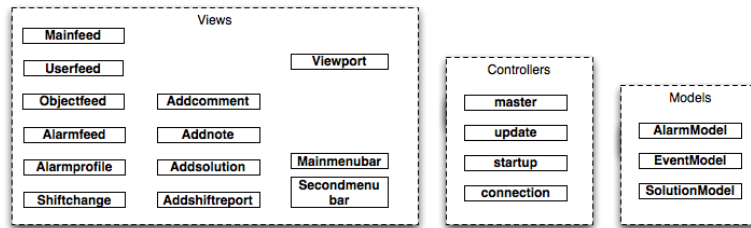


Figure 5.8: All the Models, Views and Controllers included in the application

event members. The event list was transferred to the application using JSON-P format.

JSON is an acronym for JavaScript Object Notation and is a standard for transferring data in an human-readable form. As a security measure browsers do not allow a page to request JSON data from a server in a different domain. This restriction limits how and where JSON data can be placed. JSON-P, or "JSON with padding" evades this limitation by padding the JSON with html script tags. Therefore the data is not perceived as JSON, but instead as a script, ready to be evaluated by a JavaScript interpreter.

JSON-P hence opens the possibility for cross-domain communication, this was important during development since the database was hosted on a separate server and during development the application was stored on separate machines, thus making cross-domain communication necessary.

As the application receives the event list it starts looking through it for unique id numbers. All the found id numbers are then sent to the server and the corresponding object is returned. When every id has been requested and returned the application will store the fetched data in a DataStore and then display this data on the screen. It will then start polling the server for updates, see Figure 5.9.

In short, the application would load data from the server and store this in the Model using a Sencha Touch DataStore. The View then asks the Model for data to display on the screen. When the user interacts with the application or something needs to be calculated the Controller holds the logic to do that, including the ability to tell the View what to display on the screen.

5.4 The Evaluation phase

The Communication and Collaboration application was tested on three operators from Kraft Värme Verket in Västerås that works with ABB process control software on a daily basis. All three operators had been working at the plant at least 10 years. The evaluation was conducted utilizing "The Think Aloud protocol" and "The System Usability Scale" (SUS) [13]. Moreover, with inspiration from the Get Satisfaction application where users are prompted to reply on how content they are regarding a concern, the informal method "The mood in here" was applied in order to get an overall view on users feelings towards the application.

The "Think Aloud protocol" is a method where the users are asked to think aloud when testing an design. This is a qualitative method although the collected data is time consuming to conclude and overview.

The SUS questionnaire is a fast and efficient evaluation method where users are prompted with questions concerning a solution. The data is then analyzed by using a framework.

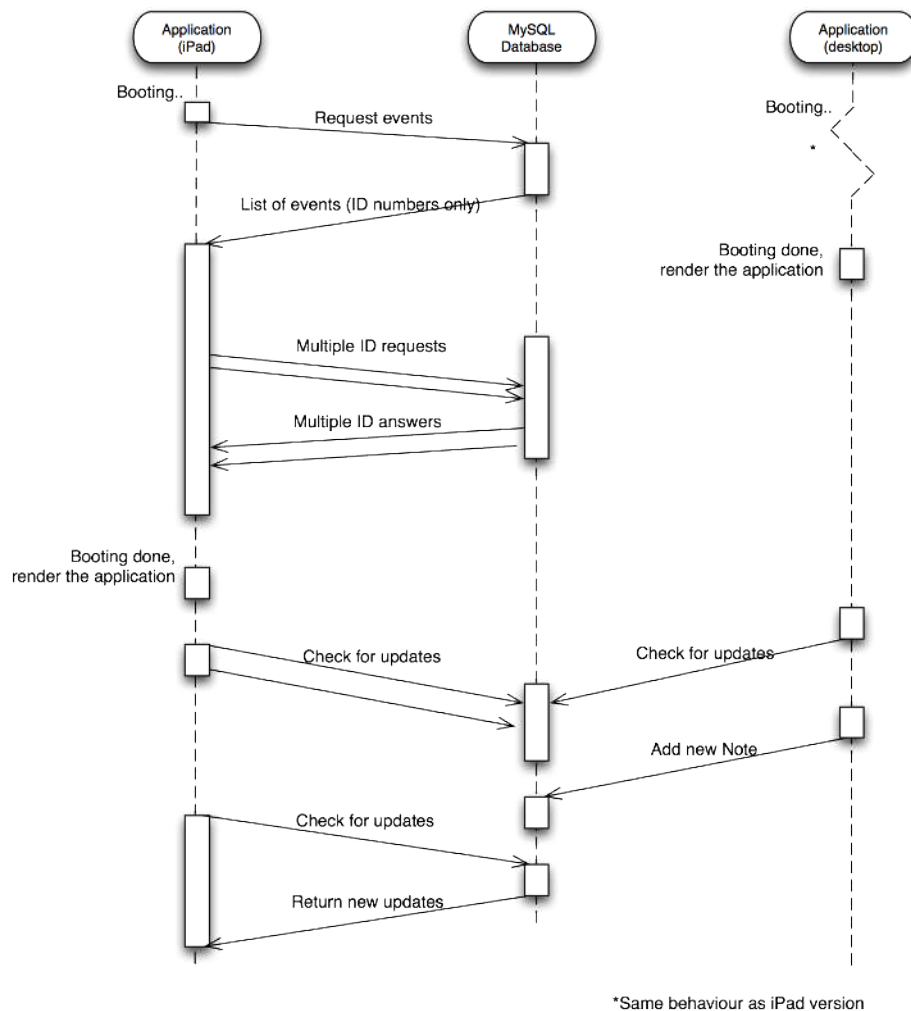


Figure 5.9: UML-sequence diagram of the application flow

This section will describe how the user tests were conducted and how the tasks were designed. The prototype was not feature-complete during the evaluation but it was considered important to on an early stage get input from plant operators about the functionality.

5.4.1 The Think Aloud protocol

This method aims to identify users thoughts and concerns about a solution. The user are given a set of tasks in which they are asked to solve by thinking aloud. This protocol can provide a variety of cognitive and qualitative information such as errors made, information used to solve the task, applied strategies or rules.

The comments from the users are either documented by writing and/or by using a

recorder. The transcribing is time consuming, for each hour of interview the conductor must spend 10 hours of documentation, analysis and reviewing. The talking in it self interferes with users effective thinking. The advantages are that it is easy to conduct, no special tools or or methods to learn. The Think Aloud protocol often provides a lot of information in short time [13].

5.4.2 SUS - the System Usability Scale

SUS is a cheap and fast way to measure usability. Cheap and fast in the sense that SUS evaluations are both easy and fast to perform once the SUS statements are formulated. The user is asked to rate provided statements on a Likert scale, from 1 (I disapprove) to 5 (I approve). In this evaluation the original SUS statements were used, see Appendix B.

SUS questions are designed in pairs, for example the statements *"I found the various functions in this system were well integrated"* is paired with *"I thought there was too much inconsistency in this system"*. If the user does find the various functions in a system to be well integrated (a high number on the Likert scale) the same user should also give the second statements a lower score on the Likert scale. If the user should rate high scores on both statements, the result would be contradictive.

An algorithm is used to process the unbalanced data from the finished SUS score and the result can then be evaluated and compared using the SUS scoring scale, see Figure 5.10 [32]

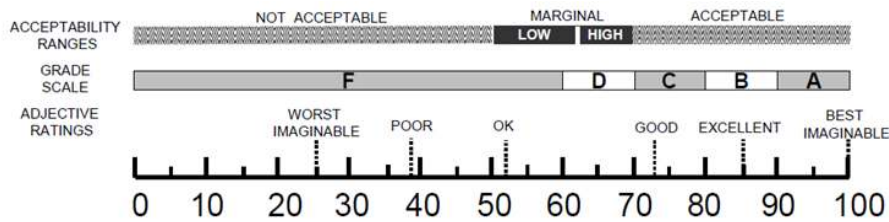


Figure 5.10: The SUS scoring scale

SUS questionnaires are often used to measure effectiveness, efficiency and user satisfaction.

5.4.3 The mood in here

With inspiration from the Get Satisfaction application a section was added to the evaluation that simply displayed four smileys (see Figure 5.11) in which the operators were asked to choose the one that represented their overall feeling regarding a feature. The operators were asked to choose a smiley after finishing each task. In Get Satisfaction (see Figure 5.4) they use this as an indicator of how content or displeased the users are regarding a stated concern. This simple approach can possibly be a good complement to the SUS form and the Think Aloud protocol.

5.4.4 The setup

The operators were given a general introduction about how staff from ABB had visited plants around the globe, interviewing operators and taking notes on their needs and what

they thought about the current software system. Based on these data certain areas were identified where operators either seemed to miss functionality or the existing functionality was not used for different reasons. This introduction was given to the operators in order to show them how important the users had been during the evaluation, that their voice would be heard and their inputs would make a difference.

