

Design og udvikling af Android site specific browser

Diplom -IT Eksamensprojekt 2011/2012

Forfatter:

Shaymaa M. Yassen s082710

Vejledere:

DTU – Mads Nyborg

NetDesign a/s – Nino Martinez

Danmarks Tekniske Universitet

Informatik og Matematisk Modellering

Bygning 321, DK-2800 Kongens Lyngby, Danmark

Telefon +45 4525 3351, Fax +45 4588 2673

reception@imm.dtu.dk

www.imm.dtu.dk

IMM-B.Eng-2010-92

Forord

Denne rapport er udarbejdet af Shaymaa M. Yassen i forbindelse med afgangsprojektet på linjen diplom IT på Danmarks Tekniske Universitet (DTU). Rapporten dokumenterer det arbejde, som er blevet gennemført i forbindelse med eksamensprojektet. Projektet har forløbet i perioden 05-09-2011 til 01-02-2012, som svarer til 18 uger og giver 20 ECTS point. Mine vejledere på projektet har været Mads Nyborg fra Institut for Informatik og Matematisk Modellering IMM på DTU samt Nino Martinez fra virksomheden NetDesign a/s.

Abstract

A lot of enterprises have seen the possibilities of mobile applications, since they provide previously unseen potentials in terms of improvements in productivity, increasing mobility, and reducing costs. By the means of mobile applications, information and tasks are provided to the coworker anytime at any location. This ensures that the coworkers are no longer dependent on computers for resources such as the internet. There are however certain challenges in having mobile applications, one of them are the smaller screen size of the phone.

In this report I will review the design and development of the Wallboard mobile application, which is a mobile version of NetDesign's wallboard desktop system. In the report I will start with an introduction to the project where subsequently I will discuss the different technologies in the development of mobile applications. On the basis of the discussion I will choose a technology to use with the Wallboard mobile application. The report will also describe the development phases that I have worked through to design and develop the Wallboard mobile application including analysis, design, implementation, and testing. I will conclude the paper with a comparison of other development tools, which can be used for the development of mobile applications.

Resumé

Mange virksomheder har åbnet øjnene op for mulighederne indenfor mobilapplikationer, da de giver usete muligheder for at forbedre produktiviteten, øge mobiliteten og reducere omkostningerne. Ved hjælp af mobilapplikationer stilles informationerne og opgaverne til rådighed for medarbejderne når og hvor som helst, hvilket betyder at medarbejder ikke er afhængig af pc'er længere. Der er visse udfordringer i udvikling af mobilapplikationer, som mindre skærmstørrelse.

I denne opgave gennemgår jeg design og udviklingen af Wallboard mobilapplikation, som er en mobilversion af virksomheden NetDesign's wallboard desktop system. I rapporten starter jeg med at give et introduktion til projektet, dernæst diskuteres de forskellige teknologier indenfor udviklingen af mobil applikationer, på baggrund af denne diskussion vælges det teknologi, som jeg bruger til at udvikle wallboard mobil applikationen. Rapporten beskriver også udviklingsfaserne, jeg har været igennem for at designe og udvikle wallboard mobilapplikationen herunder, analyse, design, implementering og test.

Indholdsfortegnelse

| | |
|---|----|
| 1. Introduktion..... | 1 |
| 1.1 Indledning..... | 1 |
| 1.2 Problemformulering | 2 |
| 1.3 NetDesign | 2 |
| 1.4 Wallboard | 3 |
| 1.4.1 Designer menu..... | 6 |
| 1.4.2 Available wallboards menu | 7 |
| 1.4.3 Setup menu | 8 |
| 1.5 Wallboard desktop Overview | 10 |
| 1.6 Motivation og baggrund | 11 |
| 2. Projektbeskrivelse | 12 |
| 2.1 Udviklingsmetoder | 12 |
| 2.2 Tidsplan..... | 13 |
| 2.3 Risikovurdering | 15 |
| 3. Teknologier | 16 |
| 3.1 Smartphone platforme | 16 |
| 3.2 iPhone..... | 18 |
| 3.3 Android | 18 |
| 3.3.1 Android arkitektur | 19 |
| 3.4 Webapplikation vs. native applikation | 21 |
| 3.4.1 Native applikationer | 21 |
| 3.4.2 Mobil web applikation | 22 |
| 3.4.3 M-power mobile web applikationer | 23 |
| 3.4.4 PhoneGap og Titanium | 25 |
| 3.4.5 Site specifik browser..... | 25 |
| 3.4.6 Opsummering på udviklingsmuligheder..... | 27 |
| 3.5 Konklusion på valg af teknologi | 30 |
| 4. Prototypeudvikling | 31 |
| 4.1 Analyse | 31 |
| 4.1.1 Inspiration fra FDB..... | 31 |
| 4.1.2 Personas | 32 |

| | | |
|---------|---|----|
| 4.1.3 | Brugerbehov | 35 |
| 4.1.4 | Use case diagram..... | 36 |
| 4.1.5 | Scenarier | 36 |
| 4.1.6 | Kravspecifikation | 40 |
| 4.2 | Design | 42 |
| 4.2.1 | Design udfordring | 42 |
| 4.2.2 | Brugervenlighed i wallboard mobilapplikation | 42 |
| 4.2.3 | Konceptuelt design | 43 |
| 4.2.4 | Overview Wallboard System med mobil | 49 |
| 4.3 | Implementering | 50 |
| 4.3.1 | Webviewer..... | 50 |
| 4.3.2 | jQuery | 55 |
| 4.3.3 | Wallboard skabeloner og CSS..... | 58 |
| 4.3.3.1 | Cisco Support Wallboard | 58 |
| 4.3.4 | Fremtidige funktioner..... | 61 |
| 4.4 | Test og evaluering af Wallboard mobil applikationen | 63 |
| 4.4.1 | Test af Wallboard mobilapplikationen | 63 |
| 4.4.2 | Brugervenligheds test..... | 64 |
| 4.4.3 | Heuristisk evaluering | 65 |
| 4.4.4 | Gestalts love | 66 |
| 5. | Konklusion | 68 |
| 6. | Perspektivering..... | 70 |
| 7. | Litteraturliste..... | 71 |
| 8. | Bilag | 72 |
| | Bilag 1 Persona spørgeskema | 72 |
| | Bilag 2 Interview spørgsmål..... | 72 |
| | Bilag 3 Kildekode | 73 |
| | Webviewer applikation | 73 |
| | Wallboard mobil skabeloner | 75 |
| | jQuery wallboard menu..... | 80 |
| | Bilag 4 main.xml | 82 |
| | Bilag 5 Webviewer BT | 82 |
| | Bilag 6 Tablet | 83 |

| | |
|--|----|
| Test på tablet (emulator)..... | 83 |
| Tablet will Grow..... | 83 |
| Bilag 7 Test på iPhone..... | 84 |
| Bilag 8 Brugervenlighedstest | 85 |
| Brugervenlighedstest spørgsmål: | 85 |
| Statistiske resultater:..... | 85 |
| Bilag 9 Heuristics Evaluation | 87 |
| Bilag 10 Installationsvejledning | 89 |
| Bilag 11 CD-ROM..... | 90 |

1. Introduktion

I dette kapitel har jeg beskrevet hvad mit projekt går ud på i detaljer, derudover har jeg givet en detaljeret beskrivelse wallboard desktop systemet. Til sidst skriver jeg hvorfor jeg har valgt dette projekt og hvad der har været min motivation.

1.1 Indledning

Dette eksamensprojekt omhandler design og udvikling af en mobilapplikation til NetDesign's wallboard system, som bruges til overvågning af kaldssituationen og serviceniveauet i kontaktcentre¹.

I takt med at antallet af smartphone ejere samt mobilapplikationer vokser, er brugen af internettet på mobilen også steget markant. Der er allerede forudsigelser om, at der er flere der vil tilgå internettet via mobile enheder frem for en klassisk computer i 2014. Det betyder at mobilen kommer til at spille en endnu større rolle i fremtiden. Derfor kan man ikke længere ignorere denne udvikling ude i erhvervslivet. Flere og flere virksomheder er begyndt at udvikle mobilapplikationer til deres medarbejdere, ikke kun fordi det er den nye trend, men også fordi det indebærer mange fordele. Brugen af mobilen er steget og behovet for at være mere fleksibel og bevægelig er blevet større.

Rapporten er opbygget i kronologiske kapitler med de faser jeg har været igennem i projektforsløbet. I denne rapport vil jeg starte med at give en grundig beskrivelse af den oprindelige wallboard desktop applikation. Dernæst vil der fremgå en undersøgelse og gennemgang af de forskellige muligheder samt teknologier der er indenfor udvikling af mobilapplikationer. På baggrund af de forundersøgelser jeg har lavet, kunne jeg træffe nogle beslutninger sammen med min vejleder fra NetDesign om, hvordan applikationen skal udvikles og hvilke teknologier der skal bruges og m.m.

For at udvikle en applikation, som brugerne får glæde og nytte af har jeg inddraget brugerne i udviklingsprocessen. Dette gjorde jeg ved at interviewe nogle forskellige brugere fra NetDesign, om hvad deres forventninger, ønsker og behov til denne applikation er, samt inddrage dem i analyse, design og test fasen, hvor de får lov til at se prototyper og give feedback på det.

Rapporten beskriver hvordan analyse, design, implementering og test faserne har forløbet og hvilke resultater det har givet. Da brugervenlighed spiller en essentiel rolle i udvikling af mobile applikationer, har jeg udarbejdet brugervenlighedstest.

¹ Har til formål at servicere kunder: Kundeservice, kundesupport, marketing etc.

1.2 Problemformulering

Projektet vil bestå af to dele, en teknisk og en analytisk del:

1. Første del af projektet går ud på, at udvikle en Android webviewer applikation. Webvieweren skal kunne åbne, en hvilken som helst webapplikation, den konfigureres til at åbne. Man skal ikke kunne se at det er en webapplikation, som vieweren åbner, det skal se ud som om det er en almindelig mobil applikation og ikke en webside.

Jeg vil konfigurere applikationen til at åbne Netdesign's wallboard webløsning, som bruges til overvågning af kaldssituationen i et kontaktcenter. Derudover vil jeg designe forskellige mockups/skærm billeder, der viser/beskriver, hvordan wallboard data, som vises på en storskærm, kan præsenteres på en Android mobil skærm, på den bedst mulige måde. Derfra vælger jeg 2 eksempler og designer dem via css stylesheet.

2. Anden del af projektet går ud på at vise, hvordan designprocessen for en mobilapplikation kunne se ud. Jeg vil udføre usability test og analytisk evaluering af mobil applikationen, for at bedømme brugerens oplevelse (user experience). Her kan der evt. anvendes heuristisk evaluering og analyse af gestalts love.

1.3 NetDesign

NetDesign a/s er et af de førende virksomheder indenfor integration af sikre netværks og ip-kommunikationsløsninger. I 1988 blev NetDesign blev etableret i Farum, dengang havde virksomheden kun 5 ansatte, i dag er der 470 medarbejder. I 2004 blev NetDesign købt af TDC. De er blevet kåret til Danmarks største rådgiver og integrator af sikre netværks- og IP-kommunikationsløsninger. De har to kontorer, hovedkontoret ligger i Farum og regionskontor som ligger i Skejby (Jylland). I NetDesign arbejder de med løsninger indenfor unified communication (UC), netværk, sikkerhed, service og outsourcing. Der bliver bla. arbejdet med udvikling af kommunikationsteknologier lige fra ip-telefoni og virtuelle møder over til chat og kontaktcentre. NetDesign arbejder også med at tilpasse netværk og datacenter til virksomheder og sørger for alt netværk fungerer. En af NetDesign's spidskompetencer er sikkerhed, hvor de arbejder med at rådgive virksomheder, om hvilke sikkerheds produkter, der passer dem.

1.4 Wallboard

Netdesign's Wallboard er et webbaseret system. Det er et fleksibelt arbejdsredskab, som bruges til at vise aktuelle kalds data, historik og support sager på en computerskærm eller på en storskærm som placeres i kontorer og afdelinger. Systemet bliver primært anvendt i et "callcenter", hvor medarbejdere, ledere og supervisorer kan holde øje med kaldsituationen og serviceniveauet af kundebetjeningen i deres afdeling.

NetDesign's wallboard system har indtil videre over 15 kunder blandt dem er GF, TreFor, Molslinien og Miljøministeriet. Der er ikke besluttet tal på hvor mange brugere wallboard systemet har i alt, idet det er svært at holde styr på hvor mange brugere af wallboard kunderne har. I NetDesign er der ca. 30 brugere som anvender wallboard systemet i deres hverdags arbejde. Det er bla. teknikere, supportere og ledere.

Wallboard giver hele tiden et opdateret oversigtsbillede over kalds aktiviteten på arbejdspladsen. Systemet er også konfigurerbar, således at kunderne selv kan bestemme, hvilke data som skal vises og hvordan det skal se ud. Wallboard understøtter også flere køer² og grupper³ samtidig.

Netdesign's wallboard kan f.eks. i et oversigtsbillede oplyse detaljer omkring:

- **Antal ledige medarbejdere**
- **Antal kunder som venter i kø**
- **Antallet af mistede kald**
- **Ledige medarbejdere**

Wallboard er designet til blandt andet at give medarbejdere, supervisorer og ledere et hurtigt overblik over f.eks. "kalds" situationen. Medarbejderne og ledernes viden omkring kald situationen giver dem mulighed for at yde en bedre kundeservice, ved at planlægge deres tid bedst muligt mht. til pauser, prioritering af kald og m.m. Dette resulterer også at kald køerne reduceres, især i de belastede perioder, hvilket også hjælper virksomheden med at yde bedre service til deres kunder.

Wallboard systemet er designet på den måde, at den understøtter overvågning af flere køer samtidig, således at den kan dække virksomhedernes særlige behov. Systemet indeholder en brugergrænseflade, som designer wallboard setup. Grafisk er systemet designet til at køre et wallboard display på en skærm, som typisk vil hænge i det rum, hvor medarbejderne og lederne befinder sig, men samtidig kan det også ses

² I et kald center er der forskellige køer: telefon, e-mail, fax. Køer kan også være kunder som venter i forskellige køer baseret på f.eks. prioritet.

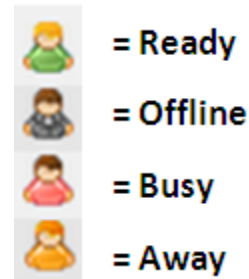
³ I et kald center kan der være forskellige grupper f.eks. kundeservice, marketing.

fra medarbejdernes computerskærm. Designet på skærmdisplayet kan tilpasses så den matcher med behovene hos de forskellige kunder. Herunder ses eksempler på wallboard skærm.

Oversigt over agenter:



Piktogram over agent status



Figur 1 Agent status wallboard

Displayet giver et hurtig statusbillede af agenterne. Piktogrammet til højre illustrere hvad status ikonerne symbolisere.

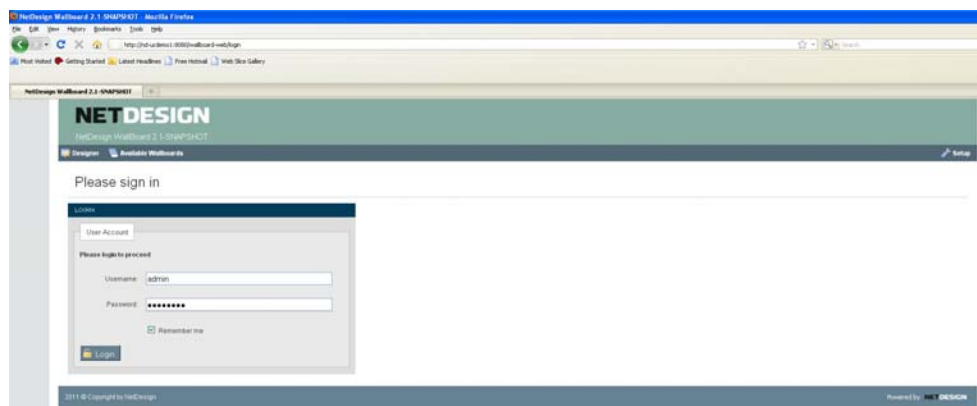
Oversigt over flere køer:

| Cisco | Asterisk | Avaya | Aastra |
|------------------|------------------|------------------|------------------|
| Logged in agents | Logged in agents | Logged in agents | Logged in agents |
| 8 | 8 | 8 | 8 |
| Service level | Service level | Service level | Service level |
| 33 | 33 | 33 | 33 |
| Calls handled | Calls handled | Calls handled | Calls handled |
| 100% | 100% | 100% | 100% |
| Total calls | Total calls | Total calls | Total calls |
| 19 | 19 | 19 | 19 |

Figur 2 Multiple Queue wallboard

Som sagt så understøtter wallboard flere køer samtidig. Ovenstående eksempel giver et overblik over de agenter, der er logget ind i hver "afdeling/gruppe", det nuværende serviceniveau, og procentdelen af opkald som er blevet håndteret i løbet af dagen og antallet af kald. Dette display kan være meget relevant for en leder, idet den vil give ham/hende et hurtigt overblik over "kald" situationen i sin afdeling. Det er selvfølgelig altid muligt at vise flere eller andre data på displayet, så den passer med ens behov.

Wallboard er som sagt et webbaseret system, som køres via en web-browser, som f.eks. (internet Explorer, Fire Fox, Chrome) Når man starter siden skal man identificere sig med et brugernavn og log in.



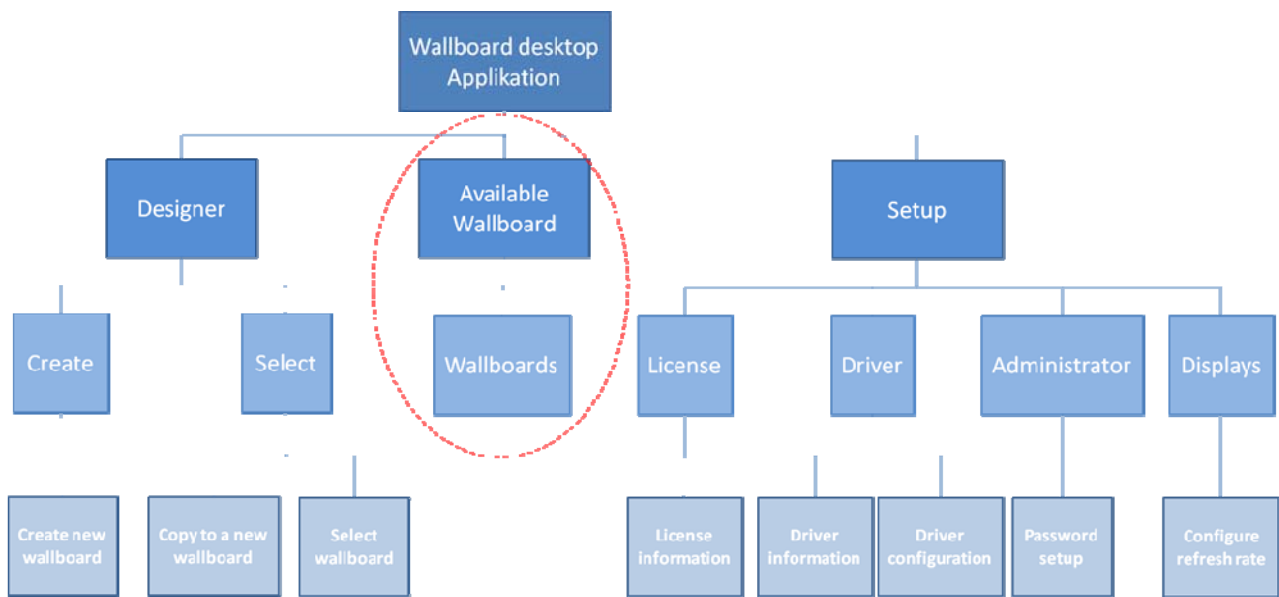
Figur 3 wallboard login

Wallboard systemet består af 3 menuer; Designer, Available wallboards og Setup, som kan ses øverst på siden:



Figur 4 wallboard menu

Figur 5 illustrerer et organisationsdiagram, som beskriver wallboard systemets menuer samt undermenuer. Diagrammet giver et bedre overblik over systemet. Jeg har markeret *Available wallboard* menuen med en rund rød cirkel, for at fremhæve, at det er den del af wallboard, som jeg kommer til at designe og udvikle til wallboard mobilapplikationen. I de kommende afsnit vil der være en detaljeret beskrivelse af de 3 hovedmenuer og deres undermenuer.



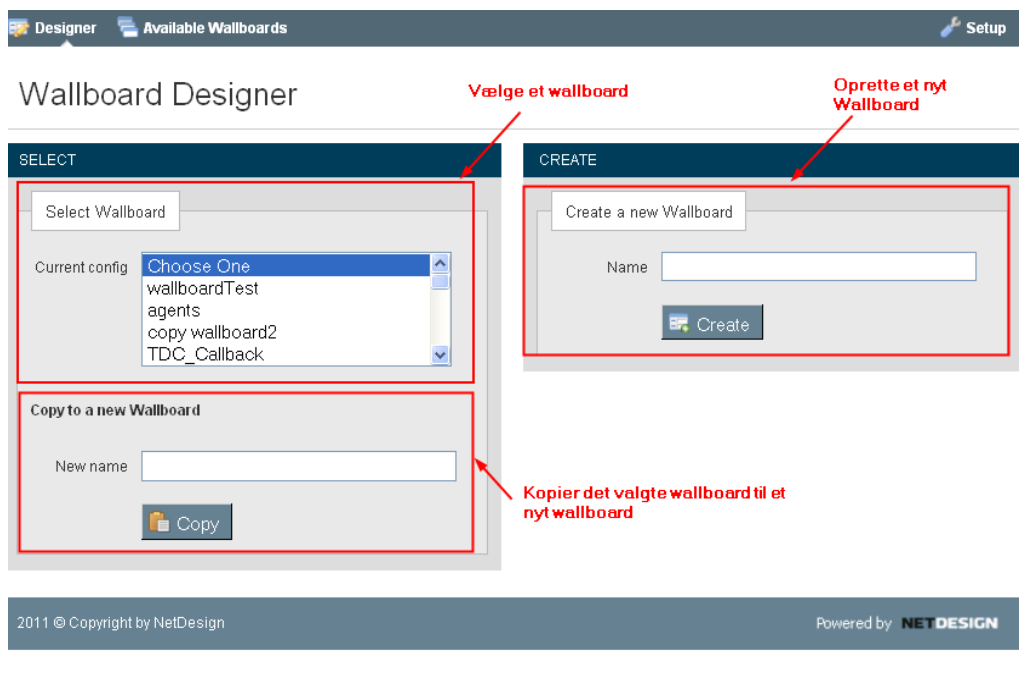
Figur 5 Organisationsdiagram

1.4.1 Designer menu

I Designer menuen kan man oprette et nyt wallboard eller modificere på et allerede eksisterende wallboard. Designer siden består af 2 primære bokse; *select* og *Create*. I *Select* wallboard kan man vælge et eksisterende wallboard, ved at klikke på den, så får man lov til at konfigurere den. Man kan også oprette et kopi af et eksisterende wallboard, ved at vælge den op i *select wallboard* og skrive navnet på den nye kopi i *copy to a new wallboard* og klikke på *copy*.

I boksen *Create* kan man oprette et nyt wallboard ved at give den et navn og klikke på knappen *Create*. Når man har oprettet et nyt wallboard popper der en ny undermenu frem "Configure Wallboard", hvor man kan tilføje de vinduer, informationer man vil have på wallboard og bestemme stylesheet.

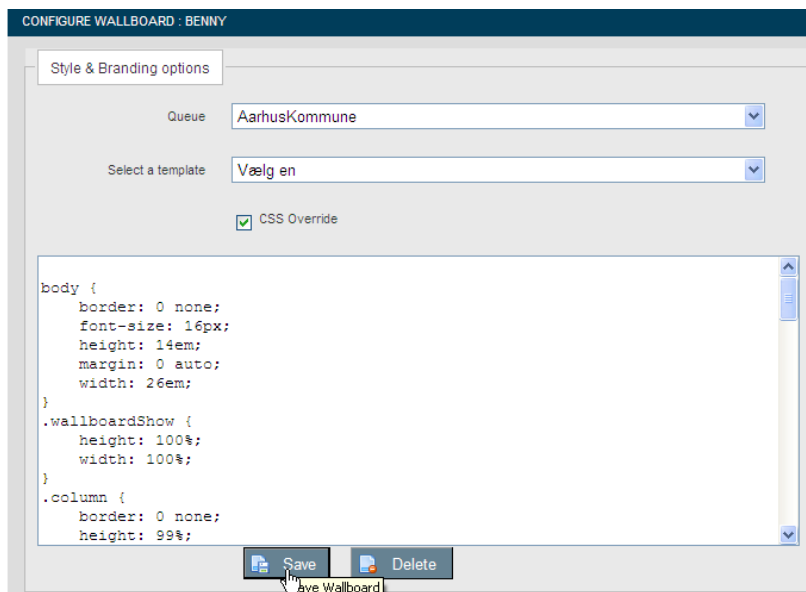
Sådan ser designer menuen ud til at starte med:



Figur 6 Designer side

Kunderne vil ikke være begrænset med faste layouts, fordi en af fordelene ved NetDesign's wallboard er, at det er muligt at foretage individuelle wallboard designs. Man har som kunde også mulighed for, at ændre i stylesheet på sin wallboard, til enhver tid. Det er meget nemt at designe og tilpasse wallboard, dette kan gøres i "Configure wallboard", hvor man kan tilføje css scripts.

