

BACHELOROPPGAVE:

PICOBASE

FORFATTER(E):

Arild Plassen
Øyvind Skattum Vesteng

Dato: 20.05.2009

SAMMENDRAG AV BACHELOROPPGAVEN

Tittel:	<u>PICObase</u>	Nr. :	
		Dato :	20.05.09
Deltaker(e):	<u>Arild Plassen</u>		
	<u>Øyvind Skattum Vesteng</u>		
Veileder(e):	<u>Øivind Kolloen</u>		
Oppdragsgiver:	<u>Sykehuset Innlandet avd Gjøvik.</u>		
Kontaktperson:	<u>Håvard Njå</u>		
Stikkord (4 stk)			
Antall sider: 101	Antall bilag: 5	Tilgjengelighet åpen	
<p>PICObase er et system som hjelper medisinsk personale å søke i medisinske databaser, videre hjelper applikasjonen til med å strukturere og vurdere resultatene søket gir. Idéen er å støtte opp om en forsknings- og kunnskapsbasert praksis i det daglige arbeidet.</p> <p>Prosjektet består av to applikasjoner; en webapplikasjon og en håndholdt applikasjon.</p> <p>Webapplikasjonen er hovedåren i systemet, mens den håndholdte applikasjonen fungerer som en bærbar miniutgave av websystemet, og er fullt synkroniserbar med en sentral database.</p> <p>Den håndholdte applikasjonen er utviklet i Qt med C++, et rammeverk som gir mulighet å raskt portere en applikasjon til andre plattformer enn primærmålet.</p> <p>I dette tilfellet Windows Mobile 5 og 6.</p> <p>Applikasjonen på web er utviklet i PHP med Zend framework som grunnstruktur, for å opprettholde Model View Controller prinsippet. Databasen er implementert ved bruk av MySQL og PHPmyAdmin.</p> <p>Vi har gjennom prosjektet fokusert på å få til en så helhetlig prototype som mulig, da dette skal brukes som en del av et større nasjonalt prosjekt videre.</p>			

Forord:

Prosjektoppgaven ble presentert av oppdragsgiver på Høgskolen i Gjøvik , oktober 2008 i sammenheng med emnet Objektorientert systemutvikling. Ingen av de nåværende medlemmene arbeidet med prosjektet i det faget. Forarbeidet som var gjort før vi tok over, var ikke i stil med det vi ønsket, slik at ikke noe av dette er blitt tatt i bruk i løsningen av oppgaven.

Oppgaven virket spennende og utfordrende, da det skulle utvikles to hele systemer, og den ene på en plattform vi aldri hadde utviklet mot før. Håndholdte enheter er noe som det brukes mye av i dag, og dette prosjektet var noe vi så på som en gyllen mulighet til å lære mer om denne typen utvikling.

Siden vi ikke kunne bruke noe av det som var gjort fra før, så måtte vi ta for oss en omfattende kartlegging i starten. Dette særskilt pga prosjektets omfattende omfang med tanke på domenekunnskap. Da ingen av oss hadde medisinsk kunnskap tilstrekkelig nok for et slik system fra før, var dette noe vi måtte lære oss mye om.

Det har derfor vært en spennende og meget lærerikt prosjekt både rent faglig og teknisk sett, men også som et innsyn i den medisinske delen av verden, og hvordan en kliniker (medisinske personale) arbeider.

Veileder for prosjektet har vært Øivind Kolloen. Kontaktperson hos oppdragsgiver (Sykehuset Innlandet) har vært Håvard Njå. Vi vil gjerne takke begge for et godt samarbeid under prosjektet.

I tillegg så vil vi også rette en takk til Per Ola Vandvik v/ Sykehuset Innlandet for godt engasjement og faglig støtte rundt kunnskapsbasert praksis.

Gjøvik, 20.mai 2009

Arild Plassen

Øyvind Skattum Vesteng

Innhold

1	Introduksjon	1
1.1	Målgrupper	2
1.1.1	Prosjektet	2
1.1.2	Rapporten	2
1.2	Prosjektets hensikt	2
1.3	Prosjektgruppens bakgrunn	3
1.4	Utviklingsmodell	3
1.5	Organisering av rapporten	4
1.6	Faglig bakgrunn	5
1.6.1	Refleksjon, spørsmålsformulering og litteratursøk	5
1.6.2	Kritisk vurdering	6
1.6.3	Anvend og evaluér	7
2	Kravspesifikasjon	8
2.1	Systemets omgivelser	11
2.1.1	Entrez Programming Utilities	11
2.2	Systemets brukere	11
2.3	Overordnede funksjonelle krav	12
2.3.1	Use case diagram	12
2.3.2	Overordnede use case beskrivelser	12

2.4	Domenemodell	16
2.5	Supplementær spesifikasjon	17
2.5.1	Funksjonalitet	17
3	Design	18
3.1	Arkitektur	19
3.1.1	Autentisering og autorisasjon	20
3.2	Overordnet design	21
3.2.1	Webapplikasjon	21
3.2.2	Autorisering	21
3.3	Deployment View	22
3.4	Sekvens diagram	23
3.5	Data View	26
3.6	Designutfordringer	27
3.6.1	Lagring av kliniske spørsmål på håndholdt	27
3.6.2	Synkronisering mellom web og håndholdt	28
3.6.3	Støtte for flere søkekomposisjonstyper samtidig på håndholdt	29
4	Realisering	31
4.1	Utviklingsmiljø og språk	32
4.1.1	Webapplikasjon	32
4.1.2	Database	32
4.1.3	Håndholdt enhet	33
4.2	Design	35
4.2.1	Autentisering og autorisasjon	35
4.2.2	Interaksjon med EUtilities	39
4.2.3	Utforming søkefunksjonaliteten på webapplikasjonen	41

4.2.4	Plugin-tilnerming til søkekomposisjoner på webapplikasjonen	44
4.3	Implementasjon - Håndholdt	46
4.3.1	Opprettelse av klinisk spørsmål	46
4.3.2	Lagre klinisk spørsmål	46
4.3.3	Gjenopprette kliniske spørsmål	48
4.3.4	Komponere søk	50
4.3.5	Lagre / gjenopprette søkekomposisjoner	54
4.3.6	Utføre søk	56
4.3.7	Vis resultat	56
4.3.8	Vis abstract	57
4.4	Implementasjon - Webapplikasjon	58
4.4.1	Opprettelse av klinisk spørsmål	58
4.4.2	Søkefunksjonalitet	58
4.4.3	Kritisk vurderer	64
4.5	Små detaljer	67
4.5.1	Formatering av abstract	67
5	Avslutning	68
5.1	Diskusjon av resultater	68
5.1.1	Drøfting av resultater	68
5.2	Videre arbeid og forbedringsmuligheter	69
5.3	Evaluering av gruppas arbeid	69
5.3.1	Organisering og fordeling av arbeid	69
5.3.2	Fordeling av arbeidet	70
5.3.3	Loggføring	70
5.3.4	Prosjekt som arbeidsform	70
5.3.5	Konklusjon	71

6	Vedlegg	74
A	Forprosjekt	75
A.1	Mål og rammer	75
A.1.1	Innledning	75
A.1.2	Effektmål	76
A.1.3	Resultatmål	76
A.1.4	Læringsmål	77
A.1.5	Rammer	77
A.2	Omfang	78
A.2.1	Oppgavebeskrivelse	78
A.2.2	Avgrensninger	79
A.3	Prosjektorganisering	80
A.3.1	Ansvarsforhold	80
A.3.2	Grupperegler	80
A.3.3	Rutiner	80
A.3.4	Øvrige roller og bemanning	81
A.4	Planlegging, oppfølging og rapportering	82
A.4.1	Hovedinndeling av prosjektet	82
A.4.2	Prosjektmodell	82
A.4.3	Plan for statusmøter	83
A.5	Kvalitetssikring	83
A.5.1	Konfigurasjonsstyring	83
A.5.2	Kildekode og dokumentasjon	83
A.5.3	Kodestandard	83
A.5.4	Risikoanalyse	84
A.6	Plan for gjennomføring	85

A.6.1	Forklaring til gantt-skjema	85
B	Statusrapporter	86
B.1	Status	86
B.1.1	Planlegging	86
B.1.2	Organisering av gruppens arbeid og ansvarsområder	86
B.1.3	Klargjøring av problemstilling	87
B.1.4	Løsningsmetode	87
B.1.5	Rapportskriving	87
B.2	Muligheter, trusler og farer	88
B.3	Avsluttet arbeid	89
B.4	Gjenstående arbeid	89
B.5	Tidsfrister	89
B.6	Motivasjon	89
B.7	Veileder	90
C	Møtereferat og timelogg	91
C.0.1	Møtereferat	91
D	XML-Strukturer	94
D.1	Lagringsformat for kliniske spørsmål	94
D.1.1	Håndholdt enhet	94
D.2	Format for indeksering av kliniske spørsmål	98
D.2.1	Håndholdt enhet	98
D.2.2	PICObaseWeb	99
E	CD'ens innhold	101

Figurer

1.1	Elementene i kunnskapsbasert praksis	1
1.2	Sentrale figurer i KBP	5
2.1	Use Case for den håndholdte applikasjonen	9
2.2	Use Case for webapplikasjonen	10
2.3	Domenemodell	16
3.1	a) Selvstendig håndholdt b) Webapplikasjon overtar felles problematikk	19
3.2	MVC - <i>Model View Controller</i> pattern	21
3.3	Deployment diagram	22
3.4	Sekvens diagram som viser kommunikasjon mellom webapplikasjon og Entrez webservice ved henting av <i>docsum-liste</i>	23
3.5	Sekvens diagram som viser kommunikasjon mellom webapplikasjon og Entrez webservice ved henting av et enkelt abstract	25
3.6	Databasemodell	26
3.7	Illustrasjon av søkekomposisjonsdesign.	29
4.1	HTTP autentisering	35
4.2	Utsnitt fra data modellen som viser <i>ACL</i> implementasjonen i relasjonsdatabasen	36
4.3	Klassediagram over EUtilities systemet i webapplikasjonen	39

4.4	Sekvens diagram som viser kommunikasjon mellom webapplikasjon og Entrez webservice ved henting av <i>docsum-liste</i>	41
4.5	Sekvens diagram som viser kommunikasjon mellom webapplikasjon og Entrez webservice ved henting av et enkelt abstract	42
4.6	Full visning av klinisk spørsmål leser	43
4.7	Det grafiske brukergrensesnittet for søk	45
4.8	Klinisk spørsmål på håndholdt.	47
4.9	Gjenopprett et klinisk spørsmål	49
4.10	Fritekst-søk på håndholdt.	51
4.11	PICO-søk på håndholdt.	52
4.12	Dialogboks for lagring av søkekomposisjon.	54
4.13	Dialogboks for navngivning av lagret søkekomposisjon.	54
4.14	Gjenoppretting av lagret søkekomposisjon.	55
4.15	Resultatvisning på håndholdt.	56
4.16	Abstractvisning på håndholdt.	57
4.17	Opprett klinisk spørsmål. Nederste viser feilrapportering ved manglende utfylling.	59
4.18	Lagring / gjenoppretting av søkekomposisjoner	60
4.19	Visning av abstract med lagring av artikkel.	61
4.20	T.v. visning av lagrede abstract. Mulighet for å rediger kommentarer. <i>Kritisk vurder</i> linken aktiverer visningen t.h.	62
4.21	T.v. visning av resyme-editor. Denne vil bli erstattet av en rik-tekst editor. T.h. Visning av aktivt klinisk spørsmål med sine søkekomposisjoner.	63
4.22	Datastrukturen som er relatert til kritisk vurdering	65
4.23	Sammenligning av pdf-sjekkliste mot interaktiv sjekkliste	66

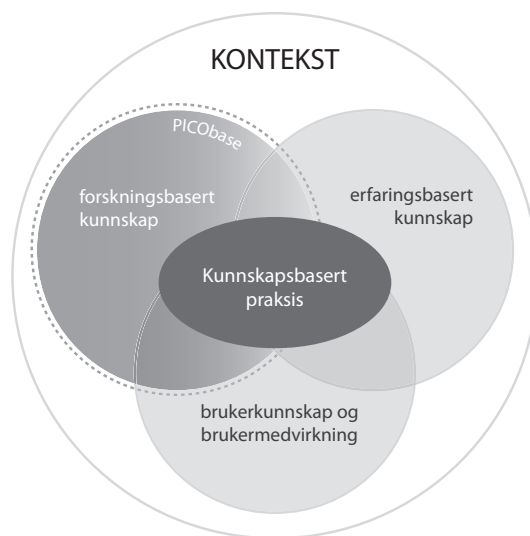
Listings

4.1	Utsnitt av felles autentiserings- og autorisasjonspunkt	38
4.2	Instansiering av stackedLayout og organisering av søkekomposisjon- swidgeter.	50
4.3	Setting av stackedLayout indeks.	53
4.4	Event handling for dynamisk endring av søkekomposisjonstype. . . .	53
	sourcecode/PICObaseWeb/abstract_format_regex.js	67
D.1	XML-format på lagret klinisk spørsmål	94
D.2	Valideringskjema for lagret klinisk spørsmål	95
D.3	Eksempel på lagret klinisk spørsmål	96
D.4	XML-format for indeksering av kliniske spørsmål på håndholdt . . .	98
D.5	Eksempel på indekseringsfil i bruk	98
D.6	XML-format for indeksliste av kliniske spørsmål fra web.	99
D.7	Eksempel på indekseringsfil fra web i bruk.	99

Kapittel 1

Introduksjon

Hverdagen til helsepersonell består av å ta beslutninger som skal bedre pasientens situasjon. Det er kritisk at disse beslutningene tas på riktig grunnlag, både direkte faglig og i henhold til pasientens ønsker og behov. Det er her kunnskapsbasert praksis kommer inn i bilde. Kunnskapsbasert praksis (KBP) er et sett med teknikker og metoder for å sikre at beslutninger tas på et vitenskapelig, oppdatert og pasientfokusert grunnlag.



Figur 1.1: Elementene i kunnskapsbasert praksis

Figuren viser hvordan KBP er bygd opp. *Forskningsbasert praksis* representerer den vitenskapelige forankringen i KBP. Her brukes systematisk innhentet oppdatert forskning aktivt for å støtte opp om helsefaglige beslutninger. *Erfarings-*

basert kunnskap representerer fagpersonens egne kunnskaper som er opparbeidet i praksis. *Brukerkunnskap og brukervedvirkning* setter fokus på å gi pasienten en aktiv rolle i behandlingen ved at de blir informert om alternativer, for eksempel presentert gjennom forskningsbasert kunnskap.

Dette prosjektet berører i hovedsak den delen som har med forskningsbasert praksis å gjøre.

1.1 Målgrupper

1.1.1 Prosjektet

Prosjektets målgruppe er klinikere som har behov for utøvelse av kunnskapsbasert praksis¹ (KBP), primært i kursvirksomhet, men også til bruk av klinikere erfarne i KBP. Disse klinikerne hører i første omgang til Sykehuset Innlandet, men prosjektet skal på sikt skaleres opp til andre kliniske institusjoner.

1.1.2 Rapporten

Målgruppe for rapporten er veileder, eksamenssensor, datastudenter og øvrige faglærere ved Høgskolen i Gjøvik. Det er derfor antatt at rapportens lesere har tekniske kunnskaper og innholdet følger deretter.

1.2 Prosjektets hensikt

Hensikten med prosjektet er å lage et system som skal gjøre det enklere for klinikere å innhente forskningsbasert kunnskap i medisinske databaser, strukturere og evaluere denne kunnskapen. Spesielt skal systemet gjøre det enklere for klinikere under opplæring i KBP. Dette skal oppnås med et systemet som skal bistå klinikere til å søke i medisinske databaser, behandle og vurdere kvaliteten og gyldigheten på artiklene som finnes gjennom disse søkene.

Dette skal oppnås med et systemet som skal bistå klinikere til å søke i medisinske databaser, behandle og vurdere kvaliteten og gyldigheten på kunnskapsbasert forskningsartikler.

¹Se seksjon 1.6 for beskrivelse.

1.3 Prosjektgruppens bakgrunn

Prosjektgruppens bakgrunn er i basis lik da begge har hatt fokus på programvareutvikling underveis i studiet. En av oss har hatt mer design og arkitekturteori i et ekstra fag, men lik utdanning på programvareutvikling. En i gruppa har tidligere erfaring med webutvikling i PHP og MySQL databaser, noe som har stor betydning for utvikling av webgrensesnittet. Ingen hadde nevneverdig kompetanse innen Qt, slik at kompetanseheving har vært nødvendig på området. Utvikling av GUI-applikasjoner på håndholdte enheter og embeddede systemer var i tillegg helt nytt for begge.

Valgt rammeverk for webgrensesnittet, Zend og jQuery var også nye for gruppa, og måtte læres fra scratch.

1.4 Utviklingsmodell

På bakgrunn av at prosjektets domene (KBP) er stort og at oppdragsgivers krav var uklare, var det viktig å unngå å låse prosjektet til en fast systemmodell som f.eks. fossefall hvor all kravspesifikasjon låses under implementering. Vi ville derfor ha en form for iterativ / inkrementell systemmodell med rom for revidering av kravspesifisering og / eller nye krav underveis.

For å bygge opp under dette vurderte vi først SCRUM, en systemmodell som lar utviklerne dele opp produksjon i korte intensive utviklingsintervall, som oftest mellom 14 og 30 dager. Dette ville gitt oss mulighet for en inkrementell utvikling, med mulighet for å dra med deler videre for flere iterasjoner, hvis målet for intervallet ikke ble nådd på alle deler. Men den gir lite eller ingen rom for å bruke mye tid på revidering av kravspesifikasjon underveis. Noe som forutsetter at de krav vi allerede har må være nesten ferdig definert før prosjektets start. I vår situasjon så hadde vi flere deler vi visste vi skulle ha, men pga avhengighet til hvordan andre krav ble implementert, og stor usikkerhet rundt hva vi fikk til på den håndholdte applikasjonen, ville ikke SCRUM sin ideologi fungere godt nok.

Vi valgte derfor å strukturere prosjektet etter en iterativ inkrementell systemmodell, hvor man kan definere hovedgrupper av funksjonalitet og jobbe mot når prototyper av disse skal være ferdig, for så og konstant iterere over alle inkremitter sekvensielt etter som utviklingen går fremover. På denne måten kunne vi ta høyde for forandringer underveis, som påvirket tidligere deler av prosjektet, i tillegg til eventuelt nye krav.

1.5 Organisering av rapporten

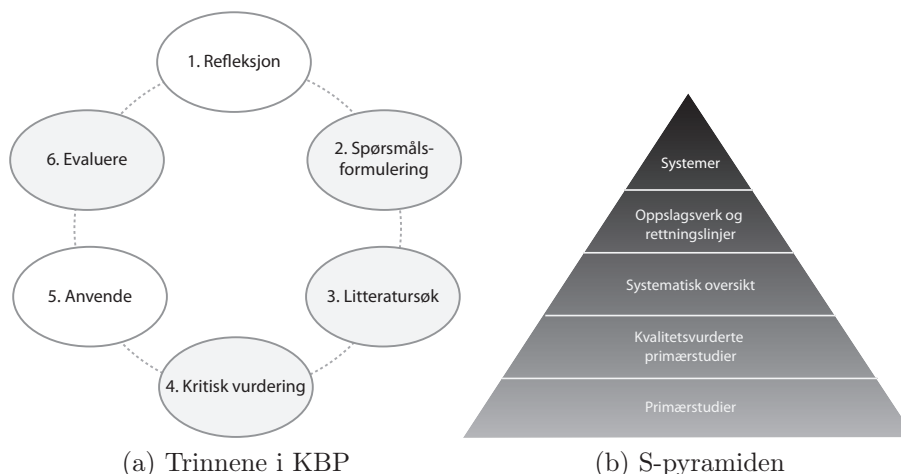
Rapporten er grovinn delt i følgende 7 kapitler:

- Introduksjon - Inneholder en introduksjon til prosjektet, og forklaring av faglig bakgrunn for prosjektets domenekunnskap.
- Kravspesifikasjon - Illustrerer de krav og retningslinjer som systemene skulle oppfylle.
- Design - Viser hvordan vi skal løse krav definert under kravspesifikasjonen.
- Realisering - Dette er selve implementasjonen, dette kapitlet fokuserer på et mer abstrakt nivå supplert med noen få viktige kodesnutter.
- Avslutning - Avsluttende kapittel om vår prosess og tanker vi sitter igjen med.
- Bibliografi - Kilder og referanser.
- Vedlegg

1.6 Faglig bakgrunn

KBP er et vidt fagfelt så vi introdusere kun det som er relevant for prosjektet. For mer detaljert informasjon om KBP kan du besøke KBP sin hjemmeside².

Som nevnt innledningsvis er KBP en kombinasjon av forskningsbasert kunnskap, egen personlig erfaring og pasientens innvirkning. Disse skal kombineres sammen til å utgjøre god kunnskapsbasert praksis.



Figur 1.2: Sentrale figurer i KBP

1.6.1 Refleksjon, spørsmålsformulering og litteratursøk

Sentralt i KBP står seks trinn som vist figur 1.2 a. Trinnene er illustrert som et sirkel fordi det er en iterativ prosess; ofte vil en gjennomføring av de seks trinnene føre til at nye spørsmål som igjen må behandles i de samme seks trinnene.

Hele prosessen starter med refleksjon som skaper grunnlaget for formuleringen av *kliniske spørsmål*. Litteraturkildene er enorme så det er avgjørende å kunne søke på en god og systematisert måte, for eksempel vil et spørsmål som er for generelt gi alt for mange resultater. Det er her definert noen systematiske teknikker for å stille kliniske spørsmål på en presis og hensiktsmessige måte. De to viktigste er PICO-søk³ og kjernes spørsmål.

²www.kunnskapsbasertpraksis.no

³(P)asient, (I)ntervention, (C)ontrol, (O)utcome

I tillegg til god utforming av søket er det også viktig å velge riktig kilde som du skal søke i. I figur 1.2 b vises *S-pyramiden* som viser hvor det er lurt å starte søket. Hovedpoenget her er at jo lengre opp i pyramiden du kommer desto bedre og sikrere er kilden fordi de er kvalitetsvurdert og oppsummert av fagpersoner eller faglige tidsskrifter. Samtidig som du beveger oppover i pyramiden blir også bredden i stoff mindre og mindre. Vis man ikke finner noe på høye nivå må man bevege seg nedover i pyramiden. Alle trinnene i pyramiden tar utgangspunkt i det nederste trinnet som består av "Primærstudier" (enkeltstudier). Dette nivået anses å være det mest usikre fordi kvaliteten på et enkelt studie kan variere sterkt.

De ulike nivåene har sine egne databaser og søkemotorer, for eksempel ligger primærstudiene i *MEDLINE*-databasen og systematiske oversikter i *Cochrane Library* databasen.