The evaluation contained three stations which each operator would visit one at a time. All stations were designed to test different functionality. The collaboration and communication application was set up as one of these stations, the two other stations had nothing to do with the collaboration and communication application and will not be further discussed. The communication and collaboration station will from now on be referred to as station 3.

Station 3 utilized a dual monitor setup and an iPad and iPhone for the testing. An audio recorder was used to save what the users said during the testing. Each station had two test leaders where one had the role of guiding the operator and give the operator the tasks. The second test leader would take notes on the test subject's behavior, comments and reactions.

5.4.5 The evaluation

The operators were one at the time given a set of simple tasks to complete while applying the Think Aloud protocol. If there were any questions the test leader would answer them. The test was held mostly in Swedish. All the presentations and questionnaires the operators would fill in had been translated in order to avoid any confusion and misinterpretation. The Communication and Collaboration application was however in English. For every task the operators would perform there was a short discussion whether the tested functionality might aid the operator in his work and how he would want to use the functionality.

Operator tasks

The following tasks are the different tasks operators were asked to perform (see Appendix C for more details):

- Add a note on a process object (This task was performed on a stationary computer)
- Choose and save information for a coming shift change (This task was performed on a stationary computer)
- Try the application on an iPad (This task was performed on an iPad)
- Use alarm assistance to help other operators (This task was performed on a stationary computer)
- Use alarm assistance to get assistance and solve alarms (This task was performed on a stationary computer)

When the operators had finished with the five tasks they were asked to fill in a questionnaire. The questionnaire was a SUS questionnaire [4]



Figure 5.11: The operators were asked to mark one smiley that best represented their overall feeling after each finished task

Chapter 6

Results

This Chapter shows the results from the implementation in the Sencha touch framework. Moreover, the results from the evaluation is shown from the user study methods: "The mood in here", "The SUS form" and the "Think aloud protocol".

6.1 The implementation

The application is written utilizing the Sencha touch framework which enables the application to run on different platforms such as a tablet, smartphone and a desktop computer. The prototype was realistic enough to enable a valid evaluation, the users could test important functions, for instance, handling an alarm, making a comment, entering the shift-change report and filtering information.

The aim of the implementation was to cover as many of the identified use cases and scenarios as possible.

The implemented features covers the following test scenarios:

- **Add an operator note** - A operator adds a note on a process object. The operator can add pictures along with the text in order to better describe the situation.
- **Add a comment on a note** - Another operator sees the above note and wants to add his own opinion on the particular note
- **Solve an alarm with help from Alarm Assistance** - An inexperienced operator like Nick is sitting at the operator workstation and receives an unknown alarm. With the assistance of earlier notes and screenshots entered by another operator at the plant Nick manages to solve the alarm by himself.
- **Learn more about alarms** - Operators with nothing to do should be able to study solutions for alarms so that when the alarm happens they are more prepared to handle the situation.
- **Shift handover** - By being able to mark certain information as extra important this marked information will then show up in a "shift log" making it easy for operators to save information that they need to discuss with the next shift, or if the next shift is delayed they can write their comments directly on the saved information.

- **Plan shift** - The senior operator is usually the one responsible for shift planning. Having a digital shift log from the previous shift with important notes and events makes it easier to do the shift planing and know that nothing is lost or forgotten.

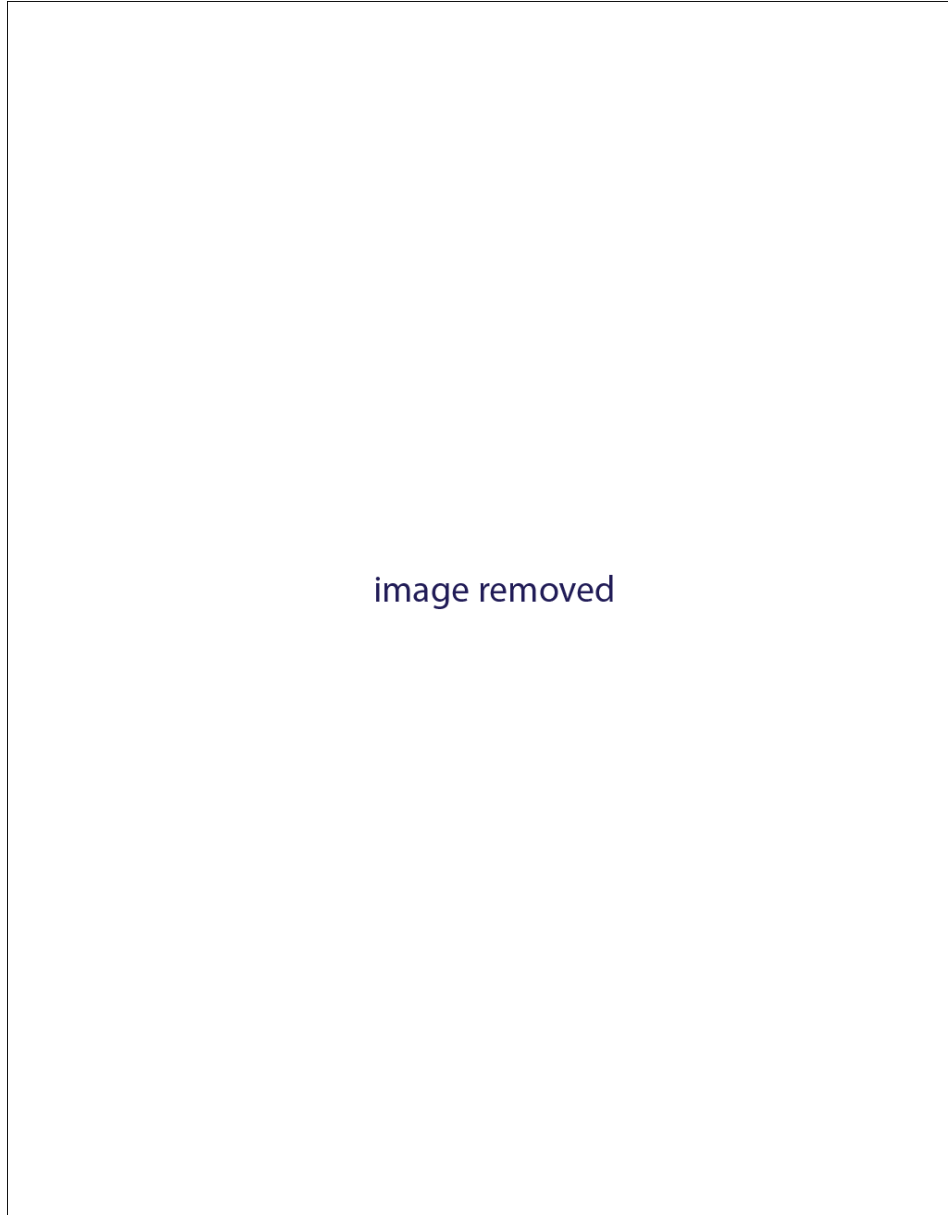


Figure 6.1: Screenshot of the application running on a stationary computer using a 27" screen (the full width of the screen was not used)

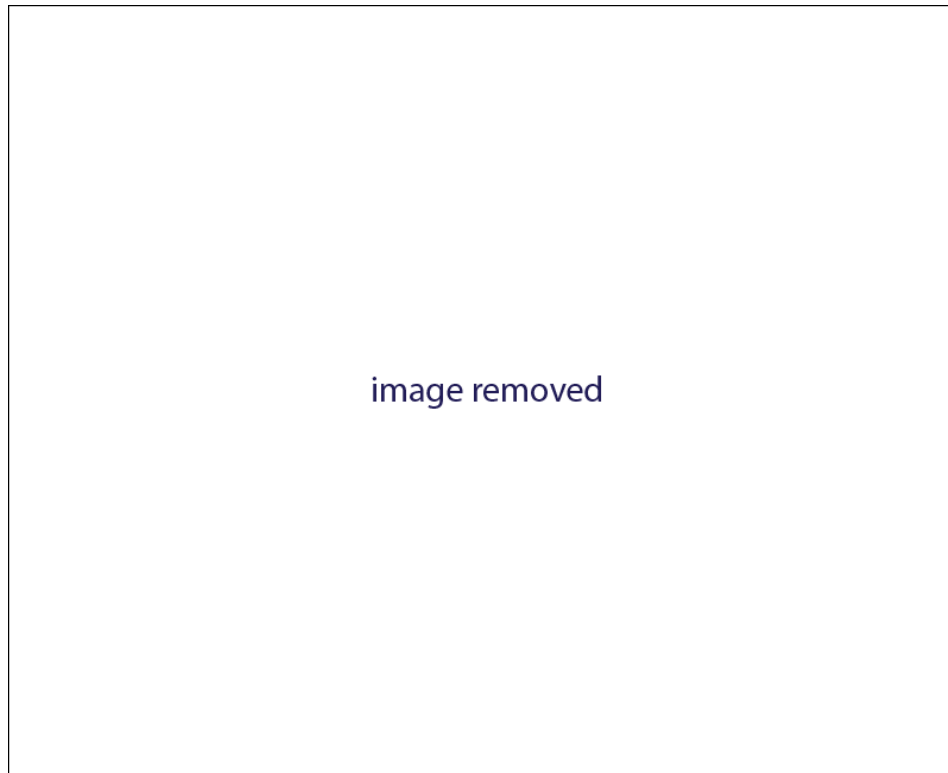


Figure 6.2: The application running on an iPad (to the left) and on an iPhone (to the right), showing the difference in available screen size

6.2 Implementing for multiple platforms

The implemented application was able to run on three different devices without any changes to the code. As a technical demonstrator this works fairly well. However a few usability and user interface problems can be identified, suggesting that even though it is possible to use the very same code across several platforms the application should be tweaked to allow for modifications to the user interface depending on the platform currently used.