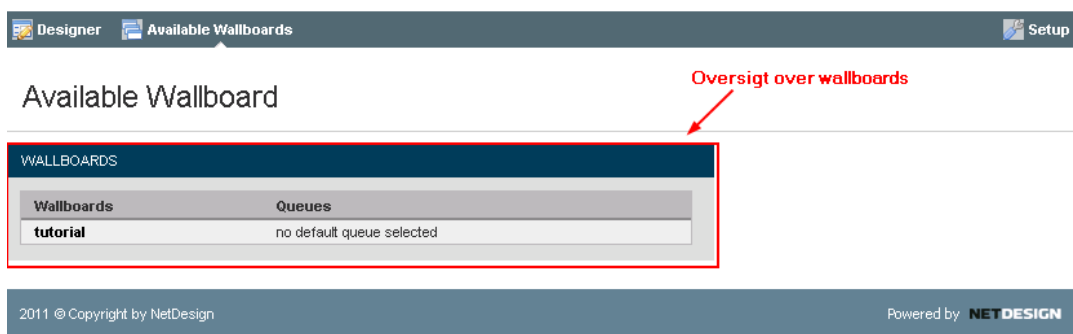
Nedenstående figur 7 viser configure wallboard boksen:



Figur 7 wallboard konfiguration

1.4.2 Available wallboards menu

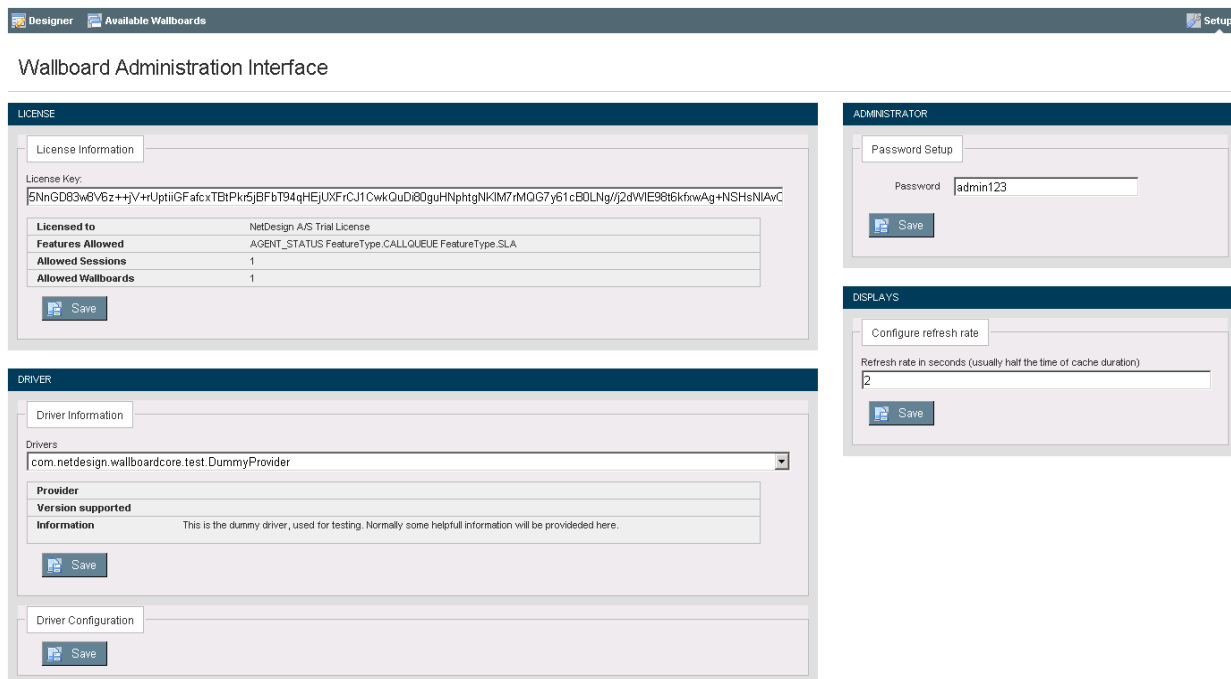
“Available Wallboards” menuen er meget simpel, den viser et oversigt over de wallboards, som er oprettet i systemet. Her kan man vælge det wallboard som man ønsker at køre, ved at klikke på den, og wallboard vil poppe op på skærmen. Herunder kan man se at der kun et wallboard, som hedder tutorial. Når man opretter et wallboard i designer menuen, kan det ses med det samme og vil kunne køres fra “Available wallboards” menuen.



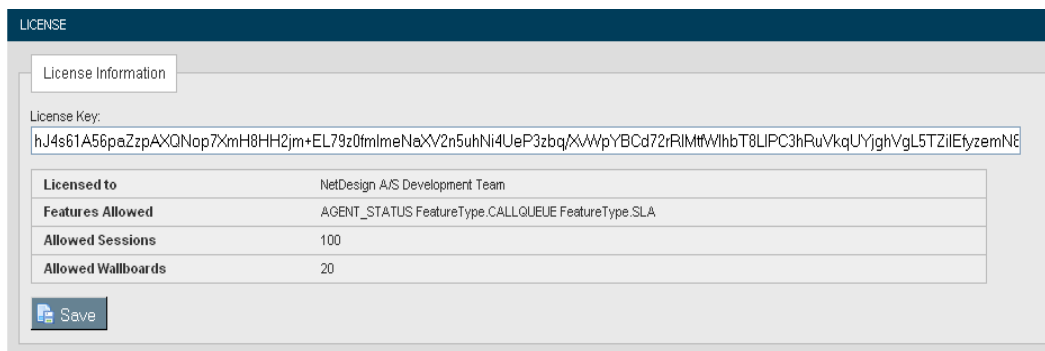
Figur 8 Available wallboard

1.4.3 Setup menu

Setup siden består af 4 blokke; license, driver, administrator og displays, som kan ses på nedenstående screenshot.



I det første blok "license", findes information vedrørende licensen som man har fået fra NetDesign, herunder er det også specificeret, hvor mange sessioner og wallboards kunden har købt licens til. På nedenstående screenshot kan du se at der er licens til 100 sessioner samt 20 wallboards.



I den næste blok "driver" kan man bestemme hvilke driver wallboard skal køre mod, ved at vælge en af driverne i drop down menuen. Hver driver har sin egen driver konfiguration.

DRIVER

Driver Information

Drivers
com.netdesign.wallboardcore.test.DummyProvider

Provider

Version supported

Information
This is the dummy driver, used for testing. Normally some helpful information will be provided here.

Save

Driver Configuration

Save

I administrator blokken har du mulighed for at se og ændre adgangskoden, som du bruger til at logge på wallboard systemet. Adgangskoden ændres ved at overskrive adgangskoden og klikke på save.

ADMINISTRATOR

Password Setup

Password admin123

Save

I den sidste blok displays, kan man angive/ændre den tid, som wallboard skal opdatere med.

DISPLAYS

Configure refresh rate

Refresh rate in seconds (usually half the time of cache duration)

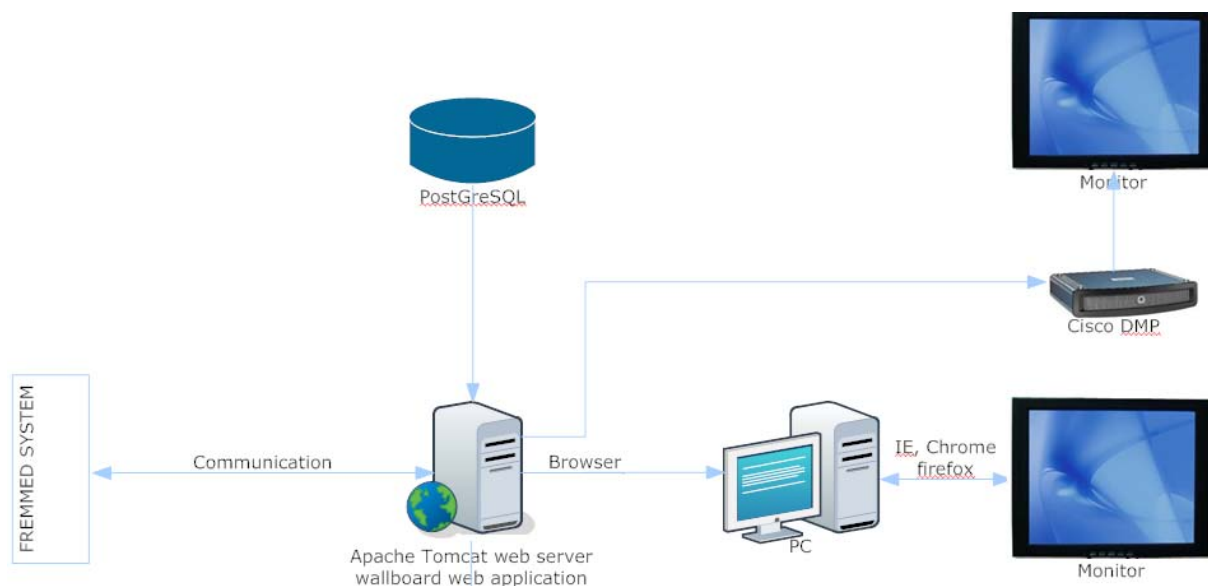
2

Save

1.5 Wallboard desktop Overview

Nedenstående diagram giver et overordnet overblik over wallboard systemet. Wallboard kører på en apache Tomcat web server. Den kommunikerer med et fremmed system, som kan indeholde informationen, der skal bruges i wallboard. Det fremmede system kan være Outlook⁴, hvor wallboard f.eks. kan hente informationer om medarbejdernes status. Det kan være UCCX⁵, som giver oplysninger om f.eks. antallet af kald, mistede kald, agenter som er logget ind, agenter som taler eller agenter som er optaget. Det kan også være Remedy system⁶, som viser de sager/opgaver som bliver oprettet i systemet. Wallboard kommunikerer med en Postgresql database server.

Man kan have wallboard kørende på en skærm i f.eks. et fællesrum, eller mødelokale, vha. DMP. Her er wallboard uafhængig af en PC. DMP'en kører wallboard ved hjælp af Mozilla browser⁷. Man kan også vise wallboard på en storskærm, som er forbundet til en computer, hvor wallboard køres ved hjælp af en webbrowser (IE, Chrome, firefox) på computeren.



Figur 9 Overview over wallboard systemet

⁴ Outlook er et program, der giver dig mulighed for at sende, modtage og organisere e-mail, men også holde styr på din kalender og dine kontaktpersoner.

⁵ Unified Contact Center Cisco

⁶ Et system som holder styr på sager og tids registrering

⁷ Mozilla browser er en webbrowser ligesom IE og Chrome

1.6 Motivation og baggrund

Jeg har været i praktik hos NetDesign A/S, i perioden februar 2011 til juni 2011. I den periode har jeg arbejdet bla. med at teste wallboard systemet, som stadig var under udvikling på daværende tidspunkt. I gennem min praktikperiode har jeg stiftet en del bekendtskab til wallboard systemet.

Udvikling til mobilapplikationer er den nyeste trend blandt virksomhederne. NetDesign nævnte også, at de snart vil begynde med at udvikle mobilapplikationer, idet det var det nyeste trend indenfor softwareudvikling og mange virksomheder var begyndt at efterspørge det. Det gav mig inspiration og motivation til at tage det første skridt og lave NetDesign's første mobil applikation.

Medarbejderne og lederne som anvender wallboard har den typisk kørende på en skærm ved deres kontor/afdeling. Dette gør medarbejderne og lederne bliver afhængige af deres plads, derfor vil en wallboard mobil applikation have mange fordele for brugerne. En leder, som typisk ikke befinder sig på sit kontor og tit er til møder vil gerne holde øge med sine medarbejdere og serviceniveauet for kundebetjening i sin afdeling og med en wallboard mobil kan han/hun altid have en realtime opdatering uanset hvor de er i virksomheden. Dvs. med wallboard mobil øges mobiliteten, så medarbejderne og lederne kan operere, når som helst og hvor som helst. brugerne får nemmere ved at planlægge deres tider, mht. pauser, møder og andre aktiviteter. Det bringer produktiviteten uden for kontorets rammer. Medarbejderne kan være produktive selvom de holder "pause" eller lavet noget andet. Det gør at de kan multitasking, dvs. lave flere ting samtidig. Medarbejderne/lederne har ved hjælp af mobilapplikationer adgang til real-time produkt data hele tiden. Kunderne har en anden måde at interagere med deres virksomhed. Listen over fordele fortsætter, men konklusionen er at flere og flere virksomheder har indset, hvor vigtigt mobile apps er for deres forretnings udvikling.

Herunder har jeg listet nogle fordele:

- **Sparer tid**
- **Produktivitet**
- **Sparer penge**
- **Øge mobilitet og multitasking**
- **Give større frihed til medarbejdere**
- **Uafhængighed af PC/storskærm**
- **Kunder kan interagere med virksomheden på en nemmere måde**
- **Øge serviceniveauet**

2. Projektbeskrivelse

I dette kapitel har jeg beskrevet, hvilke udviklingsmetoder jeg har brugt til at udvikle mobilapplikationen. Jeg har også beskrevet hvordan jeg har planlagt mit projekt og lavet en tidsplan, der indeholder de overordnede aktiviteter. Til sidst har jeg lavet en risikoanalyse.

2.1 Udviklingsmetoder

Til udvikling af wallboard mobilapplikationen har jeg fulgt en tilpasning af Unified Process⁸ UP, som er en iterativ metode, der bruges til udvikling af software processer. UP består af 4 faser: Inception, elaboration, construction og transition. Der har ikke været mange iterationer i hver fase, men der har været et par stykker i nogle af faserne. Jeg var til statusmøde i NetDesign ca. 1-2 gange om måneden, hvor vi diskuterede status og forbedringer, fejl etc. Hvor jeg tog hjem efter og arbejdede ud fra de ting mig og min vejleder blev enige om. På denne måde har der været iterationer i nogle af faserne, specielt design og implementeringsfasen.

Herunder kommer en kort beskrivelse faserne i Unified process, som jeg også har været igennem.

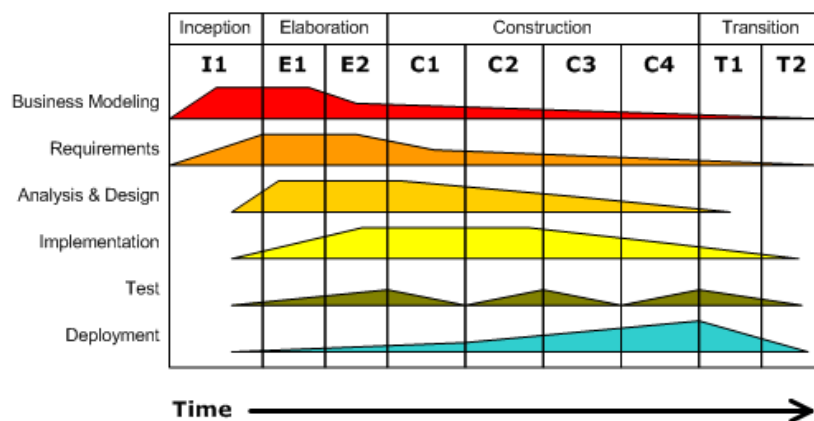
Inception: Er opstarts fasen, hvor man analyser kravene og fastlægger de centrale elementer i projektet. Målet med inception er, at forstå hvad der skal udvikles, at identificere funktionerne i systemet. Det er også her man beslutter sig om man skal fortsætte med projektet eller droppe det. Man definerer projektets plan, risici og udviklingsmetoder.

Elaboration: Under denne fase bestemmes kravene og arkitekturen fastlægges. Der laves et udkast af projektplan og risici defineres. Her arbejdes der også med analyse og design, hvilke gør projektet mere håndgribeligt og nemmere at bestemme hvordan produktet skal implementeres.

Construction: I denne fase implementeres det endelige system og testes for fejl. Formålet er her at fastslå om løsningen opfylder målene. Derudover skrives rapporten og brugermanualer.

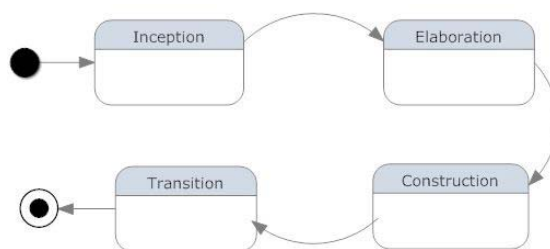
Transition: formålet med denne fase er, at levere produktet til slutbrugerne. Systemet finpudses for småfejl.

⁸ Se litteraturliste [4] s. 633-636



Figur 10 Unified process

Herunder ses faserne i Unified process. Når projektet starter befinder man sig i starttilstanden så går man igennem de 4 udviklingsfaser og slutter til sidst i sluttilstanden, hvor produktet er færdigt og kan leveres til kunden.



Figur 11 faser i Unified process [4]:fig 15-11 s. 633

2.2 Tidsplan

Der er afsat 18 uger til projektet med start d. 05-09-2011 og slut d. 01-02-2012. Jeg har lavet en overordnet tidsplan, som består af aktiviteterne i projektet og hvor dage jeg har afsat til dem. Det skal lige sige at datoformatet på tidsplan aktiviteterne er angivet i måned/dato/år. Tidsplanen har givet mig overblik over mit projektførløb. Dette hjalp mig med at holde styr på hvor langt jeg var nået med projektet, hvad jeg manglede og hvad næste opgave var. Når man kigger på tidsplanen så ligner den en tidsplan til en vandfaldsmodel⁹, fordi man kan se at når en aktivitet slutter starter jeg på et nyt aktivitet, men i virkeligheden har det ikke forløbet på denne måde. Tidsplanen er ikke blevet fulgt til punkt og prikke, men jeg har i stedet for brugt den til at holde styr på de forskellige aktiviteter.

I tidsplanen kan man også se de forskellige UP faser i forhold til aktiviteterne i tidsplanen. Den anden aktivitet i tidsplanen kan oversættes til at være forberedelsen faser fra UP modellen, hvor jeg interviewede

⁹ Vandfaldsmodel er en udviklingsmodel, hvor udviklingen betragtes som flydende faser se litteraturliste [4] s. 629

brugerne og startede med at definere kravene. Den tredje aktivitet ligner elaboration fasen, hvor jeg designede mockups af systemet og fik dem reviewet af brugerne. Fjerde aktivitet er construction fasen, hvor jeg lavede den største del af implementeringen og testede applikationen. I sjette aktivitet tester jeg applikationen med rigtige brugere, hvor de får lov til at se applikationen på en rigtig telefon og prøve den, dette aktivitet kan have træk fra transition fasen, hvor produktet leveres til brugeren og kan finpudses for evt. småfejl.

Herunder vises eksamensprojektets plan, som består af de overordnede aktiviteter i projektet. Pilene mellem de forskellige aktiviteter indikerer en afhængighed, dvs. at designfasen afhænger af at der er en analyse og udviklingsfasen afhænger af at man har overvejet noget analyse og design. Det betyder man ikke kan starte med at implementere når man ikke har gjort sig nogle design overvejelser, for ellers ender det med at man laver noget som ikke opfylder brugernes behov og ønsker. Man kan sagtens begynde starte en smule på implementeringen samtidig med at man laver nogle design overvejelser. Jeg startede f.eks. med at lave webvieweren på en simpel måde, samtidig med jeg var i gang med designfasen.

| # | Task | Assigned To | Start | End | Dur | 2011 | | | | 2012 |
|---|--|------------------|----------|----------|-----|------|-----|-----|-----|---------|
| | | | | | | Sep | Oct | Nov | Dec | Mont... |
| | Design og udvikling af android sitespecific browser | Shaymaa M Yassen | 9/5/11 | 1/22/12 | 100 | | | | | |
| 1 | Inklende analyse/læsning hvor jeg undersøgte de forskellige teknologier. Herunder litteratur søgning og Google søgning, hvor jeg finder relevante bøger og artikler. | | 9/5/11 | 10/7/11 | 25 | | | | | |
| 2 | Analysefasen: definere målgruppen, skitsere personas, definere brugerbehov ved at interviewe eksisterende bruger og tilsidst bestemme krav. | | 10/8/11 | 11/2/11 | 18 | | | | | |
| 3 | Designfasen: designe mockups og reviewe dem sammen med brugerne. | | 11/3/11 | 11/25/11 | 17 | | | | | |
| 4 | Udvikling: webviewer, wallboard skabeloner og UI | | 11/26/11 | 12/19/11 | 16 | | | | | |
| 5 | Test: Applikationen testes på AVD, HTC desire og Iphone 4. | | 12/20/11 | 12/26/11 | 5 | | | | | |
| 6 | Evaluering: her studeres og udarbejdes brugeroplevelses test, heuristisk evaluering, gestaltslove. | | 12/20/11 | 12/29/11 | 8 | | | | | |
| 7 | Rapportskrivning | | 12/6/11 | 1/22/12 | 34 | | | | | |

2.3 Risikovurdering

I dette afsnit beskriver jeg de risici, som kan skade/forhindre mig i projektførelsen. I den nedenstående tabel har jeg listet de vigtigste risici. Tabellen er delt op i risici, skade; som beskriver størrelsen på skaden og reducere skaden, hvor jeg kommenterer hvordan jeg har tænkt mig at forebygge og løse skaden.

Det har hjulpet mig med at holde styr på de problemer jeg kan komme ud for samt hvordan jeg kan forebygge dem.

| Risici | Betydning | Reducere risici |
|--------------------------------|-----------|---|
| Tidsmangel | Medium | Jeg vil planlægge mit projekt således at jeg ikke kommer i tidspres, men kan blive færdig inden deadline. |
| Ressource mangel | Medium | Det bliver svært at arbejde hvis der er ressource mangel, herunder materialer (f.eks. Android telefon, adgang til NetDesign's netværk). |
| Deadline ikke overholdes | Høj | Planlægge fornuftigt for at nå deadline og opfylde projektets mål. |
| Datatab | Høj | Tage backup af det jeg laver hver gang. Bruge evt. dropbox |
| Gå i stå | Lille | Hvis jeg går i stå vil jeg prøve at snakke med vejleder for at få hjælp/inspiration til at komme videre. |
| Overse nogle krav | Medium | Kigger på min kravspecifikation ofte for at tjekke om der noget som jeg har overset/glemt |
| Programmerings problemer | Medium | Planlægge inden jeg begynder med at programmere. Bruge Google for at finde løsning til programmeringsproblemerne. |
| 10 point kursus Web 2.0 på DTU | Medium | Jeg deltager i andet kursus løbende med mit eksamensprojekt. Jeg vil planlægge således at jeg har nok tid til eksamensprojektet. |
| Vejleder i ND/DTU ikke har tid | Høj | Det vil påvirke min plan, hvis min vejleder ikke har tid, derfor vil jeg booke møder i god tid. |

3. Teknologier

Her vil jeg redegøre for hvilke teknologier, der anvendes til udvikling af smartphone applikationer.

Herunder hvilke platforme der findes på markedet, hvilke metoder der findes til udvikling af mobilapplikationer og ikke mindst fordelene og ulemperne ved dem. På baggrund af dette vælges der en passende teknologi.

3.1 Smartphone platforme

En af de største udfordringer i udvikling af mobilapplikation er, at vælge en platform. Der findes så mange forskellige platforme ude på markedet, at man bliver forvirret. Lige nu er der seks store smartphone platforme¹⁰: Android, iOS, RIM, WebOS, Windows Phone 7, og Symbian. Ud over det, er der 4 store tablet platforme: iOS, Android Honeycomb, WebOS, og RIM. Hver platform er bygget på forskellige udviklingsprog, det gør at kan man ikke migrere den samme kode mellem de forskellige operativsystemer. Det betyder at der er brug for flere udviklere med forskellige kompetencer og kendskab til de forskellige sprog.

Apple bygger på Objective-C, hvor Android og Blackberry bygger på Java. Windows Phone 7 bruger C #, WebOS og Symbian bruger C + +. Dette betyder, at man ikke kan bruge en udvikler til hele jobbet, men skal bruge udviklere som har kompetencer indenfor hver platform. Det er en meget tidskrævende proces. Planlægning, udvikling og test af 8 forskellige versioner af en enkelt applikation er hverken nemt eller hurtigt. Det er svært at vedligeholde en applikation i 8 forskellige versioner. For hver gang man laver en ændring i applikationen, skal der rettes i de 8 versioner. Hver gang operativsystemet kommer med en ny opdatering bliver man nødt til at opdatere applikationen så den er tilpasset i alle versionerne. Dette vil koste virksomheden mange penge.

Derfor er det svært at vælge, hvilke platform man skal udvikle til. På nuværende tidspunkt er det **Android** og **iOS**, som er de store spillere på markedet, men for knap 5 år siden var det RIM og WebOS, som var de største spillere. Nu er RIM og WebOS i nedgang. Derfor er det svært at forudse, hvordan markedet kommer til at se ud om 5 år.

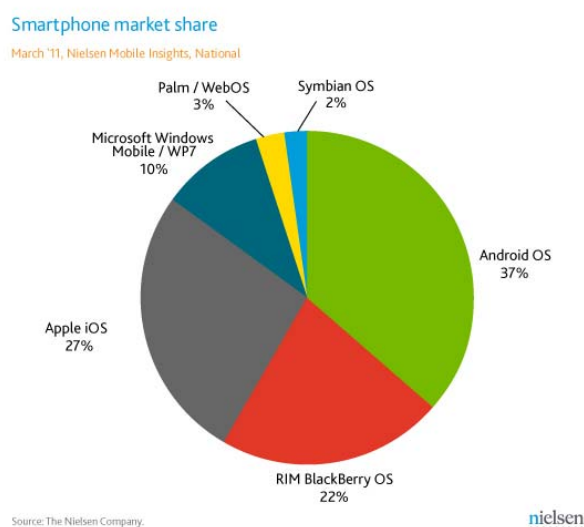
¹⁰ Kilde: se litteraturliste [18]



Figur 12 De 8 Største smartphone /tablet platforme [18]

Nedenstående diagram illustrerer de seks forskellige smartphone platforme og deres markedsandel fra marts 2011. Det kan tydeligt ses at Android og iOS, har den største andel på markedet. På verdensplan har Google's mobil styresystem Android overhalet Apples Iphone. Man kan se at Android har den største markedsandel på 37 %, dernæst kommer Apple iOS med 27 % og RIM Blackberry OS med 22 %.

Det skyldes bla. at Android er ude på forskellige devices og er ikke bundet til en type device som Iphone. Iphone er meget specifik software, som er designet til specifikt hardware, hvilke giver en meget god brugeroplevelse, men gør at brugerne ikke har forskellige smartphone varianter at vælge imellem. Med Android har brugerne til gengæld en lidt større frihed til at vælge imellem forskellige telefoner.



Figur 13 Nielsen Mobile Insights

3.2 iPhone

iPhone kører på iOS, som er Apples mobile styresystem. iOS var oprindeligt udviklet til iPhone¹¹, men nu er den blevet udvidet til at støtte andre Apple enheder, såsom iPod Touch, iPad og Apple TV.



iOS bygger på programmeringssproget Objective-C til at udvikle applikationer. En af de største fordele ved at udvikle mobilapplikationer til iPhone er, at man udvikler til én bestemt mobiltelefon og ikke flere forskellige mobiltelefoner, som man gør med Android. Dette gør det nemmere at udvikle, idet man ikke skal tænke på at det måske ikke vil køre på en bestemt mobiltelefon. Det giver også en bedre brugeroplevelse, når applikationen er udviklet til netop en telefon. Udviklerne har det også nemmere ved at teste, idet de kun skal teste det på en, højst to mobiltelefoner. Ulempen ved at udvikle til iPhone er, at det ikke er Open Source. Man bliver nødt til at registrere sig i Apple Developer program.

For 99 dollars om året kan man installere og distribuere sit software til et begrænset antal enheder gennem udviklerportalen¹².

iPhone understøtter heller ikke Adobe Flash player.

3.3 Android

Android er et styresystem som er rettet mod mobile enheder, såsom smartphones og tablet pc'er. Der er mange fordele ved at udvikle applikationer til Android. En af de væsentligste fordele er, at det er gratis Open Source. Det kræver kun at man henter Android SDK¹³ i udviklingsmiljøet, og herefter er man klar til at udvikle.