Prosjektets relevans: Det er her hovedtyngden i prosjektet ligger. Systemet skal ha funksjonalitet for å formulere og utføre søk i MEDLINE-databasen med teknikker som for eksempel PICO-formulering. Videre skal systemet forenkle resultatbehandlingen som aktuelt søkeformulering gir. Det skal for eksempel være mulig å lagre abstracts⁴ og knytte det til et klinisk spørsmål.

1.6.2 Kritisk vurdering

I dette trinnet vurderes gyldigheten av studie(ene) som er funnet. Spesielt viktig er å være kritisk til primærstudier, men man må også være kritisk selv om man bruker studier som er publisert i anerkjente tidsskrifter er det fremdeles viktig å være kritisk. Gyldigheten av forskningen må overføres til hver enkelt situasjon. For å kontrollere kvaliteten og relevansen til en artikkel er det laget sjekklister for vurdering av ulike typer studier.

Prosjektets relevans: Systemet skal ha funksjonalitet for å kvalitetssikre enkeltstudier via elektroniske sjekklister. Disse sjekklister skal bygges på sjekklister som er utgitt av Nasjonalt kunnskapssenter for helsetjenester⁵.

⁴Abstract: Sammendraget av en artikkel

⁵Sjekklister: <http://www.kunnskapssenteret.no/Verktøy/2031.cms>

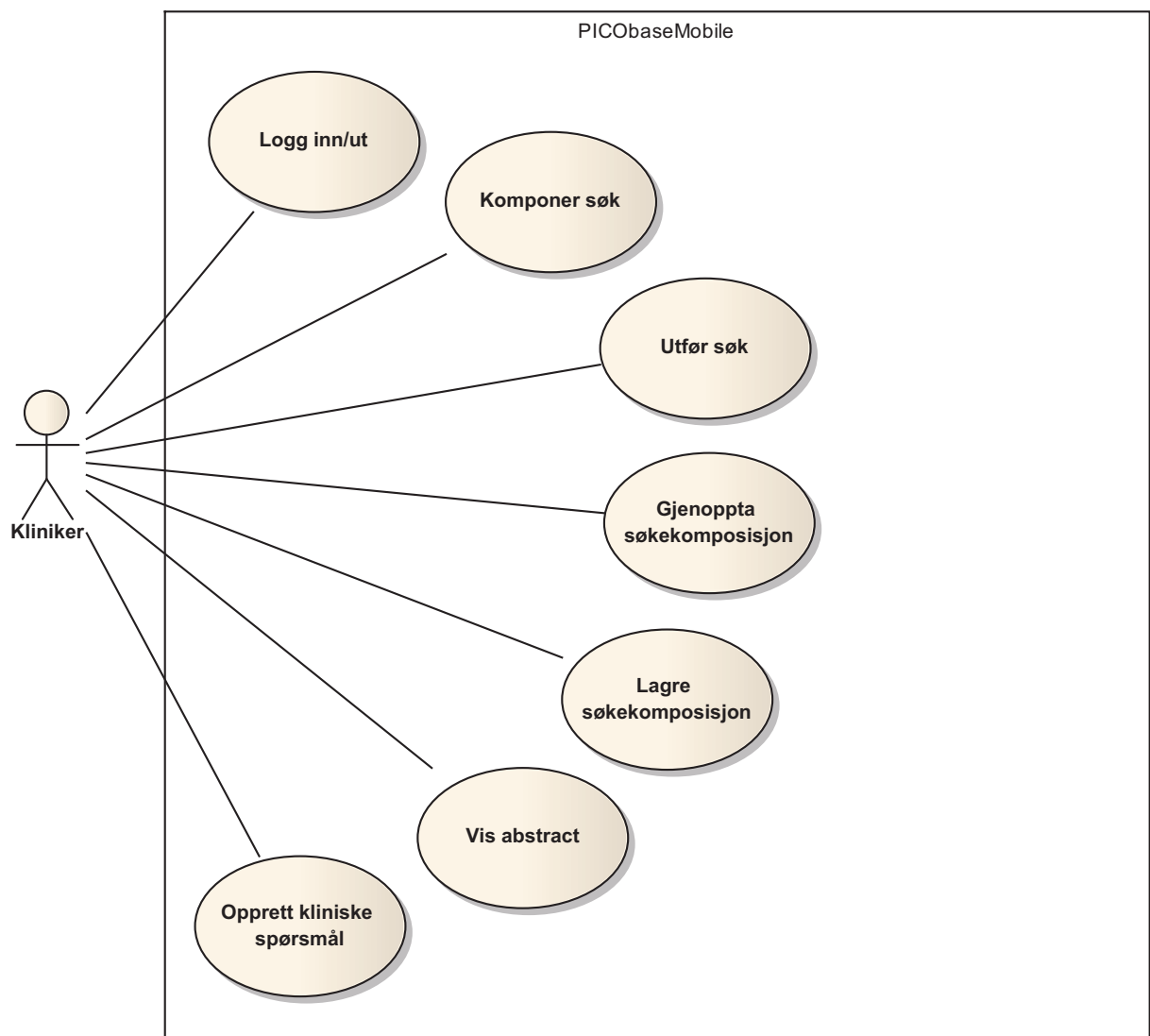
1.6.3 Anvend og evaluér

I “Anvende” trinnet skal kunnskapen fra de forskningsbaserte studiene omformes til praktisk bruk. Her må de ulike delene av kunnskapsbasert praksis (se figur 1.1) kombineres for en helhetlig vurdering på hvorvidt resultatet skal implementeres i klinikerens hverdag. Ved implementasjon er det også viktig å evaluere resultatet av en eventuell implementasjon.

Prosjektets relevans: Det skal i systemet være mulighet for å legge til et oppsummerende resymé under hvert klinisk spørsmål. Her kan klinikeren evaluere eller fritt diskutere resultatene av litteratursøket.

Kapittel 2

Kravspesifikasjon



Figur 2.1: Use Case for den håndholdte applikasjonen



Figur 2.2: Use Case for webapplikasjonen

2.1 Systemets omgivelser

2.1.1 Entrez Programming Utilities

Entrez er portalen til de medisinske databasene som PICObase skal utføre søk i etter forskningsbasert kunnskap.

Entrez er et system for å søke i flere forskjellige fagdatabaser, medisinske databaser inkludert [10]. En artikkel i en database kan også være tilknytning flere artikler spredd over flere databaser. Det er også mulig å utføre søk som går på kryss av databasene¹.

For at eksterne programmer kan ta i bruk Entrez er det utarbeidet et grensesnitt for eksterne systemer som kalle *Entrez Programming Utilities (EUtilities)*. Dette grensesnittet kan fritt aksesseres via vanlig HTTP grensesnitt eller SOAP [9].

EUtilities er bygd opp av flere “verktøy;” i hovedsak *ESearch*, *ESummary* og *EFetch*.

ESearch brukes for å søke i de ulike databasene, *ESummary* brukes for å hente artikkeloppsummeringer (docsums) og *EFetch* for å hente mer utfyllende data (abstract).

En viktig egenskap ved *EUtilities* er at *ESearch* kun returnerer primærnøkler på artikler som passet søket. Disse primærnøklerne brukes i sin tur av *ESummary* eller *EFetch* for å hente ned konkrete resultater i form av for eksempel artikkeloppsummeringer eller abstracts. Entrez kaller denne serien av “funksjonskall” for pipelines² som er kombinasjoner av de ulike verktøyene i *EUtilities*. For eksempel for å søke og hente ut resultatet i artikkeloppsummeringer må du kjøre gjennom denne pipelinen: *esearch* → *esummary*.

2.2 Systemets brukere

Systemets brukere er fordelt over to systemer, den håndholdte enheten og web-grensesnittet. De håndholdte enhetene brukes av kliniske personale mens de er i arbeid, dette for å sikre at kliniske spørsmål og / eller søk kan gjøres på kort tid uten at det går utover den ansattes arbeidsoppgaver.

¹<http://www.ncbi.nlm.nih.gov/sites/gquery>

²<http://www.ncbi.nlm.nih.gov/bookshelf/br.fcgi?book=coursework&part=utils#basic-pipelines>

Webgrensesnittet sin bruk er todelt, på den ene siden har man deltagere i kurs på kunnskapsbasert praksis, hvor kursdeltagerne bruker det som en trinnbasert læringsmodul for utøvelse av KBP. Den andre brukermassen er klinikere som er erfarne i KBP og bruker det på en mer administrativ måte med kritisk vurdering av artikler og skriving av resymér.

2.3 Overordnede funksjonelle krav

2.3.1 Use case diagram

2.3.2 Overordnede use case beskrivelser

Use Case: Opprett klinisk spørsmål

Aktør: Kliniker

Mål: Opprette nytt klinisk spørsmål.

Beskrivelse: Klinikeren oppretter nytt klinisk spørsmål i systemet. Med egen tittel og beskrivelse. Nye søkekomposisjoner legges til ved bruk av use case *Komponer søk*.

Use Case: Komponer søk

Aktør: Kliniker

Mål: Komponere et hensiktsmessig søk i forhold til det kliniske spørsmålet.

Beskrivelse: Klinikeren komponerer et søk gjennom å velge søkemethode, og type som dikterer måten søket blir bygd opp på, utført og hvordan resultatet blir fremstilt.

Use Case: Utfør søk

Aktør: Kliniker

Mål: Søke og hente resultat fra medisinsk database.

Beskrivelse: Initieres av use case *Komponer søk* og søker i medisinsk(e) database(r) med den komponerte søkestringen.

Use Case: Lagre søkekomposisjon

Aktør: Kliniker

Mål: Lagre uferdige komposisjoner og / eller gode komposisjoner man vil ta vare på.

Beskrivelse: Klinikeren bygger opp søkekomposisjonen gjennom *Komponere søk* use caset, men initierer denne hvis klinikerens vil lagre det han har gjort til nå og / eller har brukt komposisjonen og vil ha mulighet for å søke med samme komposisjon om igjen uten å måtte lage den på nytt.

Use Case: Kritisk vurderer

Aktør: Kliniker

Mål: Kritisk vurdere en studie / artikkel ved bruk av elektronisk sjekkliste i henhold til studiedesignet som er brukt i studien / artikkelen.

Beskrivelse: Klinikeren arbeider seg gjennom sjekklisten, punkt for punkt og for hvert punkt avgis et svar på kontrollspørsmålet; enten via avkrysning eller kommentar. Use caset avsluttes ved at bruker eksplisitt lagrer vurderingen.

Use Case: Gjenoppta søkekomposisjon

Aktør: Kliniker

Mål: Gjenoppta søkekomposisjoner som er lagret.

Beskrivelse: Klinikeren ser gjennom en liste med sine lagrede søkekomposisjoner, søkekomposisjonen som ønsket gjenopptatt aktiveres og systemet fyller inn aktuelle søkefelt i use caset *Komponer søk*.

Use Case: Administrer resymé

Aktør: Kliniker

Mål: Opprette og vedlikeholde resyméer av studier / artikler.

Beskrivelse:

1. Klinikere gis mulighet til å opprette resyméer.
 2. Klinikeren ser gjennom en liste med allerede opprettede resyméer og kan endre, slette og lese enkeltresyméer.
-

Use Case: Administrer brukere

Aktør: Administrator

Mål: Opprette og vedlikeholde brukere i systemet

Beskrivelse:

1. Administrator oppretter ny bruker ved å angi personalia og annen brukerinformasjon, systemet generer passord og sender dette til bruker.
2. Administrator sletter eller blokkerer bruker direkte i brukerlisten.

Use Case: Endre personalia

Aktør: Kliniker

Mål: Endre personalia og brukerinformasjon på egen bruker

Beskrivelse: Kliniker endrer ønsket informasjon om seg selv og lagrer.

Use Case: Godkjenn / publiser resymé

Aktør: Veileder

Mål: Godkjenn et resymé til publisering på PICObase sin presentasjonsside

Beskrivelse: Når veileder leser et resymé i use caset *Les interne resymé* gis en identifisert veileder muligheten til å godkjenne aktuelt resymé for publisering. Systemet publiserer dette resyméet for allmennheten.

Use Case: Les publisert resymé

Aktør: Nettbruker

Mål: Nettbruker gis mulighet til å se på publiserte resymé

Beskrivelse: Nettbruker ser gjennom en oversikt og ønsket resymé presenteres med studierelatert data.

Use Case: Les interne resymé

Aktør: Veileder

Mål: Gi tilgang til alle resyméer, både offentlige og interne.

Beskrivelse: Veileder ser gjennom en oversikt og ønsket resymé presenteres med studierelatert data.

Use Case: Logg inn / ut

Aktør: Kliniker, veileder, administrator

Mål: Autentifisere brukere og tilgjengeliggjøre lagret data (som søkekomposisjoner, resymé, forumposter, med mer).

Beskrivelse:

1. Bruker logger seg inn med brukernavn og passord. Systemet autentifiserer bruker.
 2. Bruker logger av. Systemet “glemmer” brukeren.
-

Use Case: Vis resultat

Aktør: Kliniker

Mål: Vise resultatet etter et søk på en hensiktsmessig måte i forhold til søkemetode, søketype og andre innstillinger.

Beskrivelse: Use caset viser et resultatsett basert på brukt søkekomposisjon, kliniker kan så velge en artikkel i resultatsettet for å se på abstract. På PICObaseWeb kan kliniker velge å lagre abstractet, lagre en kommentar, eller kritisk vurdere (initierer use caset *Kritisk vurder*) funnet.

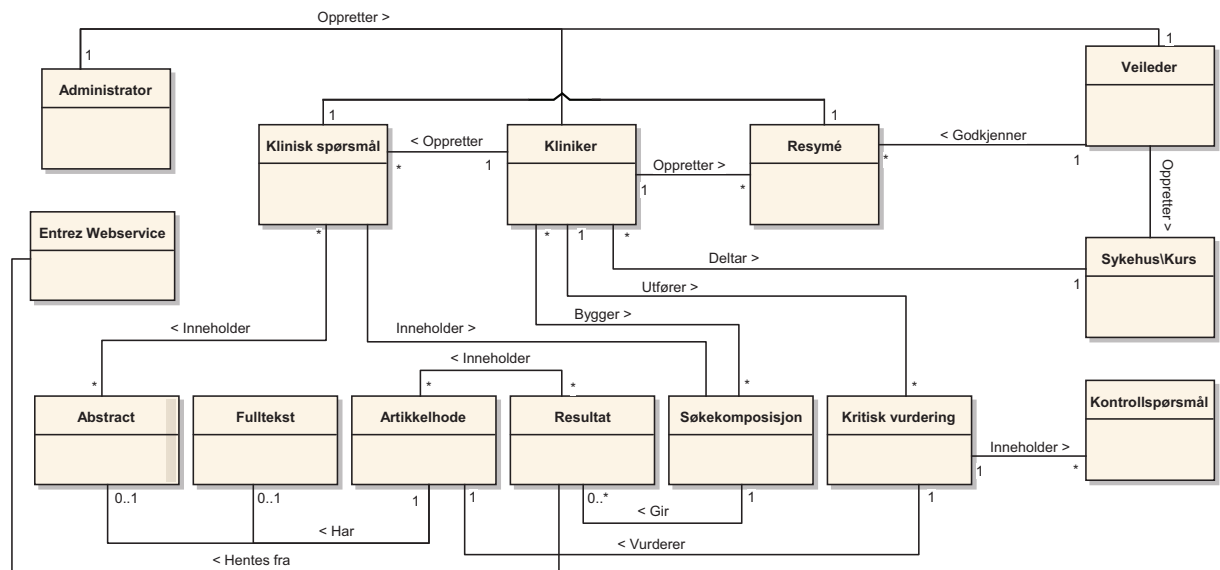
Use Case: Vis artikkel i fulltekst

Aktør: Kliniker

Mål: Vis hele artikkelens innhold og ikke bare artikkelhode / abstract

Beskrivelse: Hvis en artikkel er tilgjengelig i fulltekstformat vises den i ønsket format (hvis kun url, så åpnes denne i et browservindu).

2.4 Domenemodell



Figur 2.3: Domenemodell

Domene som beskrives er i hovedsak knyttet til *forskningsbasert kunnskap* i tillegg til noen administrative aktiviteter som for eksempel oppretting av kurs og klinikere. Den sentrale aktøren er klinikeren, alle handlinger som er direkte knyttet til kunnskapsbasert praksis utføres av denne aktøren.

Kliniske spørsmål besvares av en eller flere artikler. Disse artiklene består av 3 deler; et artikkelhode, et abstract og artikkelen representert i fulltekst.

Entrez Webservice er grensesnittet til de medisinske databasene hvor artiklene ligger.

Artikkelhode inneholder metainformasjon om artikkelen, slik som tittel, forfatter utgivelse osv. Når søk utføres, er det artikkelhodene som kommer opp i resultatlisten.

Et **abstract** er en oppsummeringen av en artikkel og inneholder vanligvis elementer som bakgrunn, problemstilling, resultat og konklusjon. Det er imidlertid ikke alle artikler som har abstractet tilgjengelig, dette er i så fall angitt i artikkelhode. For de artiklene med tilgjengelig abstract kan dette hentes inn via Entrez webservicet.

Fulltekst er artikkelens fulle versjon. Fulltekst er ikke tilgjengelig gjennom Entrez webservicen pga. ulike rettigheter knyttet til utgiveren av artikkelen.

Artiklene finnes gjennom ulike **søkekomposisjoner** (PICO-søk, fritekst etc). Søkekomposisjoner gir så et resultat i form av artikkelhoder.

Kritiske vurderinger brukes i kunnskapsbasert praksis for å kontrollere kvaliteten og relevansen til en artikkel. Hvert skjema består av ulike kombinasjoner av såkalte kontrollspørsmål, som enten er en tekstlig kommentar, eller avkryssingsfelt.

Et **resymé** er en oppsummerende drøfting av et kliniske spørsmål. Her kan klinikeren presentere sine egne konklusjoner, oppdagelser, drøftinger på det kliniske de kliniske spørsmålene.

Det blir gjennomført **kurs** i kunnskapsbasert praksis, deltagerne av disse kursene er klinikere. Det er veilederen som oppretter disse kursene og definerer hvilke klinikere som er deltagere.

2.5 Supplementær spesifisering

2.5.1 Funksjonalitet

Sikkerhet

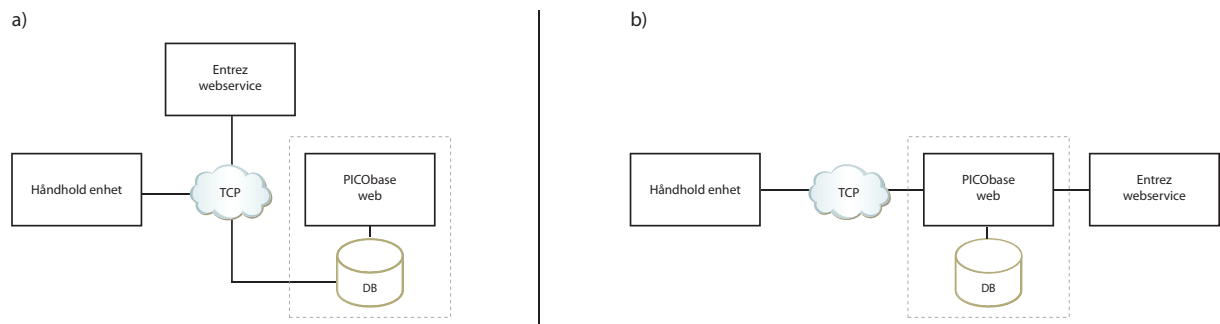
Autentisering- og autorisasjonssystemet i PICObase må håndtere følgende problemstillinger:

- Autentisering av webbrukere
- Autentisering av brukere via håndholdt enhet
- Autorisering mot ressurser som
 - Kliniske spørsmål
 - Kategorier
 - Resymé
 - Søkekomposisjoner
 - Lagrede abstracts
 - Brukerprofiler

Videre skal det kunne styres på roller hvorvidt en bruker har tilgang til en ressurs eller ikke. Det skal også være mulig å begrense en brukers tilgang til ressurser med eget eierskap.

Kapittel 3

Design



Figur 3.1: a) Selvstendig håndholdt b) Webapplikasjon overtar felles problematikk

3.1 Arkitektur

Systemet skal ha to grensesnitt; et webgrensesnitt og et eget grensesnitt gjennom håndholdt applikasjon. Begge grensesnittene skal operere på den samme databasen og de samme dataene.

Siden vi har to forskjellige grensesnitt med to forskjellige programmeringsprinsipper vil vi måtte løse og implementere mye av funksjonaliteten to ganger. Dette er noe vi i størst mulig grad vil forhindre. Strategien er å flytte felles domeneproblematikk over på en sentral plassering. I stede for at den håndholdte enheten selv tar ansvaret for operasjoner som manipulering av databasen, kommunikasjon mot Entrez webservice osv., vil disse domeneoperasjonene kjøres gjennom et felles grensesnitt som illustrert i figur 3.1 Den beste plasseringen for dette grensesnittet vil være på samme server og samme programvare som det øvrige webgrensesnittet. Vi legger altså opp til at webapplikasjonen også skal fungere som en interaktiv “gateway” for den håndholdte enheten.

En annen positiv effekt av denne strategien er at den håndholdte enheten blir “koblet løs” fra databasen og vi kan gjøre endringer uten at det påvirker de håndholdte enhetene. Dette er en stor fordel siden den håndholdte applikasjonen er distribuert blant mange brukere. Skulle det være nødvendig å gjøre endringer i databasen trenger man kun å forholde seg til grensesnittet som den håndholdte applikasjonen bruker.

3.1.1 Autentisering og autorisasjon

Autentisering er prosessen med å fastslå identiteten til brukeren. **Autorisasjon** er i neste omgang prosessen med å verifisere at aktuell bruker har tilgang til forespurt ressurs [1, side 851].

Fra kravspesifikasjonen har vi at vi skal autentisere brukere fra to grensesnitt; webgrensesnittet og håndholdt applikasjon. Disse to grensesnittene skal begge ha tilgang til de samme dataene som ligger i databasen. Den håndholdte enheten skal for eksempel ha tilgang til de samme kliniske spørsmålene som webgrensesnittet. I avsnitt *3.1 Arkitektur* introduserte vi et felles grensesnitt for den håndholdte applikasjonen og webgrensesnittet. Dette klare “kuttet” i arkitekturen er en velegnet plassering for autentisering- og autorisasjonssystemet. Legges sikkerhetsmekanismen her kan vi gjøre bruk av den samme mekanismen (samme kode) for klarering av brukere.

Autentisering

Dette designet forutsetter imidlertid at klareringsmekanismen kan håndtere flere forskjellige måter å autentisere på og velge passende metode dynamisk i henhold til hvilke grensesnitt som er aktivt. For eksempel vil det være naturlig å autentiseres webgrensesnittbrukere via HTML-skjema, mens fra en håndholdt vil det være mer aktuelt å bruke HTTP autentisering.