Identified areas where problems might occur when implementing for multiple platforms

- Screen space is valuable - As seen in the right display in Figure 6.2, a large part of the screen on the iPhone is occupied by the filter function. Although the filter functionality is not as important as the menu buttons at the top, it is still an important feature that should still be accessible on every platform. A possible solution on the iPhone could be to only display the filter functionality when the user is at the top of the current page, see Figure 6.3. If the user scrolls down, the filter menubar should also scroll away, thus leaving more screen space for what the user is really interested in; the content. Moreover, the filter function occupies all space on the second menubar, leaving no room for an indication of where in the user interface the user is at the moment, this could have been solves by replacing the big filter button with a smaller

one, more suited for the small display of the smartphone.

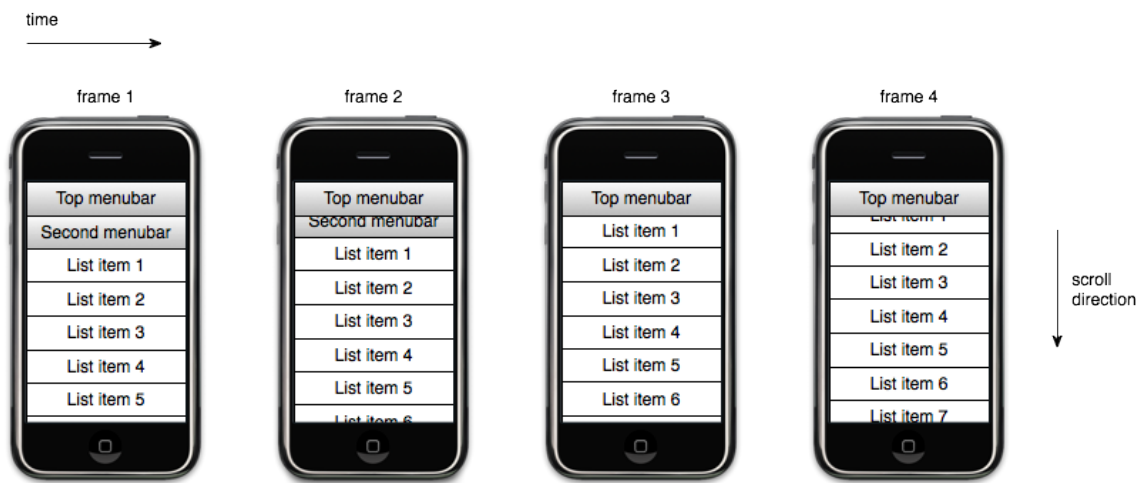


Figure 6.3: Illustration of how to better utilize screen space on small devices such as an iPhone

- Performance will vary - If several devices should be able to run the same application, performance should be tuned to give users of all platforms a good user experience. The Communication and Collaboration application works best on a desktop computer as the user interface is smoother and not as choppy as it can be when using the mobile devices (iPad and iPhone). If the application should be further developed the programmer should take performance issues into consideration as well as to provide modified code depending on device. An example could be to disable gradient effects and smooth corners on less powerful devices in order to provide the user with a fluid, but however graphically less intense experience.

6.3 What have not been implemented

One of the scenarios involved operators who experienced their work as extremely boring and difficult to focus on for long periods of time. To approach this, games and online forums were studied. These types of games/applications often succeeds to encourage their user's participation and also to make them want to return to the game/application.

With this as background the design of the Communication and Collaboration application was to award users with achievements for performing certain tasks. This design would hopefully inspire users to be more engaged and focused at work, instead of feeling bored and not involved. This was however never implemented due to time constraints.

The graphical component that enables filtering has an unwanted glow. This was unfortunately not possible to remove.

6.4 The evaluation

Results from the user evaluation included both SUS scores and user comments utilizing the "Think aloud" protocol.

6.4.1 SUS questionnaire

A SUS score above 55 is considered minimum and the result should ideally be above 80 to safely indicate that the users found the tested application satisfying.

The SUS scores were calculated and plotted, see Figure 6.4. No operator scored below 55; operator 1 scored 55 points, which is translated to "good", operator 3 scored 80 points, which is translated to "excellent" and operator 2 scored, 96 points, which is translated to "best imaginable".

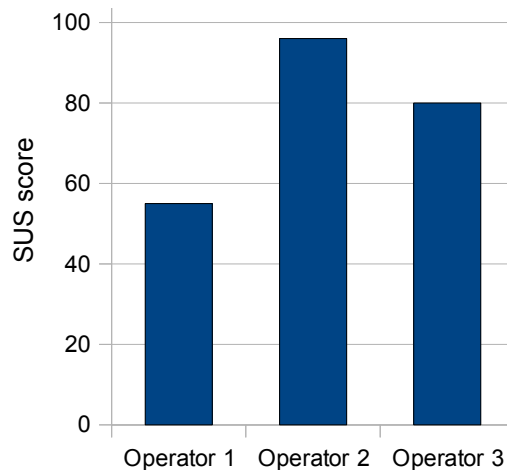


Figure 6.4: The SUS scores for the three operators participating in the evaluation. On the SUS scale, 0-25 corresponds to "worst imaginable", 25-39 to "poor", 39-52 to "okay", 52-73 to "good", 73-85 to "excellent", and 85-100 to "best imaginable". Operator 1 scored lowest, 55 points, which is mapped to "good", operator 3 scored 80 points, which is mapped to "excellent" and operator 2 scored highest, 96 points, which is mapped to "best imaginable"

6.4.2 The Think Aloud protocol

This section presents the core essence of the users opinions and concerns regarding each given task, for a full summary of the user quotes, see Appendix D.

Make an operator note on a process object

The possibility to add pictures along with notes was highly appreciated. The operators were concerned about timestamps, they should be editable by the users. Some form of filter functionality should be implemented, some plats have a high load of occurring events. User generated content should be more visible than auto generated content. Mainfeeds should be renamed to Logbook.

Choose information to send to Shift report

The operators saw an added value in selecting important feeds for the next shift to take part of. Some form of feedback would be necessary when feeds were added to the shift report. Maybe trend support could be integrated with this feature.

Mobility for increased process awareness

If operators would have the ability to control the process from an mobile device this would be very beneficial for them. The operators still saw some value in accessing alarm information and updates outside the process control room.

Alarm assistance

Operators saw a need of having a default solution for every alarm in addition to user added solutions. A better integration could be beneficial, show solutions to alarms in the faceplate. The operators saw no need for video, pictures were however highly regarded.

Alarm library

The opinions were diverse regarding this feature. Some thought it would be good for novice users, while some did not see the need for it at all.

6.4.3 The mood in here

The operators were asked to choose one of four smileys that best represented their overall feeling about the design and functionality of the Communication and Collaboration application, see in Figure 5.11. The aim of each task was to investigate how relevant the corresponding feature was for the operators. The smileys was mapped to values 1 - 4 points with the "very unsatisfied smiley" scoring 1 point, "unsatisfied smiley" scoring 2 points, "happy smiley" scoring 3 points and the "very happy smiley" scoring 4 points. A mean value was calculated and plotted (see Figure 6.5). When studying Table 6.1 one can see that operator no. 2 had scored 3 on all tasks which lowers the mean value.

In general the operators seemed to like the alarm assistance and the alarm library feature. A slightly lower mean can be noted regarding the features "Add an operator note", but it is still in the positive judgement range (≥ 3), see chapter 7.2.2 for a full analysis.

The tasks "Add to shift report" and "Mobility support" received a rating of "2" from two operators. However, the mean value was "3" which is mapped to a "happy smiley", see plot in Figure 6.5.