¹¹ Kilde: Se litteraturliste[20]

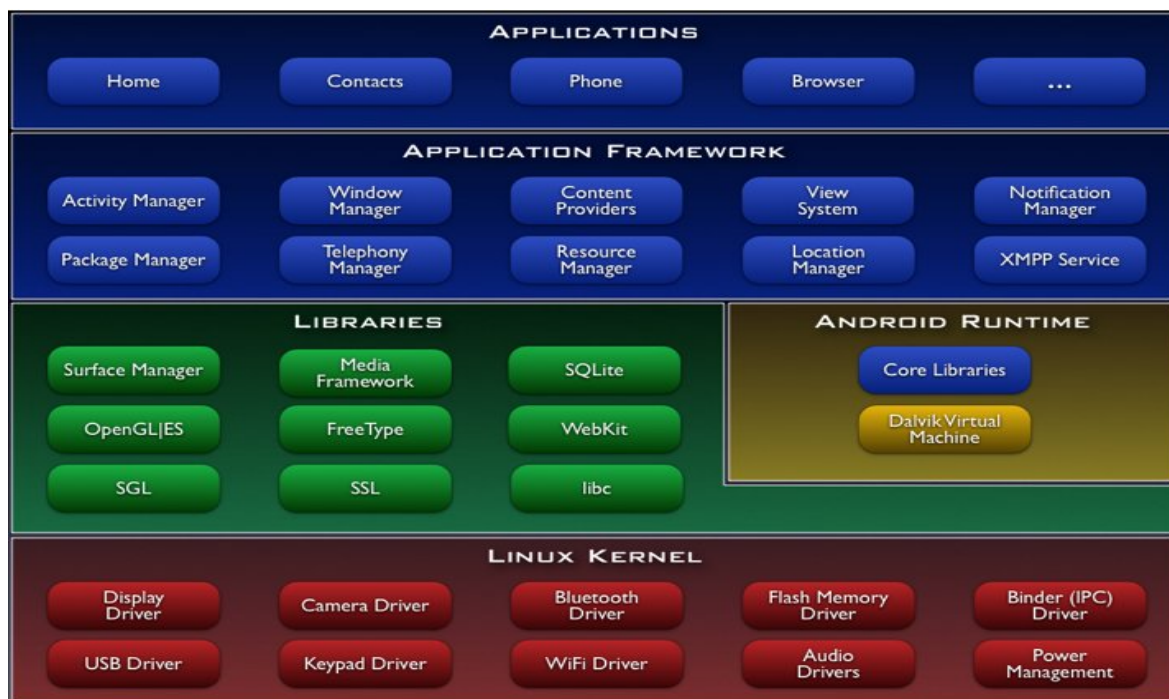
¹² Kilde: Se litteraturliste[21]

¹³ Se litteraturliste[22]

Android bruger Java programmeringsproget. En anden fordel er at Android understøtter Adobe flash player, hvilket kan være en fordel hvis der skal bruge flash til f.eks. animationer/videoer. Android styresystemet findes i mange forskellige mobiltelefoner, hvilket er en fordel idet det giver brugeren og virksomheden bredere valgmuligheder, når de skal vælge en telefon. Når der skal udvikles mobilapplikationer til Android, skal man tænke på at der findes mange forskellige mobiltelefoner og derfor skal man gøre sig nogle overvejelser om udseendet og skærmstørrelsen, idet skærmen variere fra telefon til telefon. Det er også besværligt når applikationen skal testes på så mange telefoner. Android har også den fordel at den skalerer automatisk tekst og billeder til enhedens skærmstørrelse.

3.3.1 Android arkitektur

Nedenstående figur 15 illustrerer komponenterne i Android styresystemet. Styresystemet består af 5 lag. Her kommer en kort beskrivelse af hvert lag.



Figur 14 Android Arkitektur[13]

Applications

Applikationslaget består af de centrale applikationer, som bliver leveret på forhånd, som kalender, mobilbrowseren, kort, e-mail, telefonbog og sms. Alle applikationerne er skrevet i programmeringsproget Java.

Application framework

Android application framework støtter direkte applikationsudviklingen. Det består af en bred set af Java klasser, som bruges til at udvikle applikationer. Udviklere har fuld adgang til de samme framework API'er,

som bruges af de centrale applikationer. Applikation arkitektur er designet til at simplificere genbrug af komponenter. Det omfatter alt fra brugergrænseflader, lokalitetstjeneste, kamera, notifikationsystem og m.m.

Libraries

Android består af et sæt af C / C + + biblioteker, der bruges af forskellige komponenter af Android-systemet. Disse biblioteker er tilgængelige for udviklere gennem Android Application Framework.

Android runtime

Android bruger Dalvik virtuelle maskine, hvilke gør at hver Android applikation kører sin egen proces, med sin egen instans af Dalvik virtuelle maskine. Dalvik gør at en enhed kan køre flere virtuelle maskiner effektivt.

Linux kernel

Android bygger på kernesystemets tjenester såsom sikkerhed, hukommelse ledelse, procesledelse, netværk, og driver model. Kernen fungerer også som et abstraktions lag mellem hardware og resten af software stack.

3.4 Webapplikation vs. native applikation

Mobilapplikationer kan defineres som software på en mobil enhed; smartphone, tablet, e-reader eller en anden form for mobil enhed, som brugerne kan tage med sig hvor som helst og når som helst.

Herunder er der en kort gennemgang af de mest udbredte udviklings muligheder man har ved udvikling af mobilapplikationer samt fordele og ulemper ved de forskellige tilgange.

3.4.1 Native applikationer

Native applikationer er programmer, som er lavet decideret til et bestemt platform, ligesom når man laver programmer til computer, vil de virker på en computer, men ikke på en MAC. Det samme gælder for native mobilapplikationer, hvis de er lavet til en Android så virker de ikke på en Iphone.

Native applikationer har den styrke, at man kan lave nogle meget effektive apps, som man kan få til at se pæne ud. Ved at udvikle native applikationer, har man fuld adgang til enhedens funktioner og sensorer, såsom kamera, mikrofon og højttaler. Applikationen bliver placeret på brugerens startskærm, så brugeren har nemmere ved at få adgang til applikationen. Applikationen kan markedsføres gennem applikation markedet. Native applikationer kan udvikles til flere forskellige platforme, som f.eks. Android og iOS, men man skal være opmærksom på, at hver platform anvender vidt forskellige teknologier og udviklingssprog. Sagt på en anden måde, hvis det tager 50 timer at udvikle en applikation til Iphone, så tager det lige så lang tid at importere den samme applikation til Android. Der vil måske være små enkelte dele fra applikationen, som kan genbruges, men i princippet skrives koden fra bunden til hver platform.

Hvis man vælger at udvikle native applikation til et bestemt mobil platform f.eks. Android, vil det være billigt og hurtigt, men samtidig risikabelt. Fordi udviklingen er meget hurtig inden for den mobile verden. For hvad nu, hvis man ønsker at skifte platformen? Hvad hvis platformen taber markedsandelen eller udgår? Man kan også vælge at udvikle til alle eller flere platforme, men det vil blive for dyrt og meget tidskrævende. Denne mulighed sørger for at dække alle baser. Typisk er det tilegnet virksomheder, som ønsker at appellere til en meget større målgruppe. Der findes mange native applikationer, som findes til forskellige platforme såsom WhatsApp, Tango, Viber, Skype, Facebook og mange andre.

Som tidligere nævnt, er der 8 store smartphone / tablet platforme, og det betyder, at man skal udvikle 8 forskellige versioner af hver applikation. Hvert platform understøtter som sagt et forskelligt udviklingssprog, derfor er der brug for flere udviklere med forskellige kompetencer og kendskab til de forskellige sprog.

3.4.2 Mobil web applikation

Et andet alternativ, som er mobil web applikationer. Dette alternativ giver en mulighed, som er isoleret fra mobilplatformen, så applikationen virker med hvilken som helst enhed uanset styresystemet. Opbygningen af mobil web applikationer, som er formateret til brug på en mobilbrowser, giver en billig og sikker alternativ til native mobilapplikationer.

webapplikationen er den eneste mulighed, som automatisk fungerer på tværs af alle enheder og alle platforme. Webapp er en app, som i princippet kan det samme som native app, den eneste forskel er at webapp kræver en browser for at køre. Webapp kører igennem telefonens mobilbrowser. Webapp bruger de samme teknologier, som bliver brugt til at lave desktop webapplikationer.

Med de seneste fremskridt inden for HTML, CSS og JavaScript, giver internettet en brugeroplevelse, der ligner meget native applikationer, og brugeroplevelsen udvikler sig bedre og bedre med tiden.

Opbygningen af mobil web applikationer er en fantastisk måde, at komme ind i mobilapplikationer og samtidig undgå hele mobil platform beslutningen. Mobil web applikationer kan bruges på alle mobilplatforme, den kræver som sagt kun en mobilbrowser, med andre ord kan mobil webapplikationer anvendes på tværs af alle mobilplatforme.

3.4.2.1 Fordele ved mobile web applikationer

En af de vigtigste fordele ved mobil Web applikation er, at de ikke er begrænset af platforme. Idet mobil web applikationerne kører inde i en mobilbrowser, er de kompatible med alle platforme og operativsystemer. Men det skal også siges at der kan være en lille forskel på udseendet i de forskellige web browser, dvs. at en webapplikation kan se pæn ud i firefox men ikke i IE, men det er som sagt en lille forskel vi snakker om her.

Antag, at du vil udvikle en smartphone applikation. I stedet for at udvikle en applikation til hver platform, dvs. at du får 8 forskellige projekter, vil en webapplikation arbejde på tværs af alle platforme, så du kun behøver at udvikle én enkel applikation.

Når man udvikler mobile webapplikation, udvikler man hovedsagelig til én platform, nemlig webben. Men man er ikke begrænset til et bestemt udviklingsprog, som man er med native applikationer. Derfor kan man udvikle i det sprog man har kompetencer i.

Vedligeholdelse af en mobil webapplikation svarer til at vedligeholde en standard native mobil applikation. Vedligeholdelse vil ikke være tidskrævende, idet det kun er en applikation man skal kigge på og ikke 8 forskellige versioner. En anden vigtig fordel ved webbaseret applikationer er, at ingen data kan gå tabt, hvis man mister telefonen. Ingen kritiske informationer kan hentes. Alle informationer er sikret oppe i databasen eller på webben og ikke på enheden. Med mobile webapplikationer har man altid den nyeste

version af applikationen, så man ikke behøver at opdatere applikationen, hver gang der kommer en ny opdatering. Det betyder også at omkostningen hos producenten og kunden er billigere.

3.4.2.2 Ulemper ved mobile web applikationer

Der er også ulempe ved mobile webapplikationer. I native applikationer kan man tilgå hardware sensorerne fra mobilenheden, såsom mikrofonen, højttaler og kameraet, men det kan man ikke gøre med mobile webapplikationer. Derfor kan man ikke udvikle mobile webapplikationer, hvis ens applikation skal kunne udnytte hardware sensorer, som er på enheden. Brugeroplevelsen er heller ikke på lige fod med native applikationer, som er designet til et bestemt enhed/platform.

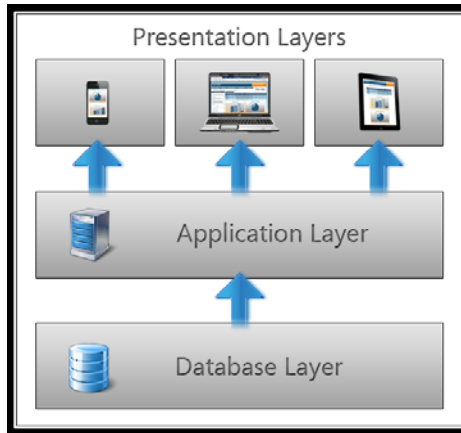
3.4.3 M-power mobile web applikationer

Som sagt så er mobile webapplikationer den simpleste, billigste og hurtigste måde at udvikle mobile applikationer på, men det er ikke alt. Der findes en simplere, billigere og hurtigere måde, at udvikle mobile web applikationer på. Med M-power teknologien, kan man udvikle web og mobile webapplikationer meget hurtigere og uden ekstra programmering. M-power virker således, at applikationen ligner og føles som en native applikation og den skifter automatisk layout alt efter hvilke enhed (Smartphone/ tablet/ pc) den køres fra. Denne løsning er optimal, hvis det indhold man ønsker at vise på enheden allerede ligger på ens almindelige hjemmeside.

3.4.3.1 Hvordan fungerer M-power?

En applikation som er bygget på M-power teknologien, består af flere HTML præsentations lag, der er formateret til at virke på de forskellige enheder. Når en bruger anvender applikationen, kan den automatisk registrere brugerens enhed og programmet vil vise den passende præsentations lag. Dette giver brugeren en god brugeroplevelse på applikationen, lige meget hvilke enhed brugeren anvender. Der findes mange applikationer som anvender M-power, det kan man se på når man ved hjælp af mobilbrowseren, går ind på en hjemmeside og man kan se at url adressen til hjemmesiden skifter til m. url adressen. Hvis man f.eks. skriver www.bt.dk i sin mobilbrowser, vil adressen skifte til m.bt.dk. Et godt eksempel på en applikation anvender M-power teknologien er social netværks applikationen facebook. Logger man på facebook via en computer vises computer præsentations lag, logger man via mobilbrowseren vises mobil præsentations lag.

Nedenstående figur 15 illustrerer opbygningen af en M-power applikations struktur. Der er 3 præsentations lag, en for hver enhed (smartphone, laptop og tablet). Som man kan se bruger alle præsentations lagene den samme applikations lag og database lag.



Figur 15 M-power [18]

Når man anvender en smartphone browser til at logge på facebook, ved at skrive adressen i browseren, finder applikationen ud af, at enheden er en smartphone og viser derfor præsentationens lagen for smartphones. Applikationer som bruger M-power teknologien får et m. foran url adressen, når man åbner applikationen fra en mobilenhed.

Facebook applikation på pc browser vs. Mobile browser



Figur 16

3.4.4 PhoneGap og Titanium

Som tidligere beskrevet er der mange platforme at vælge imellem, og det er noget, som hele tiden er i udvikling. De platforme som er nr. 1 i dag er ikke nødvendigvis nr. 1 om et år. Derfor er mange udviklere gået til at udvikle webbaserede applikationer fordi de virker på alle platforme, men de har også nogle begrænsninger, men med PhoneGap og Titanium kan man udvikle native applikationer til alle de populære platforme uden at skrive implementere i flere forskellige sprog.

Både PhoneGap og titanium er populære opensource værktøjer til udvikling og test af mobilapplikationer, som fungerer på tværs af mobilplatforme.



PhoneGap er et webbaseret løsning, som kører native webbrowser view. Her bruges HTML, CSS og JavaScript og dernæst installeres applikationen til de forskellige platforme. Man kan vha. JavaScript få adgang til mange af telefonernes hardware sensorer ligesom når man udvikler native apps.

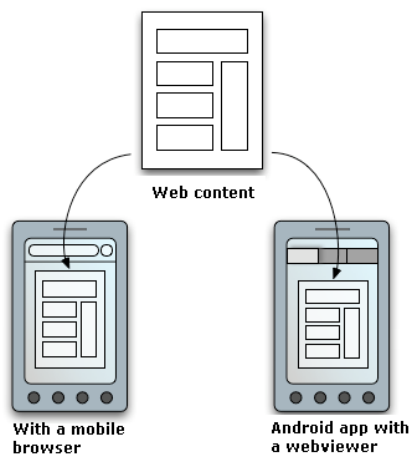
Titanium er ren JavaScript, der kompileres til native applikationer. Det fungerer både på Android og iOS platformen. Titanium mobil SDK giver mulighed for at udvikle, køre og teste native applikationer som bruger JavaScript API' er.

Fordelen ved PhoneGap er at den fungerer på flere mobilplatforme, hvorimod Titanium kun fungere på Android og iOS på nuværende tidspunkt. Derfor er der flere applikationer som bliver udviklet igennem PhoneGap.

3.4.5 Site specifik browser

En site specifik browser (SSB) er en software program/applikation, som er dedikeret til at give adgang til websider. Normalt hvis man vil have adgang til en webside, åbner man en webbrowser eller en mobilbrowser og skriver websidens url i browserens adressefelt. SSB har til formål, at vise webindholdet på en mere forenklet måde, hvor komplekse funktioner, scrollbar, menuer, værktøjslinjer fra browseren kan skjules. SSB har også den fordel at de ligger på startskærmen og har typisk et ikon, som man bare klikker på for at tilgå applikationen ligesom native applikationer. Site specifik browser bruges også i mobilapplikation udvikling, hvor den kombinere en blanding af native applikation samt webapplikation. SSB Applikationer

fungere som en "skal", så den ligner en native app, den har et ikon, som man kan have på startskærmen, ligesom native applikationer, men henter sit indhold fra nettet. SSB giver brugerne en bedre brugeroplevelse, fordi de ikke behøver at skrive url adressen i browseren. Selvom man udvikler site specific browser til et bestemt platform, kan webapplikationen stadig tilgås af andre enheder/platforme vha. mobilbrowseren, hvilket også er illustreret i figur 17.



















Figur 17 Webapp Overview [19]

3.4.6 Opsummering på udviklingsmuligheder

Der er som nævnt flere udviklingsmuligheder og teknologier, som kan anvendes, når man skal udvikle mobilapplikationer. Det er vigtigt, at man træffer den rigtige beslutning, om hvilke udviklingsmuligheder, man vælger. Derfor skal man være opmærksom på hvilke fordele og ulemper der er i hver mulighed. Man kan udvikle native applikation til netop én platform, men det vil være meget risikabelt, for hvad nu hvis platformen udgår? Der ligger også en udfordring i at brugernes muligheder vil være begrænset, hvis de ønsker at skifte platform. Man kan vælge at udvikle native applikation til alle platforme eller flere platforme. Dette er en fordel, hvis man ønsker en stor målgruppe. Man er ikke begrænset af platform og har større fleksibilitet, men det er meget dyrt både i udvikling og vedligeholdelse. Mobile web applikation er et rigtig godt alternativ, hvis man ikke skal bruge hardware sensorerne. Det er billigt og kan anvendes på tværs af alle platforme.

Nedenstående skema illustrerer forskellen mellem udvikling af web app og native app baseret på forskellige elementer som; omkostning, tid, udviklings ressourcer, risiko og antal platform. Det er billigt at udvikle til en platform, det er ikke særlig tidskrævende og det kræver ikke flere udviklere eller udviklingskompetencer i forhold til udvikling til alle platforme, men til gengæld er det meget risikabelt da man ikke kender platformens fremtid.

| | Develop for one platform | Develop for all platforms | Build mobile web apps | Build mobile web apps with m-Power |
|-------------------|--|---|---|---|
| Cost |  Moderate Expense |  High Expense |  Moderate Expense |  Low Expense |
| Time Required |  Moderate time requirement |  Large time requirement |  Moderate time requirement |  Low time requirement |
| Developers Needed |  One Developer |  Many Developers |  One Developer |  One Developer |
| Risk Involved |  High Risk |  Low Risk |  Low Risk |  Low Risk |
| Platforms Reached | Just one | All Platforms | All Platforms | All Platforms |

Figur 18 [18]

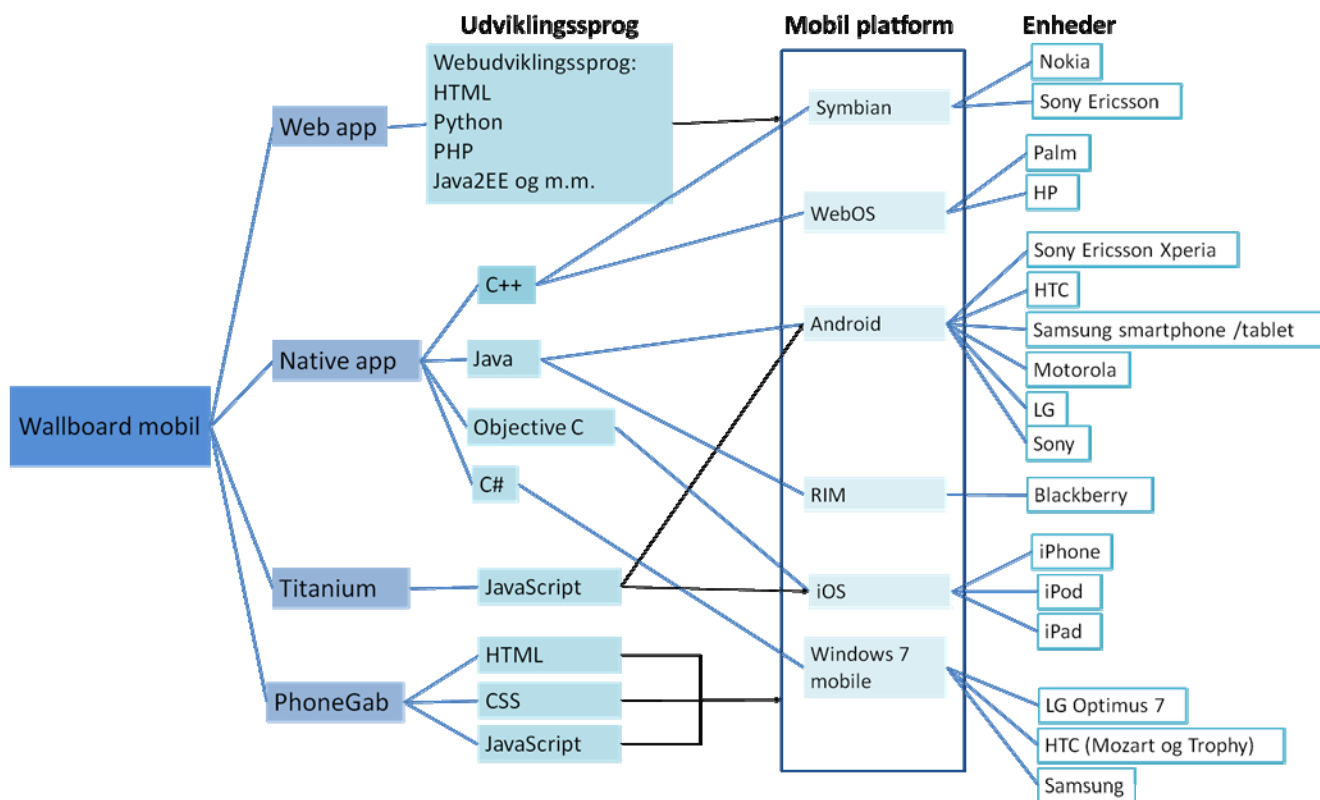
Herunder kan man se en tabel, som jeg har lavet for at illustrere de fordele og ulemper som er i de forskellige udviklingsmuligheder. Det er en opsummering af hvad jeg har diskuteret tidligere i dette afsnit.

| | Fordele | Ulemper |
|----------------------------------|---|---|
| Web app | <ul style="list-style-type: none"> - Ingen platform begrænsning - Ingen udviklingssprog begrænsning - Nem i vedligeholdelse - Informationer på app er sikret oppe i webben. - Altid nyeste version - Realtime data | <ul style="list-style-type: none"> - Ingen adgang til hardware sensorer - Ikke udviklet til specifik device, hermed mindre brugervenlighed. - Brugeren skal skrive url i mobilbrowserens adresse felt. - Ingen ikon |
| Native app | <ul style="list-style-type: none"> - Fuld adgang til enhedens funktioner og sensorer - Placeres på brugerens startskærm. - Markedsføres gennem applikation markedet. | <ul style="list-style-type: none"> - Mange forskellige platforme. - Forskellige udviklingssprog. - Risikabelt at udvikle til en bestemt platform |
| Titanium og PhoneGap | <ul style="list-style-type: none"> - Opensource værktøjer til udvikling og test. - Placeres på brugerens startskærm. - Markedsføres gennem applikation markedet. - Understøtter de populære platforme. - Nem i vedligeholdelse - Mindre kodning | |
| Site specifik browser app | <ul style="list-style-type: none"> - Ingen platform begrænsning på selve webapplikationen. - Ingen udviklingssprog begrænsning på webapp. - Nem i vedligeholdelse - Informationer på app er sikret oppe i webben. - Altid nyeste version. - Realtime data. - Placeres på brugerens startskærm. - Markedsføres gennem applikation markedet | <ul style="list-style-type: none"> - Ingen adgang til hardware sensorer - Ikke udviklet til specifik device, hermed mindre brugervenlighed. |

I nedenstående diagram har jeg lavet en oversigt over de forskellige udviklingsmuligheder som jeg har undersøgt. Wallboard mobil kan udvikles vha. 5 forskellige teknologier. Vælger man web app, bestemmer man selv hvilke webudviklingsprog man ønsker at bygge sin applikation på. Webapp kan anvendes på tværs af alle platforme hvilket betyder at det kan bruges på alle mobile enheder: iPhone, iPad, iPod, HTC, Blackberry, LG etc.

Hvis man vælger at udvikle native app, skal man først beslutte sig for, hvilke platform man ønsker. Hvert understøtter bestemte enheder, f.eks. så understøtter iOS iPhone, iPad og iPod, mens Symbian understøtter Nokia og Sony Ericsson.

Titanium er bygget på JavaScripts og understøtter på nuværende tidspunkt Android og iOS, mens PhoneGap er bygget på HTML, CSS og JavaScript og understøtter flere smartphone platforme; iOS, RIM, Android, WebOS, Windows 7 og Symbian. Både titanium og PhoneGap er smarte teknologier til udvikling af mobilapplikationer, fordi de kører native webbrowser view og udvikleren har adgang til mange af telefonernes hardware sensorer ligesom når man udvikler native apps.



Figur 19 Oversigts over 4 udviklingsmuligheder

3.5 Konklusion på valg af teknologi

Baseret på denne analyse og de fordele og ulemper ved de forskellige udviklings muligheder har jeg i samarbejde med NetDesign bestemt, at wallboard mobil applikationen skal være en Android site specifik browser. Hvor applikationens indhold er webbaseret, men sættes ind i en android webviewer app, så brugerne ikke kan se at det er en webapplikation.

Herunder understøttes alle platforme. Det er en fordel, idet wallboard kunderne ikke kan forventes at have et bestemt mobilplatform. Brugere vil også have den nyeste version af mobilapplikationen hele tiden, dvs. hvis der kommer en opdatering på applikationen, behøver udviklerne ikke, at informere brugerne om, den nye opdatering samt få dem til at opdatere til den og bruge tid/ressourcer på det. M-power teknologien vil ikke være en fordel i denne situation, netop fordi der er en del informationer og funktioner på wallboard desktop applikationen, som ikke behøves på wallboard mobilapplikationen.

Af følgende årsager har jeg i samarbejde med NetDesign valgt at det skal være en Android site specifik browser applikation og ikke en anden platform.