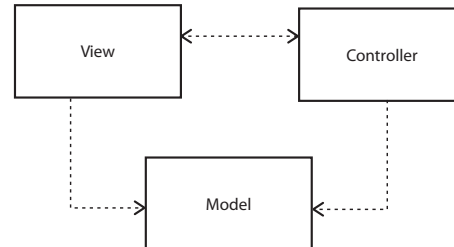
Autorisering

Hvis designet som er skissert ovenfor møtes trenger ikke autoriseringsmekanismen å vite noe om hvilke grensesnitt som er i bruk. Målet er at på det tidspunktet autoriseringen av bruker starter er brukeren allerede autentisert og identifisert.

3.2 Overordnet design

3.2.1 Webapplikasjon

Et av hovedmålene for designet bak webapplikasjonen er å skille applikasjonslogikk, domenelogikk og presentasjon på en hensiktsmessig og god måte for å ha en applikasjon som er lett å vedlikeholde. Et annet mål er å skille presentasjonskode (HTML/CSS) fra programkode (PHP, ASP etc.) for å få en mer homogen og ren presentasjonskode som lettere kan vedlikeholdes. Videre vil det også bidra til man tvinger applikasjonslogikk vekk fra presentasjonslaget.



Figur 3.2: MVC - *Model View Controller* pattern

Det er flere måter å oppnå disse rene skillene i applikasjonen på, for eksempel kunne man brukt *Template View* [2, side 350] til å skille presentasjonskode og applikasjonskode. Det løser imidlertid bare deler av problemet, vi trenger et design som også skiller applikasjonslogikk fra domenelogikk. For å løse dette har vi let oss frem til *Model View Pattern (MVC)* [2, side 330][4, side 4] som et direkte svar på problemstillingen.

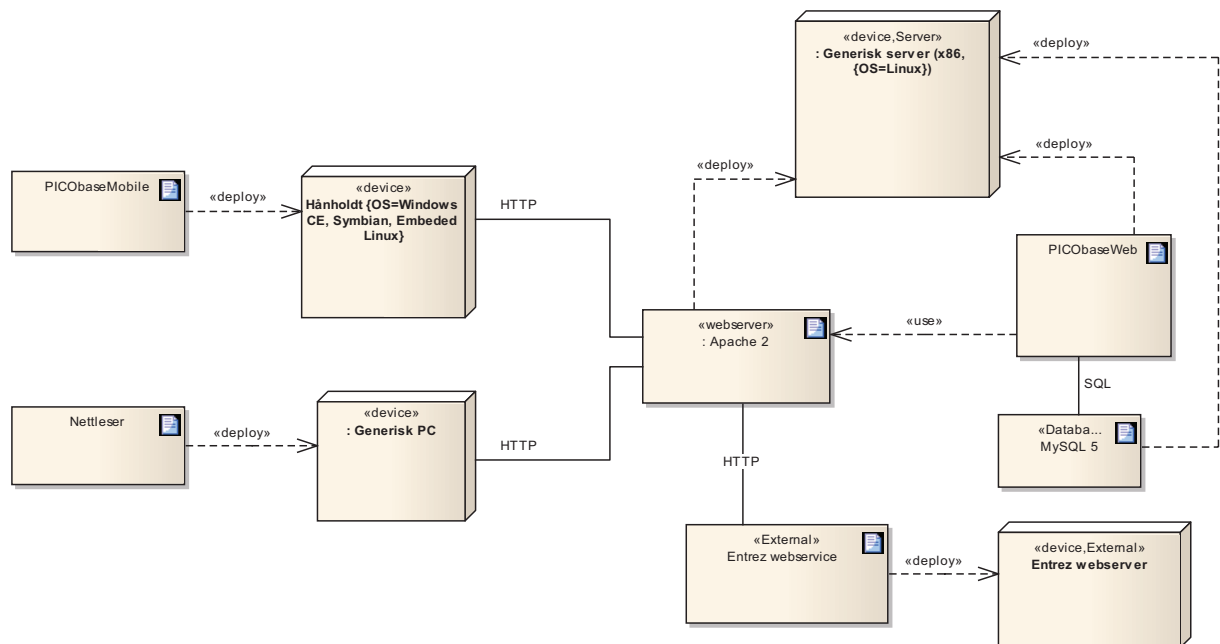
3.2.2 Autorisering

Det er viktig at autoriseringen hverken ligger i *controllerene* eller modellene. Grunnen til det er at vi ønsker å unngå å distribuere sikkerheten utover i applikasjonen, noe som ville ha gitt større sjanse for en eller annen sikkerhetsglipp. Dette i tillegg til den økte kompleksiteten som autentisering av brukere fra to grensesnitt introduserer (se 3.1.1) vil vi være tjent med en sentralisert sikkerhetsmekanisme.

Vi tar utgangspunkt i *Action Control List (ACL)* [6, side 624] for autoriseringen av brukerne. Vi trenger derfor en uniform måte å identifisere en ressurs og operasjonen som skal utføres på den aktuelle ressursen. Slik kan sikkerhetssystemet bruke ressurs, operasjon og (eventuelt) innlogget brukerinformatjon til å autorisere på.

For å få til en 100 % sentralisering vil det være aktuelt å bruke *Front Controller* [2, side 344] patternet for få et inngangspunkt for hele webapplikasjonen.

3.3 Deployment View



Figur 3.3: Deployment diagram

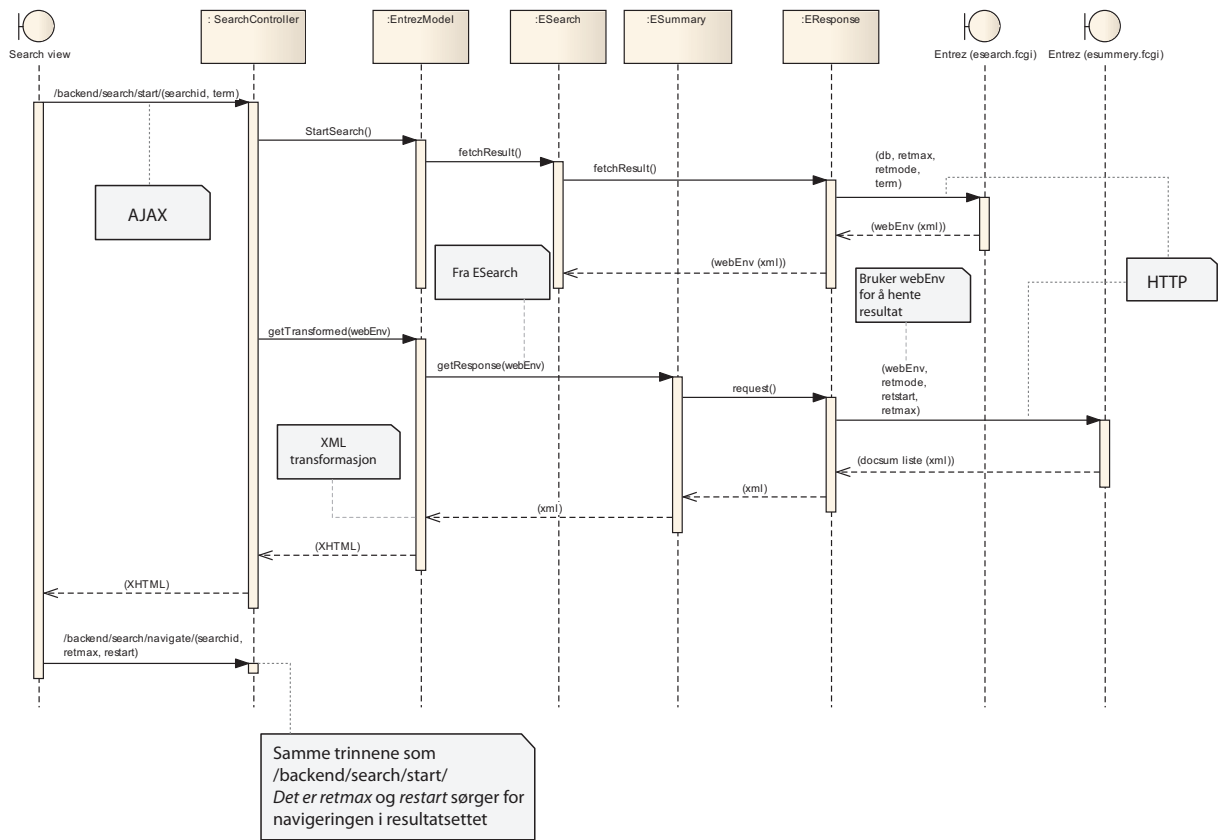
Diagrammet viser den fysiske plasseringen av de konkrete programvarene og fysiske noder som server og håndholdt. Som introdusert i seksjon 3.1 er webapplikasjonen senteret for hele systemet. Dette medfører at alle fysiske noder i systemet leder til webserveren som hoster webapplikasjonen.

Hvorvidt database serveren og webserver kjører på samme fysiske node er ikke kritisk for designet. Ofte vil disse to tjenestene ligge på hver sin fysiske node.

*Entrez webservice*¹ er grensesnittet til de medisinske databasene. De aksesseres via HTTP protokollen som antydnet i kommunikasjonslinjen. Dette grensesnittet brukes til å hente ut de medisinske artiklene (se seksjon 2.3).

¹Entrez Programming Utilities

3.4 Sekvens diagram



Figur 3.4: Sekvens diagram som viser kommunikasjon mellom webapplikasjon og Entrez webservice ved henting av *docsum-liste*

Figur 3.4 viser hvordan kommunikasjonen mot Entrez kan bygges opp. Det det er tatt utgangspunkt i en MVC implementasjon (se 3.2.1) jf. *Search View*, *SearchController* og *EntrezModel*.

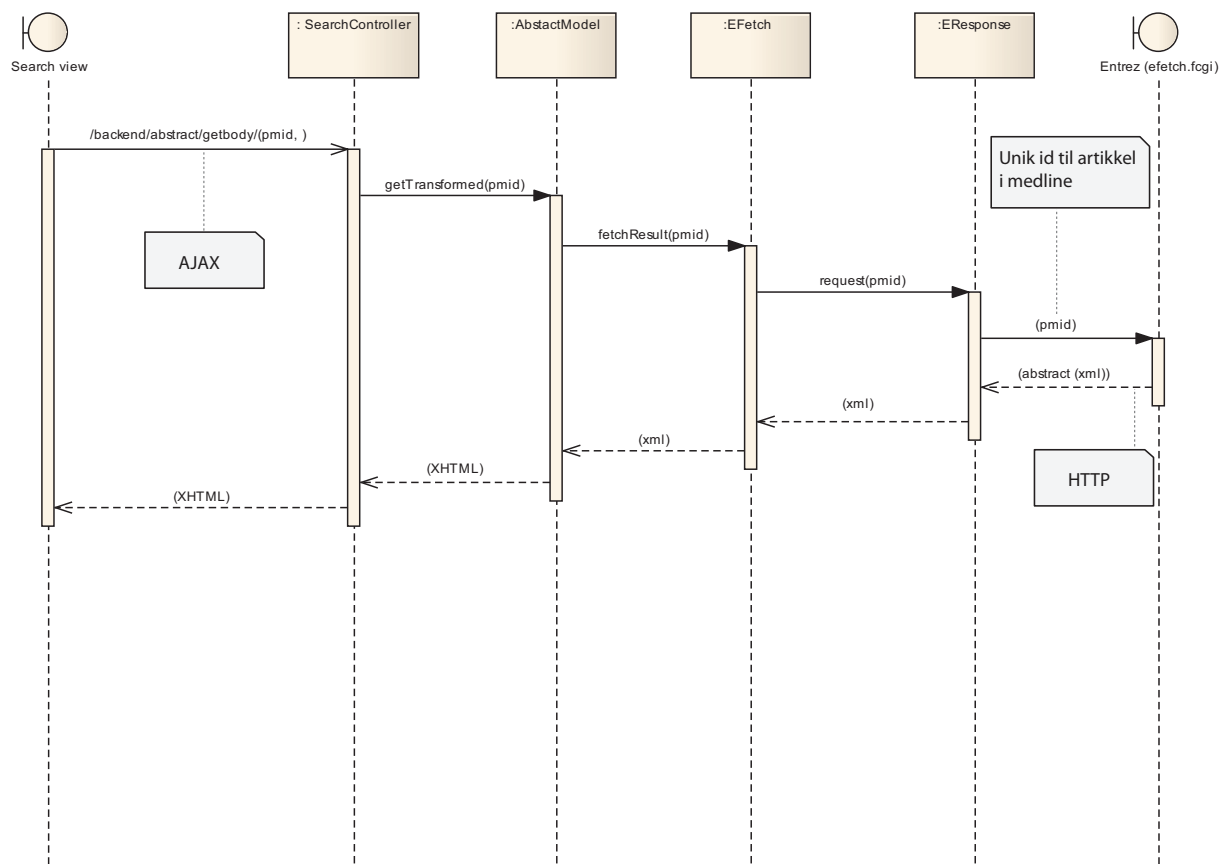
Det tas utgangspunktet i denne pipelinen: *esearch* → *esummary*

EResponse (i diagrammet) er tenkt å være en felles transport-og-retur-klasser for både ESearch og ESummary. For eksempel kunne en løsning være at ESummary klassen kunne ta en URL som parameter slik at den kunne brukes pekes til ønsket URL av klassen som aggregerer på den. I all hovedsak trenger ikke EResponse vite noe hva den kan komme til å inneholde, den trenger kun å koble seg til Entrez og returnere responskroppen.

Sekvensen starter ved at brukergrensesnittet (Search View) sender en forespørsel

(via Ajax) til SearchController som delegerer videre til EntrezModel. EntrezModel bruker i sin tur et ESearch objekt for å starte søket. ESearch bruker så videre et EResponse objekt til å utføre den faktiske forespørselen til Entrez sitt *esearch* grensesnitt. EResponse objektet fører ut forespørselen og henter tilbake en liten xml-struktur med blant annet *webEnv*. Denne xml-strukturen returneres så tilbake til ESearch objektet som lagrer unna *webEnv* slik at ESummary kan bruke denne nøkkelen til å hente resultatet med. På dette tidspunktet her kan vi bruke den returnerte *webEnv* nøkkelen til å hente ned resultatene i form av artikkeloppsamminger. Vi vil ha resultatet som XHTML så vi ser for oss at vår EntrezModel har funksjonalitet for dette. Vi bruker ESearch sin *webEnv* som argument for å hente ned resultatene. Nå er det ESummary som bruker EResponse for å hente ned informasjon fra Entrez, nå gjennom *esummary.fcgi*. Gangen blir her den samme, EResponse får tilbake en XML struktur som ESummary gir tilbake til modellen som transformerer til XHTML.

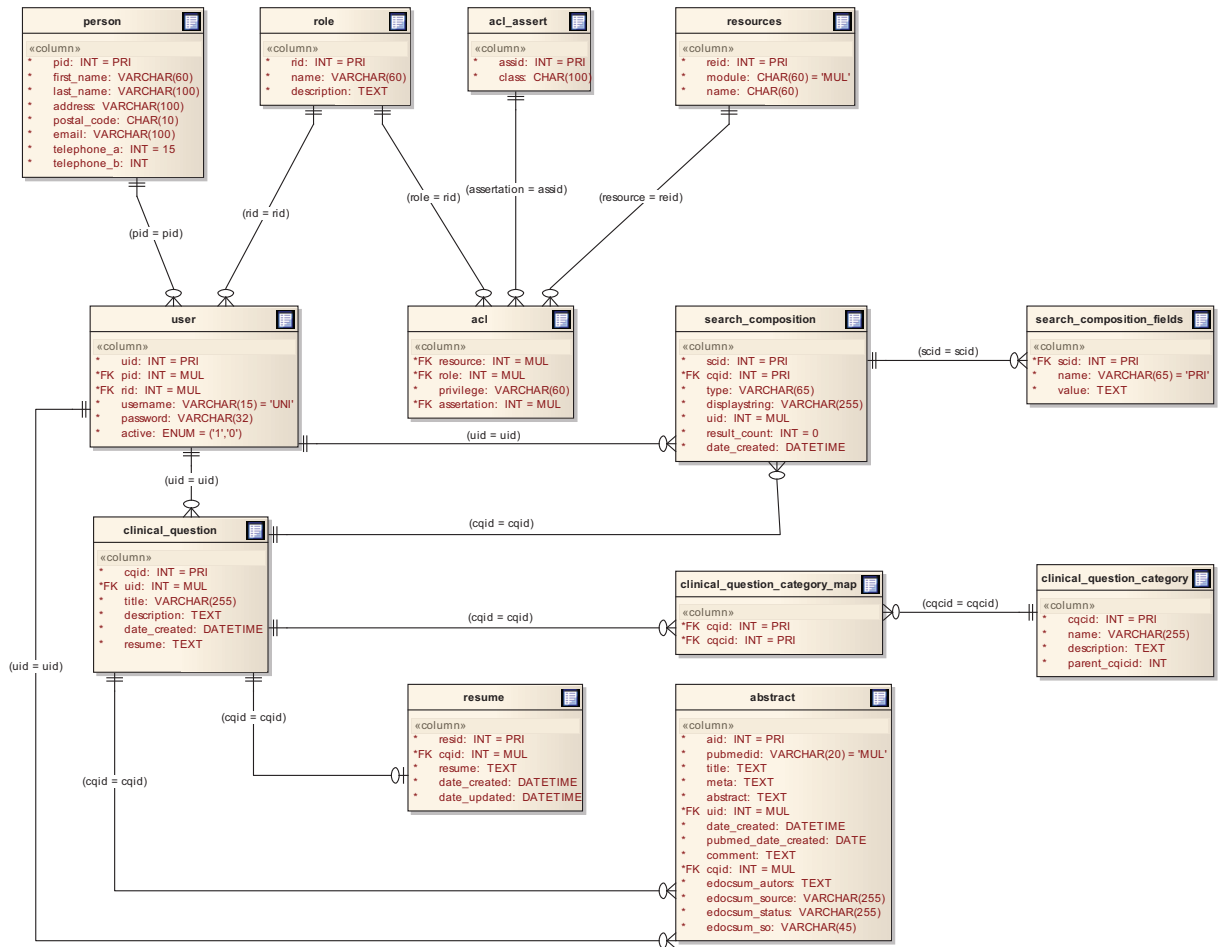
I implementasjonen må vi også ta høyde for navigering og konfigurering, men denne sekvensen av interaksjon mellom de ulike objektene kommer til å være sentral i realiseringen.



Figur 3.5: Sekvens diagram som viser kommunikasjon mellom webapplikasjon og Entrez webservice ved henting av et enkelt abstract

Vi legger også ved et sekvensdiagram over hvordan henting av et abstract kan tenkes (se figur 3.5) . Da får vi følgende pipeline: *esearch* → *efetch*

3.5 Data View



Figur 3.6: Databasemodell

3.6 Designutfordringer

3.6.1 Lagring av kliniske spørsmål på håndholdt

Problemstilling:

Det må designes en optimal løsning for lagring av kliniske spørsmål på håndholdte enheter som gir et effektivt format for synkronisering med webgrensesnittet. Formatet bør også være så plattformuavhengig, plassbesparende og skalerbart som mulig. Det skal også være enkelt å legge til nye måter å søke på (søkekomposisjonstyper), uten at lagringsstrukturen må endres betydelig.

Vurdering:

Et problem med håndholdte enheter er begrenset lagringskapasitet, vi ville derfor ha minst mulig lokal lagring. Vi så på å ha en helhetlig runtime løsning i klient / tjener arkitektur med server, og lagde en slik applikasjon i Qt og vurderte å ha en Qt server backend for overføring. Dette ville gitt oss mulighet for et proprietært og sikkert objektformat. Men låst oss til Qt på server og fjernet muligheten for et plattformuavhengig overføringsinterface. Samtidig så finnes det få portable og lite krevende databaseløsninger for håndholdte, vi måtte derfor finne en annen måte å lagre kliniske spørsmål lokalt på. En annen vurdering var å ha en form for aktiv fil med automatisk lagring ved jevne mellomrom for å hindre tap av data ved batteritap o.l. En multithreading løsning ble derfor brukt en del tid på for å teste muligheten for dette. Med autolagring og autosynkronisering mot webgrensesnittet i brukerdefinerte tidsrom.

Løsning:

I vår endelig løsning valgte vi å bruke XML, et standardisert format som er lett å håndtere, validere, og overføre mellom applikasjoner som et unisont format. Noe som var viktig i forhold til synkronisering. Men den største fordelen er skalerbarheten og den lave størrelsen. Dette fordi outputten vi skal lagre i seg selv er ren tekst, og XML er et tekstbasert format.

3.6.2 Synkronisering mellom web og håndholdt

Problemstilling:

Synkronisering av kliniske spørsmål mellom håndholdte og webgrensesnittet er viktig at skjer på en stabil, og mest mulig feilfri måte. Synkroniseringen må sikres mot redundant lagring av data, og feiloppdatering av enkelt elementer. Samtidig så må kladd lagret lokalt på håndholdte som skal synkroniseres, fjernes etter vellykket oppdatering, og gi feilmelding til bruker hvis oppdateringen mislyktes. Det er i hovedsak to spørsmål som må besvares:

- “Når og hvordan skal overføringen skje?”
- “Hva slags data skal overføres mellom systemene?”.

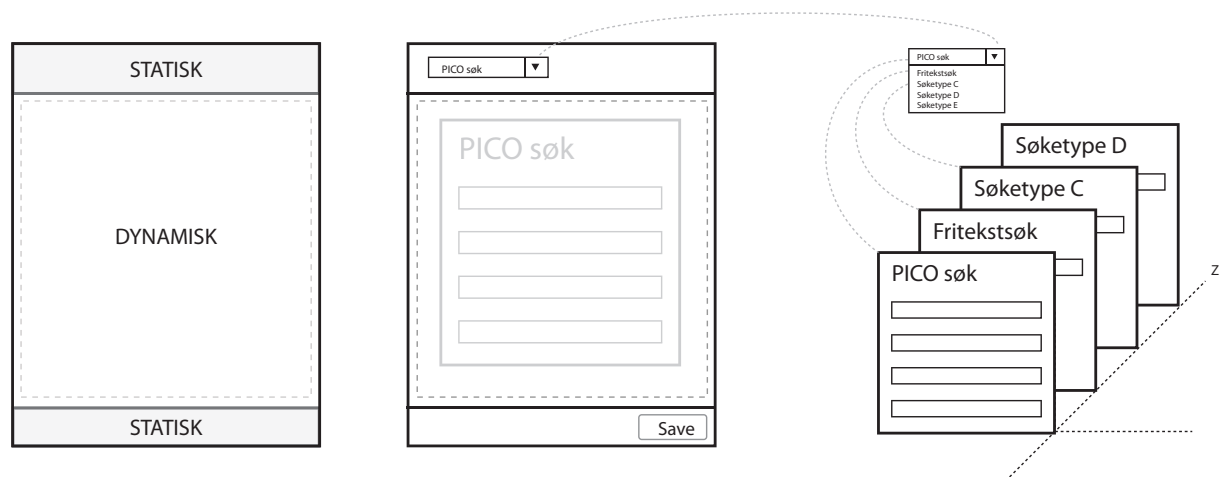
Vurdering:

En bruker må ha tilgang til alle sine kliniske spørsmål, både lagret lokalt og på server for effektivt å bruke det håndholdte systemet. En vurdering her var da å la brukeren hente inn alle kliniske spørsmål fra server, noe som ved oppskalert bruk kan vise seg å ta for lang tid og / eller fylle opp minnet til den håndholdte. Det kan derfor være en idé og kun hente en liste basert på tittel av de lagrede spørsmålene, for så å hente kliniske spørsmål per request, fremfor å laste alle inn i systemet. Synkronisering fra håndholdt til server kan skje på flere måter, det har blitt vurdert en autosynkroniseringsmetode, som brukeren selv kan velge tidsintervall på.