	Task 1	Task 2	Task 3	Task 4	Task 5
Operator 1	4	2	4	4	4
Operator 2	3	3	3	3	3
Operator 3	3	4	2	4	4

Table 6.1: How the operators experienced the features after solving task 1 - 4. Task 1 = Enter a operator note, Task 2 = Send to Shift report, Task 3 = Mobility support, Task 4 = Alarm assistance, Task 5 = Alarm library

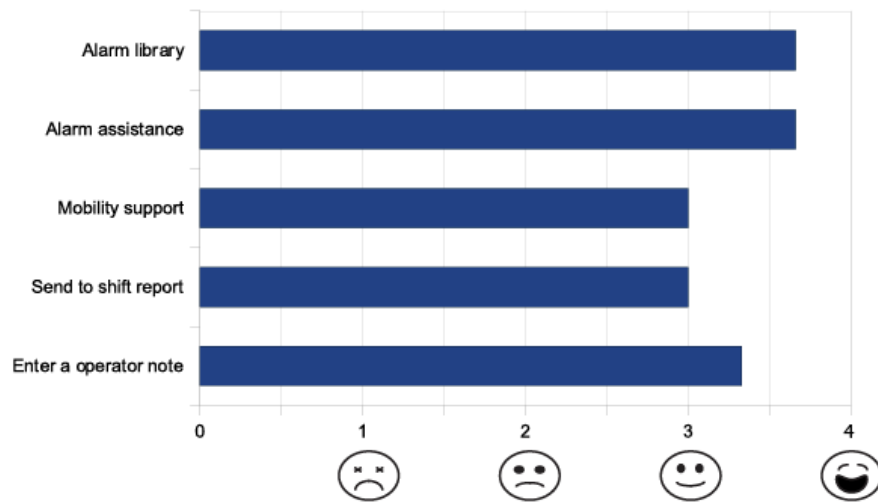


Figure 6.5: The plot shows mean values of how the three operators felt regarding the features of the Communication and Collaboration application

Chapter 7

Conclusions

In this chapter the framework Sencha Touch is evaluated. Moreover, this chapter also discusses and compares the material from the three user study methods "The SUS form", "The mood in here" and "The Think Aloud protocol". This thesis ends with comments about future work and a general discussion.

7.1 Implementation

When implementing a web application it is crucial to investigate what platforms the application will be used on. As discussed in Chapter 6, an application that can be deployed on different devices must adjust its user interface to suit the different screen sizes.

7.1.1 Optimization

The implemented application, even though it is functional, is still a bit choppy and slow. The main reason for not doing more code optimization was due to time constraints. Another factor was the lack of proper code documentation and examples of how certain optimization techniques should be implemented.

Several areas of code optimization were identified during the implementation and evaluation phase:

- Fewer database requests - The database was never written with performance as the highest priority. Because of this the available API is not optimized for efficiency. When starting the application large amounts of data have to be transferred from the database, this data is not sent to the clients in an efficient way.
- Minimize the DOM tree - As the number of elements in the DOM will have a large effect of how much memory the application consumes [28] it would be wise to keep the DOM-tree to a minimum. This was not really done as neither of us had previous experience working with the DOM and keeping elements to a minimum.
- A more efficient list - List items not shown should not be stored in RAM at all times. A native implementation for iOS using a list view would only store the objects currently showing in RAM: as the user scrolls the list objects outside the screen will be trashed and new ones will be loaded, see Figure 7.1 It is unclear if this is possible to implement in a web application.

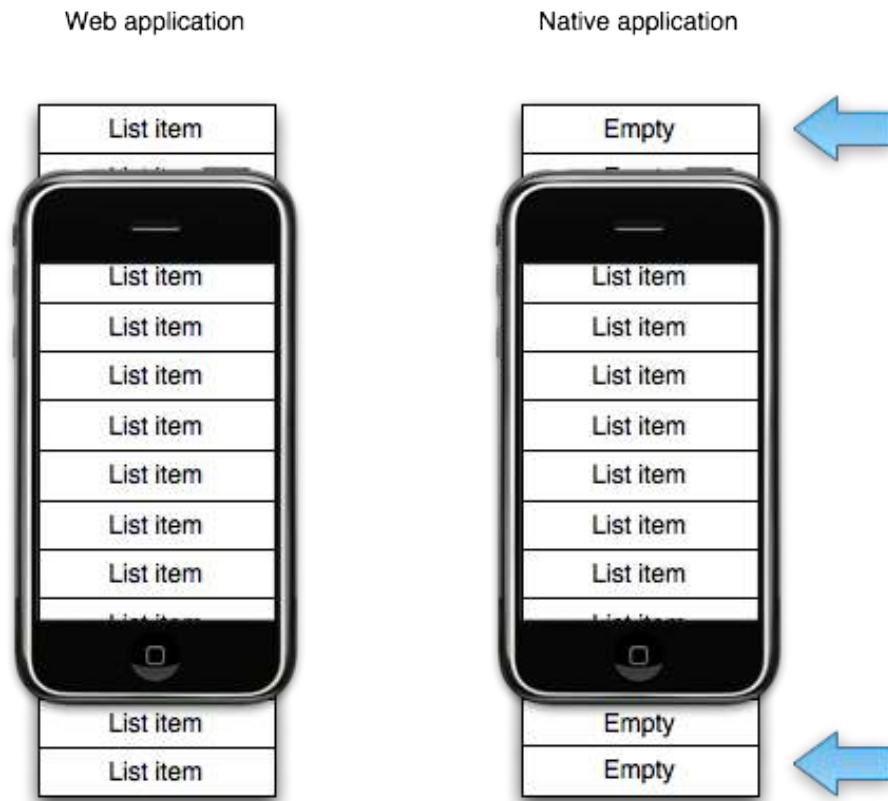


Figure 7.1: List items in a native implementation vs a web application and the differences of how memory is handled

- Polling - The autoupdate is just a simple javascript timer that automatically checks if the server contains any new events after a certain period of time. While this had a small negative impact on this project where the maximum number of connected clients never exceeded four, a real-world application should consider another way to communicate updates. One example could be to use a push solution where the client register for notifications and then receive data only when the server is updated. One ready-to-use push implementation is PubNub [24], this solution could not be used, due to the secrecy of the project.

7.1.2 Working with frameworks

Before starting any project where frameworks are considered, it is important to first evaluate the possible advantages and disadvantages. Sencha provides the possibility to use the same code on different platforms; no dual development is required in order to get a prototype that runs on both a PC and a mobile device. However, many drawbacks were also encountered, shown here in no particular order:

- Performance issues - The application does not look like a performance-demanding application, but in reality it shows that a big DOM-tree¹ will quickly ruin performance on mobile devices. While the application runs smoothly on a PC it suffers from slowdowns and animations that do not work as flawless as users of iOS devices have come to expect.
- Lack of documentation - The documentation for Sencha is not sufficient. Often links in the API would prove to be dead links. The lack of any books in the subject would also serve as a reminder of how fresh the field still is: reading tutorials online from different blogs is a good way to get started, but they rarely contain any knowledge about the foundation and core of Sencha Touch.
- Framework not finished - Even though release 1.0 of Sencha was released shortly before the start of this project and the updated version 1.1 during the development process, some parts of the framework were simply not implemented. One of these unimplemented methods was found in the `getSortState` method for Store, defined in the Sencha Touch API. This method did not return the expected result. When examining the Sencha Touch source code it turned out that some methods have not yet been implemented, rendering some of the Sencha Touch features useless. Also, there were no documentation of this whatsoever in the API documentation
- Bugs in the framework - Apart from having unimplemented methods, the Sencha Touch 1.1 framework also contained quite a few bugs. Many of the bugs encountered had to do with the data store², Sencha's preferred way of handling data. If a filter was applied to the Store and new updates were loaded from the server, resetting the filter would cause the updated data in the store to be discarded. This behavior was clearly not desired and several bug reports had been filed on the subject. That bug was possible to replicate, but other bugs were encountered that occurred seemingly at random: searching for entries in a data store would work 99 times out of 100, but every once in a while the method `findExact` would return -1 even though the element could manually be identified in the store.

An implemented performance optimization was to delete views when they were no longer visible. When a user pressed something that would trigger the screen to change content the previous screen should be deleted immediately in order to reduce memory usage. This however would not work if the user started to scroll before the screen had been completely exchanged for the new one. Should the user try to start scroll the new page before the transition was done this sometimes resulted in the previous screen not being destroyed but instead kept and rendered behind the active screen.

7.2 The evaluation

"The mood in here" and the SUS form have been applied on three operators. For that reason no valid statistical conclusions can be drawn. However this material combined with the qualitative data gathering method "the Think Aloud protocol" highlights what functionality in the application that are regarded as important and less important and for what reason. This material gives a more diverse view than if a form had been sent out to a large number

¹Document Object Model - A way to represent html and xml as objects.