- Android er Open Source, dvs. det er gratis at udvikle og hente udviklingsmiljøet.
- Android er bygget på programmeringssproget Java, som jeg allerede har kendskab til og derfor behøver jeg ikke at starte med at lære et nyt programmeringssprog.
- NetDesign har tilbudt sine medarbejdere Android telefoner, hvilke betyder at de fleste medarbejder i NetDesign har en Android telefon, og hvis wallboard brugerne fra NetDesign skal anvende/teste wallboard mobilapplikationen er det smartest at webvieweren er til Android.

Både Titanium og PhoneGap er velegnede værktøjer til udvikling af mobilapplikation, men på grund af, at jeg først fandt frem til disse to teknologier midt i projektføreløbet, hvor jeg allerede havde truffet beslutning om hvilken metode jeg ville anvende og begyndt på den kunne jeg ikke bruge disse teknologier til dette projekt.

4. Prototypeudvikling

Dette kapitel omhandler processen for udviklingen af wallboard mobil applikation. Ligesom i udvikling af softwaresystemer starter jeg med analysefasen, hvor jeg definerer målgruppen, kravene for projektet, scenarier og m.m. Dernæst går jeg videre til designfasen, hvor jeg omformer kravene til simple prototyper, som vises til brugerne og vurderes inden jeg påbegynder implementeringen. I implementeringsafsnittet beskriver jeg hvordan jeg har udviklet applikationen ude fra det valgte design. Her forklarer jeg noget af koden. Sidste fase i denne proces er testfasen, hvor jeg beskriver de forskellige test og vurderingsmetoder jeg har anvendt.

4.1 Analyse

I analysefasen undersøges forskellige ting, herunder finder man ud af hvad applikationen skal kunne og sikrer at applikationen er nytteværdig, hvilket er en vigtig del i brugervenligheden. Det er vigtigt at man udvikler et produkt, som løser de opgaver brugerne ønsker at få løst. I dette afsnit starter jeg med at definere målgruppen for applikationen. Dernæst analyserer jeg brugernes behov og brugerkaraktistika. Dette gør jeg ved at anvende forskellige metoder som interview og spørgeskema. Ude fra det kan jeg definere kravene til applikationen.

4.1.1 Inspiration fra FDB

Virksomheden **FDB**¹⁴ har udviklet applikationen **Aftensmad**¹⁵ ud fra hjemmesiden: www.viskalspise.dk. De oplyser at det har ca. taget dem 4-4½ måneder for at udvikle applikationen, hvoraf analysen af kundernes behov og krav har været den længste fase i udviklingsprocessen. Jeg har taget kontakt til dem fordi jeg ville undersøge hvilke metoder de brugte for at prioritere data og funktioner fra hjemmesiden www.viskalspise.dk i mobilapplikationen Aftensmad. FDB oplyser at de interviewede forskellige mennesker og fandt ud af hvad de ønsker applikationen skal indeholde i store træk. Ud fra disse undersøgelser, har FDB dannet idéer om hvad det er brugerne vil have samt foretrækker, på baggrund af dette kunne de udvikle en målrettet applikation, som opfylder brugernes behov. Jeg har anvendt samme metode i udviklingen af wallboard mobil applikationen, ved at interviewe eksisterende bruger og finde ud af hvad de kunne ønske sig mobilversionen skulle indeholde.

¹⁴ www.fdb.dk

¹⁵ Mobilapplikation til Android og Iphone, som indeholder 2.600 aftensmad opskrifter.

4.1.2 Personer

For at designe en god brugeroplevelse, skal man have en stor forståelse for sine brugere, herunder deres behov, ønsker, drømme, ambitioner, holdninger, og mål til den applikation man leverer. For at finde ud af hvad brugerne finder nyttigt, brugbart og ønskeværdigt skabes en persona, som repræsenterer de rigtige brugere. Når man skal definere brugerens behov, er det meget vigtigt at finde ud af, hvilke slags brugere der udvikles til. F.eks. vil uerfarne brugere have behov for flere instruktioner i forhold til eksperter brugere. Her er der en række elementer, der skal undersøges såsom, hvilke egenskaber og kompetencer tilegner brugeren sig, er brugeren vant til at anvende smartphone og mobilapplikationer, anvender brugerne internettet til dagligt etc. Applikationen skal skabe en værdi for brugerne. Derfor har jeg lavet en analyse af brugerens karakteristika.

Da det ikke er et helt nyt produkt, findes der allerede eksisterende brugere, som anvender wallboard desktop applikationen. Målgruppen for denne applikation er de allerede eksisterende brugere af wallboard desktop applikationen, som vil anvende mobilversionen for at forbedre deres arbejdsproces.

En persona er en fiktiv, repræsentativ person, som er beskrevet i detaljer, f.eks. navn, køn, alder, fritidsinteresser, uddannelse, arbejde, mål, færdigheder, holdninger, adfærd og miljø.

Når man siger at en persona er "repræsentativ", det skal forstås sådan, at en persona repræsenterer et segment af brugergruppen, og bruges som guide i brugeroplevelses design.

For at udvikle min persona har jeg udarbejdet et spørgeskema¹⁶, som 8 medarbejdere i NetDesign har svaret på.



Michael Andersen, 46 år, supporter for cisco callmanager.

Michael arbejder som supporter for cisco callmanager, i cisco support afdelingen i NetDesign. Han er uddannet som elektronik mekaniker. Michael har arbejdet i NetDesign i snart 6 år. Han startede som konsulent, men for 2 år siden blev han supporter for Cisco support callmanager.

Michael holder meget af sit arbejde, han kan godt lide at hjælpe kunderne med at løse diverse tekniske problemer.

Michael bor sammen med sin kone Lise, lidt uden for Hillerød. De har to børn, Mads på 3 og Maria på 6. De har også en kat som er navngivet Luna. Michael er en familie mand, han bruger meget af sin fritid sammen

¹⁶ Se bilag 1 Spørgeskema

med familien. Han kan godt lide at lave havearbejde. Han går meget op i at motionere. Han løber 3 gange om ugen i Farum, sammen med andre kollegaer fra NetDesign lige efter arbejdet.

Michael er meget glad for wallboard løsningen, han kan overvåge systemet ved blåt at se på skærmen. Han kan se statussen på sine medarbejdere i supportafdelingen i Farum og Skejby. Han kan danne sig et overblik over hvilke sager, som er afsluttet og hvilke som stadig mangler. Gennemsnitligt så kigger Michael op på wallboard skærmen 20-25 gange på en time.

Michael arbejder primært foran computer skærmen, hans opgaver er typisk at holde styr på sin afdeling, supportere de kunder som har problemer med deres cisco systemer. Han er meget kompetent inden for IT. Han bruger internettet til at læse nyheder, tjekke arbejdsrelateret og personlige mails, sidde på facebook og meget mere. Michael har en HTC Desire, som han har fået fra firmaet. Han bruger sin smartphone til at tjekke mails, kalender, mobil netbank, anvende diverse smarte applikationer, såsom Runapp, aftensmad, DBA, teamviewer, facebook, dropbox etc.

I en artikel af Forrester research "*Define your mobile development strategy*" [8] har de defineret de typer medarbejdere, som anvender mobilapplikation på arbejdet. Hver gruppe anvender mobilapplikationerne/ smartphones på deres egen måde under arbejdstiden.

Herunder har de opdelt medarbejderne i 4 overordnede grupper:

- *Opgave medarbejder*
- *Informative medarbejder*
- *Wannabe medarbejder*
- *Superkommunikatorer og kommunikator*

Opgave medarbejder bruger applikationen til deres daglige arbejde. Findes i det lodrette marked, som inkluderer kemikalier, råolie, fremstilling, sundhedssektor. Eksempler på denne type kan være leverance chauffør som leverer varer det ene sted til det andet, eller kontrolmedarbejder som kontrollerer forskellige ting.

Informativ medarbejder er ikke involveret i produktionen af fysisk produkt, i stedet for bruger de tid på at analysere informationer, træffe komplekse beslutninger baseret på deres bedste bedømmelse og interagerer med andre medarbejdere og kunder leverandører. De bruger deres personlige arbejdstelefon, i arbejdet, og virksomheden betaler for det. De har krav om at få adgang til oplysninger og data for at kunne gøre deres job.

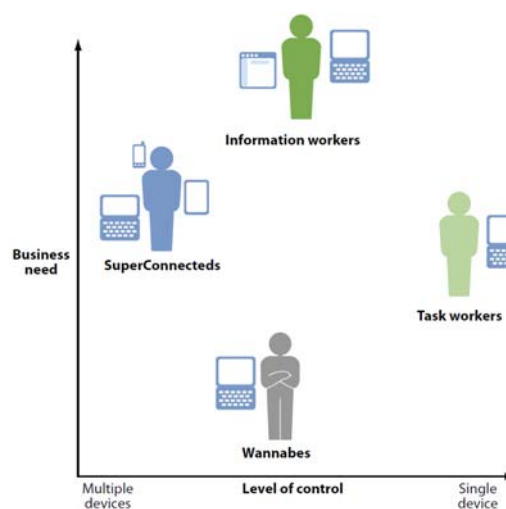
Wannabe medarbejder har ikke behov for en mobiltelefon i arbejdet. De bruger typisk deres personlig mobiltelefoner til at modtage personlige e-mails, notifikationer fra sociale medier som facebook på deres smartphones, de forstår dog ikke hvorfor virksomheden ikke giver dem samme mulighed, selvom der ikke er nogen forretningsmæssig begrundelse for at støtte disse ikke mobile medarbejdere.

Superkommunikator og kommunikator (superkommunikator: bruger deres telefon for at få information og udforske produkter, tjekke aktiekurser, kontrollere finansielle konti og er konstant i kontakt.)

Kommunikatoren er ikke en regelmæssig bruger af datatjenester, men er bekendt med beskeder (sms, mms og IM)

Jeg vil mene, at brugerne af wallboard mobil applikationen vil være **informative medarbejdere** samt **Superkommunikator og kommunikator**. Da der er to typer af bruger; agenterne som sidder ved telefonen og overvåger systemet samtidig med at de kigger på sager og tager imod henvendelser fra kunder og lederne, som har interesse i at se hvordan arbejdet fungerer i den afdeling og om kunderne får en god kunde betjening. De informative medarbejdere er agenterne/ supporterne, fordi de bruger tid på, at analysere informationer, som de ser på skærmen (mobil skærm), træffer beslutninger baseret på de oplysninger og data de kan se. F.eks. når de prioriterer hvilke kunder som skal betjenes først. De Superkommunikator er lederne som typisk er til møder og befinder sig forskellige steder, de har interesse i at holde øje med afdelingen samtidig med at de laver andre ting.

Nedenstående figur 20 illustrerer hvordan virksomheden ser på de forskellige typer medarbejdere og deres behov. Det ses at forretningsbehov er høj for informative medarbejdere og er meget lav for wannabes. Figuren illustrerer også hvilke enheder de forskellige typer medarbejdere har behov for i deres arbejde.



Figur 20 4 typer af smartphone bruger [8]

4.1.3 Brugerbehov

Man kan udvikle en smart og høj teknologisk produkt, men som ikke kan bruges til noget, fordi den ikke dækker brugernes behov, ønsker og mål. Derfor er det meget vigtigt at definere brugerbehov inden man begynder at designe og udvikle sit produkt. Når der tales om brugerbehov, er det oplagt, at se på de allerede eksisterende brugere, her menes der slutbrugere. De kan være medarbejdere i NetDesign eller en anden virksomhed, hvor wallboard desktop applikationen bliver anvendt. I den forrige afsnit personas, beskrev jeg brugerne vha. en persona beskrivelse. Det er vigtigt at man ikke antager, at man har forstået brugerne, for ved mangel reel information om brugere, vil udviklerne bruge fantasien til at beskrive, hvem de tror, at brugeren kunne være. Så de vil designe en applikation, som opfylder behovene for deres imaginære bruger, men ikke nødvendigvis behov for den rigtige bruger. Når man definere brugerbehov skal man ikke bare lytte og observere hvad kunderne siger, man skal prøve at sætte sig ind i kundens situation og finde ud af hvad de har brug for.

Steve Jobs¹⁷ sagde: "Man kan ikke bare spørge kunderne, hvad de ønsker, og derefter forsøger at give det til dem. I den tid man udviklet det, vil de have noget nyt".

Til opstilling af brugerbehovene, tog jeg udgangspunkt i nogle medarbejdere i NetDesign, som anvender wallboard desktop versionen samt udvikleren af den. Jeg interviewede 5 forskellige brugere og fandt ud af hvad de kunne tænke sig applikationen skal kunne. Inden interviewet havde jeg forberedt hvilke emner interviewet skal berøre. Jeg stillede åbne spørgsmål, for at få så mange informationer så muligt fra brugerne. Svarene noterede jeg på computeren. Her ses nogle af interview spørgsmålene¹⁸:

- *Hvordan anvender du wallboard applikationen?*
- *Hvad kunne du tænke dig mobilapplikationen skal have fokus på?*
- *Tror du at du vil bruge wallboard mobilapplikationen?*

Derfra vurderede jeg sammen med min vejleder Nino Martinez, hvilke elementer, der ville være relevant, at have med i wallboard mobil applikationen i sin 1.version.

Da applikationen er meget simpel og ikke skal kunne så meget andet end at vise de vigtigste data fra wallboard på en lille skærm, er der ikke mange behov. Herunder har jeg skrevet de væsentligste brugerbehov:

- *Skal vise de vigtigste data fra wallboard på en mobil, så man som bruger altid kan overvåge kaldssituationen og serviceniveauet uden nødvendigvis at befinde sig på kontoret.*

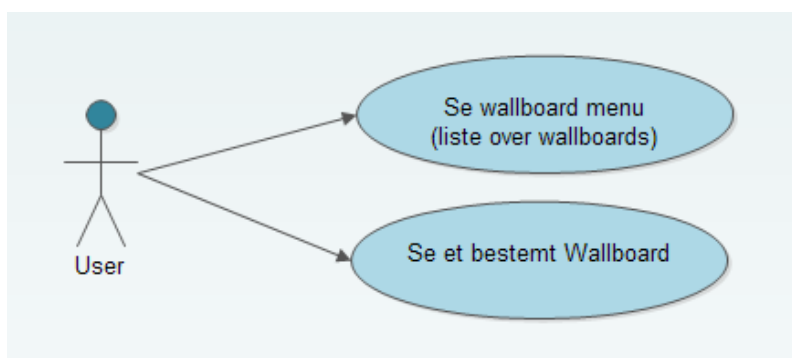
¹⁷ Steve Jobs var medstifter af Apple Inc.

¹⁸ Resten af interview spørgsmål findes i bilag 2 interview spørgsmål

- Følge samme guidelines som desktop versionen, så medarbejderne føler sig bekendt med applikationen.
- Wallboard skal vise realtime data hele tiden.
- Give notifikation, når der kommer fejl i systemet fx ved mange mistede kald eller mange i kø, så behøver medarbejderen ikke, at tjekke applikationen konstant, for at holde øje om der sker noget.¹⁹

4.1.4 Use case diagram

Use case diagrammer bruges til at illustrere de funktioner/ muligheder de forskellige aktører har i systemet. Jeg har lavet et simpelt usecase diagram, som kan ses herunder. Use case diagrammet illustrere de muligheder jeg forestiller mig en bruger af wallboard mobilapplikationen har. Som man kan se så har brugeren 2 muligheder. Den ene mulighed / use case er at brugeren kan se listen over wallboard i en menu. Den anden mulighed er at brugeren/aktøren kan vælge et bestemt wallboard, ved at vælge et wallboard kan brugeren de det ønskede wallboard.



Figur 21

4.1.5 Scenarier

Efter at have defineret brugerkaraktistika, skitseret persona og defineret simple brugerbehov, vil jeg uddybe det i en scenario beskrivelse. Et scenario er en konkret beskriver af en episode fra brugernes liv, som handler om brugen af det kommende produkt. Det skal beskrive hvordan brugerne kommer til at anvende det kommende produkt, eller hvordan de arbejder uden produktet. Jeg har lavet 2 forskellige scenarier, et scenarie hvor brugeren ikke har wallboard mobilapplikation og et tilsvarende scenarie, hvor brugeren har fået wallboard mobilapplikationen. Scenarierne tager udgangspunkt i min persona Michael Andersen.

¹⁹ Dette brugerbehov /ønske er kommet i slutningen af forløbet, der jeg var i gang med at lave brugervenlighedstest med brugerne.

Scenario før wallboard mobile applikationen:

Michael sidder på sit kontor, han er i gang med at kigge på en sag som Gentofte kommune har oprettet. De har problemer med at logge ind systemet ikke er ensrettet på alle deres server. Sagen har lav prioritet p4. Kl. er blevet 12 og nu har kantinen åben for frokost. Michael går op til kantinen sammen med sine medarbejdere fra Cisco support. De spiser deres frokost hurtigt, fordi der ikke er nogen som sidder på kontoret og holder øje med om der sker noget på wallboard. Efter de har spist frokost tager de tilbage på deres kontor, de kaster et blik oppe på wallboard skærmen og opdager at SKAT har oprettet en vigtig sag med prioritet P1, som udløber om 30 minutter. De bestemmer sig for at Michael tager sig af sagen, selvom han var i gang med Gentofte kommune, så går han i gang med SKAT, fordi de har fået et stort problem og han har kun en time til at løse det.

Scenario efter wallboard mobile applikationen:

Michael sidder på sit kontor, han sidder med en sag fra MEDCOM. De skal have hjælp til oprettelse af VPN. Det bliver frokost tid og Michael går op til kantinen sammen med 4 af sine medarbejdere fra Cisco support. Mens de sidder til frokost sidder de med deres smartphone på bordet, Michael tjekker wallboard applikationen et par gange under frokosten, han opdager, at SKAT har oprettet en kritisk sag som har høj prioritet (P1) og de har brug for hjælp hurtigst muligt. Han tager sin telefon og ringer til teknikker, som sidder i Skejby afdelingen og tildeler ham opgaven. Han oplyser blot sagsnr, som han finder på wallboard mobil applikationen. Michael fortsætter med at spise frokost mens han engang imellem tjekker wallboard applikationen på sin telefon.

4.1.5.1 Storyboard / sketch

Storyboarding er en slags low-fidelity prototype²⁰, som typisk laves i forbindelse med scenarier. Storyboard består af en serie af tegninger, som fortæller en kort historie om hvordan brugeren interagerer med det udviklede produkt. Storyboarding hjælper udviklerne samt kunden med at forstå hvordan produktet kommer til at blive brugt og i hvilke situationer vil applikationen blive brugt. Storyboard tager også udgangspunkt i min persona Michael Andersen.

²⁰ Interaction Design s. 531

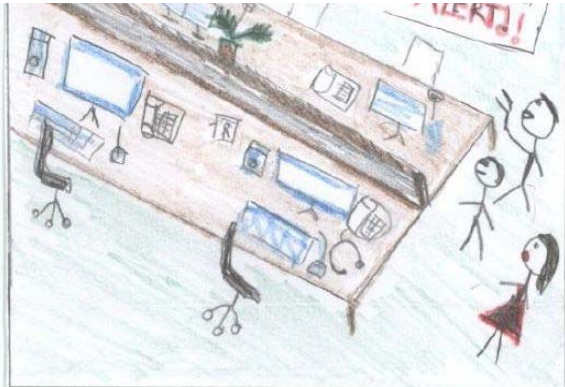
Storyboard 1 uden wallboard mobil applikation



Michael sidder på sit kontor og arbejder. Han er blevet sulten og det er frokost tid. Han går i kantinen med sin medarbejder.



Michael skynder sig med at spise, fordi han er bange for der kommer en vigtig sag på wallboard, mens han er i kantinen.



Michael tager tilbage på sit kontor og ser på wallboard. Han opdager at der er kommet en kritisk sag fra SKAT.

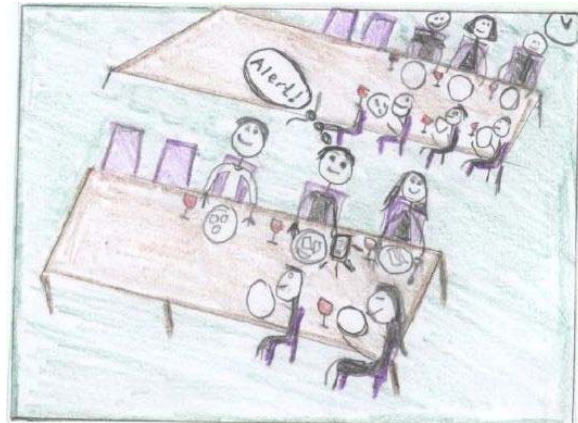


Michael går i gang med at kigge på sagen med det samme.

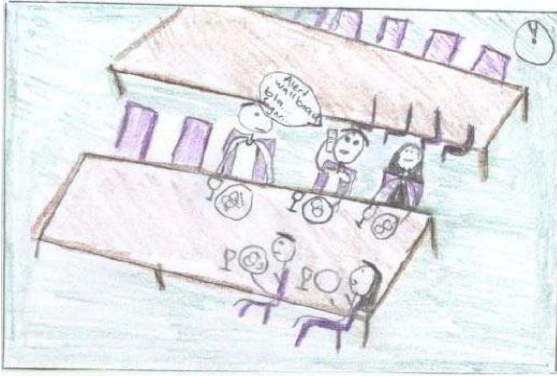
Storyboard 1 med wallboard mobil applikation



Igen sidder Michael på sit kontor og arbejder. Det er blevet frokost tid. Han spørger sin medarbejder om de vil med til kantinen og spise.

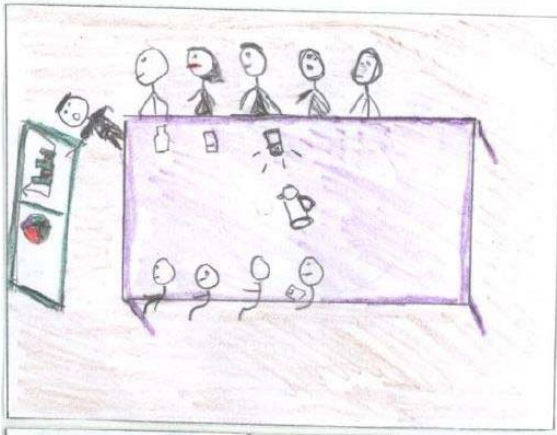


Mens Michael sidder og spiser i kantinen, tjekker han wallboard på sin HTC desire. Han kan se at der er kommet en sag op, som en tekniker fra Skejby har arbejdet på før.

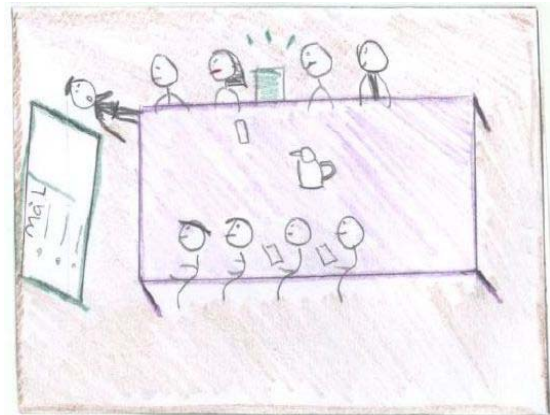


Han ringer til teknikeren og tildeler ham opgaven, ved blot at oplyse sagsnr, som kan ses på wallboard mobilen skabelonen.

Storyboard 2 med wallboard mobil applikation



Michael sidder til afdelingsmøde sammen med nogle andre medarbejder. Han holder samtidig øje med wallboard på mobilen engang i mellem. Han opdager at der er et problem/opgave.



Michael forlader mødet, for at løse problemet.

4.1.6 Kravspecifikation

Kravspecificering er en af de essentielle dele i udarbejdelsen af et design, når man arbejder med brugerorienteret udvikling. Disse krav er set fra brugernes side. Det vil altså sige, overordnede krav, som løser brugernes behov. Produktet/systemet skal leve op til brugerens forventninger og ønsker. Mange gange kan man lave et rigtig godt system, som ikke kan bruges til noget, fordi det ikke er noget som brugerne har behov for. For at definere kravene er det vigtigt at forstå så meget som muligt om brugerne, deres ønsker, deres behov og hvordan det udviklede produkt løser problemer i deres hverdag. Det er ikke altid at brugerne ved hvad de præcis vil, derfor skal udvikleren hjælpe brugeren med at definere hvad de har behov for. I de forrige afsnit har jeg defineret brugernes behov, udarbejdet nogle scenarier, usecase diagrammer og storyboard. Det har hjulpet mig med, at finde ud af, hvad brugerne vil og hvad systemet skal kunne for, at opfylde deres behov. Der findes to typer krav; funktionelle krav samt ikke funktionelle krav, herunder specificeres de.

4.1.6.1 Funktionelle krav

De funktionelle krav beskriver de funktioner, som produktet skal kunne. Wallboard mobilapplikationen er en simpel applikation, med få funktioner, men mange fordele.

1. Mobilapplikationen skal vise de wallboards, som bruger har brug for i en menu.
2. Wallboard skabelonerne skal vises på en fornuftig måde på telefonen, så brugerne ikke mister vigtige data som var på den originale wallboard desktop applikation.
3. Applikationen skal kunne åbne en hvilken som helst browser den sættes til at åbne, dvs. hvis brugeren ønsker at skippe menuen og ønsker at applikationen direkte skal linke til en bestemt wallboard skabelon, kan han/hun få mulighed for det. Enten via menu indstillingerne på applikationen eller udvikleren kan sørge for at modificere applikationen så den passer med det kunden vil have.

4.1.6.2 Ikke funktionelle krav

Ikke-funktionelle krav beskriver de forskellige begrænsninger, som sættes til løsningen og den måde løsningen kan bygges på. Formålet er at begrænse løsningsmulighederne, netop fordi der er mange måder at løse de funktionelle krav på. Herunder har jeg beskrevet de ikke funktionelle krav til wallboard mobil applikationen.

Brugervenlig:

- Let at lære at bruge.
- Let at bruge.
- Overskuelig menu struktur.
- Kan bruges uden fuld opmærksomhed.

- Genkendelig og logisk for brugeren.

Datakapacitet:

- Ingen begrænsninger på antallet af wallboard skærme, der kan vises på mobilapplikationen, ligesom på desktopapplikationen.

Datasikkerhed:

- Idet Wallboard mobilapplikationen er en slags webapplikation som henter sin data fra nettet er der ingen risiko for at data kan gå tabt på applikationen. Der er ingen grund til at have adgangskontrol på applikationen, da den ikke viser fortrolige data og brugerne kan ikke udfører handlinger med mobilapplikationen.
- Applikationen kan kun tilgås hvis telefonen har adgang til virksomhedens netværk.

Effektivt:

- Applikationen skal være hurtig til at svare tilbage. Når en bruger trykker på wallboard mobilapplikationen på sin smartphone skal han/hun ikke vente i lang tid, før applikationen åbner og viser menuen.