Eller en eksplisitt brukersynkronisering hvor brukeren selv bestemmer når han vil synkronisere. Ved offline situasjoner ble det vurdert en form for utboksfunksjonalitet lik mailklienter, hvor systemet operer som før, men gir beskjed om at man er i offline modus, og sender automatisk alt i utboksen når enheten er online igjen.

Løsning:

Etter lenge å ha holdt på en utboksfunksjonalitet med automatisk trådstyrt synkronisering og automatisk sjekk om man er i online eller offline modus. Hellet vi til slutt mot en brukereksplisitt løsning hvor brukeren selv avgjør om det kliniske spørsmålet skal lagres lokalt som kladd, eller synkroniseres direkte mot server. Brukeren kan da gjenopprette en kladd og synkronisere denne senere når han er fornøyd med definisjonen. Dette gir mindre overhead, særskilt på håndholdte enheter med mulighet for multitasking, hvor brukerne ofte lar applikasjonene kjøre lenge om gangen i bakgrunnen. En annen faktor er den antatte bruken, hvor det er mer hensiktsmessig å jobbe på ett og ett klinisk spørsmål og så enten lagre som kladd, eller synkronisere direkte. I stedet for en batchorientert arbeidsmetode, hvor en utboksfunksjonalitet ville vært mer effektiv.



Figur 3.7: Illustrasjon av søkekomposisjonsdesign.

3.6.3 Støtte for flere søkekomposisjonstyper samtidig på håndholdt

Problemstilling:

Det er essensielt for brukerverdien av applikasjonen at den håndholdte har mulighet for flere måter å søke på (søkekomposisjonstype). Denne skal også ha mulighet for lagring / gjenoppretting av lagrede søkekomposisjoner for hver type. Presentasjonen må være brukervennlig og samtidig gi god mulighet for skalerbarhet på generelt nivå.

Vurdering:

Siden det er umulig å vite hvordan fremtidige søkekomposisjonstyper skal se ut og fungere, så må designet gi rom for at typen kan kreve noe helt annet enn de to som vil følge med prototypen (PICO og fritekst). Dette i form av plass, oppsett og virkemåte (businesslogikk). Vi brukte derfor mye tid på å prøve og se for oss hvordan dette skulle se ut. En førende faktor var at på håndholdt så har man begrenset plass i form av lite mulighet for sidestilte vinduer og andre typer for samtidig design. Slik at tanken om å fremstille søkekomposisjonstypene samtidig på skjermen ble forkastet tidlig. Det måtte også vurderes om de skulle fremstilles som egne widgets, eller som en form for funksjonskall som endret den visuelle fremstillingen avhengig av hva som var valgt. Andre føringer var om de skulle fremstå som egne tabs, eller om det var en form for overskrivelse av den eksisterende tab'en.

Løsning:

Illustrert av *figur 3.7* går løsningen ut på å lage en dynamisk plassholder for søkekomposisjonene, med statisk topp og bunn. Dette ved å ha en basewidget med topp og bunn og en sentrert containerwidget som kan holde et udefinert antall widgets. Hver søkekomposisjon er sin egen widget. Dynamikken skjer dermed at den statiske toppen holder rede på hvilken søkekomposisjon som til enhver tid er i fokus. Mens bunnen lytter på operasjoner som save og slett. Det er derfor mulig å skalere opp til nye søkekomposisjoner uten å vite hva slags format den har.

Kapittel 4

Realisering

4.1 Utviklingsmiljø og språk

4.1.1 Webapplikasjon

Siden vi sto fritt til å velge utviklingsmiljø og språk på webapplikasjonen var gruppens tidligere erfaringer (se seksjon 1.3) førende. Vi endte derfor opp med å basere oss på en PHP implementasjon i et Linux basert miljø med Apache 2 som webserver.

Valg av PHP rammeverk

Vi var i starten usikre på hvilken tilnærming vi skulle ta på webapplikasjonen; skulle vi starte helt fra bunnen av, eller bygge applikasjonen oppå et eksisterende rammeverk?

Under designdelen (se seksjon 3.2.1) la vi sterke føringer på hvordan designet skulle være i webapplikasjonen. Vi var ganske beviste på at vi ville organisere webapplikasjonen etter Model View Controller patternet [2, side 330] for å skille domene-logikken fra applikasjon og presentasjonskoden. Vi tok utgangspunkt i denne føringen og samlet sammen noen rammeverk som baserte seg på MVC. Rammeverkene vi vurderte var *CodeIgniter*, *CackePHP* og *Zend Framework*.

Zend Framework fremstod som det beste alternativet i forhold til prosjektet vårt; det har for eksempel flere pakker som dekker sentral funksjonalitet som autentisering¹, autorisering² og webservice³. I tillegg er det veldig fleksibelt; du kan selv diktere hvordan applikasjonstrukturen skal se ut.

4.1.2 Database

Fra tidligere fag på Hig har vi brukt både InterBase og MySQL, begge har god støtte i PHP.

Det er viktig at databasen har støtte for å definere fremmednøkler og transaksjonstøtte slik at databasen kan sikre dataintegritet. Både InterBase [7] og MySQL [1, side 262] har disse egenskapene.

Til slutt valgte vi å bruke MySQL siden vi har mer praktisk erfaring med MySQL i

¹Zend_Auth: <http://framework.zend.com/manual/en/zend.auth.html>

²Zend_Acl: <http://framework.zend.com/manual/en/zend.acl.html>

³Zend_Rest: <http://framework.zend.com/manual/en/zend.rest.html>

kombinasjon med PHP. Dessuten sto vi også mye friere i valg av et PHP rammeverk da MySQL har bredere støtte blant PHP rammeverk. For eksempel har populære rammeverk som CakePHP ⁴ og CodeIgniter ⁵ ikke støtte for InterBase.

4.1.3 Håndholdt enhet

Utvikling på håndholdte enheter er betydelig annerledes enn å utvikle standard desktopløsninger. Dette gjorde at det måtte taes hensyn til problemstillinger som portabilitet, størrelse, ytelse, og funksjonelle muligheter i utviklingsmiljøet og språket vi skulle bruke. I tillegg så måtte applikasjonen fungere på enheter med Windows Mobile 5.0 (WM5.0) og nyere, da det er det som brukes mest hos oppdragsgiver.

Det ble derfor sett på hvilke programmeringsspråk og utviklingsmiljø som kunne fungere på slike enheter. Da kom vi frem til at C#⁶ med WM5.0 SDK⁷, Java ME⁸, Qt⁹, og en webløsning for mobile enheter ble vurdert som aktuelle for videre utprøving.

Vi valgte å se på native C# og Java ME først, fordi vi hadde mest erfaring med Javautvikling fra før og C# fordi en nativeapplikasjon er enklere å implementere. Det ble laget en testapplikasjon i C# for håndholdte som demonstrerte enkel bruk av GUI og en database komprimert til bruk i embeddede / constrainede systemer. En fordel var direkte implementasjon av Visual Studio sitt forms system med WM, noe som forenkler GUI-prosessen betraktelig og gir bedre kontroll over koblingen mellom presentasjons- og businesslaget. Deployment fungerte også i én enkel prosess, både som preview og hvis en enhet var koblet til PC'en - direkte på enheten.

Under research av Java ME fant vi raskt ut at for å få denne til å fungere på WM så måtte det brukes en tredjeparts JVM [fotnote: Java Virtual Machine] for å kjøre bytecoden til Java. Dette fordi det ikke fantes noe native JVM for WM / CE. Dette fører til restriksjoner i kodebasen skreddersydd til den enkelte JVM'en. Siden man da i tillegg er avhengig av uoffisielle JVM'er med forskjellige subset av Java sin totale funksjonalitet for hver type enhet, ble det i samråd med veileder vurdert å ikke bruke Java ME.

En helhetlig webløsning på håndholdt ville gjort at vi kunne samkjøre webgrens-

⁴<http://www.phpframeworks.com/php-frameworks/index.php?id=2>

⁵<http://www.phpframeworks.com/php-frameworks/index.php?id=9>

⁶C#: <http://msdn.microsoft.com/en-us/vcsharp/aa336809.aspx>

⁷Software Development Kit

⁸Java Platform Micro Edition:

⁹www.qtsoftware.no

esnippet på serversiden og applikasjonen på den håndholdte bedre. Med et unisont design og gjenbruk av businesslogikk. Men etter research på browserteknologi for håndholdte, og egen erfaring fra portabilitetsproblematikk og lik brukeropplevelse på forskjellige browsere. Ble det vurdert at det ville være for problematisk å tune applikasjonen til å fungere godt nok uavhengig av browser for håndholdte enheter. Dette i tillegg til at ikke alle browsere i dagens smarttelefoner og PDA'er har god nok støtte for javascript til å bygge opp under et effektivt brukergrensesnitt.

Ingen av oss hadde programmert i Qt på håndholdte / embeddede systemer og visste lite om hvordan dette fungerte i praksis. Men etter omstendelig research på forumer, Qt sin hjemmeside og utgitt litteratur pekte alt på at det med ingen eller få justeringer enkelt kunne porteres til forskjellige formater. Samtidig som den opprinnelige kodebasen var intakt. Dette var noe vi var veldig opptatt av, at oppdragsgiver ikke skulle ha problemer i fremtiden å tilby applikasjonen til brukere av forskjellige håndholdte enheter. Qt er et rammeverk laget i C++, et språk vi hadde erfaring med fra før, slik at vi så ikke overgangen på som et større problem enn å lære oss C#. Det ble derfor vurdert at vi skulle bruke Qt fremfor å lage applikasjonen i C# og binde oss til WM som plattform. En annen fordel Qt hadde var at det ved rekompilering kan tilbys til Symbian brukere i nær fremtid; det mest utbredte mobile OS'et på smarttelefoner per fjerde kvartal 2008 ¹⁰.

Lisensiering

Qt krever lisensiering for å kunne brukes, man kan velge mellom en kommersiell og en open source lisens. Den kommersielle gir full integrasjon med Visual Studio, som igjen gir enklere støtte for deployment til WM. Mens open source benytter seg av GNU GPL ¹¹ lisensiering, som gjør at man forholde seg til en del retningslinjer for utgivelse av kode. Vi ville i minst mulig grad tvinge vår oppdragsgiver til å opprettholde open source krav, og heller ikke bruke penger på en kommersiell lisens. Men det ble vurdert at det var fordelaktig for oss å få Visual Studio integrasjon, noe open source lisensen ikke tilbyr. Det ble derfor utarbeidet en avtale med Qt Software v/ Knut Yrvin og oppdragsgiver om å ha en midlertidig studentlisensiering tilsvarende kommersiell; med forbehold om ny avtaleutarbeidelse ved bachelorprosjektets endelse.

Utviklingsmiljø for håndholdt har derfor vært Visual Studio 2008 med Qt integrasjon.

¹⁰Symbian OS: http://news.cnet.com/8301-1035_3-10195697-94.html

¹¹GNU GPL: <http://www.gnu.org/copyleft/gpl.html>

4.2 Design

4.2.1 Autentisering og autorisasjon

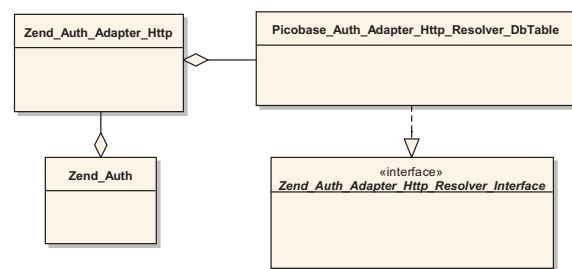
I designfasen (se seksjon 3.1.1 og 3.2.2) skisserte vi hvor og hvordan autentiserings- og autoriseringsmekanismen kunne implementeres. Dette designet har vært førende for vår endelige implementasjon og var også en faktor i valg av rammeverk for webapplikasjonen.

Zend Framework har to pakker som vi bygget løsningen på; *Zend_Auth* for autentisering og *Zend_Acl* for autorisering.

For å få et felles punkt i webapplikasjonen å autentisere og autorisere på implementerte vi dette i en plugin¹² som kjører før selve applikasjonen starter (`preDispatch[15]`).

Autentisering

Zend_Auth støtter flere metoder å autentisere brukere på. Disse er implementert i forskjellige *adapterklasser* [11], for eksempel finnes det et eget adapter for *HTTP autentisering* og et for databasetabellautentisering. Dette utnyttet vi i webapplikasjonen når vi autentiserer brukere gjennom webgrensesnittet og brukere fra den håndholdte i et felles autentiserings- og autorisasjonspunkt.



Figur 4.1: HTTP autentisering

For at *HTTP autentiseringsadapteret* skal ha noe å autentisere på trenger den en såkalt *resolver* [12] for å verifisere brukernavn og passord. Figur 4.1 viser hvordan *Zend_Auth_Adapter_Http* aggregerer på en *strategi*¹³ som den bruker når den skal verifisere brukernavn og passord. Videre bruker *Zend_Auth* adapteret til å forespørre brukeren etter brukernavn og passord på korrekt måte. Selv tilbyr *Zend_Auth* funksjonalitet for å lagre brukerinformasjon og holde denne i livet slik at bruker kan identifiseres ut gjennom applikasjonen.

Selv om *Zend_Auth* har støtte for både *HTTP autentisering* og databasetabellautentisering er bruken av de to ganske forskjellig. Den viktigste forskjellen er at når

¹²Kildere f: `/picobaseweb/library/Picobase/Controller/Plugin/Authenticate.php`

¹³`Picobase_Auth_Adapter_Http_Resolver_DbTable`

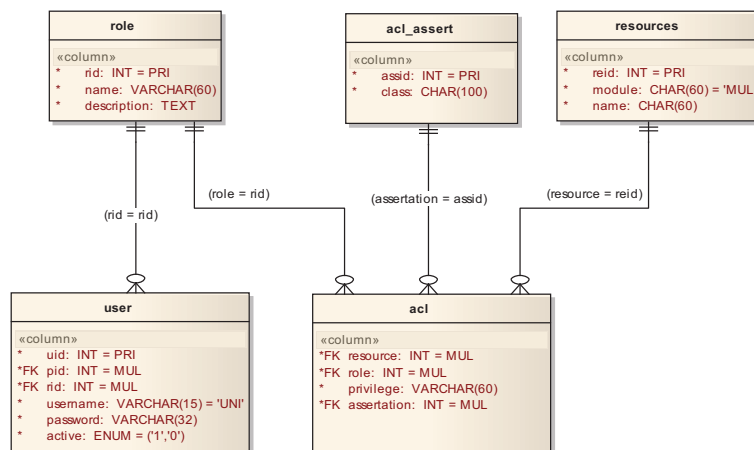
klienten logger seg inn via databasetabellautentisering holdes autentiseringsresultatet i livet mellom de ulike HTTP forespørslene, mens ved *HTTP autentisering* må klienten autentiseres for hver eneste forespørsel (autorisasjonskoden blir liggende i HTTP hode til klienten).

Konsekvensen av dette er at når autentisering mangler blir brukere av webgrensesnittet tatt videre til en login-side, mens den håndholde autentiseres direkte.

Sikkerhetsteknikker

`Zend_Auth_Adapter_Http` har støtte for å bruke både *basic* og *digest HTTP autentisering*. Vi valgte her å bruke *digest* fordi den er den sikreste av de to [17, side 239]. *Resolveren* som vist i figur 4.1 har som eneste oppgave å returnere brukernavn, *realm*, og passordet kolonseparert og MD5 hashet, slik: `MD5(username : realm : secret)` [12, 5.4.4. Resolvers]

For å unngå å lagre passordet i databasen som klartekst innførte vi den samme operasjonen når en ny bruker opprettes slik at passordkolonnen til *user*-tabellen (se databasemodellen i figur 4.2) i virkeligheten er har formen `MD5(brukernavn:picobase:passord)`.



Figur 4.2: Utsnitt fra data modellen som viser *ACL* implementasjonen i relasjonsdatabasen

Autorisering

`Zend_Acl` inneholder støtte for å definere roller og rollenes tilgang til ressursene [13]. Pakken har ikke direkte støtte for å definere regler via databasetabeller, så dette var noe vi måtte implementere selv.

Roller

Vi har valgt å implementere en løsning som baserer seg på generiske roller og ikke

enkeltbrukere. For eksempel har du roller som heter *kliniker* og en som heter *veileder*. Det fine med med `Zend_Acl` er at brukere kan arve rettigheter av hverandre [13, 2.1.2. Roles]. Vi har brukt denne teknikken for å bygge opp rettighetene; en *gjest* arver ingen, en *kliniker* arver *gjest* og en *veileder* arver *kliniker* og dermed også *gjest* også. Slik unngår vi å eksplisitt å legge til nye rader i ACL-listen vår for hver eneste bruker.

Ressurser og autorisering

I vår implementasjon definerer vi en ressurs som *controllere* i en modul, for eksempel er `/backend/clinicalquestion/` en ressurs og `/webservice/clinicalquestion/` er en ressurs.

Privileget på den aktuelle ressursen er gitt av *controllerens* action. For eksempel vil forespørselen `/backend/clinicalquestion/update/22` tyde: “Oppdater ressursen *clinicalquestion* under modulen *backend*”.

Dette gjør det enkelt å intuitivt å definere rettigheter i forhold til URI-strengen som brukes for å identifisere en ressurs og action i Zend Framework.

ACL-listne bygges opp fra databasen. Figur ?? viser relaterte tabeller i databasen som har med autentisering og autorisering å gjøre. En bruker har en rolle som den blir autorisert på.

Systemet er laget slik at all tilgang til ressursene er sperret til det eksplisitt blir klarert gjennom rader i *acl*-tabellen. Tabellen *acl_assert* inneholder klassenavn på spesial tilgangsregler, såkalte “Assertions” [14]. Dette er egendefinerte klasser som brukes for å utføre mer avanserte former for autorisering. For eksempel bruker vi en slik klasse¹⁴ når en bruker skal oppdatere et klinisk spørsmål fordi vi må forsikre oss om at innlogget bruker har eierskap over aktuelt klinisk spørsmål.

Tabellstrukturen honorerer virkemåten til *Zend_Acl*.

¹⁴`Picobase_Acl_Assert_ClinicalQuestionOwnership`


```
1  /*
2  * Prøv å autentiser bruker på aktiv ressurs */
3  if(!$this->_authorize()) {
4  /*
5  * Skjekker om bruker i det hele tatt er innlogget */
6  if( !$this->_auth->hasIdentity() ) {
7  /*
8  * Skiller mellom grensesnittene ved hjelp av aktiv modul */
9  if( $request->getModuleName() == 'webservice' ) {
10 /*
11 * Henter inn model som brukes for å HTTP autentisere direkte */
12 require_once 'models/AuthModel.php';
13 $authModel = new AuthModel();
14 /*
15 * Utfør HTTP autentisering direkte */
16 if($authModel->httpAuthenticate($request, $this->getResponse())
17 {
18 /*
19 * Autoriser bruker på nytt */
20 if( !$this->_authorize() ) {
21 /*
22 * Autentisert, men ikke autorisert til ressurs:
23 * gi en semantisk tilbakemelding med HTTP kode */
24 $response = $this->getResponse();
25 $response->setHttpResponseCode(Picobase_HTTP::Unauthorized);
26 $response->setBody("You are not authorized to use this
27 resource!");
28 $response->sendResponse();
29 exit();
30 }
31 } else {
32 /*
33 * Vanlig webgrensesnitt er aktivt, ikke autentisert
34 * send bruker til login */
35 $request->setControllerName('login');
36 $request->setActionName('index');
37 }
38 } else {
39 /*
40 * Brukeren er autentisert men har ikke tilgang til ressursen:
41 * send videre til login side */
42 $request->setControllerName('login');
43 $request->setActionName('noauth');
44 }
45 }
```

Listing 4.1: Utsnitt av felles autentiserings- og autorisasjonspunkt

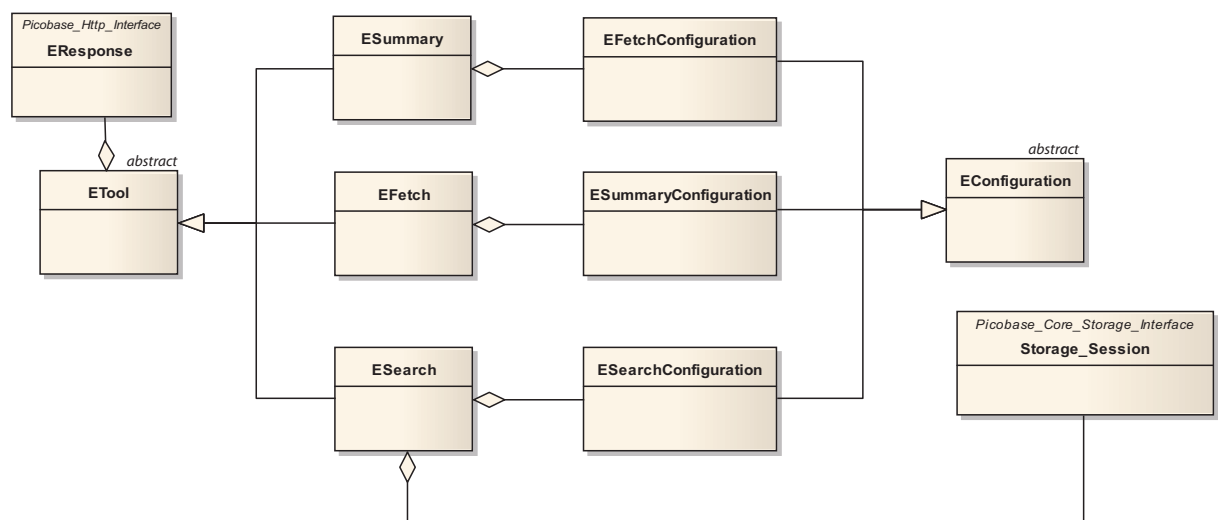
4.2.2 Interaksjon med EUtilities

Kommunikasjon mot EUtilities er en sentral del av prosjektet, all informasjon som applikasjonen bruker kommer gjennom dette grensesnittet (se seksjon

2.1.1). Vi designet og implementerte denne pakken tidlig i prosjektet. Det var opprinnelig designet for den håndholdte siden vi var tenkt på å la den kommunisere med EUtilities selv. Denne implementasjonen var asynkron siden vi brukte *signals* & *slots* sammen med Qt sin klasse for HTTP kommunikasjon. I webapplikasjonen har ikke dette noe poeng jf. den tekniske karakteren websider har så implementasjonen ble noe forskjellig.