²Sencha data store is the preferred way to work with the raw data in an Sencha application. Applications built using the MVC pattern will use the data store to organize and handle the raw data.

of operators. In this early phase where only a prototype has been developed, in depth interviews are more useful.

In this section the gathered data from the SUS form, "The mood in here" and "The Think Aloud protocol" is commented, discussed and evaluated.

7.2.1 The SUS form

Operator 2 scored 96 on the SUS form which is mapped to "Excellent" and operator 3 scored 80 which is mapped to "Good", however operator 1 scored 55 which is a weak "OK" but is still in the Acceptable range (see Figure 5.10). Operator 1 who scored lowest, rated the provided statements in an illogical way. For example, he marked "4" on both the question "I think I need help from a more experienced operator to manage this part of the application" and "I think this part of the solution is easy to use". These results are contradictive. He repeated this pattern for all but one sets of questions. His result must for that reason be taken in special consideration and possibly even rejected.

However, these results shows a generally good evaluation of the Communication and Collaboration application.

7.2.2 The mood in here

The operators seemed to have an overall positive attitude to all features of "The Communication and Collaboration" application. However, one of the three operators only gave the mobility aspect a rating of "2". This operator mentioned that he had a heavy alarm load in his work place, he simply could not see the added value for the mobility support. The other operators scored "3" and "4" which is within the content range ≥ 3 . The Alarm assistance and the Alarm library scored highest (2 operators marked "4", and one "3") which suggests that these features are important to further develop. The features "Add to shift report" and "Enter operator note" had the next highest mean which also suggests that these should be further developed.

The lack of top scores could depend on inabilities to map the features of the Communication and Collaboration application to their specific work environment and also to fully understand the information structure. It is possible that operators that are used to interact with social media applications that are built on feeds would have had easier to grasp this type of concept. The information design might feel complicated for inexperienced users, considering all the connected and related information that can be filtered in various ways.

Operator 1 marked "3" on all tasks which reinforces the suspicion that he did not make a serious attempt to evaluate the features. If his results would here be rejected the result would be much higher with for example top scores on Task 4 and 5 (Alarm assistance and Alarm library).

7.2.3 The Think Aloud protocol

Even though the documentation was time consuming, valuable aspects were highlighted after conducting the Think Aloud protocol.

An analysis and discussion is presented in the following section.

Make an operator note

Operator 1 stated that the user-generated feeds should be more highlighted than the automatically generated feeds. This is preferable in plants that on a regular basis experience

large amounts of alarms. However, the application should be adapted to both conditions. One possible solution could be to implement a filter that only displays the user-generated feeds.

The same operator wanted to be able to change the date on the feeds; he said that some alarms can take up to 24 hours to fix and it is important to file it as a recent alarm. One suggestion is to allow the operators to manually pick time and date when adding a operator note.

Operator 2 wanted to be able to add pictures because that would allow him to write less text and instead refer to the added pictures. The other two operators also expressed their unwillingness to document events in writing.

Operator 3 wanted the timestamps on the feeds to be more visible than the event thumbnail-pictures³. The feeds in the prototype is listed by time (the latest feed is listed first). This way of communicating time is possibly not enough. It can however also be the operator's, inability to grasp new concepts and tendency to relate to the existing solution. All the operators have 5 to 25 years experience using the current process automation software, which can affect their ability to imagine new work flows. This operator also suggested not to use any form of "Log in" functionality because they doesn't use the existing "Log in". He wanted it to be an active action to connect users to information, he suggested some sort of sign feature. This is possibly a good solution, because activities in a plant can result in process stops which can lead to large expenses and stress, a more selective sign feature might spare individuals from naming and shaming. On the other hand this action takes time and if information can't be connected with users many of the features of the prototype can't be used. Operator 3 also suggested that the name "Main feeds" should be renamed to "Log book", the metaphor of a log book is a better way to communicate the function of the feeds list and should simply be changed.

Choose information and send to "Shift report"

Operator 1 wanted this feature to be more integrated with the existing trend support. Information about different objects in trend formats is often used to discuss around. This needs, as mentioned in the "Make an operator note", to be further developed to be better integrated with the current existing software. A suggestion is to add the "Add to shift report" button to the trend support function.

Operator 2 got a bit confused about what really happened when he used this feature. Some sort of feedback can answer to this issue, which possibly also can enable the operation "Cancel" which should be offered to the user, this according to interface guidelines, see Chapter 4 section 4.1 "Reversal of actions" .

Operator 3 expressed a very positive feedback about the functionality.

Mobility for increased process awareness

Operator 1 thought that this feature indeed had an added value.

Operator 2 was initially a bit more sceptic because in his plant they experienced many alarms during a working shift which meant that they could rarely leave the control room, for that reason he saw no use for a mobility support. But after a while when he started to think about this feature he stated that they do need to go out and do some maintenance work on a regular basis and a mobile device that displays related information would indeed be useful. To address this the application should support guidance and related information for

³A thumbnail picture is a reduced size version of a picture [?]

these tasks. However, further studies needs to be done to fully understand what information is needed. The operators today must talk through a walkie talkie to communicate with the control room.

This operator also realized the value of being able to go to the toilet or go out and do some paper copies and at the same time being notified through a mobile phone about the priorities of incoming alarms.

Operator 3 wanted to be able to mute the alarms. A suggestion is to add an icon for muting the alarm in the feeds, or just the tap action can possibly have a muting effect.

Alarm assistance

Operators 1 and 3 had nothing to remark on. Operator 2 thought that the solutions should be able to be sorted by date. The operators are influenced by the existing system where date is important. The concept of sorting by relevance is possibly more efficient in this context than sorting by date. The concept of rating solutions hopefully engages the operators to be more active.

Operator 2 also suggested that the alarms should have highlighted default solutions. He also wanted the Alarm assistance functionality to be better integrated in the main process window of the existing software.

Operator 3 suggested that there should be some indication of how prioritized a specific alarm is.

He also wanted to know how prioritized the alarms are. A possible solution could be to use different colors, this is also in line with the concept of the next generation software, which uses the color red for high prioritized alarms and orange for less prioritized alarms.

Alarm library

Operator 1 and 2 thought this was a good feature for novice users, operator 3 was however more sceptic.

7.3 Limitations

Because of limited experience with implementation in the Sencha Touch framework and by the constrained time frame the Communication and Collaboration application could be improved in terms of functionality and design.

The implemented functionality of automatically generated feeds was appreciated but possibly sometimes hard to understand. A simple and efficient tool is of utmost importance to support participation and contribution with this application. Several operators had problems with evaluating and understanding the concept of a not fully functional application. It might be difficult for operators who have worked with the same system for up to 25 years to take on other perspectives. A solution that differs too much from the existing solution is possibly hard to grasp and the concept of the old system might interfere.

The integration of the application in the current software has been approached in terms of design by using the same color scheme and the same kind of retro and simple icons that are used in the current software. However the application should be better integrated in the process control system, it is now evident that it is an additional layer.

The evaluation of the Communication and Collaboration tool was only tested on three operators with similar background and experience; no statistical conclusions can be drawn. The result must only be considered as a indication of users perception and opinions regarding

this application. However as this was a prototype aiming to communicate functionality it was more valuable with in-depth interviews as a first approximation. A more formal and statistical valid study is possible more suitable in a more high fidelity or a fully implemented application.

7.4 Discussion

The kind of information structure that has been used in this prototype with inspiration from social media communities is possibly also applicable in other domains. Communication and collaboration are relevant aspects in many areas. The usage of this information design is however slowly starting to emerge, for example the Get Satisfaction application has successfully been implemented in the Customer support domain. Other environments could also possibly benefit. However, a possible concern could be the "privacy" issue. This has been solved in the Facebook community by enabling change of privacy settings. In the company domain this must possibly be more visually highlighted, easy to modify and easy to understand.

The strength of this information architecture is how information is filtered with both the filter function and also for example when browsing to a profile page of a process object, all information that is relevant for that object is displayed, this is a intuitive and powerful way of filtering information. All displayed information is connected and easy to access which is important for this domain. In some plants operators experience a lot of alarms, the time aspect is therefore crucial. A tool that is hard to grasp will most likely not be used.

When studying user behavior in communities it is interesting to identify what motivates people to contribute and participate. A community in which no user is active is useless. In communities like Stack Overflow people show a behavior of helping each other in a very altruistic manner by providing support information etc. It would be interesting to further test if the operators would contribute with alarm solutions in the alarm assistance feature and if they would help their coworkers when they for example were unable to solve an alarm.

But perhaps most of all; would the information architecture in the Communication and Collaboration application enable the operators to be more active, engaged and efficient. Would the operators experience a feeling of being able to influence in their own work situation.