Pålidelighed:

- Applikationen skal være stabil. Hyppige fejl vil svække brugernes tillid til applikationen.
- Brugeren skal kunne stole på at de data, som wallboard mobilapplikationen viser.

Guidelines:

- Skal følge NetDesign's og wallboard design guidelines. Dvs. der skal bruges de samme farver, skrifttyper, som NetDesign anvender.
- Wallboard skabelonerne som er lavet til mobilen skal matche med skabelonerne som bliver vist på storskærmen.

4.2 Design

Dette afsnit beskrives designfasen og reviewet af design i samarbejde med brugerne.

Designfasen tager udgangspunkt i analysen af brugernes behov, krav og de teknologier som er blevet fremlagt. I Design fasen arbejdes der med at beskrive, hvordan applikationens brugergrænseflade kommer til at se ud og fungere. Det er derfor vigtigt at designet er velovervejet fra starten af, så man er klar over hvad der skal implementeres og hvordan. Her vil der fremgå, hvilke design metoder og teknikker som er blevet anvendt. Når designet er bestemt kan man gå i gang med implementering.

4.2.1 Design udfordring

Udfordringen ligger, hovedsageligt i smartphones skærmstørrelse, som formindsker de muligheder man har på en storskærm. Telefonens skærmstørrelse²¹ udgør ca. 8 % af en storskærm²², som hænger i afdelingen. Det kræves derfor detaljerede overvejelser angående skærmens udseende. Forskellige aspekter er derfor nødvendige at undersøge og tage højde for, inden man udvikler applikation. Man bliver nød til at begrænse design og informationer til det absolut nødvendige. Men det store spørgsmål ligger i hvem bestemmer, hvad der er nødvendigt og ikke nødvendigt? Det er primært brugerne, som bestemmer hvad de vil se på wallboard mobil skærmen og hvilke data som er vigtigt for dem at se når de ikke sidder på deres kontor, samt hvilke data der er mindre vigtigt. I de følgende afsnit vil man se, hvordan jeg har inddraget brugerne i at bestemme designet.

4.2.2 Brugervenlighed i wallboard mobilapplikation

På en applikation eller webside er det ikke nok med, at teknikken er smart og virker. Brugere skal også kunne finde ud af at bruge applikationen/websiden fornuftigt. Det er vigtigt, når man udvikler en applikation, at det er en applikation, som brugerne er glade for og kan finde ud af at bruge, med andre ord at applikationen er brugervenlig. Her kommer begrebet brugervenlighed (eller på engelsk usability) på spil.

Brugervenlighed sætter fokus på mange aspekter som: Tilgængeligheden, synlighed, effektivitet, ensartethed.

Tilgængelighed og synlighed indikere, hvor nemt det er for en bruger at navigere rundt i applikationen og rent faktisk bruge funktionerne. Det er ligeegyldigt, hvor flot applikationen er, hvis brugerne ikke kan finde ud af at bruge det optimalt. Det er derfor meget vigtigt, at have brugernes bedømmelse på applikationen.

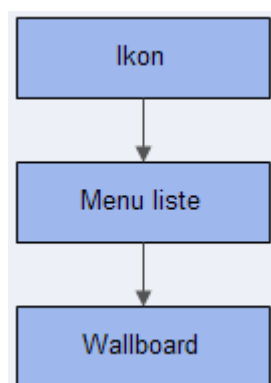
²¹ HTC Desire 3.7 tommers.

²² Ca. mellem 42 og 52 tommers.

Ensartethed forstærker koncepterne easy to learn og easy to use, hvilket er vigtig for brugervenlighed. Ensartet operationer gør det nemmere for brugerne at lære applikationen.

Noget af det væsentligste ved et brugervenligt design er, er effektiviteten og tidsbesparelsen for brugerne. Det spiller en afgørende rolle, hvor mange gange brugeren skal trykke og hvor mange menuer han/hun skal igennem inden de lander på dét, der reelt søges efter. Dette kaldes også for antal niveauer i designet. For mange niveauer gør, designet uoverskueligt, og det kan måske være svært for brugerne at huske, hvordan de kom frem til "et bestemt funktion". Derfor har jeg været opmærksom på at have så få niveauer så muligt. Da wallboard mobilapplikationen ikke er så kompleks og ikke indeholder mange funktioner, var det nemt at begrænse niveauerne. Wallboard brugerne skal trykke 2 gange inden de kommer frem til det wallboard de ønsker. Først trykkes på applikation ikonet i mobilen hovedskærm, når applikationen åbner kan man se menuen med alle wallboards, her klikker brugeren endnu en gang på det wallboard han eller hun vil se.

Figur 22 illustrerer niveauerne i applikationen.



Figur 22

4.2.3 Konceptuelt design

En prototype er en skitse af applikationen. Skitsering er en fremragende måde man kan fremvise idéerne på. Skitsering behøver ikke at være smarte tegninger, de kan være simple håndtegninger, som illustrerer de idéer og tanker man går med. Der er mange fordele ved skitsering. Det illustrer idéen for brugeren / kunden, hvilke gør man kan tage nogle design beslutninger tidligt. Man undgår mange design misforståelser, som kan opstå mellem kunden og udvikleren. Skitsering hjælper med at gøre kunderne mere opmærksomme på deres behov og ønsker. Man får brugeren til at udtale sig om forslag, da det er lettere for brugeren at forholde sig til tegninger end f.eks. til kravspecifikation. Alt i alt vil skitsering hjælpe med at give en bedre brugervenlighed. Det er heller ikke tidskrævende at skitsere.

Skitsering har hjulpet mig en del, idet jeg fik feedback fra brugerne på hvad de synes kunne være anderledes, hvad jeg evt. kunne tilføje i designet, hvilke farver, ikoner som skulle bruges og hvad der var irrelevant.

4.2.3.1 Skitseringer

Der findes som sagt forskellige typer wallboard skabeloner og man kan altid skræddersy nye skabeloner, som tilpasser kunden behov. Jeg har valgt, at skitsere nogle af de skabeloner, som allerede bruges i NetDesign, fordi de er nyttевærdige for de bruger, som kommer til at teste og vurdere dem.

Herunder ses 4 skærbilleder af wallboard desktop skabeloner, som jeg har valgt at tage udgangspunkt i. Udfordringen lå som sagt i hvordan designet skulle ændres således at jeg kunne få de vigtigste data og samtidig få designet til at se pænt ud på en mobil telefon.

Figur 23 illustrerer wallboard skabelon, som bruges til at holde øje med kaldsituationen. Som man kan se, er designet på denne skabelon alt for bred til at være på en mobilskærm selvom man vender den på vandrette side, men informationerne på selve skabelonen er ikke så mange. Derfor har jeg tænkt mig at beholde de samme informationer men ændre i designet således at den bliver mere aflang og passer med en mobilskærms højde og bredde.

| Test15 | | | | |
|---------------|-------|---------|------------|---------|
| 0 | | 0:00:00 | | |
| Logged in | Ready | Talking | Not Ready | Working |
| 0 | 0 | 0 | 0 | 0 |
| Today's calls | | Handled | Percentage | |
| 0 | | 0 | 0% | |

2012-1-12 19:29:51
Powered by NetDesign

Figur 23

Figur 24 var min 1. bud på hvordan denne skabelon kunne se ud på en mobil.

A hand-drawn sketch of a mobile wallboard layout. The sketch is titled 'Test' and contains the following information:

- I ke: 5
- Løgstid i ke: 05:03
- Logged ind klar later ikke klar: 6 3 2 1
- venter talkald Handlet: 0 10 8
- Procent: 80%

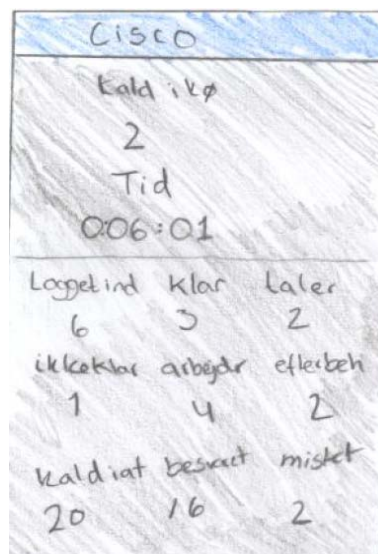
Figur 24

Figur 25 bruges også til at holde øje med kaldsituationen og historikken. Designet fylder både i længden og bredden og der er en masse informationer og data. Jeg interviewet et par brugere af denne applikation og spurgt dem, hvilke data behøver de mest, når de ikke befinder sig på kontoret. Det viste sig at de kun behøvede ca. halvdelen af informationerne som antal i kø, længst tid i kø, antal, klar, taler, ikke klar etc. Især informationerne som står under displayet i dag var ikke nødvendige. Ud fra det kunne jeg designe en skabelon som indeholder de vigtigste oplysninger samt har et design som passer med mobilskærm.



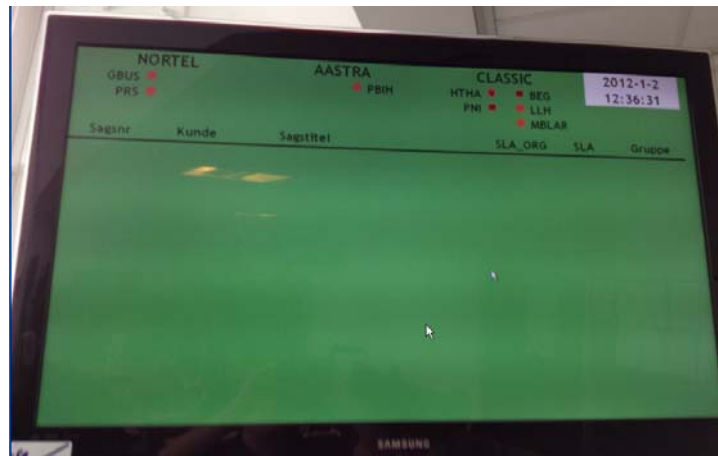
Figur 25

Figur 26 illustrere, hvordan denne skabelon kunne se ud på en mobil, som man kan se så er det ikke informationerne jeg har taget med.



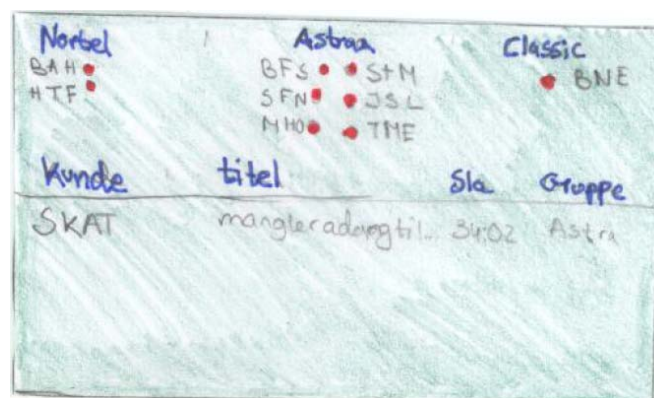
Figur 26

Skabelonen som ses på figur 27 bruges til at holde øje med Remedy sagerne, som kommer i systemet og som har brug for support. Der findes 3 grupper: Nortel, AAstra og Classic. Under hvert gruppe ses de medarbejdere som er på og deres status. Derunder ses oplysninger fra Remedy som; sagsnr, kunde, titel, sla_org, sla og gruppe. Dette design fylder mest i bredden, især pga. sags oplysningerne, derfor har jeg tænkt mig at designe denne skabelon vandret, således at brugerne skal vende telefonen vandret når de skal se den.



Figur 27 Wallboard koordination

Figur 28 illustrere mit bud på hvordan denne wallboard koordination kunne se ud på en mobil, den ligner den oprindelig desktop skabelon, forskellen er kun at jeg har fjernet Sla_org dato og klokkeslæt.



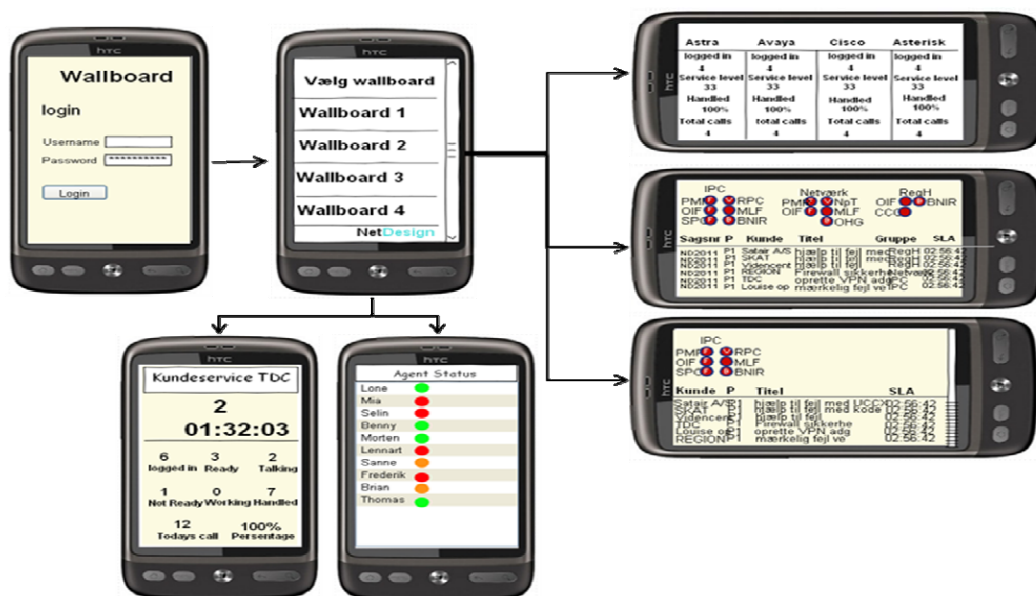
Figur 28 prototype wallboard koordination

4.2.3.2 Prototype design

Nedenstående prototype skitsering, var mit andet bud til brugerne af hvordan jeg har tænkt mig at designe wallboard mobil applikationen. Her har jeg lagt skærbillederne ind i en HTC smartphone så brugerne kan få en illustration af, hvordan det kommer til at se ud i virkeligheden. Designet af prototyperne har jeg lavet i animationssoftwaret Pencil²³.

²³ <http://www.pencil-animation.org/>

Når applikationen starter bliver brugeren bedt om at logge ind ved at skrive sit brugernavn og password og trykke på login knappen ligesom på wallboard desktop applikationen. Når brugeren er logget ind vises en menu, med wallboards, hvilke er lidt magen til menuen Available wallboard²⁴ fra wallboard desktop applikationen. Jeg har valgt at NetDesign logo skal ses nederst i menuen for at skabe en association til NetDesign. Fra menu siden kan brugerne trykke på den wallboard de vil se, vha. tilbage knappen kan de gå tilbage til menuen og vælge et nyt wallboard.



Figur 29 Prototype

Designet er blevet reviewet af 2 brugere af wallboard desktop fra NetDesign samt min vejleder fra NetDesign Nino Martinez. De kom med forskellige kommentarer og feedback. De var alle 3 enige om, at login siden ikke var nødvendig, og skulle skippes, idet der ikke er nogen fortrolige data, som ikke må ses af uvedkommende. Der kan heller ikke modificeres på data. Det vil også være enklere, hvis man uden at logge ind direkte kan vælge den wallboard man ønsker. Desuden virker applikationen kun på mobiler som er blevet godkendt til at være på virksomhedens netværk. Brugerne synes også at sags nr. mangler i nedenstående wallboard. De oplyser at det er vigtigt for dem at have sags nr. hvis de skal ringe til en tekniker og tildele ham/hende opgaver. Det er svært at slå op på kundens navn, derfor vil det være en fordel hvis sags nr. blev tilføjet til venstre for kunde navn.



Figur 30 prototype wallboard cisco IPC

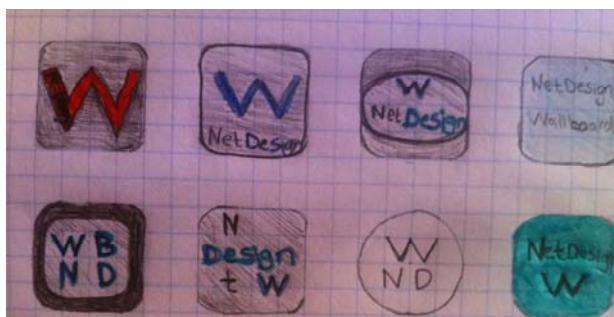
²⁴ Se afsnit 1.4.2 "Available wallboard menu".

Brugerne vil gerne have at wallboard mobil designet er magen til den som de ser på storskærm, derfor skal der bruges samme farver og skrifttyper.

4.2.3.3 Ikon

Når man søger på applikationer inde på applikation markedet (Appstore, android market etc.), er det ikonet man først får øje på. Ikonet er det samme som et bogomslag. Ikonet er det første indtryk man får af applikationen. Hvis ikonet er sigende samt interessant er sandsynligheden for at brugeren trykker på ikonet for at læse om applikationen større og dermed er chancen for at applikationen bliver downloadet / købt også større. Smartphone bruger har typisk mange applikationer liggende på deres skærm, derfor skal applikationens ikon kunne adskille sig fra de andre ikoner så brugeren ikke bliver forvirret eller bruger lang tid på at finde applikationen.

Ikonet skal sige noget om wallboard. Det skal stadig have en relation sig til NetDesign, idet det er et NetDesign produkt. Man kan relatere applikationen til NetDesign ved at bruge NetDesign's logo samt NetDesign's design guideline, som er de farver og skrifttyper, som bliver anvendt i NetDesign, bla. på deres hjemmeside²⁵. Herunder ses prototype skitseringer af, hvordan ikonet kan se ud. Jeg har primært fokuseret på farverne lyseblå og sort, som er NetDesign's farver. Jeg har anvendt W og WB, som forkortelse på wallboard og ND for NetDesign.



Figur 31 prototype ikoner

Efter at snakke med forskellige personer fra NetDesign og udenfor NetDesign om hvad de synes om ikon skitseringerne. Kom jeg frem til at nedenstående ikon var den mest simple, enkle og samtidig sigende ikon. **W** er en forkortelse for wallboard, skrevet med en lyseblå farve, baggrunden er sort gradient og i bunden er der et NetDesign logo, hvor Net er skrevet med hvid og Design med lyseblå ligesom **W**.

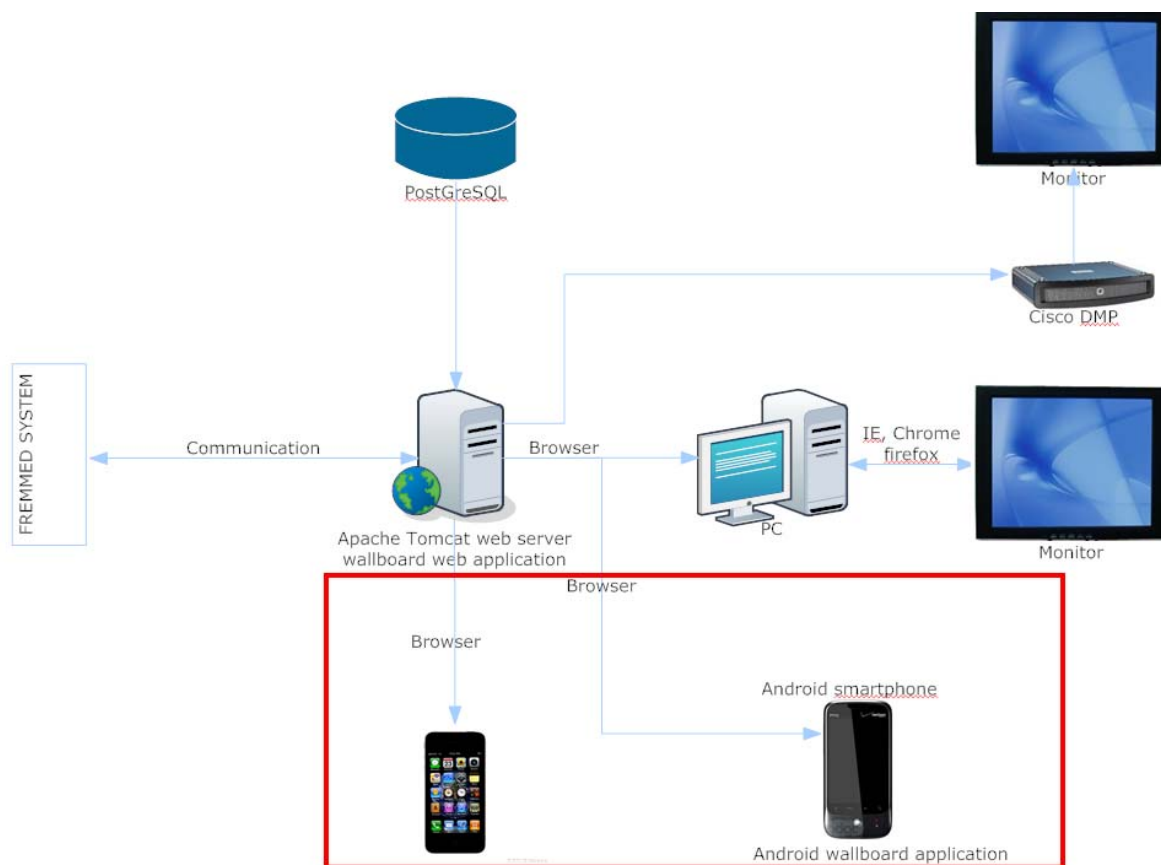


Figur 32 prototype valgte ikon

²⁵ <http://www.netdesign.dk/>

4.2.4 Overview Wallboard System med mobil

Essensen i projektet er at tilføje en mobilapplikation til wallboard systemet. Så brugerne kan tilgå wallboard systemet via deres smartphones. Applikationen vil kunne tilgås på to forskellige måder. Hvis man har en smartphone, som kører Android kan man installere applikationen på sin telefon. Idet jeg har lavet en Android webviewer applikation, som fungerer som en "skal", så man ikke kan se at det er en webside, som vieweren åbner, men i stedet ser det ud som om det er en ganske almindelig mobil applikation. Man kan også tilgå applikationen ved hjælp af mobilbrowseren ved at skrive webapplikationens url adressen i mobilbrowseren. Det vil selvfølgelig ikke give den samme brugeroplevelse, idet brugeren selv skal skrive url adressen i stedet for bare at røre applikationen ikon på telefonens skærm. Det eneste krav er at telefonerne har adgang til NetDesign's netværk.



Figur 33 Overview wallboard system med mobilversionen

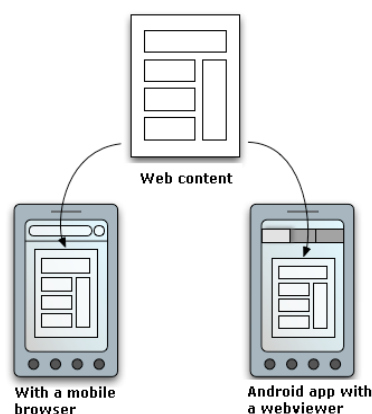
4.3 Implementering

Implementeringsfasen tager udgangspunkt i de design beslutninger som er blevet truffet i designfasen, hvor jeg og brugerne blev enige om hvordan produktet kommer til at se ud. Kildekoden findes i det vedhæftede Eclipse projekt, på den vedlagte cd samt i bilag 3. Jeg har inddraget noget af koden i dette afsnit for at det skal være mere læsevenligt og forståeligt. Jeg har brugt forskellige teknologier til at udvikle applikationen. Selve wallboard mobil applikationen er udviklet i Eclipse²⁶ med Android SDK og skrevet i programmeringssproget Java. Jeg har brugt CSS scripts til at designe wallboard skabeloner, som passer til smartphones skærmstørrelse. Applikationens hovedmenu er implementeret vha. JQuery. I de næste afsnit vil jeg beskrive de forskellige teknologier jeg har anvendt samt hvordan jeg har anvendt dem.

4.3.1 Webviewer

Jeg har tidligere diskuteret fordelene ved at lave webbaserede applikation frem for native applikation. I dette afsnit vil jeg beskrive, hvordan jeg har lavet en Android WebView²⁷ applikation, som er det samme som android site specifik browser. Webvieweren kan vise, en hvilken som helst webapplikation, den konfigureres til at åbne uden at man kan se at det er en webapplikation, som vieweren åbner.

Webindholdet kan leveres på to forskellige måder. Den ene måde, hvor det er en applikation, udviklet ved hjælp af Android SDK også installeres den på brugerens enheder, som APK. Den anden måde sker via en mobilbrowser, hvor brugerne ikke skal installere noget, men kun skal skrive url adressen i mobilbrowseren. Figur 34 illustrerer at man kan gøre webindholdet tilgængeligt for brugere på de to måder. Den traditionelle mobile browser og i en Android applikation ved at bruge WebView.



Figur 34 Webapp overview [19]

²⁶ Eclipse er et udviklingsmiljø, kan hentes fra <http://www.eclipse.org/downloads/>

²⁷ Inspiration til webviewer applikation fra: Se litteraturliste [25]

6.3.1.2 Android webviewer applikation

Jeg startede med at sætte Android SDK op i udviklingsmiljøet Eclipse, dernæst oprettede jeg en Android projekt og derefter gik jeg i gang med at udvikle webvieweren. Med 3 steps kunne jeg udvikle en simpel applikation, som viser indhold fra en webside. Dernæst tilføjede jeg forbedringer til applikationen såsom hvordan applikationen skal reagere når brugeren trykker på tilbage knappen. Herunder kommer en beskrivelse af de 3 steps.

Step 1: Det første man starter med når man skal udvikle en Android Webview applikation er at sætte Webview widget op. Dette gøres i main.xml. I denne fil kan man tilføje forskellige grafiske elementer til brugergrænsefladen, som f.eks. applikationens højde og bredde, man kan også tilføje forskellige ikoner og knapper. Main.xml har to visninger XML visning, hvor man direkte kan skrive XML kode og en grafisk visning²⁸, hvor man kan tilføje forskellige elementer til brugergrænsefladen vha. "drag and drop".

Herunder ses main.xml koden. Under LinearLayout element, har jeg tilføjet en webviewer i stedet for TextView. I lineære layout, definere jeg bredden og højden til at være fill_parent, som betyder at skærmen på den enhed der kører applikationen fyldes ud.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <WebView
        android:id="@+id/webview"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent" />

</LinearLayout>
```

Step 2: AndroidManifest.xml indeholder væsentlige oplysninger om applikationen til Android systemet. Oplysninger som systemet skal have, før det kan køre applikationen. Idet applikationen skal kunne indlæse en URL i Webview widget, skal man sørge for at anmode om internet tilladelse i AndroidManifest.xml filen. Dette gøres ved hjælp af nedenstående linje:

```
<uses-permission android:name="android.permission.INTERNET" />
```

Applikationen indeholder en enkel aktivitet, med klassenavn WallboardMobil. Når @-symbolet anvendes i AndroidManifest.xml filen refereres der til oplysninger som er gemt i ressource filer.

²⁸ Se bilag 4 figur 1 grafisk visning af main.xml

En vigtig del af android udvikling er Intent og Intent-filter. De bruges i Android udvikling til at beskrive hvad man vil gøre. Herunder beskriver Intent-filter at applikationen skal kunne køres.