Siden dette er en viktig del av systemet er det designet for å være så testbart som mulig gjennom bruk av såkalte “Dependency Injection” [3] slik at vi unngår at klasser som skal testes selv oppretter klasser den er avhengig av. I stede får de den injeksert ved initialisering. Slik er det lett å *mmocke*¹⁵ ved testing.

Klassediagram



Figur 4.3: Klassediagram over EUtilities systemet i webapplikasjonen

Klassene er designet for å gjenspeile EUtilities på en logisk og intuitiv måte. For eksempel har *ESearch*-klassen sitt begrensede ansvar rundt søk og *ESearch*-klassen har ansvaret for å hente og bearbeide resultater fra EUtilities sin *ESearch*.

¹⁵Etterligning av klassen som inngår i klassen som skal testes

`ESearch`-klassen aggregerer på et `Storage_Session`-objekt for at søkeresultatreferansen (`webEnv`) holdes i live over flere kall. For eksempel trenger `ESummary`-klassen søkeresultatreferansen både når søkeresultat skal hentes for første gang og senere når bruker navigerer i resultatsettet.

Konfigurasjonsklassene som verktøyklassene (`ESummary`, `EFetch` og `ESearch`) aggregerer på er har som oppgave å tilby et enkelt grensesnitt for å parametrisere de ulike verktøy-klassene. Hver av disse klassene vet også hvilke parametre som er tillat for det enkelte verktøyet som de er ment å aggregere i, de bruker dette for luke ut parameterverdier som ikke er gyldige. Når en respektiv konfigurasjonsklasse mottar et array, for eksempel `GET` variabelen, luker den ut ugyldige verdier. Når verktøyene skal utføre operasjoner mot `EUtilities` henter de innstillinger fra sin respektive konfigurasjonsklasse.

Klassen som konfigurasjonsklassene arver fra, `EConfiguration`, inneholder blant annet felles funksjonalitet for å filtrere ut ugyldige verdier.

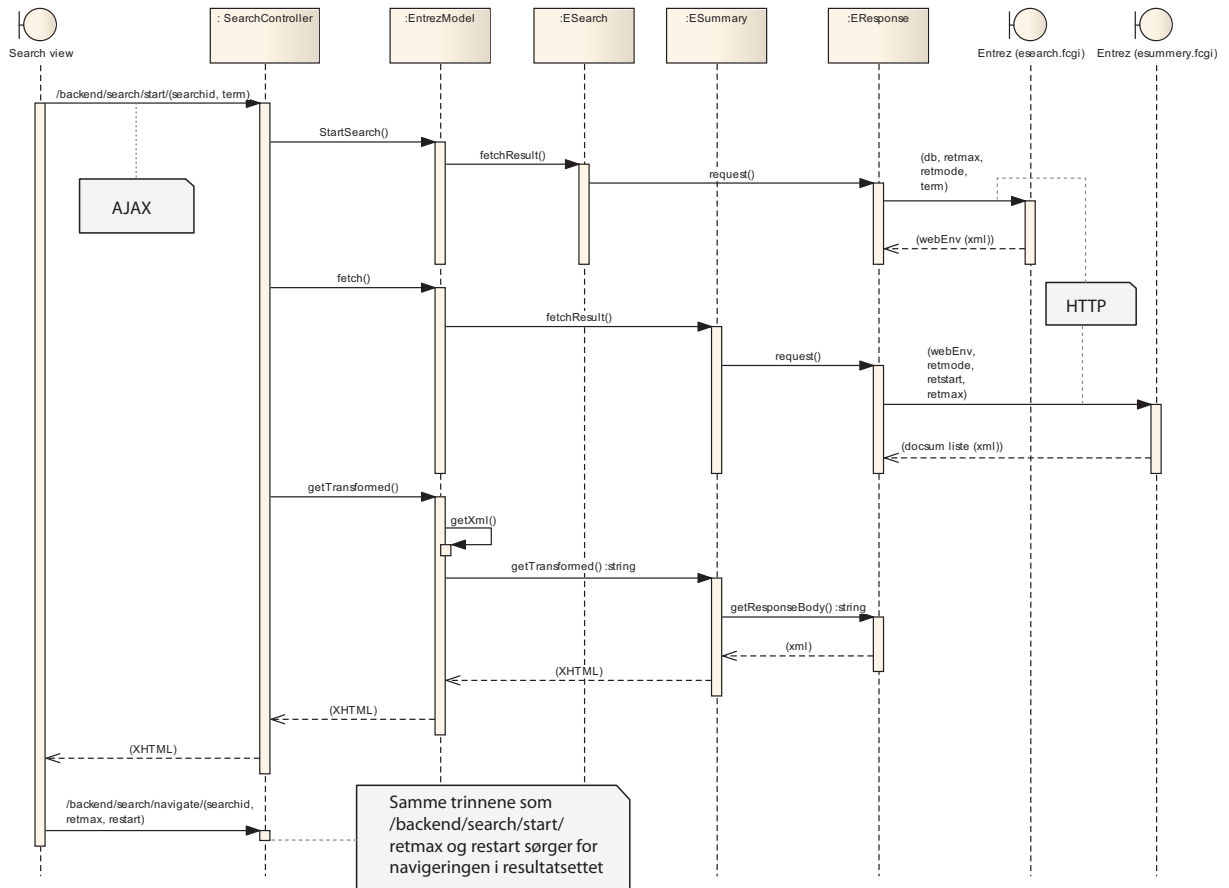
Konfigurasjonsklassene er ikke konstruert inne i verktøyklassene for å holde verktøyklassene testbare. De blir istedet gitt som parameter ved initialisering av aktuelt verktøy.

`EResponse` er klassen som står for den faktiske kommunikasjonen mot `EUtilities` og kan best beskrives som en transportklasse. Den har kun ansvar for å ta imot en URL-adresse, hente ned innholdet på aktuell denne og tilby klassen som aggregerer på den dette innholdet. Hvilket innhold den henter vet den ingenting om.

Denne pakken er også gjort uavhengig fra Zend ved hjelp av *Strategy* patternet [4, side 315]. Som det går frem av diagrammet aksepterer `ETool` og `ESearch` hvert sitt interface, henholdsvis `Picobase_Http_Interface` og `Picobase_Core_Storage_Interface`.

Sekvensdiagram

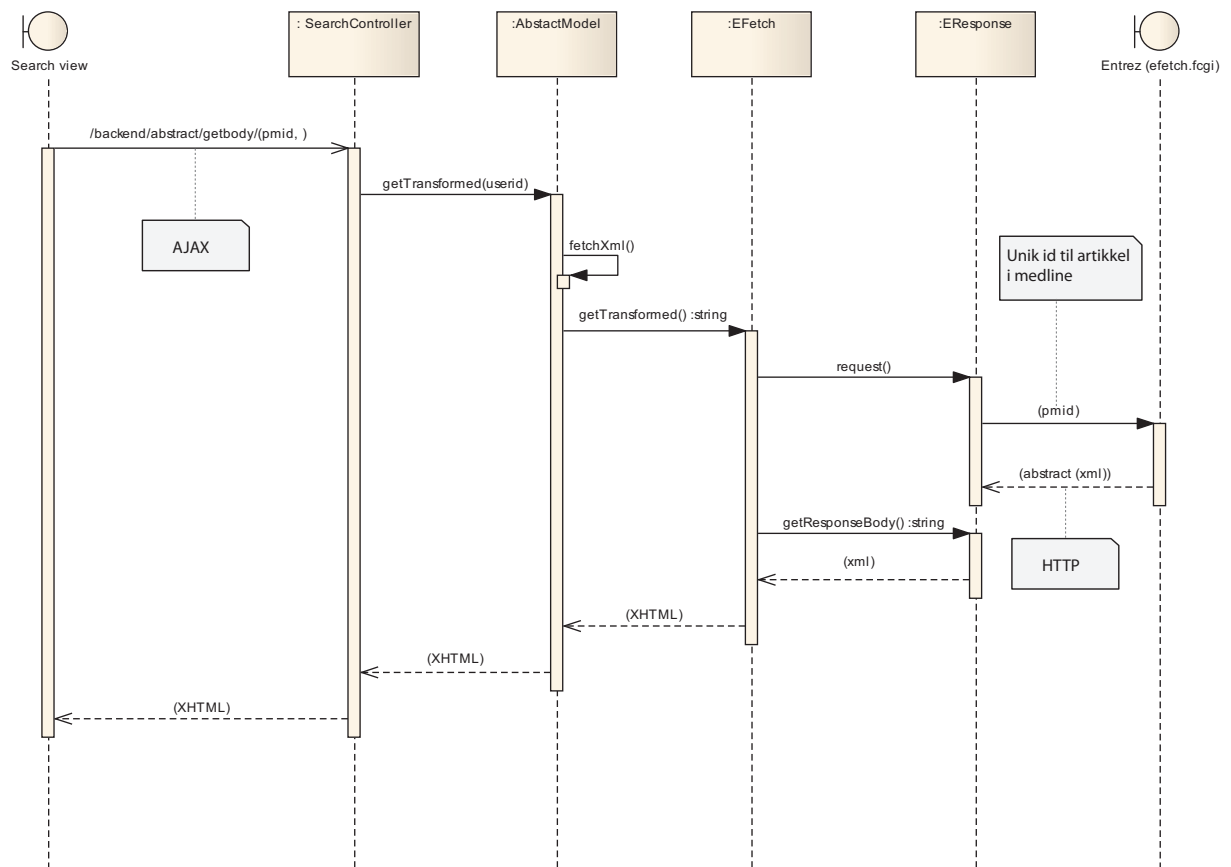
Dette avsnittet er i direkte relasjon med avsnitt 3.4 under design. Diagrammet er ment som en sammenligning mellom design og implementasjon.



Figur 4.4: Sekvens diagram som viser kommunikasjon mellom webapplikasjon og Entrez webservice ved henting av *docsum-liste*

4.2.3 Utforming søkefunksjonaliteten på webapplikasjonen

Denne delen av systemet er den viktigste for klinikerne. Her skal han kunne utføre alle operasjonene som har med søk, behandling av søkeresultater, kritisk vurdere artikler, skrive resyme, se over lagrede abstracts, i det hele tatt gjøre alt av kjerneaktiviteter.



Figur 4.5: Sekvens diagram som viser kommunikasjon mellom webapplikasjon og Entrez webservice ved henting av et enkelt abstract

Vi har derfor puttet en god del tid og har prioritert denne delen mot mindre trivielle *Use Case*. Det har vært en utfordring å tilby så mye funksjonalitet på en relativt begrenset størrelse.

Målet med søkedelen i webapplikasjonen er først og fremst at du skal ha alt du trenger tilgjengelig uten å måtte laste siden på nytt. Vi har med andre ord basert oss på tung bruk av *Ajax* teknologi. Vi valgte *jQuery* som javascript rammeverk for å bygge under både *Ajax* funksjonalitet og også grafiske brukerelementer som tabber.

Figur 4.7 viser søkedelen av webapplikasjonen. Figuren er delt i to får å vise hvordan den såkalte *klinisk spørsmåls leseren* ser ut når brukeren trykker på linken med samme navn. Når denne linken aktiveres, glir hele siden ned og bruker blir presentert med *leseren*. Fra denne leseren kan brukeren søke i sine eksisterende kliniske spørsmål på fritekst, filtrere etter kategorier eller en kombinasjon. Ettersom bruker

trykker inn søker *leseren* fortløpende etter resultater og viser de i kolonnen som heter *RESULTATER*. Hvert enkelt resultat skjuler detaljer om det enkelte kliniske spørsmålet. Disse detaljene som for eksempel beskrivelse og lagrede søkekomposisjoner vises når bruker trykker på pil-ikonen til venstre i resultatvisningen. Her kan bruker kopiere inn søkekomposisjoner fra andre kliniske spørsmål. Figur 4.6 viser full visning av leseren.

The screenshot displays a web interface for searching clinical questions. It is divided into three main sections: 'SØK' (Search), 'VELG KATEGORI' (Select Category), and 'RESULTAT' (Result).

- SØK:** A search input field.
- VELG KATEGORI:** A list of medical categories with checkboxes, including: Uncategorized, Pediatri, Cardiologi, Indremedisin, Kirurgi, ØNH, Psykiatri, Gynekologi, Nevrologi, Lungemedisin, Endokrinologi, Nefrologi, Geriatri, Gastroenterologi, En ny kategori, Ny kategori, and Ny kategori.
- RESULTAT:** A list of search results. The first result is expanded to show details:
 - My only, single clinical question:** 1 17
 - Poststep, glomerulonefritt:** 1 2The expanded view for 'Poststep, glomerulonefritt' includes:
 - Beskrivelse:** Effekt av penicillin hos asymptotiske barn med poststrep.glomerulonefritt
 - Kategorier:** Pediatri
 - Søkekomposisjoner:** A table showing search compositions.

#	SPØRSMÅL	TYPE	TREFF	X
94	Poststreptococcal glomerulonephritis /// recurrence	pico	12	

Figur 4.6: Full visning av klinisk spørsmål leser

4.2.4 Plugin-tilnerming til søkekomposisjoner på webapplikasjonen

Vi har tatt høyde for det kan komme nye måter å søke på. Vi har derfor laget søkekomposisjonsdelen så generisk som mulig. Databasen har støtte for å håndtere generiske strukturer med den forutsetningen at søkekomposisjonen består av (nøkkel, verdi) par.

Når en ny søkekomposisjon skal lagres trengs en enkel javascript class for å håndtere søkefeltene, sette samme disse og sende søket til en generisk funksjon.

I datamodellen, figur 3.6, vises denne strukturen på databasenivå under tabellnavn `search_composition` og `search_composition_fields`. Sistnevnte inneholder (nøkkel, verdi) parene.

For å håndtere lagring av abstract fra *Ajax* (flatt array), lagring fra håndhold (xml) og samme veien tilbake fra database til array og xml laget vi en klasse som har ansvaret for disse operasjonene¹⁶. Dette var også den eneste klassen vi tok oss tid til å *unit teste*¹⁷.

¹⁶`Picobase_Search_CompositionParser`

¹⁷Testen ligger `/PICObaseWeb/tests/`

The image displays two screenshots of the PICObase web application interface, annotated with yellow boxes and arrows to highlight specific design elements and user interactions.

Top Screenshot:

- SØK:** Search input field.
- VELG KATEGORI:** A list of medical categories (e.g., Uncategorized, Pediatri, Cardiologi, Indremedisin, Kirurgi, ØNH, Psykiatri, Gynekologi, Neurologi, Lungemedisin) with checkboxes. An annotation 'AJAX' points to this list.
- RESULTAT:** A table of search results with columns for title, count, and status. An annotation 'AJAX' points to the table.
- Navigation:** Buttons for 'Kliniske spørsmål', 'Søk', 'Profil', and 'Klinisk spørsmål leser'. An annotation 'Aktiveres her' points to the 'Klinisk spørsmål leser' button.
- Annotation:** 'Klinisk spørsmål leser' points to the search results area.

Bottom Screenshot:

- Søk / Søkekomposisjoner:** Search tabs and a search bar containing 'cancer'. An annotation 'AJAX' points to the search bar.
- Filters:** Fields for 'Medisinsk tilstand', 'Intervention', 'Compared to', and 'Outcome'.
- Buttons:** 'Lagre' and 'SØK' buttons.
- Result List:** A list of search results with titles, authors, journals, and PMIDs. An annotation 'VERKØY' points to the search results.
- Annotation:** 'Dokumenthoder' points to the document headers in the results.

Figur 4.7: Det grafiske brukergrensesnittet for søk

4.3 Implementasjon - Håndholdt

Merk: Skjermbilder fra håndholdt enhet i denne seksjonen er en testversjon på windows desktop. Hver håndholdt platform har sin egen layout, slik at utseendet vil variere fra platform til platform. Mens organiseringen av innhold og komponenter vil være lik.

4.3.1 Opprettelse av klinisk spørsmål

Et klinisk spørsmål opprettes som vist i figur 4.8 ved at brukeren fyller inn tittel og beskrivelsesfeltet i CQ tab'en. Den håndholdte enheten har ikke egen database, slik at ingenting mellomlagres før brukeren eksplisitt trykker på 'Save'. Det blir derfor ikke opprettet et fysisk klinisk spørsmål objekt før brukeren velger å lagre det.

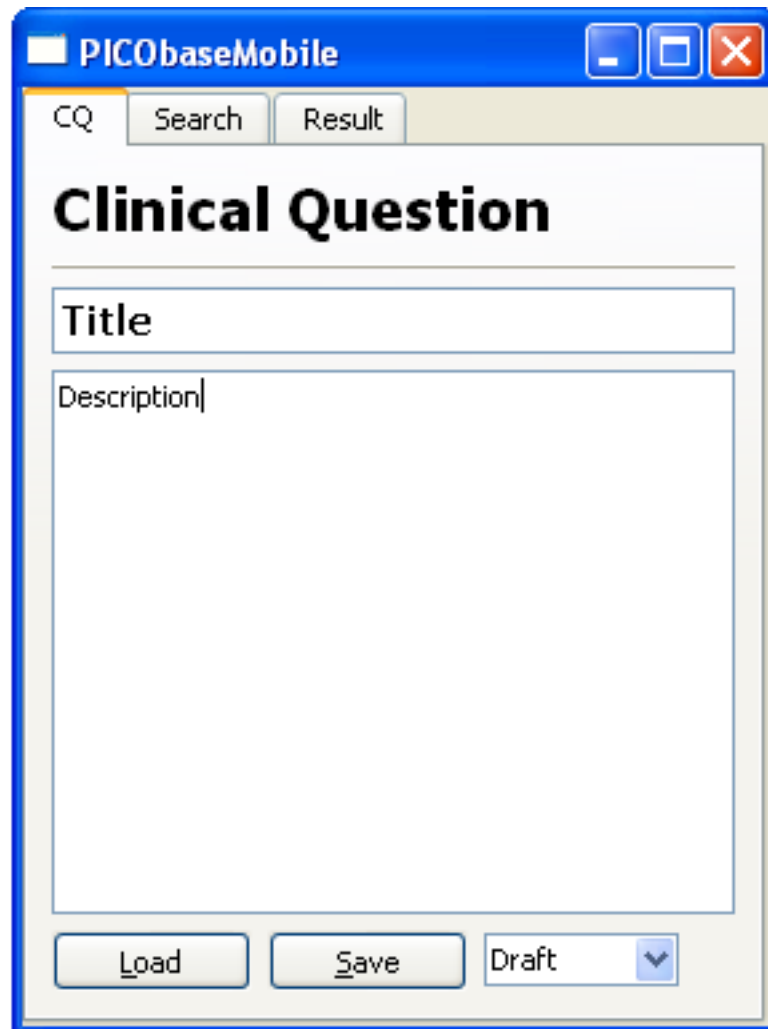
4.3.2 Lagre klinisk spørsmål

For å lagre det kliniske spørsmålet har brukeren to muligheter. Enten lagre lokalt som kladd (draft) på egen håndholdt, eller å lagre direkte til server. Alt som lagres til server blir værende i databasen hvis overføringen er vellykket. Mens kladden holdes usynkronisert så lenge brukeren ønsker det, og kan slette dette om han ikke ønsker å ha det lengre. Det gjør at brukeren hele tiden selv er klar over når et klinisk spørsmål blir opprettet / oppdatert i databasen.

Brukeren velger mellom kladd og synkronisering til server i rullegardinfeltet til høyre. For å lagre trykkes det enkelt på save. Applikasjonen går så gjennom tittel, beskrivelse og alle søkekomposisjonene (se figur 4.10 og 4.11) og lagrer dette i en xml-struktur (Se seksjon D.1 for eksempel).

Ved **synkronisering** til server gjøres samme prosess som ved kladd, men istedenfor å lagre som varig fysisk fil brukes følgende webservice metoder:

- `../webservice/clinicalquestion/addclinicalquestion/` - Et nytt klinisk spørsmål har blitt opprettet.
- `../webservice/clinicalquestion/updateclinicalquestion/cqid/` - Oppdatering av et klinisk spørsmål som allerede eksisterer i databasen med primærnøkkelen cqid (Clinical Question ID).



The image shows a mobile application window titled "PICObaseMobile". The window has a blue header bar with standard mobile OS window controls (back, home, close). Below the header, there is a navigation bar with "CQ" selected, and "Search" and "Result" buttons. The main content area is titled "Clinical Question" in large bold text. Below the title, there is a form with two input fields: "Title" and "Description". The "Description" field is currently empty and has a cursor at the end of the text "Description". At the bottom of the form, there are three buttons: "Load", "Save", and "Draft" with a dropdown arrow.

Figur 4.8: Klinisk spørsmål på håndholdt.

For å autentisere og oppdatere på riktig bruker benyttes HTTP Authentication (digest metoden). Denne spør bruker om brukernavn og passord hvis brukeren ikke allerede er innlogget.

4.3.3 Gjenopprette kliniske spørsmål

For å gjenopprette et klinisk spørsmål kan brukeren velge mellom å gjenopprette en kladd (draft) lokalt eller et klinisk spørsmål lagret i den sentrale databasen. Ved å trykke på loadknappen i CQ-vinduet så kan brukeren gjenopprette et tidligere lagret klinisk spørsmål.