7.5 Future work

The alarm assistance feature was very appreciated by the operators. However one possible risk is that not enough solutions are added because of too few operators. But if this support were implemented in similar plants, allowing users from several plants to access solutions and support from each other; the desired effects of large amounts of data, contributed by several users, could then possibly be achieved.

When it comes to functionality and integration, the interface of the current software is connected to the Communication and Collaboration application when for example an alarm is triggered or when an operator wants to enter a note on an object. However this design work can be developed to furthermore reinforce the experience of a fully integrated application. A suggestion is to integrate the trending feature, this information is relevant for the operators to share and discuss around.

The learning curve for managing the current process control software is too steep which only leaves room for experienced operators to handle many plant duties. This leaves the

novice users feeling useless and unable to influence their work situation which increases stress and depression, this can in the long run create a competence vacuum. With this as background the probability that the experienced operators actually will contribute with information is possible large, it is also in the experienced operators interest not having to take additional responsibility for the novice user. If a default solution could be added as a first suggestion this would increase the probability that the operators would use and eventually expand the knowledge bank.

Chapter 8

Acknowledgements

We would like to thank the following people:

Susanne Timsjö for her always supporting and positive spirit. You showed great patience in explaining the complexity of the existing software and users needs, and you gave us valuable feedback throughout our thesis work.

Lars-Erik Janlert for helping us structure and giving us valuable feedback on the report.

Saad Bin Azhar and **Nils Johansson** for supporting us with a database and for your time supporting us in how to integrate it in our implementation.

Magnus Larsson for making us feel welcomed and appreciated at ABB and in Västerås.

Henning Hagman for helping us with latex syntax and highlight interesting and relevant aspects to include in the report.

Our families and friends for being awesome.

References

- [1] Apple. Appstore passed 350 000 apps, October 2010. http://www.macworld.com/article/154980/2010/10/jobs_transcript.html (visited 2011-08-04).
- [2] Nurul Zakiah binti Ayob, Ab. Razak Che Hussin, and Halina Mohamed Dahlan. Three layers design guideline for mobile application. *Information Management and Engineering, International Conference on*, 0:427–431, 2009.
- [3] J. Blake. *Multitouch on Windows and NUI Development with WPF and Silverlight*. Manning, 2010.
- [4] John Brooke. Sus a quick and dirty usability scale, May 2011. [http://hell.meiert.org/core/pdf/sus.pdf/?/](http://hell.meiert.org/core/pdf/sus.pdf?/) (visited 2011-08-03).
- [5] J. Cartman and R. Ting. *Strategic Mobile Design Creating Engaging Experiences*. New Riders, 2009.
- [6] Cloudebug. Remote debugging for titanium appcelerator, August 2011. <http://www.cloudebug.com/> (visited 2011-08-04).
- [7] W3C Consortium. The world wide web consortium, February 2011. <http://www.w3.org/> (visited 2011-08-03).
- [8] Nicole B Ellison, Charles Steinfield, and Cliff Lampe. The benefits of facebook friends social capital and college students use of online social network sites. *Software Practice and Experience*, 29(5):457–478, 1999.
- [9] Roger Entner. Smartphones to overtake feature phones in u.s. by 2011, May 2011. <http://blog.nielsen.com/nielsenwire/consumer/smartphones-to-overtake-feature-phones-in-u-s-by-2011/> (visited 2011-08-03).
- [10] J. Gong and P. Tarasewich. Guidelines for handhelds mobile device interface design. *Proceedings of DSI 2004 Annual Meeting*, pages 3751–3756, 2004.
- [11] L. Gorlenko and R. Merrick. No wires attached: Usability challenges in the connected mobile world. *IBM Systems Journal*, 42(4):639 –651, 2003.
- [12] John Gruber. New iphone developer agreement bans the use of adobes flash-to-iphone compiler, 2010. http://daringfireball.net/2010/04/iphone_agreement_bans_flash_compiler (visited 2011-08-04).

- [13] D. Jonassen, M. Tessmer, and W. Hannum. *Task analysis methods for instructional design*. Lawrence Erlbaum Associates Inc, 2011.
- [14] J. Jose, T. Granollers, T. Granollers, and P. Latorre. *New trends on human-computer interaction: research, development, new tools*. Springer Verlag, 2009.
- [15] W. Karwowski. *Handbook on standards and guidelines in ergonomics and human factors*. Human Factors and ergonomics.
- [16] L. Keeble and B. Loader. *Community informatics - Shaping computer-mediated social relationships*. Routledge, 2001.
- [17] Joseph Lampel and Ajay Bhalla. The role of status seeking in online communities giving the gift of experience. *Journal of computer-mediated communication*, 12(2):1–10, 2001.
- [18] Hsiu-Fen Lin and Gwo-Guang Lee. Determinants of success for online communities: an empirical study. *Behavior and Information Technology Journal*, 25(6):479–488, 2006.
- [19] PC magazine. Definition of a smartphone. <http://www.pcmag.com> (visited 2011-08-03).
- [20] J. Nielsen and R.L. Mack. *Usability inspection methods*. Wiley, 1994.
- [21] I. Poupyrev, S. Maruyama, and J Rekimoto. Ambient touch: Designing tactile interfaces for handheld devices. *Proceedings of DSI 2004 Annual Meeting*, pages 3751–3756, 2004.
- [22] ABB Power and productivity for a better world. <http://www.abb.se/cawp/seitp202/c8ad1a3458dd675dc12574b0003e408f.aspx> (visited 2011-08-03).
- [23] Jenny Preece. Sociability and usability in online communities: Determining and measuring success. *Behavior and Information Technology Journal*, pages 1–15, 2001.
- [24] PubNub. Pubnub real-time data push, August 2011. <http://www.pubnub.com> (visited 2011-09-15).
- [25] J. Rabin. W3c mobile web best practices, 2008. <http://www.w3.org/TR/mobile-bp/> (visited 2011-08-03).
- [26] Howard Rheingold. *A slice of life in My Virtual Community*. Cambridge: MA: MIT Press, 1994.
- [27] D. Saffer. *Designing gestural interfaces*. Oreilly Media, 2009.
- [28] Sencha. Optimizing memory usage, August 2011. <http://www.sencha.com/blog/sencha-touch-optimizing-memory-usage/> (visited 2011-09-15).
- [29] StatCounter. Browser statistics, August 2011. <http://gs.statcounter.com/> (visited 2011-08-04).
- [30] The Sunday Times. Facebook ousts google in us popularity, 2010. <http://technology.timesonline.co.uk> (visited 2011-08-03).

-
- [31] TiPb. Apple relaxes cross-compiler restrictions, publishes app review guidelines, September 2010. <http://www.tipb.com/2010/09/09/apple-relaxes-crosscompiler-restrictions-publishes-app-review-guidelines/> (visited 2011-08-04).
- [32] T. Tullis and B. Albert. *Measuring the user experience: collecting, analyzing and presenting usability metrics*. Morgan Kaufmann, 2008.
- [33] w3schools. Javascript unshift() method, an example that all functionality in javascript is not supported everywhere, September 2010. http://www.w3schools.com/jsref/jsref_unshift.asp (visited 2011-08-04).
- [34] w3schools. Browser statistics, August 2011. http://www.w3schools.com/browsers/browsers_stats.asp (visited 2011-08-04).
- [35] B. Wellman and C.A. Haythornthwaite. *The Internet in everyday life*. The Information Age series. Blackwell Pub., 2002.
- [36] E Wenger. *Communities of practice: Learning, meaning, and identity*. Cambridge University press, 1998.