```
<application
  android:debuggable="true"
  android:icon="@drawable/wallboard_mobil"
  android:label="Wallboard" >
  <activity
    android:label="Wallboard"
    android:name=".WallboardMobil"
    android:theme="@android:style/Theme.NoTitleBar"
  >
  <intent-filter >
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>
</application>
```

Step 3: Nu mangler vi kun at loade websiden. Dette gøres i applikationens Java klassen WallboardMobil, hvor aktiviteten bliver deklareret.

Nedenstående kode, er en simpel kode, som jeg har brugt til at åbne hjemmesiden www.cn.dtu.dk i webvieweren.

```
public class WallboardMobil extends Activity {
  private static final String MAIN_PAGE = "http://www.cn.dtu.dk";
  WebView webview;
  /** Called when the activity is first created. */
  @Override
  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    webview = (WebView) findViewById(R.id.webview);
    webview.loadUrl(MAIN_PAGE);
    webview.setWebViewClient(new WebViewClient());
  }
}
```

Først startede jeg med at deklarere WebView i toppen af klassen. Inde i onCreate() metoden loader jeg url'en i webvieweren. onCreate() metoden bliver kaldt, når aktiviteten er begyndt. Nu kan man køre applikationen i Android AVD emulator eller på Android telefon, applikationen vil vise campusnet siden²⁹.

Android aktivitet er relateret til brugergrænsefladen, oftest er der én-til-én forhold mellem aktivitet og UI. Aktiviteten anvendes til at præsentere konkrete elementer i brugergrænsefladen til brugeren. Android-applikationer indeholder oftest mere end én aktivitet. Hver aktivitet viser et UI og reagerer på system og bruger hændelser. En af de primære opgaver en aktivitet udfører, er at vise UI elementer, der er implementeret som Views og er typisk defineret i XML-layout-filer.

²⁹ www.cn.dtu.dk

Sådan ser det ud når man kører applikationen på en Android telefon. Man kan se at skærmen er for lille til layoutet til Campusnet. Campusnet hjemmesiden er ikke designet til at vises på en mobil som f.eks. andre sider som facebook og BT.³⁰



Figur 35 Campusnet i WebView applikation

Ligesom i step 3 starter jeg med at importere de biblioteker, der er behov for. Dernæst erklæres klassen som extender superklassen Activity. Der starter jeg med at definere main page, som er hovedsiden til wallboard mobil applikationen. Jeg opretter også en instans af WebView, som jeg kalder webview.

```
package com.example.wallboard;
import android.app.Activity;
import android.os.Bundle;
import android.view.KeyEvent;
import android.webkit.WebView;
import android.webkit.WebViewClient;

public class WallboardMobil extends Activity {
    private static final String MAIN_PAGE = "http://172.23.1.188:8080/wallboard-
web/mobilUI2.html";

    WebView webview;
```

onCreate(Bundle savedInstanceState) er, hvor jeg initialisere min aktivitet. Her kalder jeg setContentView som definerer min UI. findViewById bruges til at hente widget i den UI, som jeg interagerer med via applikationen.

Dernæst bestemmer jeg, at der hverken skal vises lodrette eller vandrette scrollbar ved at kalde webview.setVerticalScrollBarEnabled(false) og webview.setHorizontalScrollBarEnabled(false). Dette gør at jeg sparer plads på skærmen.

³⁰ Se bilag 5 figur 1 hvor jeg har testet webvieweren med BT.

Her undersøges `savedInstanceState`, hvis den er forskellig fra null, hentes den sidste `savedInstanceState`, som blev gemt sidste gang metoden blev kørt. Hvis `savedInstanceState` er lig med null, hvilket den er første gang man kører applikationen bliver siden loadet i webvieweren.

```
/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    webView = (WebView) findViewById(R.id.webview);
    webView.getSettings().setJavaScriptEnabled(true);
    webView.setVerticalScrollBarEnabled(false);
    webView.setHorizontalScrollBarEnabled(false);
    if (savedInstanceState != null) {
        webView.restoreState(savedInstanceState);
    } else {
        webView.loadUrl(MAIN_PAGE);

        webView.setWebViewClient(new WebViewClient());
    }
}
```

```
@Override
protected void onSaveInstanceState(Bundle outState) {
    webView.saveState(outState);
}
```

Herunder ses `onKeyDown` metoden. Metoden bruges til at bestemme hvad der skal ske når brugeren trykker på tilbage knappen. Når man har trykket på tilbage knappen kommer brugeren tilbage til den side han/hun var på før. Dvs. hvis brugeren er på en af wallboard siderne og trykker på tilbage knappen går applikationen tilbage til siden før, som er hovedmenuen.

```
@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
    if ((keyCode == KeyEvent.KEYCODE_BACK) && webView.canGoBack()) {
        webView.goBack();
        webView.setWebViewClient(new WebViewClient());

        return true;
    }
    return super.onKeyDown(keyCode, event);
}
```

6.3.1.3 Wallboard mobil ikon

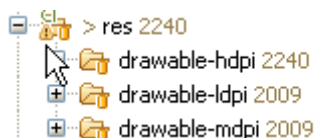
Ikonet er grafiske elementer der er placeret i hovedmenuen som vises til brugerne. Når brugerne trykker på ikonet åbnes wallboard applikationen. I design afsnittet besluttede jeg mig for hvordan ikonet skal se ud, ved at lave forskellige ikon tegninger og vælge den som passer bedst.

Jeg startede med at designe ikonet i Photoshop. Jeg har anvendt NetDesign's ikon samt en sort gradient baggrund. Ikonet skal have filtypen PNG. I Android skal ikoner oprettes i 3 forskellige skærmopløsninger.

- Low density ldpi: 36x36

- Medium density mdpi: 48×48
- High density hdpi: 72×72

Dette gør at ikoner og billeder ser pænt ud lige meget hvilke enhed man bruger. Android finder selv ud af hvilket billede, som skal bruges alt efter hvilke enhed applikationen køres på. Når jeg har ikonet i de 3 opløsninger, tilføjer jeg det til mit Android projekt. Dette gør jeg ved først at tilføje ikonet til mappen /res/drawable. Som man kan se på figur 36 så har hver opløsning en mappe; drawable-hdpi, drawable-ldpi og drawable-mdpi.

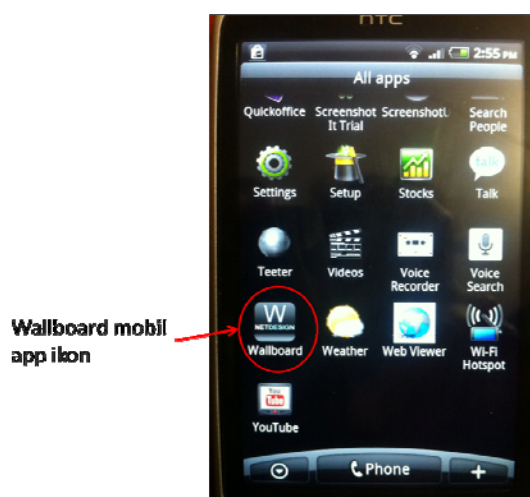


Figur 36

Dernæst bestemmer jeg i AndroidManifest.xml at ikonet som jeg tilføjede i mappen /res/drawable skal være applikationen hoved ikon. Dette gøres ved at sige:

```
android:icon="@drawable/wallboard_mobil"
```

Sådan ser ikonet ud på Android skærmen:



Figur 37 Wallboard icon på HTC desire

4.3.2 JQuery

Til design af brugergrænsefladen /menu siden har jeg anvendt en teknologi, som hedder jQuery³¹. jQuery går ud på at skabe en elegant HTML- baseret brugergrænseflade, som virker på alle de populære mobil platforme. Det virker ikke kun på smartphones, men kan også bruges på tablet og desktop platforme, hvilket gør teknologien meget populær. Alle sider i jQuery mobil er bygget således at den sikre

³¹ Du kan læse mere på <http://jquerymobile.com/demos/1.0/docs/about/getting-started.html>

kompatibilitet på stort set alle web enheder. Dette giver brugeren en god bruger oplevelse, lige meget hvilke platform han/hun anvender. Teknologien gør det nemmere for folk, som ikke har store udviklings kompetencer at få lavet et robust og flot system, vha. lidt HTML-kodning. Man sparer en del ressourcer, tid og man får en bedre kvalitet.

4.3.2.1 Procedure jQuery:

jQuery mobil indeholder forskellige brugergrænseflade skabeloner. Jeg valgte en UI skabelon og kopierede den ind i en teksteditor. Herfra begyndte jeg med at modificere i HTML koden, ved at tilføje nye elementer og slette unyttige dele, så den menuen passer med det jeg vil lave.

4.3.2.1.1 HTML kode

jQuery IU skabelonen består af 2 dele; header og body. Herunder har jeg beskrevet de 2 dele og hvad jeg har ændret i dem.

I **header**, sættes meta viewport tag skærmens bredde til pixel bredden af enheden. Det er en fordel idet layoutet ikke bliver påvirket af skærmens størrelse. Herunder tilføjes også henvisninger til jQuery mobil og tema stylesheet som er blevet valgt.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Wellcome to Wallboard</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="http://code.jquery.com/mobile/1.0/jquery.mobile-1.0.min.css" />
    <script type="text/javascript" src="http://code.jquery.com/jquery-1.6.4.min.js"></script>
    <script type="text/javascript" src="http://code.jquery.com/mobile/1.0/jquery.mobile-1.0.min.js"></script>
  </head>
```

I **body** er hver visning eller side på mobil enheden identificeret med et element (som regel en div). Her opdeles siden i forskellige dele. Første del identificeres med **data-role="header"** attribut, som indikerer at det er sidens header. Herunder bestemmes også overskriften dvs. headeren og baggrunden. jeg har givet headeren overskriften "wallboard". Til baggrunden har jeg anvendt `-moz-linear-gradient(#87ADA2, #87ADA2)`, som også bruges på NetDesign's hjemmeside.

```
</head>
<body>
  <div data-role="page">
    <div data-role="header"
      style="background-image: -moz-linear-gradient(#87ADA2, #87ADA2);">
      <h1>Wallboard</h1>
    </div>
    <!-- /header -->
```

Dernæst identificeres en **data-role="listview"** attribut, som indeholder linket liste elementer. Herunder kan man indsætte de wallboard skabeloner man vil, ved at linke til dem og give dem en overskrift, som bliver

vist på brugergrænseflade. Med jQuery mobil får man en brugervenlig listevisioning med højre pil indikator, som fylder bredden af browservinduet. Når man trykker på et element i listen vil man blive først videre til siden bag elementet.

```
<ul data-role="listview" data-inset="true" data-filter="false">
  <li><a class="availableWallboard"
    href="http://nd-wallboard1:8080/wallboard-web/viewer/ciscoSupportNetvaerk_shayma">CiscoSupport Netvaerk</a>
  </li>
  <li><a
    href="http://nd-wallboard1:8080/wallboard-web/viewer/ciscoSuppportSikkerhed_shayma?11">CiscoSupport Sikkerhed</a>
  </li>
  <li><a
    href="http://nd-wallboard1:8080/wallboard-web/viewer/shaymaa?12">CiscoSupport IPC</a>
  </li>
  <li><a
    href="http://nd-wallboard1:8080/wallboard-web/viewer/NywallboardShayma">Support_TDC</a>
  </li>
  <li><a
    href="http://nd-wallboard1:8080/wallboard-web/viewer/NykopiKoe">SikkerhedsSupport</a>
  </li>
</ul>
```

Footer har næsten den samme struktur som headeren. Her vælger jeg samme baggrund som i headeren. I bunden til højre tilføjer jeg NetDesign ikon, for at skabe association til NetDesign.

```
<div data-role="footer"
  style="background-image: -moz-linear-gradient(#87ADA2, #87ADA2);">
  <h4>
    
  </h4>
</div>
<!-- /footer -->
</div>
<!-- /page -->
```

Sådan ser brugergrænsefladen ud når den køres med AVD emulator:



Figur 38 brugergrænseflade testet med AVD

Efter at modificere på jQuery skabelonen, gemte jeg den som en HTML og lagde den op i NetDesign's Netværk system.

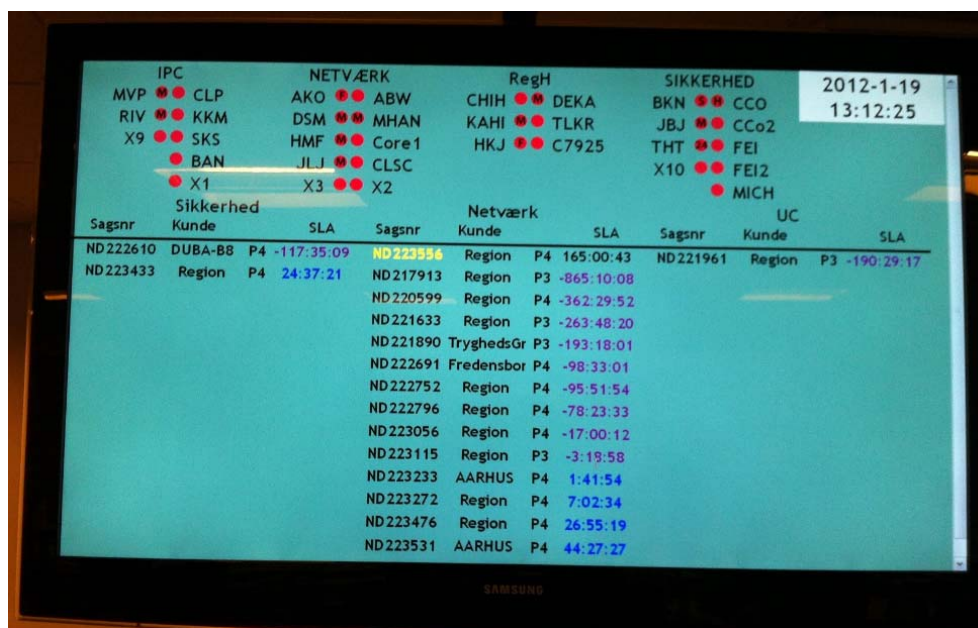
4.3.3 Wallboard skabeloner og CSS

I afsnit 1.4.1 "Designer menu" har jeg beskrevet, hvordan man opretter et nyt wallboard. Udfordringen ligger i at designe wallboard skabeloner, som passer til en mobiltelefon og stadig indeholde de oplysninger som brugeren har brug for at se. Wallboard plejer normalt at køre på en storskærm, som fylder mellem 23 og 52 tommers, for at få informationer på sådan en storskærm ned på en mobilskærm, der er knap 3.7 tommers kræver efter tanke og fantasi.

I dette afsnit vil jeg tage udgangspunkt i et af de wallboard skabeloner jeg redesignet til en mobilskærm vha. CSS³² og wallboard desktop applikation.

4.3.3.1 Cisco Support Wallboard

Cisco³³ support er en afdeling i NetDesign som er delt op i 4 underafdelinger: IPC, Netværk, RegH og Sikkerhed. Herunder ses wallboard skabelonen som afdelingen har hængende på en storskærm midt i deres kontor. Øverst på wallboard kan man se de forskellige afdelinger samt agenterne og deres status. Agenterne er delt op i øst og vest. De agenter som sidder i NetDesign Farum er øst, er placeret på højreside og de agenter som sidder i Skejby (vest), er placeret på venstreside.



Figur 39 cisco support wallboard

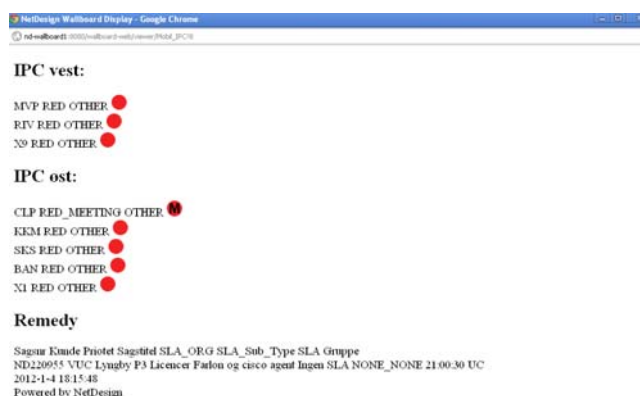
Da der er mange informationer på denne wallboard, som umuligt vil komme til at se pænt ud på en mobilskærm, valgte jeg at opdele cisco support wallboard op i 4 forskellige wallboards, en for hver underafdeling. Medarbejderne i IPC har ikke brug for at se de sager, som har brug for support fra Netværk,

³² Cascading Style Sheets

³³ Man kan læse mere om Cisco på: <http://www.cisco.com/web/DK/index.html>

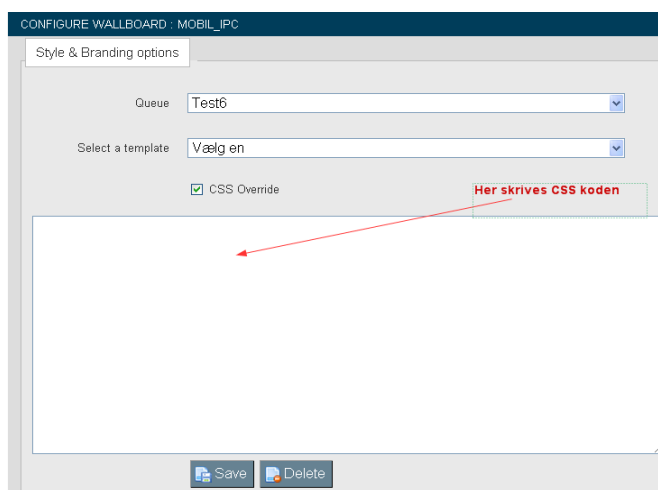
RegH eller sikkerhed. Derfor vil brugeren af denne wallboard ikke miste information ved denne opdeling. Når wallboard deles op i de forskellige afdelinger, er feltet gruppe unødvendig, fordi wallboard for den bestemte gruppe kun vil indeholder sager som er tildelt denne gruppe.

Jeg starter med at oprette et nyt wallboard i wallboard desktop applikationerne og oprette de forskellige displays, som der er behov for. Til dette wallboard skabelon er der behov for 3 displays i alt. To displays med agentstatus, en for IPC øst og en for IPC vest. Det sidste display består af Remedy sager. Når jeg har oprettet displays og tilføjet indhold til displayene ser wallboard således ud. Bemærk at der ikke er noget stylesheet.



Figur 40 wallboard uden stylesheet

Det er svært at læse hvad der står, især under Remedy og det ser heller ikke pænt ud. Derfor har jeg tilføjet stylesheet til dette design inde i wallboard designer siden. Herunder ses designer siden, hvor man kan konfigurere wallboard. I den hvide boks skrives CSS koden.



Figur 41 designer side

Herunder ses noget af CSS koden for denne wallboard. Resten af koden samt koden for de andre wallboard kan ses i bilag 3 Wallboard mobil skabeloner.

I HTML body bestemmes generelle ting som højde, bredde, tekststørrelse skrifttype og m.m. Det man skriver i body rammer alle elementer. Dvs. Hvis jeg vælger en bestemt skrifttype I body, vil alt tekst på siden have den skrifttype medmindre jeg senere overskriver den, ved at pege på komponenten og give den en anden skrifttype.

```
html, body {
    background-attachment: scroll;
    background-color: #60A08A;
    font-family: Trebuchet MS;
    text-align: center;
    font-size: 13px;
    font-weight: bold;
    height: 98%;
}
```

Herunder peger jeg direkte på de runde cirkler, som står ved agenternes navn og bestemmer højden og bredden på dem samt margin.

```
div.uccx_exchange_user img.displayAgentStatusImg {
    height: 10px;
    width: 10px;
    margin-bottom: 0;
    margin-right: 0.3em;
    margin-top: 0;
}
```

For at pege på de forskellige elementer og vide hvad jeg præcis skal skrive for at ramme dem har jeg brugt Firebug³⁴ i firefox samt Chrome, ved at højreklikker på elementet og klikke på *inspect element* eller *vis detaljer om elementet*.

De elementer som jeg ikke vil visse i wallboard, såsom Remedy gruppe, powered by ikonet fjerner jeg ved at pege på dem og skrive display:none. Herunder ses hvordan jeg har skjult Remedy gruppen.

```
div.remedyItem span.remedyGroup {
    display: none;
}
```

I Remedy displayet skal alle felterne modificeres således, at der er nok plads til dem og de informationer som står under dem. Derfor har jeg prøvet at tilpasse en fornuftig stylesheet til dem. Herunder ses stylesheet for sagsnr, hvor jeg har bestemt fontstørrelsen og margin.

```
span.remedyId {
    font-size: 0.8em;
    margin-left: 0.1em;
    text-align: left;
    width: 5em;
    white-space: nowrap;
}
```

Jeg har som sagt ikke taget alt koden her, men kun dele af koden idet det vil fylde en del. Resten kan man læse i bilag 3 wallboard mobil skabeloner. Herunder ses wallboard mobil cisco IPC, når den køres med AVD emulator.

³⁴ Her kan man læse nærmere om firebug <http://getfirebug.com/whatisfirebug>



Figur 42

4.3.4 Fremtidige funktioner

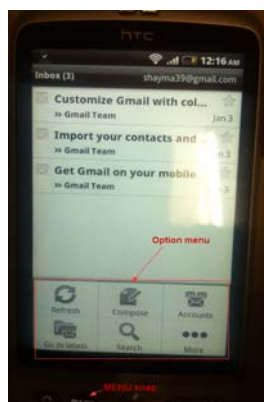
Under dette afsnit vil jeg gennemgå de elementer, som ikke er blevet implementeret, men som kunne implementeres i fremtiden. Der vil blive argumenteret for hvorfor de ikke er blevet implementeret og jeg vil give et forslag til hvordan man evt. kunne implementere det.

4.3.4.1 Notifikation/ eventhandler

Notifikation funktionen er ikke blevet implementeret eftersom denne idé, kom sent i projektforløbet, hvor der ikke var tid til at kigge på det. Notifikation kunne bruges til at give alarm til brugeren, når der sker noget vigtigt på wallboard. Denne alarm kan være lyd eller vibration samt en besked som vises oppe i status baren. F.eks. kan brugeren få en notifikation, når SLA er på en time, eller hvis der er over 4 mistede kald. Dette gør at brugeren ikke behøver at holde øje med wallboard applikationen hele tiden. Notifikation bør skræddersyes efter kundens behov.

4.3.4.2 Android Option menu

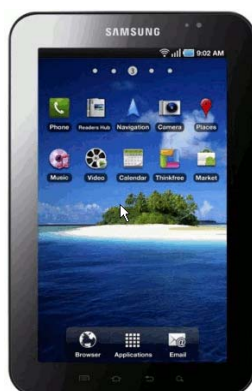
På nogle Android applikationer er der options menuer, som indeholder forskellige ting. Option menuen kan ses når brugeren trykker på menu knappen på Android telefonen mens han er inde i applikationen. Udvikleren af applikationen kan selv bestemme hvad option menuen skal indeholde. Herunder ses et eksempel af option menuen for Gmail:



På nuværende tidspunkt har det ikke været noget krav eller et behov at have en option menu til wallboard mobil applikationen. Hvis der bliver et behov for at have nogle indstillinger i wallboard mobil applikationen i fremtiden, kan man udvikle en option menu. Man kunne f.eks. have den mulighed at brugerne kunne bestemme at det wallboard de har brug for vises med det samme ligeså snart man trykker på start ikonet på skærmen. Det vil være hurtigere for dem at se de oplysninger de har brug for i stedet for at gå igennem applikationens hovedmenu og vælge det ønskede wallboard. Der findes forskellige måder man kan lave option menuer til sin applikation på.

4.3.4.3 Wallboard til Android tablet

Idet brugen af Tablets er i konstant udvikling³⁵ kan man også tænke sig at tilpasse wallboard mobil applikationen til Android tablet. Jeg har ikke kunnet teste applikationen på en tablet enhed da jeg ikke har haft en Android tablet, jeg kunne teste på. I stedet for har jeg prøvet at ændre skærmopløsningen³⁶ på AVD emulatoren så den passer til Android tablet. I bilag 6 kan man se, hvordan wallboard skabelonerne vil se ud på en tablet. Idet skærmstørrelsen på en tablet er større end på en smartphone, er der meget tom plads. Man kan tilpasse wallboard skabelonerne så de passer til en tablet skærmstørrelse, ved f.eks. at ændre på tekststørrelsen, placeringen, layoutet og informationer.



Figur 43 Android Tablet Samsung Galaxy

4.3.4.4 Site specifik browser applikation til andre platforme

Hvis wallboard mobilapplikationen bliver populært hos wallboard kunderne og der kommer en efterspørgsel på at have wallboard mobilapplikationen til f.eks. iOS eller andre platforme, kunne man lave site specifik browser til andre platforme, man kunne evt. anvende PhoneGap, så behøver man kun at skrive koden flere gange også har man en sitespecifik browser til alle de populære platforme.

³⁵ Se grafen i bilag 6 Tablets well grow as fast as MP3 player

³⁶ Opløsningen er på Samsung Galaxy tablet er 1024x600 og på en Motorola XOOM 1280x800

4.4 Test og evaluering af Wallboard mobil applikationen

Test er en essentiel fase i udviklingsprocessen, da man skal teste produktet for fejl inden man leverer det til kunden. I dette afsnit har jeg beskrevet test af applikationen. Da brugervenligheden spiller en vigtig rolle i mobilapplikationer har jeg anvendt forskellige evalueringsmetoder til at bedømme brugervenligheden i applikation, såsom brugervenlighedstest, heuristisk evaluering og gestaltslove.

4.4.1 Test af Wallboard mobilapplikationen

Jeg har lavet en simpel "test" af applikationen med screenshots, som viser om applikationen reagerer efter hensigten, dvs. hvis den viser det brugeren forventer. Jeg har testet applikationen både på emulatoren og på en Android telefon (HTC Desire). Alt i alt vil jeg sige at applikationen gjort som forventet. Herunder ses de screenshot af testen på en HTC Desire. I midten kan man se hovedmenuen, som indeholder 6 wallboard, pilene indikere hvad der sker når brugeren trykker på et bestemt wallboard i menuen. Jeg har også testet applikationen på en iPhone 4, dette kan ses i bilag 7.