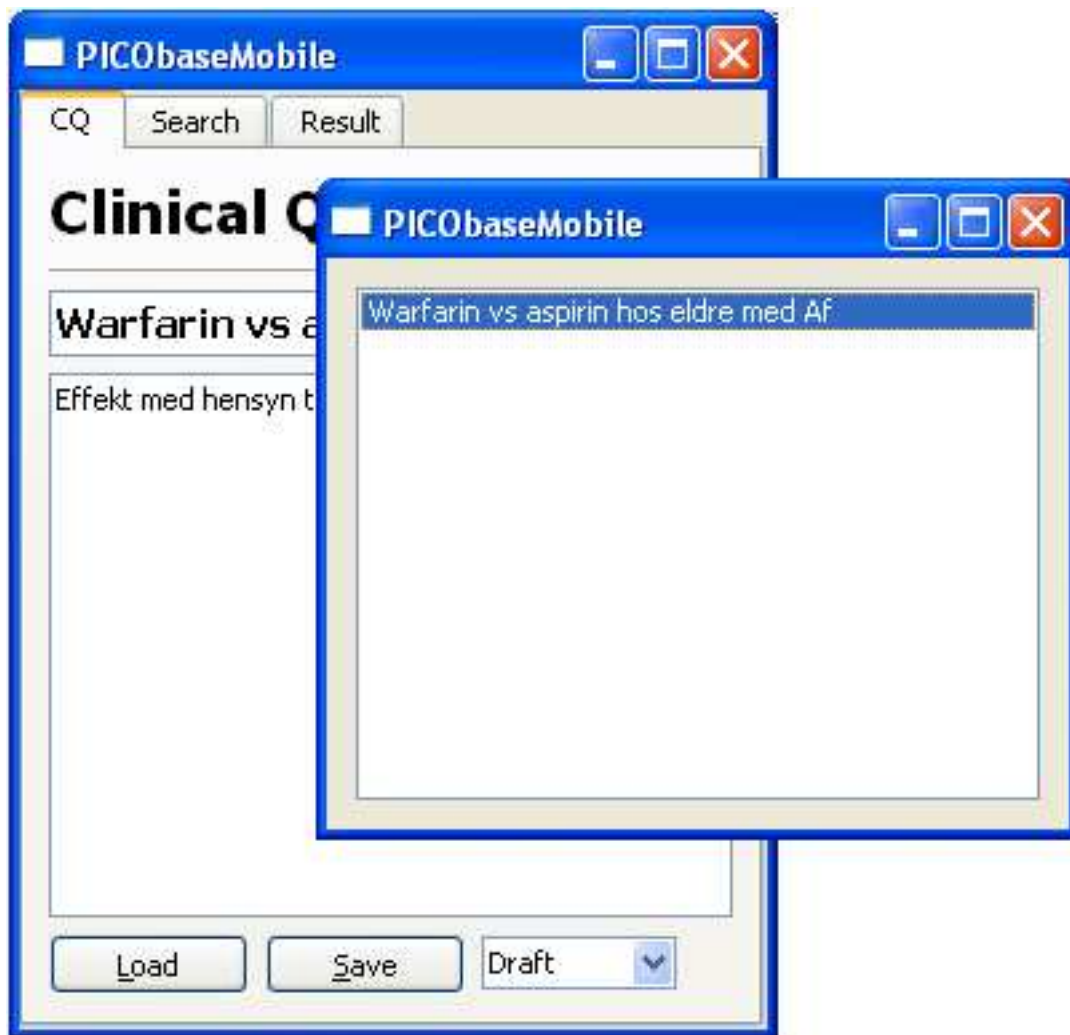
Som ved lagring velges også her kilden i rullegardinfeltet, enten “Draft” eller “Server”. Bruker klikker så på load og en widget med en liste lik *figur 4.9* popper opp. For å lage denne listen brukes et indekseringssystem i XML format (Se seksjon D.2.1) , hvor en updatefunksjon kalles som parser alle drafts. Denne tar så tittel og filreferansen og legger i en indeksfil. “LoadWidget’en” bruker så alle titlene for å lage en klikkbar liste. Når bruker så dobbeltklikker på ønsket klinisk spørsmål, oppdateres resten av applikasjonen dataene til

Listen som kommer fra server hentes via følgende webservicekall: `../webservice/-clinicalquestion//getlist/`.

Formatet er i XML (se seksjon D.2.2) og følger en tilnærmet lik struktur som den for håndholdte. Forskjellen er noen ekstra identifikatorer for å vedlikeholde verifikasjonen av innholdet.

Autentifisering gjøres på samme måte som ved lagring av kliniske spørsmål.

Merk: Listerepresentasjonen i *figur 4.9* er fortsatt i prototypisk stand, og vil ha flere attributter i den endelige versjonen; som date modified, cqid og uid.



Figur 4.9: Gjenopprett et klinisk spørsmål

4.3.4 Komponere søk

PICObase tilbyr i den første prototypen to søkekomposisjonstyper; PICO (se figur 4.11) og fritekst (se figur 4.11).

Implementasjonen har fulgt designskissen i figur 3.7 så godt som mulig. Ved å bruke en `QStackedLayout`¹⁸ som layout manager og plassholder-widget, løses den dynamiske problematikken.

Søkekomposisjonswidgetene legges til som vist nedenfor.

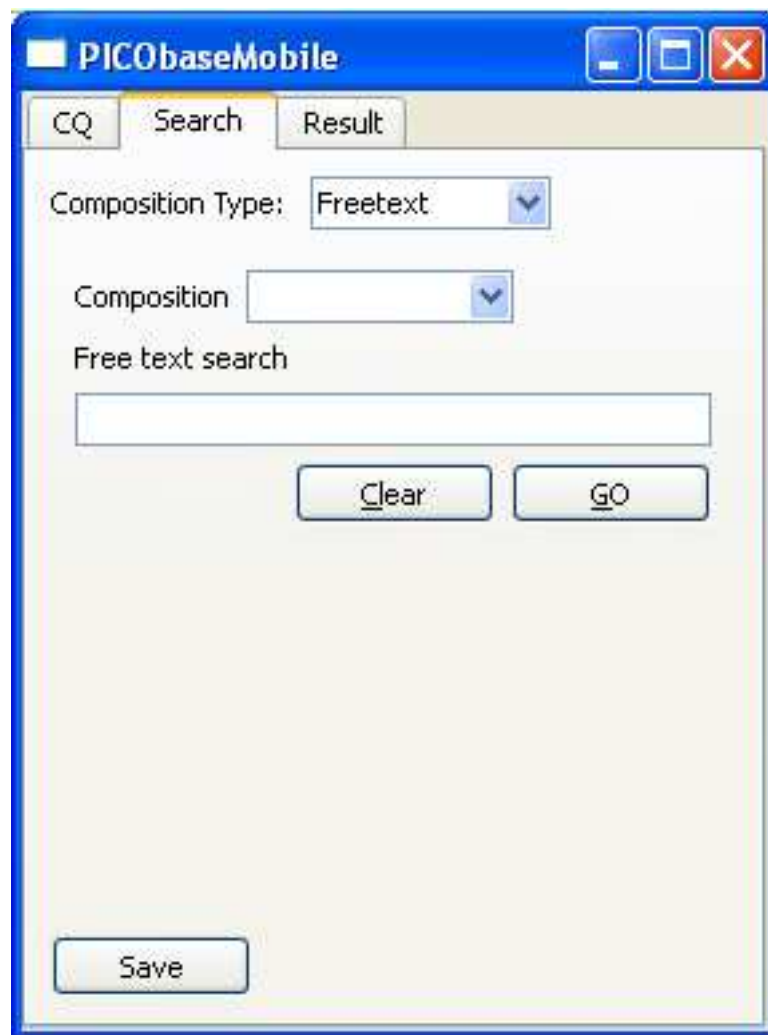
```
1      // Create dynamic containerwidget.
2      QStackedLayout *stackedLayout = new QStackedLayout();
3
4      // Add search widgets[PICO, Freetext].
5      stackedLayout->addWidget( picoSearchWidget );
6      stackedLayout->addWidget( freeTextSearchWidget );
7
8      // Adds the stackedLayout to basewidget.
9      mainLayout->addLayout( stackedLayout );
```

Listing 4.2: Instansiering av `stackedLayout` og organisering av søkekomposisjonswidgeter.

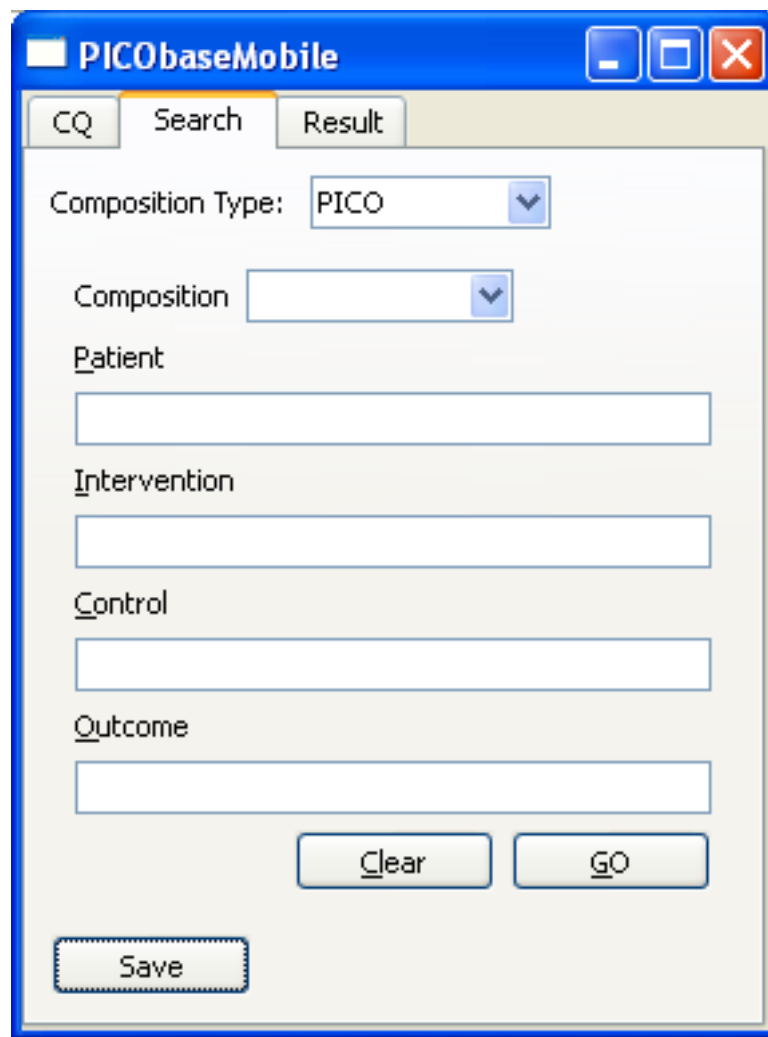
Hver søkekomposisjonswidget er ansvarlig for implementasjon av sin egen layout. Selve objektene som brukes for å holde runtimeinformasjon om søkekomposisjonene er realisert som en polymorfisk baseklasse med en derivert klasse for hver søkekomposisjonstype (`ETerm` -> `EPicoTerm` | `EFreeText` osv). Disse har så ansvar for å strukturere og danne søketerm basert på sin egen skreddersydde representasjon.

F.eks. så lager PICO sin term av å konkatenerer Pasient, Intervention, Control, og Outcome strengene. Hvert `ETerm` objekt har også ansvar for å både lese inn egen data og lage egen xmlstruktur basert på *name / value* nøkkelparet i lagringsformatet (se seksjon D.1). Dette sikrer en generisk struktur, som både kan skaleres opp uavhengig av hvor mange nøkkelpar den inneholder og hvordan de er organisert i søkekomposisjonen.

¹⁸`QStackedLayout`: <http://doc.qtsoftware.com/4.5/qstackedlayout.html>



Figur 4.10: Fritekst-søk på håndholdt.



The screenshot shows a mobile application window titled "PICObaseMobile". At the top, there are three tabs: "CQ", "Search", and "Result", with "Search" being the active tab. Below the tabs, the interface is organized into sections for defining a PICO query:

- Composition Type:** A dropdown menu currently set to "PICO".
- Composition:** A dropdown menu.
- Patient:** A text input field.
- Intervention:** A text input field.
- Control:** A text input field.
- Outcome:** A text input field.

At the bottom of the form, there are three buttons: "Clear", "GO", and "Save". The "Save" button is located at the bottom left, while "Clear" and "GO" are side-by-side at the bottom right.

Figur 4.11: PICO-søk på håndholdt.

I den statiske toppdelen kan brukeren velge mellom de tilgjengelige søkekomposisjonene fra en rullegardinboks. Denne sørger for å endre aktiv søkekomposisjon. Dette gjøres ved å sette aktiv indeks lik indeksen i rullegardinboksen i `stackedLayout` objektet.

```
1  stackedLayout->setCurrentIndex(  
    searchCompositionComboBox->currentIndex() );
```

Listing 4.3: Setting av `stackedLayout` indeks.

For å samkjøre endringen når bruker skifter søkekomposisjon og oppdateringen av `stackedLayout`, brukes Qt sitt innebygde eventhandling-system: Signals & Slots¹⁹.

```
1  // When user changes searchCompositionType,  
2  // Activate changed-to index in stackedLayout.  
3  connect(searchCompositionComboBox, SIGNAL( activated(int) ),  
4          stackedLayout, SLOT( setCurrentIndex(int) ));
```

Listing 4.4: Event handling for dynamisk endring av søkekomposisjonstype.

¹⁹Qt Signals & Slots: <http://doc.qtsoftware.com/4.5/signalsandslots.html>

4.3.5 Lagre / gjenopprette søkekomposisjoner



Figur 4.12: Dialogboks for lagring av søkekomposisjon.

PICObase tilbyr flere søkekomposisjoner på hver enkelt søkekomposisjonstype - knyttet til hvert enkelt kliniske spørsmål. For å realisere dette så må brukeren kunne mellomlagre flere søkekomposisjoner på hver enkelt søkekomposisjonswidget. Både i runtime og i lagret tilstand.

Dette gjøres ved at bruker fyller inn feltene i valgt søkekomposisjon og trykker på 'Save'. Tittelen til hver lagrede søkekomposisjon legges i en rullegardinliste med referanse til rett ETermobjekt i en vector (basert på indeks).



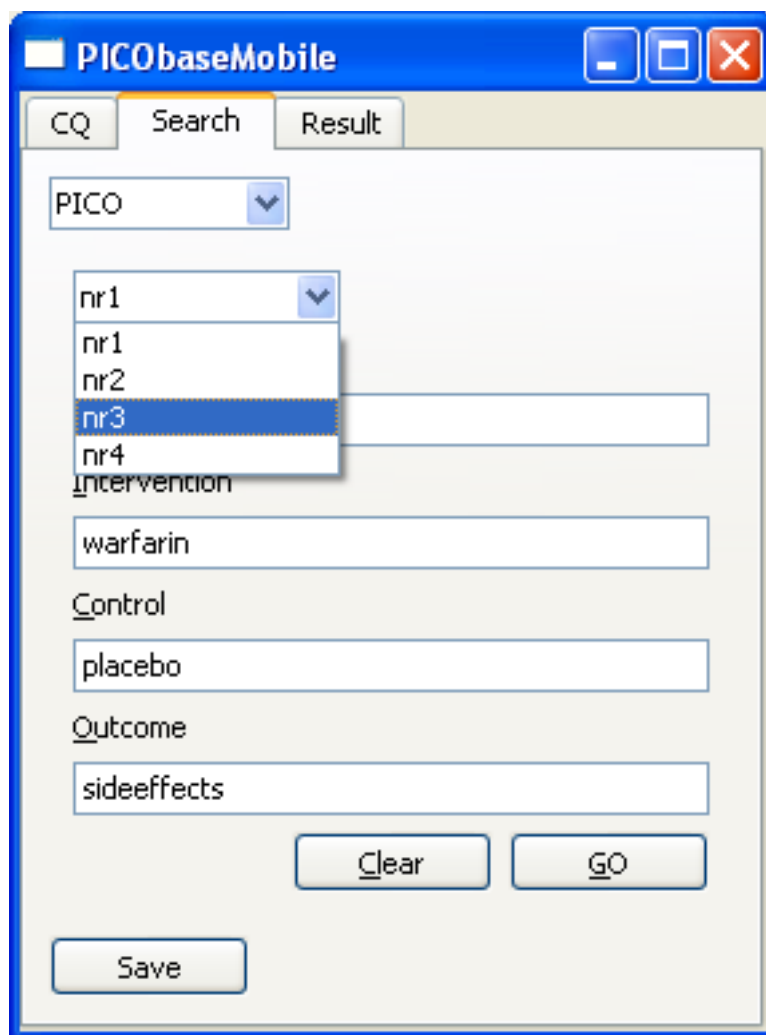
Figur 4.13: Dialogboks for navngivning av lagret søkekomposisjon.

Velges "Save New" åpnes dialogboksen vist i *figur 4.13*. Bruker skriver så inn ønsket lagringstittel.

Hvis ingen lagrede søkekomposisjoner eksisterer fra før åpnes dialogen direkte når bruker trykker på "Save".

Måten implementasjonen er gjort på kodemessig fører dog med seg noe redundands, da metoden for lagring må ta hensyn til hver enkelt søkekomposisjonsobjekt innad

i widget'en. Dette krever derfor at hver søkekomposisjonswidget reimplementerer lagstrukturen og gjør at for hver endring så må hver type søkekomposisjon oppdateres. Bakgrunnen for at implementasjonen står som den gjør er at en restrukturering ville vært for omfattende og tidkrevende og det var viktigere å få til en fungerende prototype i løpet av prosjektperioden. En videreutvikling bør derfor refaktorere til en implementasjon hvor søkekomposisjonswidget'en kun konsentrerer seg om å representere en søkekomposisjonstype, og lar lagringsfunksjonaliteten være av generisk art.

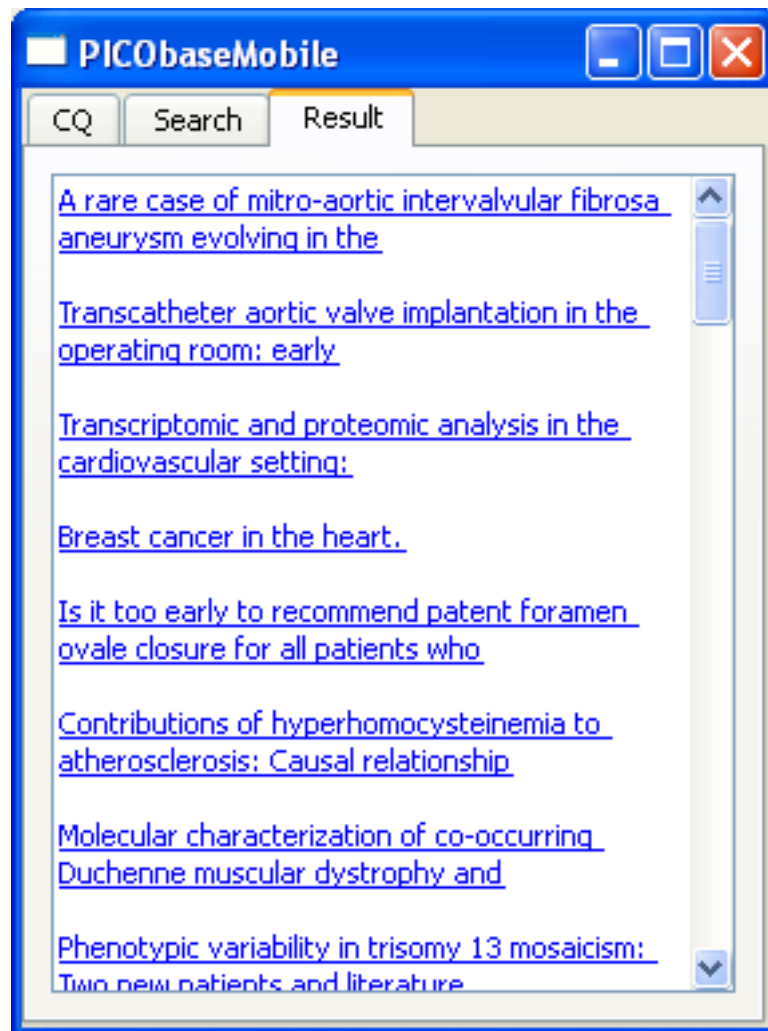


Figur 4.14: Gjenoppretting av lagret søkekomposisjon.

Gjenoppretting skjer ved at bruker velger ønsket søkekomposisjon illustrert av *figur 4.14*. Applikasjonen oppdaterer så feltene med valgt søkekomposisjon sine data.

4.3.6 Utføre søk

4.3.7 Vis resultat



Figur 4.15: Resultatvisning på håndholdt.

Resultatet fra søkekallet blir her parset og transformert til hypertextlinker. Dette blir representert i tekstformat ved å bruke en QTextBrowser²⁰ widget. Denne har naturlig støtte for navigering av hypertextlinker og en rekke andre hjelpesjonyer rundt dette.

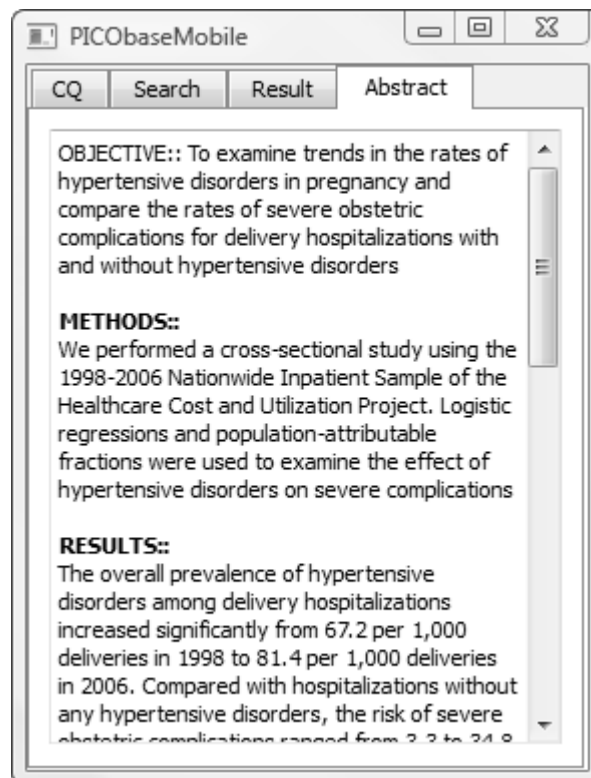
²⁰QTextBrowser: <http://doc.qtsoftware.com/4.5/qtextbrowser.html>

De blå tekststrengene illustrert i *figur 4.15*, er alle klikkbare hypertextlinker med referanse til artikkelens abstract.

4.3.8 Vis abstract

Etter at et søk er utført og får presentert resultatet har de fleste artikkelhodene fått klikkbare linker (de med tilgjengelig abstract). Når bruker trykker på en av disse forespør den håndholde applikasjonen webapplikasjonen om aktuelt abstract, webapplikasjonen henter dernest ned fra EUtilities og XML-transformerer resultatet over til et enkelt html-format som den håndholde applikasjonen kan bruke direkte. Slik kan utseende om ønskelig endres i en eneste *XSLT*-fil på server.

Figur 4.16 viser hvordan et abstract blir presentert. Legg også merke til den formaterte utseende som er modifisert av en lite *regular expression* funksjon.



Figur 4.16: Abstractvisning på håndhold.