Appendix A

Screenshots

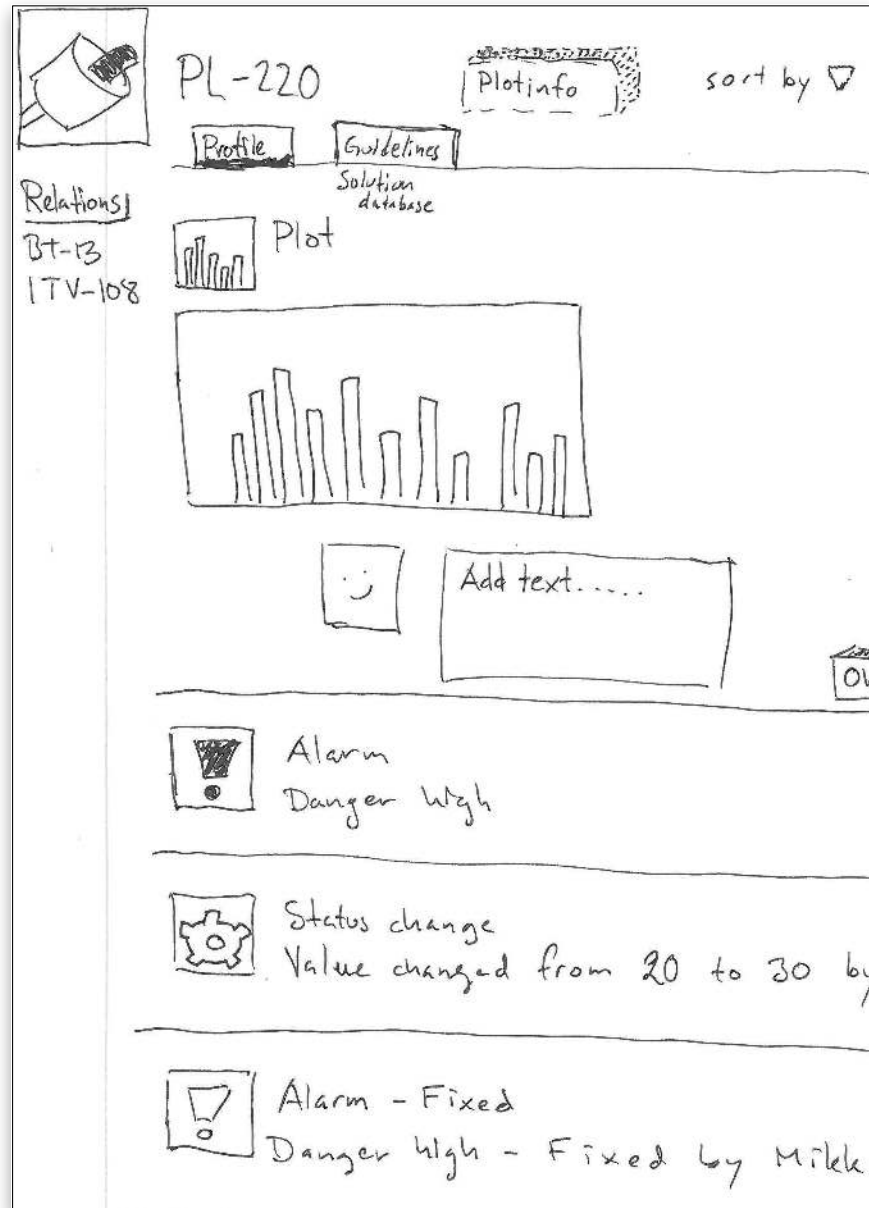


Figure A.1: Low fidelity paper prototype. Profile page for process object PL-220A, the list consists of feeds that belongs to categories like alarms and changes.

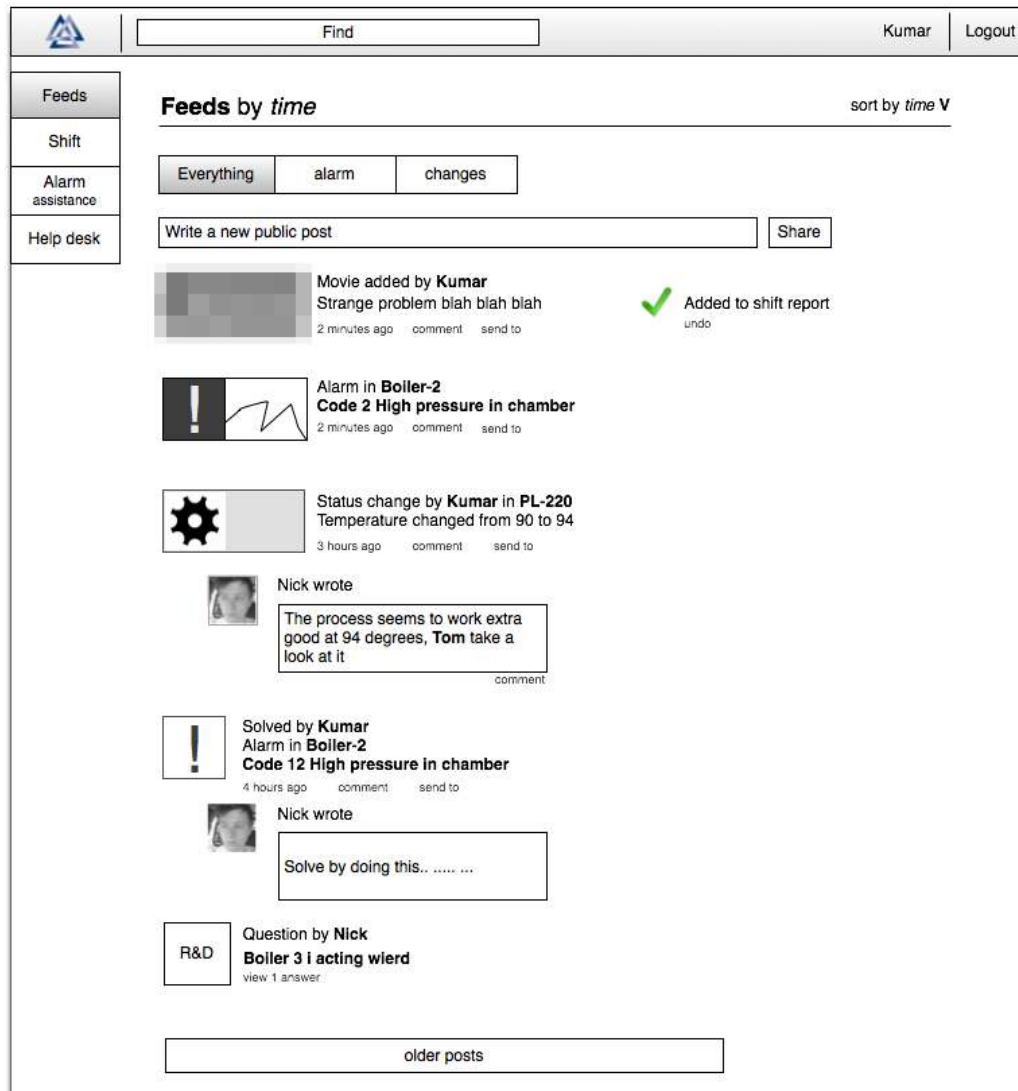


Figure A.2: Low fidelity prototypes produced in Omnigraffle. The graphical components are clickable, this makes it easier to communicate the functionality of the Communication and Collaboration application