Idet applikationen kun kan tilgås via godkendte enheder inden for NetDesign's netværk, har jeg filmet en video, mens jeg tester applikationen på en HTC Desire, videoen kan ses via dette link:

<http://www.youtube.com/watch?v=OLypP5ITCrY&feature=youtu.be>

4.4.2 Brugervenligheds test

Der er forskellige måder man kan lave en brugervenlighedstest på. Det afhænger af applikationen og de funktioner den tilbyder. Man kan lave en brugervenlighedstest ved at definere forskellige brugeropgaver, som testpersonerne kan løse via applikationen. Her kan man observere om testpersonerne løser opgaverne rigtig, hvor lang tid det tager dem at løse opgave og hvilke løsninger de vælger. Her måles bla. Om funktionerne i applikationen er tilgængelige og nemme at bruge.

Det forholder sig lidt anderledes med wallboard mobil applikationen, idet der ikke er nogen funktioner i applikationen andet end at man kan vælge de forskellige wallboards, hvilket er lige til. Derfor vil jeg ikke kunne lave brugeropgaver, men i stedet for har jeg lavet et spørgeskema, som har til formål, at beskrive brugernes mening om applikationen. Det har hjulpet med at finde ud af om brugerne er glade for applikationen, om de synes den er nyttевærdig og forståelig. Jeg har også fået en del feedback på hvad de synes kunne forbedres og hvad man ellers kunne tilføje til applikationen i fremtiden.

4.4.2.1 Proces for brugervenligheds test

Jeg har testet applikationen med 10 personer fra NetDesign. Mine testpersoner havde forskellige baggrund der var supportere, afdelingsleder, chefer og konsulenter. Nogen af dem brugte wallboard til hverdag andre havde kendskab til det. Processen foregik således, at jeg aftalte en halvtimes møde med hver person, hvor jeg havde applikationen installeret på en HTC Desire. Jeg startede med at fortælle at det er en wallboard mobil app som viser wallboard skabelonerne som de er vant til at se på storskærmen. Så gav jeg telefonen til testpersonen og bad ham/hende om at finde wallboard mobil applikationen og gå ind på den. Her vil jeg observere om testpersonen har nemt ved at finde applikationen på mobilskærmen. Når testpersonen har fundet applikationen og gået ind på den vælger de at se de forskellige wallboard skabeloner en efter en og kommer med evt. kommentarer, som jeg notere ned. Når brugerne er færdige med at teste applikationen og har set de forskellige skabeloner igennem, stiller jeg dem de brugervenlighedsspørgsmål³⁷, som jeg havde forberedt. Herunder ses et par af de spørgsmål, jeg stillede brugerne, resten kan læses i bilag 4 under *Brugervenlighedstest spørgsmål*.

*Synes du applikationen er nem og bruge?
Vil du bruge wallboard mobil applikationen frem for storskærmen?*

Resultaterne for usability testen kan ses i bilag 8 under *Statistiske resultater*.

³⁷ Se brugervenlighedstest spørgsmål i bilag 8

4.4.3 Heuristisk evaluering

Heuristisk evaluering også kaldt ekspertvurdering er en slags brugervenligheds test, som har til formål at identificere problemer. Her er det ikke brugerne, som foretager evalueringen, men "eksperter", som evaluerer produktet i forhold til nogle anerkendte principper. Til forskel fra brugervenlighedstest, kan man vha. heuristisk evaluering finde frem til de problemer, som en bruger kan komme ud for.

Jeg har i den heuristiske evaluering³⁸ taget udgangspunkt i Jakob Nielsens 10 berømte Usability Heuristics. Ved hjælp af denne evaluering er jeg blevet opmærksom på fejl og mangler ved applikationen. Herunder har jeg listet nogle af principperne og beskrevet om de er opfyldt eller ej. Skadeniveauet indikerer, hvor meget stor skaden er når princippet ikke er opfyldt.

1. Visibility of system status

Dette er opfyldt. Først vises menuen, hvor efter bruger vælger wallboard han/hun gerne vil se. Hvis brugeren vil se et andet wallboard går han bare tilbage ved at klikke på tilbage knappen og vælger et nyt wallboard.

Skadeniveauet: Høj

2. Match between system and the real world

Ordene som bliver anvendt i applikationen, matcher den virkelige verden. Brugere har også kendskab til ordene fra deres arbejde og fra wallboard desktop applikationen.

Skadeniveauet: Høj

3. Error prevention:

På nuværende tidspunkt understøtter mobilversionen ingen fejlmeddelelser. Hvis man prøver at tilgå applikationen i et tilfælde hvor der ikke er netadgang, giver android en fejlmeddelelse om at man ikke kan tilgå applikationen på nuværende tidspunkt.

Skadeniveauet: medium

4. Flexibility and efficiency of use

Der er ikke noget fleksibilitet på selve applikationen, men vha. Android applikationen kan brugeren vælge hvordan de opsætter applikationen, således at når de klikker på applikationen viser den direkte det wallboard de ønsker uden at vise menuen.

Skadeniveauet: Lav

5. Aesthetic and minimalist design

Dette er opfyldt. Der er ingen unødvendig informationer og knapper, som kan distrahere brugeren.]

Skadeniveauet: Høj

³⁸ Se evalueringsskemaet I bilag 9 Heuristisk evaluering.

4.4.4 Gestalts love

Gestaltlove handler om perception. De giver et bud på, hvordan den menneskelige hjerne opfatter helheder i billeder. Gestaltlovene bruges forskellige steder uden vi lægger mærke til det, de bruges i design af applikationer for at gøre dem mere forståelig for brugerne. Lovene hjælper med, at organisere informationer, så brugeren nemmere kan overskue indholdet på en skærm. Det er blevet bevist, at det er sværere, at læse på en skærm end på papir. Hvilke betyder at der er et stort behov for, at være særlig opmærksom på, hvordan informationerne organiseres på en forståelig måde. Overskuelighed hjælper brugeren med hurtigt at gennemskue funktionerne i applikationen uden at læse om dem eller få hjælp fra andre. Gestaltlovene sørger for at designet gør funktionaliteten synlig og sigende. Ved at anvende gestaltlovene kan man få brugerne til at forstå designet og gennemskue funktionerne hurtigt, hvilket øger brugervenligheden.

Derfor har jeg i design af wallboard mobil versionen været opmærksom på de afledte krav fra gestaltlovene. Herunder har jeg beskrevet nogle eksempler på gestaltlove i mit design. Loven om **nærhed**³⁹ fortæller, at symboler og tekst, som er anbragt tæt på hinanden bliver opfattet som sammenhængene. Denne lov er opfyldt mange steder i wallboard applikationen. Herunder ses screenshot, hvor loven er opfyldt.

MICH ● FEI2 ●

Figur 44

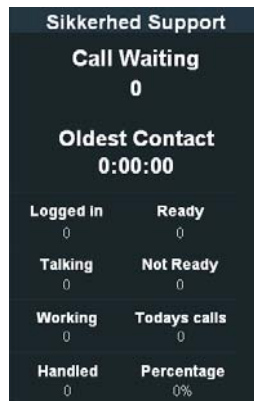
I figur 44 kan man se at loven om nærhed er overholdt. Man kan se at initialerne hænger sammen med den røde status cirkel.

| Kunde | Sagstitel | SLA |
|-------|--|-----------|
| SKAT | P3 Mail bliver droppet i LEV-FW - address spoofing | check SLA |

Figur 45

Loven om nærhed er overholdt i figur 45. overskrifterne; kunde, sagsnr og SLA er bragt tæt på teksten nedenunder. Det gør at man kan læse at hvem der er kunde, titlen og SLA.

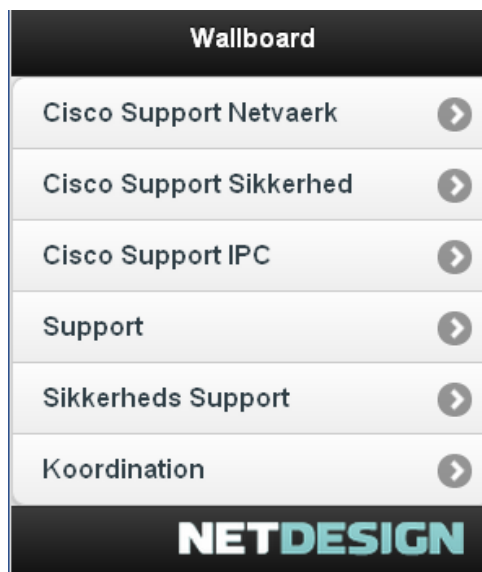
³⁹ Law of proximity: (Søren Lauresen) kap 3



Figur 46

Man kan også se at loven er opfyldt i figur 46, hvor overskrifterne er anbragt tæt på tallene som står under. F.eks. så kan man læse at call waiting hænger sammen med 0 som står under den, så brugerne kan se at der er 0 call waiting.

Menu siden overholder flere gestaltlovene. Linkene til wallboard er anbragt inde i en ramme, som fastlægger, hvor menuen starter og slutter, hvilke opfylder loven om **lukket**⁴⁰. Loven om **lighed**⁴¹ fortæller, at ting som ligner hinanden bliver opfattet som sammenhængene. Man kan se at alle link til wallboard har samme layout, hvilket gør at man opfatter at der er lighed mellem dem. Loven om nærhed er også opfyldt idet wallboard titlerne og pilene er anbragt tæt på hinanden.



Figur 47

⁴⁰ Law of closure: (Søren Lauresen) kap 3

⁴¹ Law of similarity: (Søren Lauresen) kap 3

5. Konklusion

Projektets primære mål var, at designe og udvikle en wallboard mobilapplikation. Jeg har derfor belyst forskellige teknologier og metoder til udvikling af mobilapplikationer.

Undersøgelsen viste at webapp samt Android webview app (android site specific browser) er en passende teknologi og udviklings mulighed til wallboard mobil applikation, fordi webapp understøtter alle platforme. Det er en fordel, da man ikke kan forvente at wallboard kunderne/brugerne har en bestemt mobilplatform. Webapp giver også brugerne den nyeste version af mobilapplikationen hele tiden. Vha. webapp minimeres udviklingsomkostningerne, vedligeholdelsen af applikationen er nemmere, brugerne får realtime data hele tiden. Jeg valgte en Android webview app frem for en anden platform først og fremmest fordi Android har den største markedsandel på nuværende tidspunkt. Android er Open Source og bygger på programmeringssproget Java, som jeg allerede har kendskab til. Derudover har de fleste medarbejdere i NetDesign en Android telefon, idet NetDesign har tilbudt sine medarbejder HTC telefoner, derfor er det mere hensigtsmæssigt at udvikle Android webviewer applikation.

Jeg startede med at lave brugeranalyse for at bestemme målgruppen for mobilapplikationen og målgruppens brugerkaraktistika. Jeg interviewede eksisterende wallboard brugere fra NetDesign om deres ønsker og behov til wallboard mobilapplikation. Ved hjælp af data indsamling og interview kunne jeg definere krav til wallboard mobilapplikationen.

Jeg havde succes med at anvende mockups som prototyper, scenarie beskrivelser og storyboards, som brugerne kunne give feedback på. Det hjalp mig til at finde ud af hvilke informationer brugerne prioriterer. I wallboard designet har jeg været opmærksom på at anvende afledte krav fra gestaltslove, som gør at brugerne bedre kan forstå designet, og hermed øge brugervenligheden i wallboard mobilapplikationen.

Med udgangspunkt i design kunne jeg implementerer wallboard mobilapplikationen, hvor jeg tog brugernes mening og feedback i betragtning. Udviklingen af wallboard mobilapplikationen bestod af 3 dele:

- Implementering af wallboard skabeloner vha. CSS.
- Implementering af wallboard web applikation, som konfigureres til at vise en webapplikation.
- Implementering af wallboard menu vha. jQuery.

De essentielle krav og behov til applikationen er opfyldt. Brugerne havde dog et ønske om, at applikationen skulle håndtere notifikation således, at de fik alarm på mobilen hver gang der skete noget særligt. Dette

behov kom først i slutningen af forløbet, derfor er det ikke blevet opfyldt, men det vil tages i betragtning, hvis wallboard mobilapplikationen skal videre udvikles.

At inddrage brugerne i udviklingsfaserne har hjulpet med at skabe en værdi i wallboard mobilapplikationen for brugerne, idet den lever op til deres forventninger og ønsker. Dette resulterede også i at brugerne var tilfredse med applikationen og synes den er nem at bruge, nyttевærdig og hjælpsom, som det fremgår i brugervenlighedstesten.

Idet brugervenligheden spiller en essentiel rolle i udvikling af mobilapplikationer, har jeg anvendt forskellige evalueringsmetoder til at bedømme brugervenligheden af wallboard mobilapplikation, herunder brugervenlighedstest og heuristisk evaluering. I brugervenlighedstesten var det brugernes mening om applikationen jeg fik frem, i heuristisk evaluering var det en ekspert vurdering jeg fik tilbage, som gjorde mig opmærksom på de eventuelle mangler i applikationen, som jeg ikke havde taget i betragtning.

I dag bruges wallboard mobilapplikationen af medarbejderne i NetDesign, som har vist glæde af applikationen. NetDesign har også planer om, at tilbyde mobilapplikationen til deres eksisterende wallboard desktop kunder.

Wallboard applikationen var en velegnet opgave at bygge en mobilapplikation ud fra. Opgaven var overskuelig men også kompleks nok til at tage mig igennem alle udviklingsfaserne og havde således været en lærerig opgave at gennemføre.

6. Perspektivering

Arbejdet med wallboard mobilapplikationen har været et inspirerende og spændende projekt for mig. Jeg har lært forskellige metoder og teknologier, som kan anvendes til at udvikle mobilapplikationer, som jeg ikke havde lært om i de kurser jeg har haft på mit diplom IT uddannelse. Jeg har også lært metoder og principper, som bruges til at designe og udvikle brugercentreret software, som er en primær faktor i brugervenligt design. Mobilapplikationer er i dag blevet en central del i softwareudvikling og brugervenligheden spille en større rolle på mobile enheder, fordi brugerne laver forskellige ting samtidig med at de bruger mobile apps.

Projektforløbet har givet mig en erfaring i at arbejde individuelt, hvor jeg under min uddannelsesforløb har arbejdet i grupper på forskellige projekter.

Jeg tror, at mobilapplikationer bliver en stor del af fremtidens kommunikation og vil spille en større rolle i IT verdenen. Med den erfaring jeg har fået fra dette projekt, kunne jeg tænke mig at arbejde videre med mobilapplikationer i fremtiden og lærer nye metoder og teknologier indenfor dette område.

7. Litteraturliste

Jeg har anvendt bøger, artikler og hjemmesider til informationssøgning. Jeg har sat reference til al litteratur, men det er ikke alt litteratur, som bruges direkte i rapporten, derfor er der ikke alt, jeg refererer til i rapporten.

Bøger:

- [1] Lauesen, Søren: *User Interface Design, A Software Engineering Perspective*: Person Education, 2005.
- [2] Wiley, John: *Interaction Design, beyond human-computer interaction 2nd Edition*: Wiley, 2007.
- [3] Molich, Rolf: *Brugervenligt webdesign 2*. Udgave: Nyt Teknisk Forlag, 2006.
- [4] Bernd, Bruegge & Allen H., Dutiot: *Object-Oriented Software Engineering, Using UML, Patterns, and Java 2nd Edition*: Person Education 2004.
- [5] Mier, Reto: *Android Application Development*: Wiley, 2009.
- [6] Johnson, Jeff: *Designing with the Mind in Mind, Simple Guide to understanding User Interface Design Rules*: Morgan Kaufmann Publisher, 2010.
- [7] Nilsen, Jakob: *Godt webdesign 1*. Udgave: IDG Forlag, 2001.

Artikler:

- [8] Jeffrey S. Hammond: *Define Your Mobile Development Strategy for Application Development & Delivery Professionals*: Forrester August 24, 2010.
- [9] Mike Gualtieri: *Mobile App Design Best Practices, When It Comes To Designing The Mobile User Experience (UX)*, Context Is King: Forrester April 12, 2011.
- [10] Jeffrey S. Hammond: *Mobile Development Goes Multiplatform iPhone And Android Top The List Of Devices Developers Target*: Forrester December 22, 2010.
- [11] Paul D. Hamerman: *Mobile Applications Will Empower Enterprise Business Processes An Empowered Report: Harnessing Mobile Technology For Process Acceleration*: Forrester September 30, 2010.
- [12] Brownlee Thomas, *How To Empower iWorkers With Mobile UC An Empowered Report: Sourcing Professionals Help Mobility Efforts Through Effective Supplier Due Diligence*: Forrester October 12, 2010.

Links:

- [13] <http://developer.android.com/guide/basics/what-is-android.html>
- [14] <http://developer.android.com/guide/webapps/webview.html>
- [15] http://developer.android.com/guide/practices/ui_guidelines/icon_design.html#design-tips
- [16] <http://www.cmswire.com/cms/web-engagement/the-mobile-application-browser-vs-native-010223.php>
- [17] <http://www.cmswire.com/featured/ektron-web-cms/ty-mobile.php>
- [18] http://www.mrc-productivity.com/research/whitepapers/whitepaper_mobile.pdf
- [19] <http://developer.android.com/guide/webapps/index.html>
- [20] <http://en.wikipedia.org/wiki/IOS>
- [21] <http://www.version2.dk/artikel/saadan-bliver-du-iphone-udvikler-10823>
- [22] <http://developer.Android.com/sdk/index.html>
- [23] <http://jquerymobile.com/demos/1.0/docs/about/getting-started.html>
- [24] <http://getfirebug.com/whatisfirebug>
- [25] <http://developer.Android.com/guide/tutorials/views/hello-webview.html>

8. Bilag

Bilag 1 Persona spørgeskema

Spørgeskema som er blevet brugt til at udarbejde persona beskrivelse:

Dette spørgeskema sendte jeg rundt til medarbejderne som anvender wallboard applikationen. Jeg har brugt <http://da.surveymonkey.com> til at lave et online spørgeskema, som jeg kunne sende rundt. Herunder ses spørgsmålene fra spørgeskemaet.

- **Navn**
- **Køn**
- **Alder**
- **Hvilken baggrund /uddannelse har du?**
- **Hvad er dine arbejdserfaringer?**
- **Hvilke typer opgaver sidder du med i NetDesign?**
- **Bruger du wallboard til at løse dine arbejdsopgaver? Hvis ja, hvor ofte har du brug for at kigge på den?**
- **Hvad er dine fritidsinteresser?**
- **Har du en smartphone? Hvis ja hvilken?**
- **Bruger du mobilapplikationer? Hvis ja hvilken?**
- **Har du behov for en wallboard mobilapplikation? Hvis ja uddyb.**

Bilag 2 Interview spørgsmål

Herunder ses interviewspørgsmålene, som blev brugt til at finde ud af hvad brugerne ønsker. Det har ikke været direkte spørgsmål og svar interview, det var mere en samtale, hvor jeg fik svare på nedenstående spørgsmål.

- **Hvordan anvender du wallboard applikationen?**
- **Hvad kunne du tænke dig mobilapplikationen skal have fokus på?**
- **Hvor mange gange i timen ser du på wallboard skærmen?**
- **Har du behov for en wallboard mobilapplikation?**
- **Hvilke fordele tror du der vil være hvis du kunne have wallboard med det overalt i virksomheden?**
- **Tror du at du vil bruge wallboard mobilapplikationen?**

Bilag 3 Kildekode

Webviewer applikation

WallboardMobil Class:

```
package com.exampel.wallboardmobil;
import android.app.Activity;
import android.os.Bundle;
import android.view.KeyEvent;
import android.webkit.WebView;
import android.webkit.WebViewClient;

public class WallboardMobil extends Activity {
    private static final String MAIN_PAGE = "http://172.23.1.188:8080/wallboard-web/mobilUI2.html";
    WebView webview;
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        webview = (WebView)findViewById(R.id.webview);
        webview.getSettings().setJavaScriptEnabled(true);
        webview.setVerticalScrollBarEnabled(false);
        webview.setHorizontalScrollBarEnabled(false);
        if (savedInstanceState != null){
            webview.restoreState(savedInstanceState);
        }else{
            webview.loadUrl(MAIN_PAGE);
            webview.setWebViewClient(new WebViewClient());
        }
    }
    @Override
    protected void onSaveInstanceState(Bundle outState) {
        webview.saveState(outState);
    }
    @Override
    public boolean onKeyDown(int keyCode, KeyEvent event) {
        if ((keyCode == KeyEvent.KEYCODE_BACK) && webview.canGoBack()) {
            webview.goBack();
            webview.setWebViewClient(new WebViewClient());
            return true;
        }
        return super.onKeyDown(keyCode, event);
    }
}
```

Main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
```

```

    android:layout_height="fill_parent"
    android:orientation="vertical"
  >
  <WebView
    android:id="@+id/webview"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
  />
</LinearLayout>

```

AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.exampel.helloandroid"
    android:versionCode="1"
    android:versionName="1.0" >
  <uses-sdk android:minSdkVersion="7" />
  <uses-permission android:name="android.permission.INTERNET" />
  <application
    android:debuggable="true"
    android:icon="@drawable/wallboard_mobil"
    android:label="Wallboard" >
    <activity
      android:label="Wallboard"
      android:name=".WallboardMobil"
      android:theme="@android:style/Theme.NoTitleBar"
      >
      <intent-filter >
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
  </application>
</manifest>

```

Wallboard mobil skabeloner

Skabelon 1

Denne skabelon bruges i forskellige wallboards; cisco netværk, cisco IPC, cisco Sikkerhed og Koordination.



Herunder ses css koden:

```
html, body {
  -moz-background-clip: border;
  -moz-background-origin: padding;
  -moz-background-size: auto auto;
  background-attachment: scroll;
  background-color: #60A08A;
  background-image: none;
  background-position: 0 0;
  background-repeat: repeat;
  font-family: Trebuchet MS;
  font-size: 13px;
  font-weight: bold;
  height: 98%;
  margin-bottom: 0;
  margin-left: auto;
  margin-right: auto;
  margin-top: 0;
  text-align: center;
  overflow: hidden;
}
div.uccx_exchange_user img.displayAgentStatusImg {
  height: 10px;
  width: 10px;
}
.blueBorder, .thresholds {
  width: 98%;
}
.Threshold_INFO, .Threshold_WARNING,
.Threshold_ALERT, .Threshold_NONE {
  font-size: 3em;
}
.column {
  display: inline;
  float: left;
  margin-bottom: -10px;
  margin-left: 1px;
  margin-right: 1px;
  margin-top: -1px;
  text-align: center;
  vertical-align: text-top;
}
.header {
  color: black;
  font-size: 1em;
  margin-bottom: -4px;
  margin-left: -4px;
  margin-right: -4px;
  margin-top: 3px;
}
.column {
  display: inline-block;
  line-height: normal;
  white-space: normal;
}
.columnwrapper {
  min-width: 375px;
  text-align: center;
  white-space: nowrap;
  width: 25%;
}
.displayAgent img {
  margin-right: 0.3em;
}
.displayAgentName {
  display: table;
  text-align: left;
  top: 10px;}
.displayAgentStatus {
  display: inline-block;
  width: 10px;}
div.headline span.remedyId {
  display: none;}
div.remedyItem span.remedyGroup {
  display: none;}
.displayAgentStateTime {
  display: inline-block;
  width: 50px;}
div .displayAgentStateTime {
  display: none;}
div .displayAgentStatus {
```

```

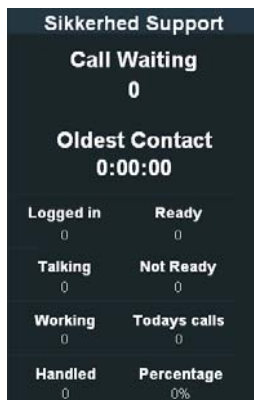
    display: none;}
.PoweredBy {
  display: none;}
.lastupdate {
  display: none;}
.viewerDisplayItemErrorPanel {
  color: red;
  font-weight: bold;}
.headline.remedyItem {
  font-weight: bold;}
div.remedyItem span.remedyId {
  display: none;}
.remedyItem span {
  display: inline-block;
  font-size: 1em;
  max-height: 2.2em;}
div.headline.remedyItem {
  border-bottom-color: -moz-use-text-color;
  border-bottom-style: solid;
  border-bottom-width: 2px;
  text-align: center;}
span.remedyCustomer {
  font-size: 0.8em;
  margin-left: 0.5em;
  text-align: left;
  width: 10.9em;
  white-space: nowrap;
  overflow: hidden;
}
span.remedyPriority {
  font-size: 0.8em;
  width: 3.2em;
}
span.remedyCaseTitle {
  font-size: 0.8em;
  text-align: left;
  width: 29em;
  white-space: nowrap;
  overflow: hidden;}
span.remedyHidden {
  display: none;
}
span.remedyShowSLA {
  font-size: 0.8em;
  width: 6.2em;}
span.remedyGroup {
  width: 5em;
}
div.column.Remedy {
  clear: left;
  margin-top: 0.8em;}
div.column.Remedy h2 {
  display: none;}
.headline.remedyItem .remedyPriority {
  display: none;}

.headline.remedyItem .remedyCustomer {
  margin-right: 2.0em;}
span.NOT_SET {
  color: white;}
div.FIX_AT_NONE .remedyId {
  color: yellow;}
span.NORMAL_ALERT, span.FIX_AT_ALERT {
  color: red;}
span.NORMAL_WARNING, span.FIX_AT_WARNING {
  color: yellow;}
span.NONE_WARNING, span.NONE_NONE {
  color: blue;}
span.NONE_ALERT {
  color: purple;}
.BURNED {
  color: red;}
.P1.remedyPriority {
  background-color: red;}
.P2.remedyPriority {
  background-color: yellow;}
div.wallboardShow div.column.OBJ1 {
  clear: left;}
div.wallboardShow div.column.OBJ0 {
  clear: right;}
div.wallboardShow div.column.OBJ0 div,
div.wallboardShow div.column.OBJ1 div {
  float: left;
  padding: 2.2px 3px;}
.uccxStatus {
  display: none;}
.exchangeStatus {
  display: none;}
span.RESPECTED {
  color: green;}
div.wallboardShow div.column.OBJ1 div,
div.wallboardShow div.column.OBJ0 div {
  float: right;
  padding: 2.1px 2.2px;
}
.column.OBJ0 .header, .column.OBJ1 .header { float:
left;
padding: 1px 2.8px;}

```

Skabelon 2

Denne skabelon kan bruges hos Astraas sikkerheds support og hos mange NetDesign kunder.