4.4 Implementasjon - Webapplikasjon

4.4.1 Opprettelse av klinisk spørsmål

Opprettelse av kliniske spørsmål er triviell operasjon. Bruker følger linken fra hovedmeny og får presentert skjermbilde som vist i figur 4.17. Brukeren kan fylle inn en eller flere kategorier, eller opprette nye dynamisk via *Ajax* uten å laste siden på nytt. Figur 4.17 viser to skjermbilder av opprett klinisk spørsmål. Det nederste bilde viser hvordan skjemaene reagerer på feil når bruker trykker på lagre.

For å få til en generisk system for å håndtere brukerverdier via skjema laget utstyrte vi *modeller*²¹ som har med brukerverdier med definisjoner av hvilke verdier som er påkrevd, hvilke verdier di skal ha og videre ulike valideringstyper. Denne implementasjonen er kjent som *Forms* (*Zend_Forms*²²) i Zend Framework. Med den implementasjonen baserer på å opprette en *Form* for hver *Controller* - eller for hver enkel form. Vi prøver å holde *Controllerene* så “tynne” som mulig ved å plassere det vi kan inn i *Modellene*. For å oppnå det opprettet vi vårt eget definisjonssystem²³ oppe på Zend sin *Zend_Form* implementasjon. For få utnyttet dette ut i *View-ene* så har vi også supplert med egne *View-hjelpere* som automatisk oppdager om sitt felt har registrert feil, som vist i figur 4.17.

4.4.2 Søkefunksjonalitet

Som nevnt tidligere (se seksjon 4.2.3) er dette den mest sentrale delen av systemet. Her skal kliniker ha han trenger for å jobbe med det *aktive kliniske spørsmålet*. For å jobbe med søk må brukeren velge et aktivt klinisk spørsmål. Dette gjøres enten via startside eller via *klinisk spørsmålsleser* ved å trykke på et ønsket klinisk spørsmål. Etter at det aktive kliniske spørsmålet er valgt aktiveres alle funksjonene i søk og bruker kan starte å lagre abstract (se figur 4.19), lagre søkekomposisjoner, (se figur 4.18), kritisk vurdere artikler og skrive resyme.

²¹M - i Model view controller pattern (MVC)

²²<http://framework.zend.com/manual/en/zend.form.html>

²³`Picobase_Model_Definition`

The figure consists of two screenshots of a web application interface for creating clinical questions.

Top Screenshot: Shows the form with the following elements:

- Title:** An empty text input field.
- Beskrivelse:** An empty text area.
- Kategori:** A list of categories with checkboxes: Uncategorized, Pediatri, Cardiologi, Indremedisin, Kirurgi, ØNH, Psykiatri, Gynekologi, Nevrologi, and Lungemedisin.
- Ny kategori:** A section for adding a new category with fields for **Navn** and **Beskrivelse**.
- Buttons:** A "Save" button is located below the description field, and a "Legg til" button is located below the "Ny kategori" section.

Bottom Screenshot: Shows the same form with error messages indicating missing data:

- Title:** A red error message: "Value is required and can't be empty".
- Beskrivelse:** A red error message: "Value is required and can't be empty".
- Buttons:** The "Save" button is now visible below the description field, and the "Legg til" button remains below the "Ny kategori" section.

Figur 4.17: Opprett klinisk spørsmål. Nederste viser feilrapportering ved manglende utfylling.

Søk

PICO Free text

Medisinsk tilstand:

Intervention:

Compared to:

Outcome:

Lagre **SØK**

Lagret tilstand [id=95]
elderly atrial fibrillation / warfarin / aspirin / side effects
[\[x\] Lukk og rens](#) | [\[x\] Lukk og la stå](#)

Resultat Lagrede abstract Aktivt klinisk spørsmål Resymè

Warfarin vs Aspirin hos eldre med AF [id=25]
Effekt mtp cerebrale insulter og blødning

#	SPØRSMÅL	TYPE	TREFF
95	elderly "atrial fibrillation" / warfarin / aspirin / side effects	pico	21
96	atrial fibrillation elderly / / /	pico	18214
97	atrial fibrillation elderly / warfarin / /	pico	1239

Figur 4.18: Lagring / gjenoppretting av søkekomposisjoner

The image displays a web application interface for viewing and saving articles. It is divided into two main panels.

Left Panel (Search Results):

- Navigation tabs: Resultat, Lagrede abstract, Aktivt klinisk spørsmål, Resymè, Kritisk vurdering.
- Page navigation: Viser side 1 av 90741, Forrige, Neste.
- Result 1:
 - Title: Preclinical and clinical studies of anticancer agent-incorporating polymer micelles.
 - Author: Matsumura Y, Kataoka K.
 - Journal: Cancer Sci. 2009 Apr;100(4):572-9
 - PMID: 19462526 [PubMed - in process]
- Result 2 (Selected):
 - Title: Active TGF-beta1 correlates with myofibroblasts and malignancy in the colorectal adenoma-carcinoma sequence.
 - Author: Hawinkels LJ, Verspaget HW, van der Reijden JJ, van der Zon JM, Verheijen JH, Hommes DW, Lamers CB, Sier CF.
 - Journal: Cancer Sci. 2009 Apr;100(4):663-70
 - PMID: 19462508 [PubMed - in process]
- Result 3:
 - Title: Piloting a psycho-education program for parents of pediatric cancer patients in Malaysia.
 - Author: Othman A, Blunden S, Mohamad N, Mohd Hussin ZA, Jamil Osman Z.
 - Journal: Psychooncology. 2009 May 22.
 - PMID: 19462470 [PubMed - as supplied by publisher]
- Result 4:
 - Title: Molecular profiling of the "plexinome" in melanoma and pancreatic cancer.
 - Author: Balakrishnan A, Penachioni JY, Lamba S, Bleeker FE, Zanon C, Rodolfo M, Vallacchi V, Scarpa A, Felicioni L, Buck M, Marchetti A, Comoglio PM, Bardelli A, Tamagnone L.
 - Journal: Hum Mutat. 2009 Mar 18.
 - PMID: 19462467 [PubMed - as supplied by publisher]
- Result 5:
 - Title: Update of cylindromatosis gene (CYLD) mutations in Brooke-Spiegler syndrome: novel insights into the role of deubiquitination in cell signaling.
 - Author: Blake PW, Toro JR.
 - Journal: Hum Mutat. 2009 Mar 27.
 - PMID: 19462465 [PubMed - as supplied by publisher]

Right Panel (Article View):

- Header: Cancer Sci. 2009 Apr;100(4):572-9, PMID: 19462526 [PubMed - in process]
- Title: Active TGF-beta1 correlates with myofibroblasts and malignancy in the colorectal adenoma-carcinoma sequence.
- Author: Hawinkels LJ, Verspaget HW, van der Reijden JJ, van der Zon JM, Verheijen JH, Hommes DW, Lamers CB, Sier CF.
- Journal: Cancer Sci. 2009 Apr;100(4):663-70, PMID: 19462508 [PubMed - in process]
- Abstract: Transforming growth factor-beta1 (TGF-beta1), a cytokine involved in various stages of cancer, is produced as a latent complex and requires processing to become active. We have determined total and active TGF-beta1 levels in homogenates of colorectal neoplasia. In contrast to total TGF-b levels, showing a stepwise increase in the mucosa-adenoma-carcinoma sequence, active TGF-beta1 levels are increased only in carcinomas but not in premalignant adenomas. Furthermore, solely active TGF-beta1 levels are associated with the stage of the carcinomas and worse patient prognosis. Active TGF-beta1 levels correlated significantly with plasminogen activator inhibitor (PAI)-1, alpha-smooth muscle actin (SMA) and several matrix-remodeling proteinases. Interestingly, SMA levels are also significantly increased in colorectal carcinomas but not in adenomas, suggesting that despite the enhanced total TGF-beta1 levels, myofibroblast accumulation is not (yet) occurring in these premalignant neoplasias. The correlation between active TGF-beta1 and SMA expression in tumors indicates that tumor-promoting myofibroblasts might arise as a result of increased TGF-beta1 activation. These data underline the significance of the interaction between malignant cells and (myo)-fibroblasts in the tumor microenvironment, modulating the biologic behavior of colorectal cancer.
- Buttons: Lagre, Skjul
- Section: Kommentar
- Text: Hvirker veldig lovende.
- Button: Save

Figur 4.19: Visning av abstract med lagring av artikkel.

The image shows two side-by-side screenshots of a web application interface. The left screenshot displays a list of three abstracts, each with a title, authors, journal information, and a text input field for a comment. The right screenshot shows a detailed evaluation form for a 'Randomisert Kontrollert Studie (RCT)'. It includes a section for 'Følgende forhold må vurderes:' with three sub-questions, a 'God, (Svært god)' button, and two numbered questions with radio button options.

Left Screenshot (List of Abstracts):

- Abstract 1:** Title: Active TGF-beta1 correlates with myofibroblasts and malignancy in the colorectal adenoma-carcinoma sequence. Authors: Hawinkels LJ, Verspaget HW, van der Reijden JJ, van der Zon JM, Verheijen JH, Hommes DW, Lamers CB, Sier CF. Journal: Cancer Sci. 2009 Apr;100(4):693-70. PMID: 19462508 [PubMed - in process]. Comment: Hvirker veldig lovende.
- Abstract 2:** Title: Active TGF-beta1 correlates with myofibroblasts and malignancy in the colorectal adenoma-carcinoma sequence. Authors: Hawinkels LJ, Verspaget HW, van der Reijden JJ, van der Zon JM, Verheijen JH, Hommes DW, Lamers CB, Sier CF. Journal: Cancer Sci. 2009 Apr;100(4):693-70. PMID: 19462508 [PubMed - in process]. Comment: A comment
- Abstract 3:** Title: Update of cylindromatosis gene (CYLD) mutations in Brooke-Spiegler syndrome: novel insights into the role of deubiquitination in cell signaling. Authors: Blake PW, Toro JR. Journal: Hum Mutat. 2009 Mar 27; PMID: 19462465 [PubMed - as supplied by publisher]. Comment: bra
- Abstract 4:** Title: Dietary fat intake and risk of skin cancer: A prospective study in Australian adults. Authors: Ibiebele TI, van der Pols JC, Hughes MC, Marks GC, Green AC. Journal: Int J Cancer. 2009 Apr 6; PMID: 19462452 [PubMed - as supplied by publisher].

Right Screenshot (Evaluation Form):

Randomisert Kontrollert Studie (RCT)

Følgende forhold må vurderes:

- Kan vi stole på resultatene?
- Hva forteller resultatene?
- Kan resultatene være til hjelp i praksis?

God, (Svært god) Lagre

Innledende spørsmål

1. Er formålet med studien klart formulert?

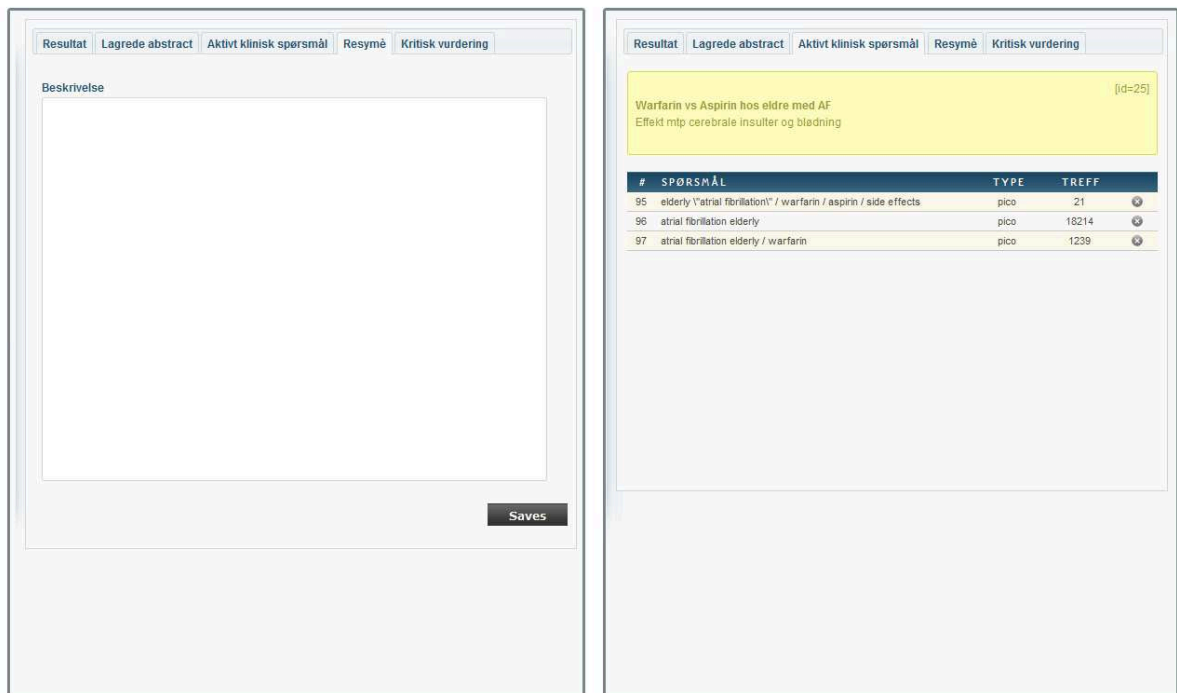
Ja
 Uklart
 Nei

2. Er en randomisert kontrollert studie et velegnet design for å besvare spørsmålet?

Ja
 Uklart
 Nei

Kan du stole på resultatene?

Figur 4.20: T.v. visning av lagrede abstract. Mulighet for å rediger kommentarer. *Kritisk vurder* linken aktiverer visningen t.h.



Figur 4.21: T.v. visning av resyme-editor. Denne vil bli erstattet av en rik-tekst editor. T.h. Visning av aktivt klinisk spørsmål med sine søkekomposisjoner.

4.4.3 Kritisk vurder

Her skulle vi ta utgangspunkt i sjekklister til Nasjonalt kunnskapsseter for *Nasjonalt kunnskapssenter for helsetjenesten* (se seksjon 1.6.2). Utfordringen var å lage et generisk systemet som kan inneholde alle typer sjekklister, med alle mulige kombinasjoner av kontrollspørsmål. Både sjekklistedefinisjoner og sjekklisteinstanser må være generiske. Det vil si at hvis det opprettes en ny type sjekklister, må systemet automatisk ha støtte for å opprette en instans av denne.

For å løse denne oppgaven med minst mulig kompleksitet gjorde vi oss den forutsetningen at et kontrollspørsmål kan enten være en (Ja, Uklart, Nei) gruppe, eller et kommentar felt, i tillegg til metainformasjon om hvert enkelt spørsmål, hint som vi har kalt det i databasestrukturen (se figur 4.22).

Datastrukturen kan deles opp i to deler; *schema* som representerer definisjonen og *instance* som representerer en eksisterende instans. De er koblet sammen på en slik måte at en definisjon kan ha mange instanser.

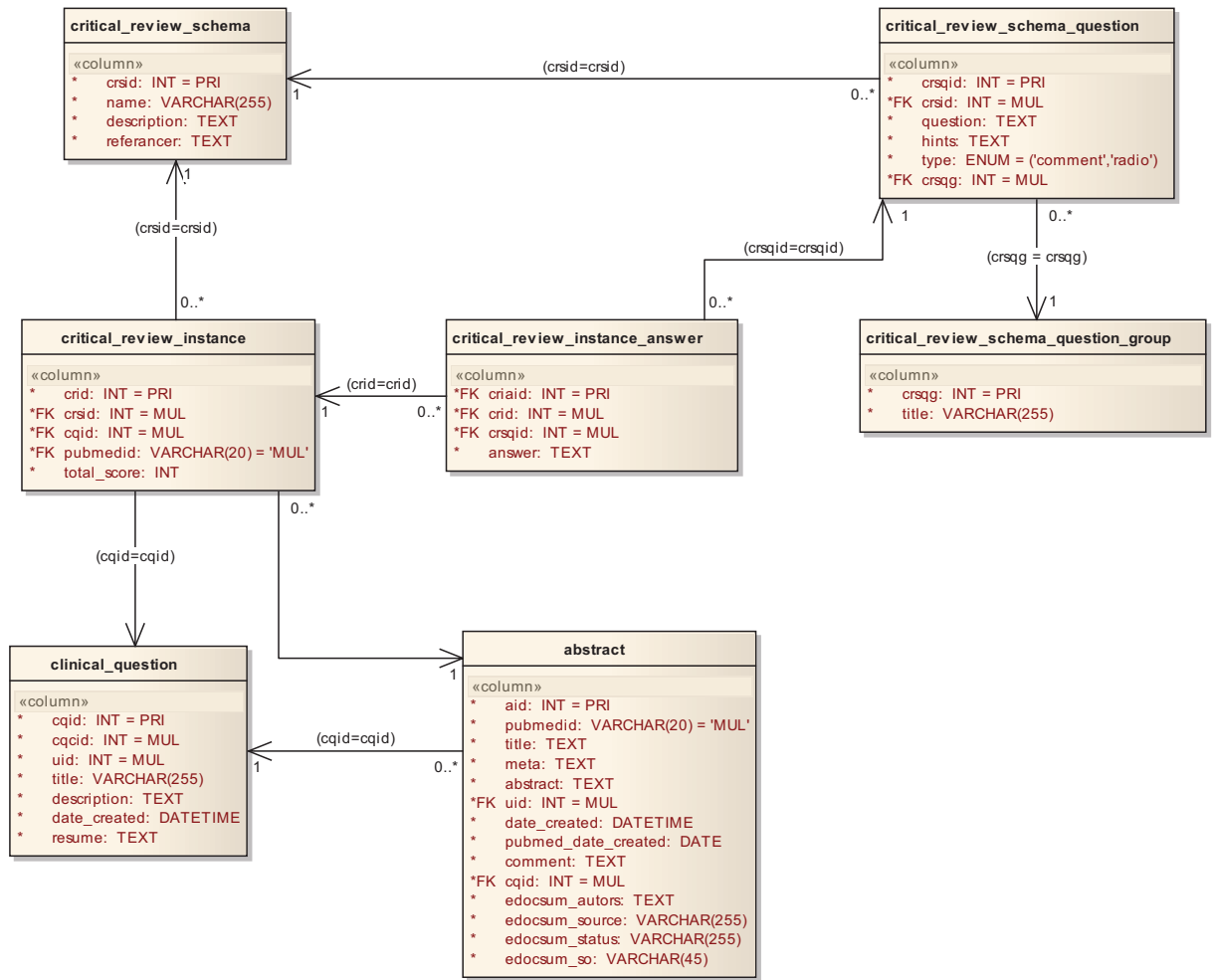
Når en instans skal opprettes tar algoritmen en definisjon id (*crsid*) som argument og går systematisk gjennom definisjonens kontrollspørsmålliste²⁴ og oppretter en nytt kontrollskjemainstans²⁵ for hvert kontrollspørsmål. For å unngå at bare noen felter blir opprettet har vi brukt databasetransaksjoner og lagt hele operasjonen *atomisk*.

Figur 4.23 viser likheten mellom *Nasjonalt kunnskapssenter for helsetjenesten* sine sjekklister og de interaktive digitale listene. I tillegg har vi også implementert en overordnet vurdering som består av en enkelt *drop down liste* som relatert til *critical_review_instance* sin *total_score*. Denne verdien kan senere brukes til å vise fargekoder på lagrede abstracts for å lettere se hvilke abstracts som er interessante.

Hele implementasjonen er lagt under “Kritisk vurder” fanen i søkedelen av webapplikasjonen. Alt foregår via *Ajax* uten å laste sidene på nytt.

²⁴`critical_review_schema_question`

²⁵`critical_review_instance_answer`



Figur 4.22: Datastrukturen som er relatert til kritisk vurdering

Kritisk vurdering - RCT

INNLEDENDE SPØRSMÅL

1. Er formålet med studien klart formulert?	Ja O	Uklart O	Nei O
2. Er en randomisert kontrollert studie et velegnet design for å besvare spørsmålet?	Ja O	Uklart O	Nei O

KAN DU STOLE PÅ RESULTATENE

3. Ble utvalget fordelt til de ulike gruppene ved bruk av tilfredsstillende randomiseringsprosedyre? <i>TIPS:</i> <ul style="list-style-type: none"> Randomiseringsprosedyren bør være beskrevet Randomiseringsprosedyren er tilfredsstillende dersom den er skjult for den som fordeler (eks lukkede konvolutter, dataprogram, tabeller etc) Se om gruppene er like ved oppstart av studien (etter randomisering) med hensyn til f.eks. alder, kjønn, sosial klasse, smerte, funksjon 	Ja O	Uklart O	Nei O
4. Ble gruppene behandlet likt bortsett fra tiltaket som evalueres? <i>TIPS:</i> <ul style="list-style-type: none"> Eventuelle tilleggstiltak bør unngås eller være like i gruppene som sammenlignes. For å unngå forskjellsbehandling, er det beskrevet om tiltakene ble gjennomført i forhold til en protokoll? 	Ja O	Uklart O	Nei O
5. Ble deltakere, helsepersonell og utfallsmåler blindet mht. gruppetilhørighet? <i>TIPS:</i> <ul style="list-style-type: none"> Det er ikke alltid mulig å blinde deltakere (pasienten) og helsepersonell (de som gir tiltaket) – men man bør vurdere om blinding var mulig. Subjektive utfallsmål (f. eks. smerte eller funksjon) har økt risiko for bias, hvis ikke blinding. Den som måler utfallet kan som regel blindes. 	Ja O	Uklart O	Nei O

Nasjonalt kunnskapssenter for helsetjenesten, 2008.

Resultat Lagrede abstract Aktivt klinisk spørsmål Resymé Kritisk vurdering

Randomisert Kontrollert Studie (RCT)

Følgende forhold må vurderes:
 - Er utvalg på resultatene?
 - Er det tilstrekkelig randomisert?
 - Er resultatene klare i lys av prosedyre?

Dårlig, (under middels) Lagre

Innledende spørsmål

1. Er formålet med studien klart formulert?

Ja
 Uklart
 Nei

2. Er en randomisert kontrollert studie et velegnet design for å besvare spørsmålet?

Ja
 Uklart
 Nei

Kan du stole på resultatene?

3. Ble utvalget fordelt til de ulike ruppene ved bruk av tilfredsstillende randomiseringsprosedyre?
 - Randomiseringsprosedyren bør være beskrevet
 - Randomiseringsprosedyren er tilfredsstillende dersom den er skjult for den som fordeler (eks lukkede konvolutter, dataprogram, tabeller etc)
 - Se om gruppene er like ved oppstart av studien (etter randomisering) med hensyn til f.eks. alder, kjønn, sosial klasse, smerte, funksjon

Ja
 Uklart
 Nei

4. Ble gruppene behandlet likt bortsett fra tiltaket som evalueres?
 - Eventuelle tilleggstiltak bør unngås eller være like i gruppene som sammenlignes.
 - For å unngå forskjellsbehandling, er det beskrevet om tiltakene ble gjennomført i forhold til en protokoll?