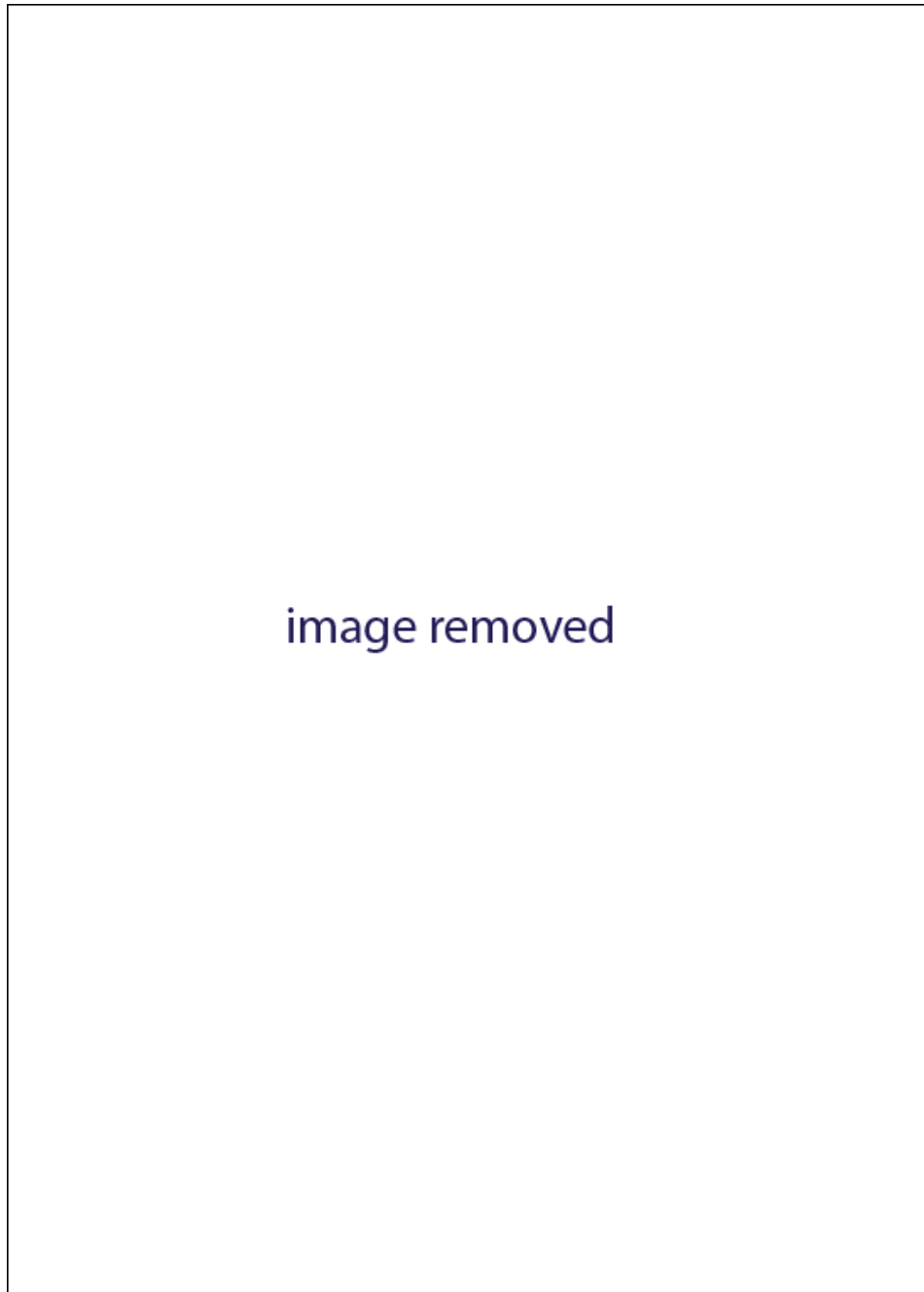


Figure A.3: A medium fidelity prototype that is only focusing on layout design



Figure A.4: The popular community Facebook, the picture shows how its list of feeds looks like

The screenshot shows a Stack Overflow page. At the top, the Stack Overflow logo is on the left, and navigation buttons for 'Questions', 'Tags', and 'Users' are on the right. The main heading of the question is 'Need to know where StartAction configuration'. Below the heading is a banner for 'CAREERS 2.0 by stackoverflow'. The question text reads: 'Hi In Silverlight project, I could right click and choose Properties. In th of-browser application to debug the application in OOB mode. Can anybody inform where is this configuration stored? I've tried to OutOfBrowserSettings.xml, and AppManifest.xml, but I don't see th Any help will be appreciated. Thanks.' There are tags for 'silverlight' and 'visual-studio-2010'. Below the question, it says '1 Answer'. The answer text is 'The changes are made to the .csproj.user file.' followed by a code block:


```
<StartAction>OfflineApplication</StartAction>
.
.
.
<OutOfBrowserProjectToDebug>YourAppProject</OutOfB
```

Figure A.5: Stack Overflow is a website for questions and answers about programming. The users can rate questions and/or answers +1 point if they think it is good or rate it down -1 point if they think it is bad



Figure A.6: The GetSatisfaction website provides a framework for applications like Facebook and Twitter to support their users when they have different technical or usability problems.

Appendix B

SUS questionnaire

- 1. I think that I would like to use this system frequently
- 2. I found the system unnecessarily complex
- 3. I thought the system was easy to use
- 4. I think that I would need the support of a technical person to be able to use this system
- 5. I found the various functions in this system were well integrated
- 6. I thought there was too much inconsistency in this system
- 7. I would imagine that most people would learn to use this system very quickly
- 8. I found the system very cumbersome to use
- 9. I felt very confident using the system
- 10. I needed to learn a lot of things before I could get going with this system

Appendix C

User evaluation scenarios

You will here get the opportunity to try the next generation of the process automation software that supports Communication and collaboration. The test leaders will present the features and how to interact with this application. This concept is based on identified user needs.

1. Enter a operator note on a process object

The user has recently performed an adjustment of the process object and wants to communicate this information to the following shift.

- Use the context menu to write a note on an process object, where an alarm has recently been triggered.
- Note how the process feed were updated

2. Communication

Some information might be important to communicate to the following shift.

- Choose relevant information and send it to the Shift report
- Note how the Shift report has been updated

3. Mobility

Use the iPad to perform the last two scenarios again.

4. Alarm assistance

An alarm has been triggered.

- Use the alarm assistance for guidance

5. Alarm assistance continued

The provided solution did not match your expectations.

- Enter a new solution.

Appendix D

Think aloud protocol

This section presents the results from the Think Aloud protocol. The findings are sorted by user. The user study was conducted and documented in Swedish, but the collected notes were later translated to English. The operators sometimes had difficulties to communicate their thoughts and concerns, in these cases a discussion was held where the operators could point at the screen etc., this discussion could often not be quoted, instead a short summary of the core of the discussion was compiled. Quotes from operators are written in *italics*.

Quotes from operator 1

Task: Make an operator note on a process object

- *”Good with pictures along with the notes!”*
- *”You should be able to manually set the timestamp on a note when you post it.”*
- *”The notes should not be so user centered, but instead shift and time centered, they would not care that much of who wrote the note, but instead when it was written and by what shift.”*
- *”In our plant there is a triggered alarm and event every minute, the number of feeds would make it hard to identify the more important user generated notes, this should be more flexible, I would like to be able to turn off the generated feeds.”*
- *”Our system have seven different alarm priorities, I would like to know what kinds of alarms are important and which are not.”*
- *”The automatically generated alarm and events feeds should be possible to make inactive, we have too much alarm and events in our plant, we are only interested in feeds, like a change on an object or a note made on an object.”*
- *”I would like to be able to change the date stamp on the feeds, an alarm can be hard to solve, sometimes it takes 24 hours, in this design that alarm has been pushed down in the feeds list.”*
- The operator wants to be able to add notes in the feed that are not bound to a specific object, i.e. wants to be able to use the feeds as digital ”post its”.

- Regarding user-generated content and automatically generated content, *"...in the main feed you should really rank the user-created content higher so that auto-created content doesn't flood the main feeds.."*
- Likes the idea of showing operator changes in the feed.

Task: Choose information to send to Shift report

- The operator was initially not very fond of the idea, he thought it seemed unnecessary to manually mark feeds as important shift change objects. But changed his mind to a more positive view of the shift change button during the later part of the evaluation.
- *"Could this be more integrated with the trend support? History trend statistics would be nice"*

Task: Mobility for increased process awareness

- Liked the idea of having an iPad to show attached pictures to related notes.

Task: Alarm assistance

- Really liked the idea.

Task: Alarm library

- *"This is a good feature for novice users."*

Comments from operator 2

Task: Make an operator note on a process object

- *"Very good if it would be possible to add pictures to every note."*
- *"The pictures of the operators and the process objects should be clickable."*

Task: Choose information to send to Shift report

- The operator likes the idea of selecting certain feeds. Uses paper and pencil every day right now to take notes during the day that could be relevant for the coming shift change.
- The operator got a bit confused about where the feed was sent, some sort of verification of that it was successfully sent. *"Have I copied this feed to the shift report?"*

Task: Mobility for increased process awareness

- Initially the operator had a quite negative view of a mobility part, very likely because the fact that they could never leave their chairs during their work. As they got so many alarms they were always required to be at their desk.
- If however the operator workstation could be on the iPad (to be able to control the process) then he would have liked it. Could give the people who made manually changes to the process objects out in the plant valuable info instead of having to talk to the operators while changing parameters.

- And if the operator workstation could be in the iPad he would have no problem with adding also the collaboration/communication part.
- However being able to go to the toilet with an iPhone in his pocket and check incoming alarms and mute them would have been much appreciated.

Task: Alarm assistance

- The operator wanted the ability to sort the solutions by the date they were added.
- He liked the idea of having text and pictures as description for certain alarms.
- He did not see any need for video.
- *"It would be good to have a default solution for the Alarm in addition to the user-generated answers."*
- *"I would like to have this more integrated in the other features as the faceplate bar etc. When I press an alarm button, I would like to have the alarm assistance information in the context where I am operating; it should not be visualized on a different screen. Could the features be integrated in the process objects menu, or on the faceplate."*
- *"It should be easy to mute an alarm, now I have to localize the mouse arrow and move it to the small space in the top corner."*

Task: Alarm library

- *"I like this."*

Comments from operator 3**Task: Make an operator note on a process object**

- *"Time is more important to show rather than the process operator picture."*
- *"Instead of using logins every note could be signed with the name of the user who wrote the note."*
- *"The name for "main feeds" could be "logbook"."*
- *"Profile page is confusing, is it not just a way to filter information."*

Task: Choose information to send to Shift report

- *"I like this idea."*

Task: Mobility for increased process awareness

- Initially had a quite negative view of a mobility part, very likely because the fact that they could never leave their chairs during the work. As they got so many alarms they were always required to be at their desk.
- *"I would like to be able to mute alarms ¹ on the mobile phone."*

¹Mute is referring to the action of silencing an audio signal

Task: Alarm assistance

- *"This is a very good idea."*
- The operator did not see the need for video.

Task: Alarm library

- *"I do not see how it should be used."*