Herunder ses css koden:

```
body, .wallboardShow {
  font-family:sans-serif;
  float: left;
  width: 98%;}
.viewerDisplayItemShowPanel,.header,
.viewerDisplayItemShowPanel div {
  color:white;
}
.column {
  font-family:sans-serif;
  float: left;
  width: 20%;
  -moz-border-radius: 11px 11px 11px 11px;
  background-color: #1B262A;
  border: 1px solid #242E38;
  margin: 1%;
  min-width:300px;}
.column .OBJ0 h1, .column .OBJ1 h1 {
  font-size: 1em;
  font-weight: bold;
  vertical-align: text-top;
}
.column .OBJ1 {
  float: right;
  font-size: 1.6em;
  font-weight: bold;
  margin-left: auto;
  margin-right: auto;
  text-align: center;
  width: 100%;}
.column .OBJ0 {
  float: left;
  font-size: 1.6em;
  font-weight: bold;
  margin-left: auto;
  margin-right: auto;
  text-align: center;
  width: 100%;}
.header {
  -moz-border-radius: 11px 11px 0 0;
  background-color: #243642;
  font-size: 1.5em;
  font-weight: bold;
  height: 1em;
  margin-bottom: -0.5%;
  margin-top: -0.4%;
  padding: 1%;
  text-align: center;
}
.viewerDisplayItemShowPanel .OBJ2{
  clear: left; }
.viewerDisplayItemShowPanel.OBJ2,
.viewerDisplayItemShowPanel.OBJ3,
.viewerDisplayItemShowPanel.OBJ4,
.viewerDisplayItemShowPanel.OBJ5,
.viewerDisplayItemShowPanel.OBJ6{
  width: 45%;
}
.viewerDisplayItemShowPanel.OBJ7,
.viewerDisplayItemShowPanel.OBJ8,
.viewerDisplayItemShowPanel.OBJ9 {
  width: 45.9%;
}
.viewerDisplayItemShowPanel.OBJ2,
.viewerDisplayItemShowPanel.OBJ3,
.viewerDisplayItemShowPanel.OBJ4,
.viewerDisplayItemShowPanel.OBJ5,
.viewerDisplayItemShowPanel.OBJ6,
.viewerDisplayItemShowPanel.OBJ7,
```

```

.viewerDisplayItemShowPanel.OBJ8,
.viewerDisplayItemShowPanel.OBJ9 {
  border-top: 1px solid #242E38;
  float: left;
  margin-left: auto;
  margin-right: auto;
  text-align: center;}
h1 {
  font-size: 1.1em;
  vertical-align: text-top;}
.PoweredBy {
display: none;
}
.lastupdate{
  display: none;}
.Threshold_NONE {
color:black;}
.Threshold_INFO {
color:green;}
.Threshold_WARNING {
color:yellow;}
.Threshold_ALERT {
color: red;}

```

Skabelon 3

| Support | | |
|---------------|-----------|---------------|
| Max ventetid | | |
| 0 | | |
| Ventende kald | | |
| 0 | | |
| Klar | Meldt ind | Spær |
| 0 | 0 | 0 |
| Taler | Tilbudt | Besvaret kald |
| 0 | 0 | 0 |

Herunder ses CSS koden for ovenstående skabelon:

```

BODY {
  float: left;
  font-family: sans-serif;
  font-size: 11px;
  width: 98%;
}
.wallboardShow {
  float: left;
  font-family: sans-serif;
  font-size: 1.8em;
  width: 98%;
}
.column {
  background-color: #F0F0F0;
  border-color: gray;
  border-radius: 11px 11px 11px 11px;
  border-style: solid;
  border-width: 1px;
  float: left;
  font-family: sans-serif;
  margin: 1%;
  width: 15em;
}
.viewerDisplayItemShowPanel.OBJ1 {
  float: left;
  font-size: 3em;
  font-weight: bold;
  margin-left: auto;
  margin-right: auto;
  text-align: center;
  margin-top: -0.5em;
  width: 100%;
  clear: left;
}
.viewerDisplayItemShowPanel.OBJ0 {
  float: left;
  font-size: 3em;
  font-weight: bold;
  margin-left: auto;
  margin-right: auto;
  text-align: center;
  width: 100%;
}
.header {
  background-color: #759975;
  border-radius: 11px 11px 0 0;
  font-size: 1.8em;
  font-weight: bold;
  height: 1em;
  margin: -0.35% auto 0;
}

```

```

padding-bottom: 0.5%;
text-align: center;
width: 100%;
}
.viewerDisplayItemShowPanel.OBJ2 {
clear: left;
}
.viewerDisplayItemShowPanel.OBJ2 {
border-top: 1px solid gray;
float: left;
margin-left: auto;
margin-right: auto;
text-align: center;
width: 33.3%;
}
.viewerDisplayItemShowPanel.OBJ3 {
border-top: 1px solid gray;
float: left;
margin-left: auto;
margin-right: auto;
text-align: center;
width: 33.3%;
}
.viewerDisplayItemShowPanel.OBJ4 {
border-top: 1px solid gray;
float: left;
margin-left: auto;
margin-right: auto;
text-align: center;
width: 33.3%;
}
.viewerDisplayItemShowPanel.OBJ5 {
border-top: 1px solid gray;
float: left;
margin-left: auto;
margin-right: auto;
text-align: center;
width: 33.3%;
clear: left;
}
.viewerDisplayItemShowPanel.OBJ6 {
border-top: 1px solid gray;
float: left;
margin-left: auto;
margin-right: auto;
text-align: center;
width: 33.3%;
}
.viewerDisplayItemShowPanel.OBJ7 {
border-top: 1px solid gray;
float: left;
margin-left: auto;
margin-right: auto;
text-align: center;
width: 33.3%;}

.viewerDisplayItemShowPanel.OBJ8 {
border-top: 1px solid gray;
float: left;
margin-left: auto;
margin-right: auto;
text-align: center;
width: 33.3%;
}
.viewerDisplayItemShowPanel.OBJ2 DIV {
font-size: 1em;
font-weight: bold;
}
DIV DIV .OBJ8 DIV {
font-size: 1em;
font-weight: bold;
}
DIV DIV .OBJ3 DIV {
font-size: 1em;
font-weight: bold;
}
DIV DIV .OBJ4 DIV {
font-size: 1em;
font-weight: bold;
}
DIV DIV .OBJ5 DIV {
font-size: 1em;
font-weight: bold;
}
DIV DIV .OBJ6 DIV {
font-size: 1em;
font-weight: bold;
}
DIV DIV .OBJ7 DIV {
font-size: 1em;
font-weight: bold;
}
H1 {
font-size: 1em;
font-weight: bold;
vertical-align: text-top;
}
.viewerDisplayItemShowPanel.OBJ0 H1 {
font-size: 0.5em;
font-weight: bold;
vertical-align: text-top;
}
.viewerDisplayItemShowPanel.OBJ1 H1 {
font-size: 0.5em;
font-weight: bold;
vertical-align: text-top;
}
.PoweredBy {
bottom: 10px;
font-size: 0.8em;
padding: 5px;
}

```

```

position: fixed;
right: 10px;
display: none;
}
.lastupdate {
display: none;
bottom: 10px;
font-size: 0.5em;
left: 10px;
position: fixed;}

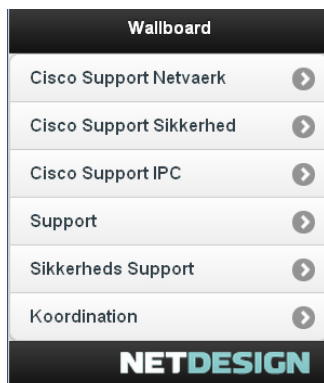
```

```

.Threshold_NONE {
color: black;}
.Threshold_INFO {
color: green;}
.Threshold_WARNING {
color: #FFCC00;}
.Threshold_ALERT {
color: red;}

```

jQuery wallboard menu



Herunder ses HTML koden for hovedmenuen. Koden er skrevet ved hjælp af jQuery.

```

<!DOCTYPE html>
<html>

<head>
<title>Wallboard</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="http://code.jquery.com/mobile/1.0/jquery.mobile-1.0.min.css" />
<script type="text/javascript" src="http://code.jquery.com/jquery-1.6.4.min.js"></script>
<script type="text/javascript" src="http://code.jquery.com/mobile/1.0/jquery.mobile-1.0.min.js"></script>
<style>
body div ul li a.test{color:red;}
</style>
</head>

<body>
<div data-role="page">
<div data-role="header" style="background-image:-moz-linear-gradient(#87ADA2,#87ADA2);">
<h1>Wallboard</h1>
</div><!-- /header -->

<ul data-role="listview" data-inset="true" data-filter="false">
<li><a class="test" href="http://nd-wallboard1:8080/wallboard-web/viewer/ciscoSupportNetvaerk_shayma">Cisco Support Netvaerk</a></li>
<li><a href="http://nd-wallboard1:8080/wallboard-web/viewer/ciscoSupportSikkerhed_shayma?11">Cisco Support Sikkerhed</a></li>

```



```
<li><a href="http://nd-wallboard1:8080/wallboard-web/viewer/shaymaa?12">Cisco
Support IPC</a></li>
<li><a href="http://nd-wallboard1:8080/wallboard-web/viewer/Support
">Support</a></li>
<li><a href="http://nd-wallboard1:8080/wallboard-web/viewer/Overview">Sikkerheds
Support</a></li>
</ul>
```

```
<div data-role="footer" style="background-image:-moz-linear-gradient(#87ADA2,
#87ADA2);">
<h4>
</h4>
</div>
<!-- /footer -->
</div><!-- /page -->
</body>
</html>
```

Bilag 4 main.xml

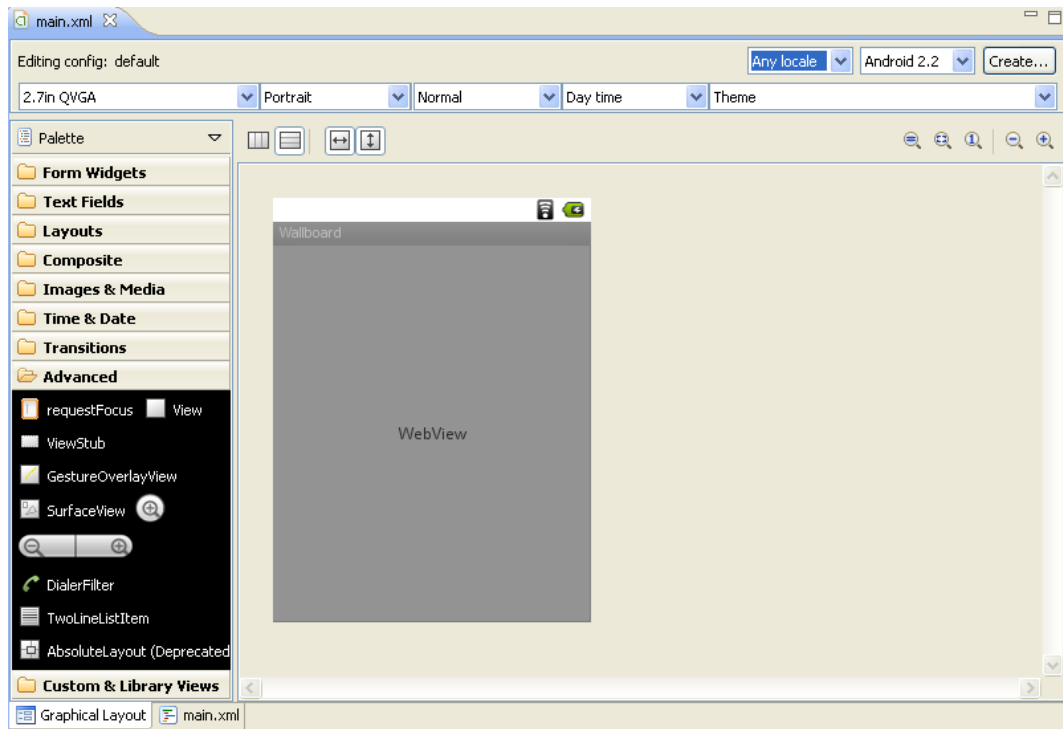


Fig. 1 Grafisk visning af main.xml

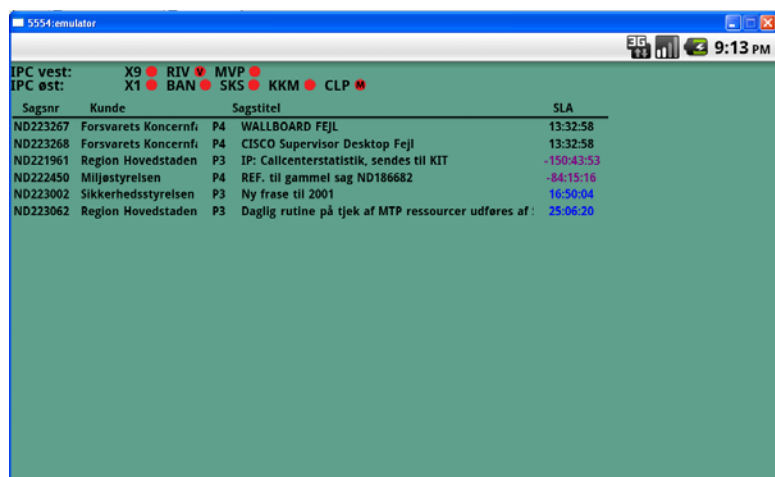
Bilag 5 Webviewer BT



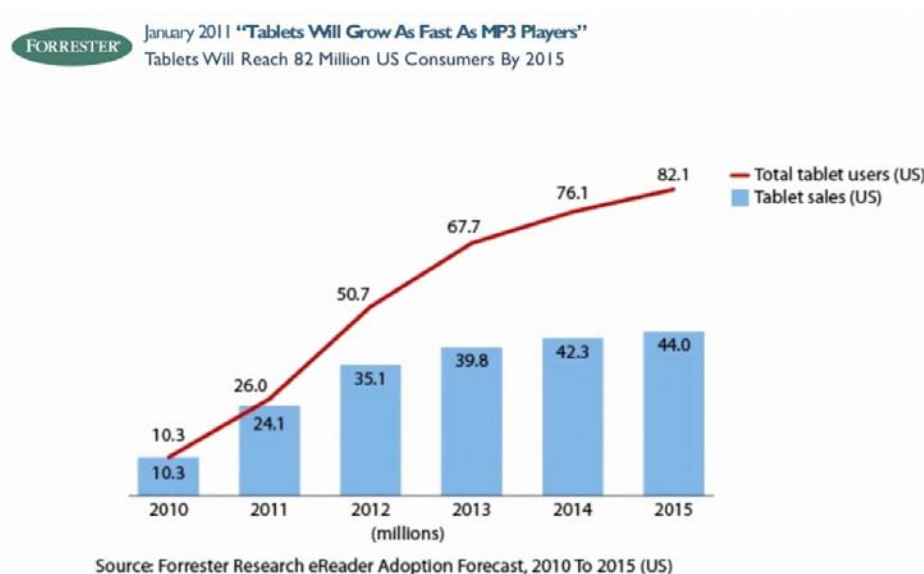
Fig. 1 www.BT.dk i webviewer applikation

Bilag 6 Tablet

Test på tablet (emulator)

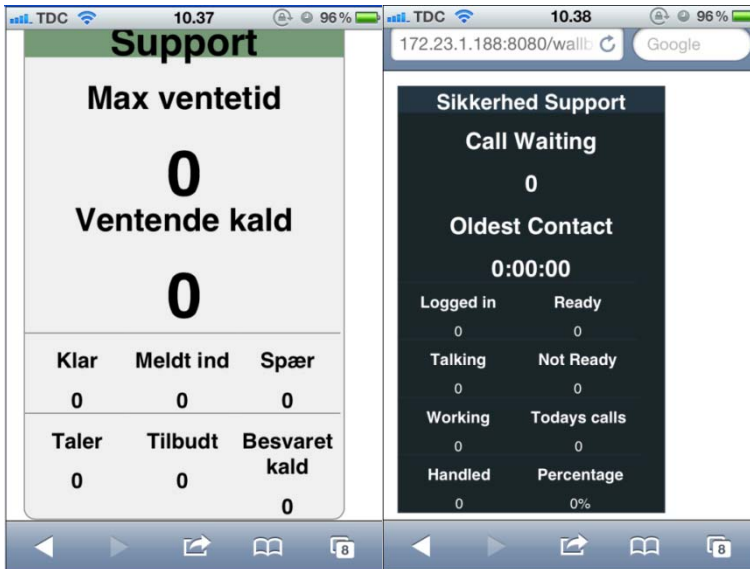


Tablet will Grow



Bilag 7 Test på iPhone

Herunder ses screenshot af wallboard skabelonerne på en iPhone 4.



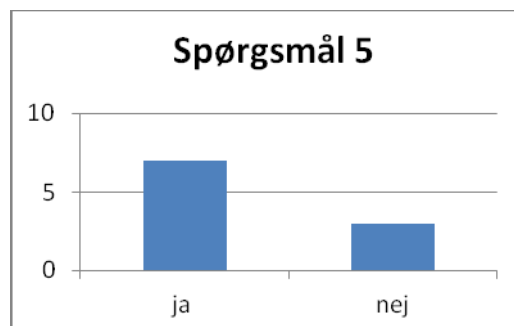
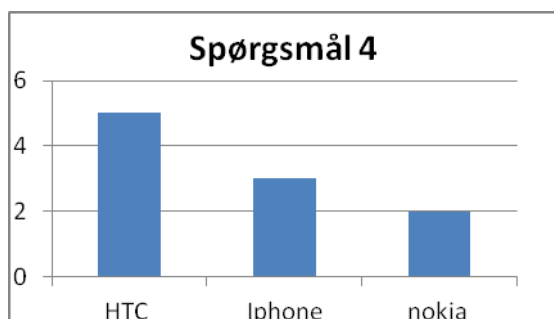
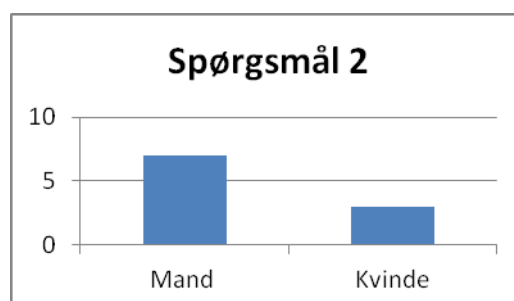
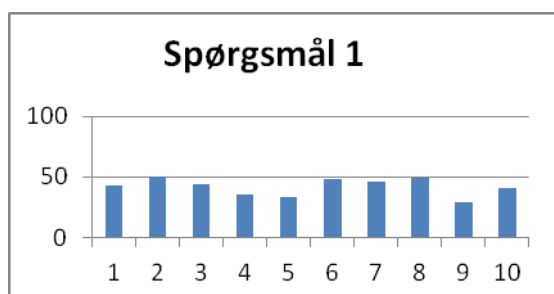
Bilag 8 Brugervenlighedstest

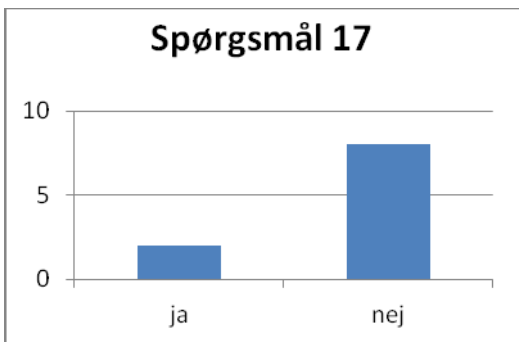
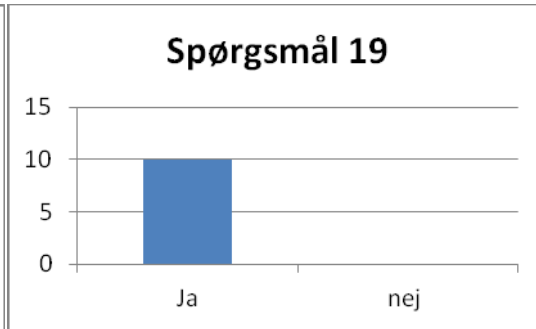
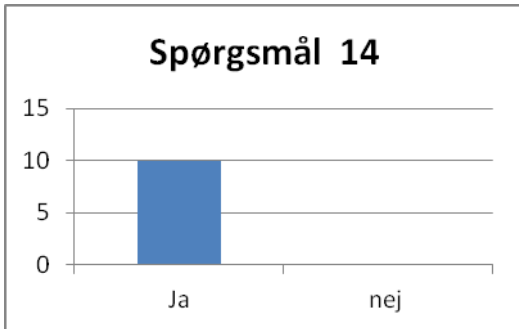
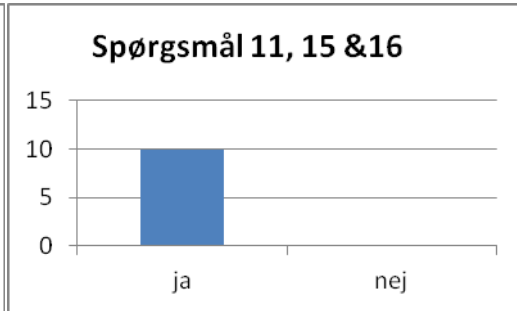
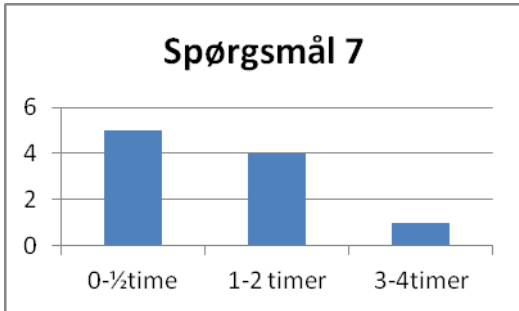
Brugervenlighedstest spørgsmål:

1. **Alder**
2. **Køn**
3. **Uddannelse/job**
4. **Mobiltelefon – årsag til valg?**
5. **Har du adgang til internet vha. mobiltelefon?**
6. **Bruger du internettet på din mobil eller mobilapplikationer? Hvis ja hvilke?**
7. **Tid brugt på internettet vha. mobiltelefonen om dagen?**
8. **Hvad synes du om at kunne komme på nettet vha mobilen?**
9. **Hvad synes du om layoutet på applikationen?**
10. **Hvad synes du om applikationens ikon?**
11. **Synes du applikationen er nem og bruge?**
12. **Mærker du en stor forskel mellem wallboard desktop og wallboard mobil, hvis ja hvilke?**
13. **Savner du noget ved mobil applikationen?**
14. **Giver wallboard mobil applikationen dig de informationer, som du har brug for at se?**
15. **Synes du applikationen er forståelig?**
16. **Synes du applikationen er nytteværdig?**
17. **Vil du bruge wallboard mobil applikationen frem for storskærmen?**
18. **I hvilke situationer vil mobilapplikationen være nyttig for dig?**
19. **Vil du anbefale wallboard mobil applikationen til dine kollegaer eller andre wallboard kunder?**

Statistiske resultater:

Herunder ses statistiske resultater af brugervenlighedsspørgsmålene, det er ikke alle spørgsmål som kunne fremvises på en statistisk måde.





Bilag 9 Heuristics Evaluation

Heuristics Evaluation Of [Wallboard mobil applikation]

By [Michael Houmaark & Shaymaa Yassen] Date [09/01/2012]

1. Visibility of system status

Evaluation

[Dette er opfyldt. Først vises menuen, hvor efter bruger vælger wallboard han/hun gerne vil se. Hvis brugeren vil se et andet wallboard går han bare tilbage ved at klikke på tilbage knappen og vælger et nyt wallboard.]

Skade: Høj

2. Match between system and the real world

Evaluation

[Mange af ordene som bliver anvendt i applikationen matcher den virkelige verden. Brugerne har også kendskab til ordene fra deres arbejde og fra wallboard desktop applikationen]

Skade: Høj

3. User control and freedom

Evaluation

[Der er altid mulighed for at komme tilbage til det forrige side eller lukke applikationen.]

Skade: Middel

4. Consistency and standards

Evaluation

[Bruger er bekendt med alle symboler begreber og farver, fra wallboard desktop applikationen.]

Skade: Middel

5. Error prevention

Evaluation

[På nuværende tidspunkt understøtter mobilversionen ingen fejlmeddelelser, det skyldes at der ikke har været behov for det, idet applikationen henter direkte information fra wallboard desktop applikationen, som håndterer fejlmeddelelser.]

Skade: Høj

6. Recognition rather than recall

Evaluation

[ikoner, handlinger og valgmuligheder er synlige. Brugerne skal være i stand til at genkende de forskellige aktioner. Brugerne behøver ikke at huske noget.]

Skade: mellem

7. Flexibility and efficiency of use

Evaluation

[Der er ikke noget fleksibilitet på selve applikationen, men vha. android applikationen kan brugeren vælge hvordan de opsætter applikationen, således at når de klikker på applikationen viser den direkte det wallboard de ønsker uden at vise menuen.]

Skade: lav

8. Aesthetic and minimalist design

Evaluation

[Dette er opfyldt. Der er ingen unødvendig informationer og knapper, som kan distrahere brugeren.]

Skade: Høj

9. Help users recognize, diagnose, and recover from errors

Evaluation

[Dette er ikke understøttet i den version af wallboard.]

Skade: Mellem

10. Help and documentation

Evaluation

[understøttes ikke. Jeg synes ikke at der er behov for en guid eller hjælp og dokumentation til denne applikation.]

Skade: Lav

Bilag 10 Installationsvejledning

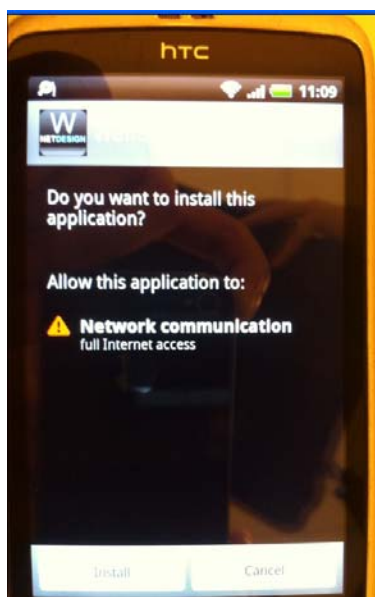
APK filen lægges op i NetDesign's produktion netværk. Jeg har valgt at distribuere wallboard mobil applikationen vha. e-mail. Det er nemmere idet medarbejderne ikke skal skrive en url i mobilbrowseren for at downloade applikationen på deres android telefon, men kan nøjes med at trykke på adressen, som står i e-mailen.

Installationsvejledning består af 2 steps.

1. Step: Åben den e-mail som du har modtaget med wallboard mobil applikationen og tryk på den angivet adresse, ligesom på nedenstående screenshot.



2. Step: Når du har klikket på adressen føres du til en ny side, som kan ses på nedenstående screenshot. Her bliver du spurgt om du ønsker at installere denne applikation. Tryk på install for at installere applikationen. Når du har trykket på install, downloades applikationen på din android telefon og du kan bruge den med det samme.



Bilag 11 CD-ROM

Knyttet til denne rapport er en cd-rom, som indeholder:

- En digital kopi af denne rapport
- Eksporteret Eclipse projekt som indeholder kildekoden til android site specifik browser app.
- Wallboard mobilapplikationen (Wallboard.APK)
- Videoptagelse af applikationen