Ja
 Uklart
 Nei

5. Ble deltakere, helsepersonell og utfallsmåler blindet mht. gruppetilhørighet?
 - Det er ikke alltid mulig å blinde deltakere (pasienten) og helsepersonell (de som gir tiltaket) – men man bør vurdere om blinding var mulig.
 - Subjektive utfallsmål (f. eks. smerte eller funksjon) har økt risiko for bias, hvis ikke blinding.
 - Den som måler utfallet kan som regel blindes.

Ja
 Uklart
 Nei

Figur 4.23: Sammenligning av pdf-sjekkliste mot interaktiv sjekkliste

4.5 Små detaljer

4.5.1 Formatering av abstract

All informasjon som kommer fra EUtilities kommer uformatert og i klartekst. En følge av dette er abstractene kan være tunge å lese, da de inneholder tekst uten linjeskift. Vi har prøvd å bøte på dette ved å bruke regular expression i javascriptet som laster inn abstractene via Ajax. Her har vi utnyttet at mange abstract inneholder mye av de samme oppsettene. For eksempel bruker mange abstract formen **BACKGROUND: (...)** og **RESULT: (...)** osv. Følgende *regular expression* formaterer titlene til fet skrift og setter inn linjeskift rundt disse overskriftene:

```
1 function injectNewLines( sourceString ) {
2   sourceString.replace(/(?:^|\.)\s*([a-zA-Z0-9\s]+:\s*)/gi, "<br
   /><br /><b>§1</b><br />");
3   sourceString.replace(/^(<br \/>)*\/gi, "");
4   return sourceString;
5 }
```

Kapittel 5

Avslutning

5.1 Diskusjon av resultater

Ved endelsen av prosjektet står vi igjen med to fungerende prototyper. Et webgrensesnitt og en håndholdt applikasjon. Utifra de krav vi fikk fra oppdragsgiver og egne tolkninger (se kapittel 2 for kravspesifikasjon) har vi klart å lage to systemer som samarbeider med hverandre.

5.1.1 Drøfting av resultater

Vårt prosjekt besto av to hele systemer, et webgrensesnitt og en håndholdt enhet. Prosjektets mål fra oppdragsgivers side var å få til prototyper som kunne vise hvordan dette kunne se ut og gi grobunn for videre satsing som en del av et større nasjonalt prosjekt. Vi har fokusert på å se på systemene som mer enn prototyper og forsøkt etter beste evne og komme så langt som mulig, slik at dette kan taes i bruk direkte. Til en viss grad føler vi at vi har lyktes med dette.

Begge systemene integrerer godt med hverandre, og fungerer tilstrekkelig slik at en bruker kan f.eks. opprette og synkronisere et klinisk spørsmål på farten. Søk gjøres gjennom webserveren, dette fordi brukeren må uansett være online for å gjøre søk, og det går raskere gjennom webserveren å behandle resultatene. For å gjøre disse systemene fullendte så kreves det mer tid enn prosjektets rammer ga, dette særskilt på bakgrunn av at utvikling på håndholdte enheter er mer utfordrende enn på desktop. Noe vi fikk erfare den vanskelige veien, mer om dette i *seksjon ??*. Resultatet er derfor en god indikasjon på virkemåte og hvordan dette bør gå videre, og skal ikke vurderes etter fullendhet. Innenfor bacheloroppgavens rammer

er det gjenstående noen mindre oppgaver, disse ble nedprioritert og det lot seg ikke gjøre å få de ferdigstilt innenfor tidsfristen. Se *seksjon ??* for detaljer om dette.

Domeneproblematikken

Den vanskeligste delen av prosjektet vårt, og noe som gjorde at vi fikk dårligere tid enn planlagt, var det diffuse domeneområdet. Dette er forsøkt illustrert i *seksjon 1.6*, hvor den faglige bakgrunnen forklares. Kunnskapsbasert praksis er et veldig vidt område, det er vanskelig å definere selv for leger og medisinsk personell som driver med dette til daglig. PICObase i seg selv er ment for å brukes til kursing av klinikere som har som mål å ta i bruk dette i sin hverdag på en bedre måte. For oss som studenter uten medisinsk bakgrunn ledet dette fort til feiltolkninger, og var en vanskelig prosess gjennom hele prosjektet. Dette har derfor hatt direkte innvirkning på resultatet, og gjort at systemene har en lengre vei å gå for å være et fullverdig medisinsk verktøy.

5.2 Videre arbeid og forbedringsmuligheter

For at applikasjonene skal være robust nok til dagelig bruk må den kvalitetssikres. Vi fikk dessverre ikke tid til å unit teste alle deler av applikasjonen, noe vi gjerne har lyst til å gjøre ved en eventuelt fortsettelse av prosjektet etter innlevering av bacheloroppgave. Vi har prioritert de viktigste delene som søk og behandling av søk. Vi har brukt mye tid på design, både programmessig og grafisk. Vi la stor vekt på at applikasjonen skulle ha en moderne look og feel. Det har vært mange designrevisjoner til endelig resultat. Dette har imidlertid ført til at vi ikke har fått tid til noe mindre, trivielle Use Case som oppretting av brukere og publisering av resyme. Men vi føler det har vært mer viktig å vie mer tid og energi på de viktigste delene, og at dette har gitt resultat.

5.3 Evaluering av gruppas arbeid

5.3.1 Organisering og fordeling av arbeid

Prosjektets ledelse har i praksis foregått i felleskap, og det har ikke vært behov for megling over avgjørelser. Disse er blitt tatt sammen og etter diskusjon. Siden vi bare har vært to i gruppa har kommunikasjon med veileder og oppdragsgiver

blitt utført av begge, men på felles grunnlag. Vi så ikke noe behov for å forsøke en streng administrativ modell på bakgrunn av dette, noe som har fungert bra. Øvrige ansvarsområder er dekket av forprosjektet (se *seksjon A.3.1*) og har blitt opprettholdt uten endringer.

5.3.2 Fordeling av arbeidet

I prosjektets første del jobbet vi mye sammen om de samme arbeidsoppgavene, dette innebærte deler som forprosjektrapporten og kartlegging av krav, utviklingsmiljø etc. Men grunnstruktur og design, særskilt på den håndholdte applikasjonen ble gjort sammen. Et resultat av at vi begge var nye på utviklingsplattformen og at designet er viktig å være helt enige om. Som illustrert i *seksjon 5.1.1* førte tidsporene til at vi måtte gjøre drastiske endringer i fordelingen av arbeidet. All utvikling på de to systemene ble derfor todelt fra slutten av mars og ut prosjektperioden. Til tross for den kritiske situasjonen så fungerte dette over all forventning.

5.3.3 Loggføring

For å holde oversikt over progresjon og føring av timelogg og møtereferater, lagde vi vårt eget backend grensesnitt til prosjektets hjemmeside. Her ble alle timer ført, og møtereferater lagt inn gjennom et enkelt brukergrensesnitt. Dette ga oss bl.a. mulighet for jevnlig å ta backup til SVN og en mer effektiv måte å føre arbeidslogg på, enn ved manuell føring i f.eks. regneark.

5.3.4 Prosjekt som arbeidsform

Prosjektarbeid eller arbeid i team er noe som er vanlig i arbeidslivet i dag, hvor det er veldig sjeldent at én person har hovedansvaret for et helt prosjekt alene. Bachelorprosjektet har derfor vært en god prøvelse i hvordan et større utviklingsprosjekt fungerer som arbeidsmetode. Til forskjell fra prosjektoppgaver av varierende omfang gjennom utdanningsforløpet, har dette vært et prosjekt med reell arbeidsgiver og andre enn seg selv å tilfredsstillere. Det mest utfordrende med prosjektgjennomføringen har vært å beregne tid i forhold til oppgaven, noe som har vært spesielt utfordrende i vårt tilfelle med et diffust domeneområde (se *seksjon 5.1.1*).

5.3.5 Konklusjon

PICObase har vært et vanskelig prosjekt å få tak på, dette basert på domeneproblematikken illustrert i *seksjon 5.1.1*. Den tidlige fasen besto av mye kommunikasjon med oppdragsgivers kontaktperson for å prøve å få tak på dette. Men det viste seg å være enda vanskeligere, da det oppsto flere misforståelser og fokuset ble overført til detaljer innen GUI. Dette er noe som påvirket prosjektgjennomførelsen, hvor vi aldri helt fikk oversikt over hva kunnskapsbasert praksis egentlig er. Den viktigste erfaringen vi har lært er derfor at man må bruke så mye tid som mulig på å avdekke domeneområdet, særlig med et ikke-konkret område som vi har måttet takle. En nyttig erfaring vi er glade for å kunne ta med oss videre.

Når vi ser tilbake på prosjektet som helhet, så føler vi at vi har lært veldig mye om det å takle utfordringer, som å gå over tidsgrenser og bruke for mye tid på noe man ikke helt forstår. Det er derfor givende å se at vi nå på slutten, har klart å se hva vi har gjort feil, og hvordan vi kan hindre dette ved senere prosjekter. Men til tross for dette føler vi at vi har et fullverdig resultat vi kan være stolte av.

Bibliografi

- [1] Tim Converse, Joyce Parks, Clark Morgan, *PHP 5 and MySQL bible* 2004.
- [2] Martin Fowler, *Patterns of Enterprise Application Architecture*, 2002
- [3] Martin Fowler, *Inversion of Control Containers and the Dependency Injection pattern*, 2004,
<http://www.martinfowler.com/articles/injection.html>
- [4] Erich Gamma, Richard Helm, Ralph Johnson, John M. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, 1994
- [5] Philippe Kruchten, *Applying UML and patterns*, 2005
- [6] Andrew S. Tanenbaum, *Modern Operating systems 3e*, 2008
- [7] Bill Todd *Understanding InterBase Transactions* , 2004,
<http://conferences.embarcadero.com/article/32280>
- [8] Jasmin Blanchette, Mark Summerfield *C++ GUI Programming with Qt*, 2006
- [9] National Center for Biotechnology Information, *Entrez Programming Utilities*, 2009,
http://www.ncbi.nlm.nih.gov/entrez/query/static/eutils_help.html
- [10] NCBI Help Manual, *Entrez Help*, 2006,
<http://www.ncbi.nlm.nih.gov/bookshelf/br.fcgi?book=helpentrez&part=EntrezHelp>
- [11] Zend Framework, *Programmer's Reference Guide - Zend Auth*,
<http://framework.zend.com/manual/en/zend.auth.html>
- [12] Zend Framework, *Programmer's Reference Guide - HTTP Authentication Adapter*, <http://framework.zend.com/manual/en/zend.auth.adapter.http.html>
- [13] Zend Framework, *Programmer's Reference Guide - Zend Acl*,
<http://framework.zend.com/manual/en/zend.acl.html>

-
- [14] Zend Framework, *Programmer's Reference Guide - Zend Acl - Advanced Usage*,
<http://framework.zend.com/manual/en/zend.acl.advanced.html>
- [15] Zend Framework, *Programmer's Reference Guide - Zend_Controller Plugins*,
<http://framework.zend.com/manual/en/zend.controller.plugins.html>
- [16] Johan Thelin, *Foundations of Qt Development*,
2007
- [17] Leonard Richardson, Sam Ruby, David Heinemeier Hansson, *RESTful Web Services*, 2007

Kapittel 6

Vedlegg

Tillegg A

Forprosjekt

A.1 Mål og rammer

A.1.1 Innledning

Daglig står helsepersonell i praksis overfor en rekke situasjoner som innebærer å måtte ta faglige beslutninger. Kunnskapsbasert praksis (KBP) er å ta faglige avgjørelser basert på systematisk innhentet forskningsbasert kunnskap, erfaringsbasert kunnskap og pasientens ønsker og behov i den gitte situasjonen.

Sykehuset Innlandet jobber mye med å innføre kunnskapsbasert praksis i sin virksomhet. De ser derfor stor nytte i å få til et hjelpemiddel til bruk gjennom håndholdte enheter og et webgrensesnitt, hvor det kan gjøres søk basert på kliniske stilte spørsmål (PICO¹) i medisinske databaser, og evaluere resultatene av disse.

PICObase skal bli løsningen på dette, en egen applikasjon utviklet for håndholdte enheter, med synkronisering mot en databaseserver og et webgrensesnitt med samme funksjonalitet i tillegg til administrative muligheter.

¹(P)asient, (I)ntervention, (C)ontrol, (O)utcome er en måte å formulere et klinisk stilt spørsmål på.

A.1.2 Effektmål

Effekten av prosjektet vil være redusert tid og arbeid med besvarelse av kliniske spørsmål. Flere klinikere vil ta i bruk kunnskapsbasert praksis mer bevisst og kliniske spørsmål blir stilt, evaluert og fulgt opp, i mye større grad enn i dagens situasjon. Kunnskapsbasert praksis som verktøy vil være mer tilgjengelig, selv utenfor kontoret (mobilitet).

A.1.3 Resultatmål

Målet er å produsere et brukervennlig verktøy, som kan tilby både enkle og avanserte kliniske søk i medisinske databaser, basert på kunnskapsbasert praksis. Ved bruk av håndholdte enheter skal verktøyet kunne brukes uavhengig av fast tilholdssted. Evaluering av resultat skal inngå som en naturlig del av systemet.

A.1.4 Læringsmål

Under prosjektets gang skal det fokuseres på å anvende kunnskap gruppen har i dag, i en større prosjektsammenheng. Det legges også vekt på modning i et reelt utviklingsprosjekt, der ny teknologi læres og effektivt taes i bruk. Det skal arbeides systematisk etter gitt prosjektmodell og strukturert tankegang. Dette bidrar til å sikre prosjektets fremgang og utførelse.

A.1.5 Rammer

Prosjektets tidsfrist er satt av Høgskolen i Gjøvik (HiG), avdeling for Informatikk og Medieteknikk, til 20.mai 2009. Det kreves dokumentasjon av alt arbeid, dette skal sammenbefattes i ett dokument ved sluttendelse av prosjektet.

Øvrige krav av oppdragsgiver utover gitt oppgavebeskrivelse etterkommes hvis forsvarlig, gjennomførbart og i tråd med HiG sine retningslinjer.

Det stilles krav til at den håndholdte applikasjonen skal fungere på enheter med Windows Mobile 5, men gruppen vil jobbe for å legge til rette for enkel portering til andre formater ved valg av utviklingsmiljø.

A.2 Omfang

A.2.1 Oppgavebeskrivelse

Gruppen skal i prosjektperioden utvikle et verktøy som skal bistå klinisk personell i daglig arbeid med KBP. Applikasjonen, heretter kalt PICObase, skal spesifikt være et verktøy for å administrere kliniske spørsmål og svar på disse gjennom medisinske databaser. Klinisk personell skal ved hjelp PICObase kunne:

- Bygge opp kliniske spørsmål med PICO-formulering
- Søke i medisinske databaser som PubMed² vha. det oppbygde søket
- Behandle og revidere søkeresultatene
- Publisere resymér basert på artikler fra medisinske databaser

PICObase skal være tilgjengelig både på håndholdte enheter og via et webgrensesnitt.

Webgrensesnittet og den håndholdte enheten skal synkroniseres så lenge begge systemer er tilkoblet Internett. Det er derfor viktig å finne en løsning som muliggjør offline lagring ved brudd på nettilgang.

Bruker grensesnittet på begge modulene skal være intuitive og styrke opp mot KBP, dette er særskilt viktig på den håndholdte enheten.

Administrasjonspanelet i webgrensesnittet skal ha mulighet for å vise full historikk både på brukernes egne og alle andres kliniske søk. Det skal også være mulighet for å lage resymé av artikler og publisere disse. Dette krever moderering av godkjent personell, og medfører krav om egen autentiseringsmodul på denne biten.

²<http://www.ncbi.nlm.nih.gov/pubmed/>

A.2.2 Avgrensninger

Gruppen skal ikke ha ansvar for drifting, backuprutiner o.l. av PICObase-server. Dette tar oppdragsgiver seg av.

I prosjektets tidsplan³ er det satt opp to inkremitter; printfunksjon og ordbokfunksjon. Disse inngår ikke i kjernefunksjonaliteten og vil om nødvendig neglisjeres til fordel for høyere prioriterte inkremitter⁴.

³Se Gantt-skjema i seksjon A.6.1.

⁴Se seksjon A.5.4 Risikoanalyse

A.3 Prosjektorganisering

A.3.1 Ansvarsforhold

Øyvind Skattum Vesteng - Prosjektleder og rapport- og dokumentasjonsansvarlig.
Arild Plassen - Ansvarlig for webside, konfigurasjonsstyring og tidsplanlegging.

I tillegg skal det være delt ansvar på initiativ ovenfor oppdragsgiver, og oppfølging av alt arbeid. Delegering av prosjektspesifikke oppgaver og utviklingsmessige forhold vil bli fordelt fortløpende. Hvert gruppemedlem har spesielt ansvar for gjennomføring av sine delegerte oppgaver, særskilt der lik funksjonalitet skal implementeres parallelt både på håndholdt enhet og i webgrensesnittet.

A.3.2 Grupperegler

- Hvert gruppemedlem skal legge ned minst 30 timers arbeid per uke. Alt skal dokumenteres i prosjektets timeloggssystem, med antall timer og beskrivelse av utført arbeid.
- Uenigheter løses innad i gruppen, men siden gruppen kun består av to personer skal kritiske uenigheter taes opp i samråd med veileder for å fatte en beslutning.
- Langvarig sykdom eller andre prosjektkritiske tilstander skal det informeres om innad i gruppen så fort det oppstår.
- Formelle møter skal det genereres møtereferat fra og legges inn i prosjektes administrasjonssystem.

A.3.3 Rutiner

- Websideansvarlig skal ukentlig produsere oppdateringer på prosjektets webside.
- Timelogg for hvert gruppemedlem skal føres ved dagsslutt i prosjektes administrasjonsside.
- Hvert gruppemedlem har ansvar for å hele tiden holde seg oppdatert på korrespondanse vedrørende prosjektet.
- Ved utsendelse av mail skal alle berørte interessenter stå som CC eller BCC.

A.3.4 Øvrige roller og bemanning

Øivind Kolloen, prosjektets veileder. Oppdragsgiver Håvard Njå ved Sykehuset Innlandet avd. Gjøvik. Kunnskapsbasert praksis v / Per Ola Vandvik. Bibliotekar ved Helsebiblioteket.

A.4 Planlegging, oppfølging og rapportering

A.4.1 Hovedinndeling av prosjektet

Som nevnt i oppgavebeskrivelsen består prosjektet av to applikasjoner; en applikasjon spesialtilpasset for håndholdte enheter og en som kjører bak et webgrensesnitt. Webgrensesnittet skal ha de samme funksjonene som den håndholdte enheten, men skal også ha mer avansert funksjonalitet for å etterbehandle de kliniske spørsmålene og fungere som et sentralt administrasjonspanel for hele systemet.

Funksjonaliteten som finnes både på håndholdt enhet og på web skal lages parallelt slik at det kan leveres et komplett inkrement på implementasjon av disse funksjonene.

Listen som følger viser grovinndeling av systemet. H indikerer håndholdt applikasjon og W web.

- Søk ($H+W$)
- Etterbehandling av søk (W)
- Brukerbehandling og autentisering ($H+W$)
- Grensesnitt mellom håndholdt enhet og PICObase-server ($H+W$)
- Printgrensesnitt (H)
- Ordbokfunksjonalitet ($H+W$)

A.4.2 Prosjektmodell

Kommunikasjon med arbeidsgiver er en viktig suksessfaktor; prosjektmodellen skal sørge for at inkrementene som leveres er i tråd med forventningene og i tillegg gi rom for endringer og/eller optimaliseringer underveis.

Spesielt på den håndholdte enheten er det kritisk at brukergrensesnittet er så effektivt og optimalisert som mulig. Vi mener den beste måten å oppnå dette på er gjennom iterasjon på de prosessene som er relatert til brukergrensesnittet og brukeropplevelse.

Prosjektmodellen som gruppen ender opp med er sterkt inspirert av den inkrementelle utviklingsmodellen med rom for iterasjoner gjennom inkrementene⁵.

Prosjektet har også avhengigheter i funksjonaliteten som den inkrementelle modellen adresserer effektivt, jmf. gantt-skjema.

⁵I gantt-skjemaet er ikke iterasjonene merket, se seksjon A.6.1.

A.4.3 Plan for statusmøter

Det er planlagt statusmøter med veileder annenhver mandag kl. 1500. Formelle møter utover dette vil bli avtalt ved behov.

Oppfølgingsmøte med oppdragsgiver skal holdes hver 3.uke, møter utover dette vil kun bli holdt ved behov.

A.5 Kvalitetssikring

A.5.1 Konfigurasjonsstyring

For å holde rede på alle versjoner av vårt arbeid, og sikre at det til enhver tid er oppdatert informasjon og kode som brukes, skal gruppa benytte Subversion (svn). HiG tilrettelegger for dette gjennom it-tjenesten, med eget svn-område. For å administrere området skal programmet TortoiseSvn brukes.

Subversion skal fungere som backup, da prosjektrapporten også legges inn i repositoryet, og skolens rutiner for backup av server anses som gode nok. Ukentlig backup av administrativ database for prosjektet (timelogg o.l.). Annet arbeid relatert til prosjektet skal ha oppdatert kopi på egen minnepinne for hvert grupped medlem.

A.5.2 Kildekode og dokumentasjon

For å generere kildekodedokumentasjon i lesbart format skal Doxygen⁶ brukes både til applikasjonsutvikling og webgrensesnittet.

A.5.3 Kodestandard

All kode skal kommenteres i henhold til Doxygenformatet, og være på engelsk. Navnekonvensjoner og øvrig kodestandard for håndholdt applikasjon skal ta i bruk Qt-style formatet⁷.

⁶Doxygen Official Website: <http://www.doxygen.org/>

⁷Qt-style konvensjoner: <http://doc.trolltech.com/qq/qq13-apis.html>

A.5.4 Risikoanalyse

- Prosjektets kritiske inkremitter blir ikke ferdig til tidsfristen⁸
 - Risiko: **Høy**
 - Målrettet arbeid i forhold til milepæler. Gantt-skjema revideres og korrigeres gjennom hele prosjektet for å gi et oppdatert bilde av prosjektstatus. I verste fall kuttes de siste modulene til fordel for kjernefunksjonalitet.⁹
- Produktets virkemåte er ikke i tråd med kunnskapsbasert praksis sitt rammeverk.
 - Risiko: **Middels**
 - Innførende kurs i KBP og utover i prosessen samarbeid med bibliotekar fra helsebiblioteket.
- Databasedesignet er ikke holdbart over tid grunnet sprenget kapasitet.
 - Risiko: **Middels**
 - Kartlegge lagringsbehovet over tid gjennom estimering, og stresstesting av databasen tidlig i prosessen.
- Et gruppemedlem kan ikke fortsette / midlertidig ute av drift grunnet sykdom e.l. Da gruppen kun består av to personer, setter dette prosjektet i kritisk fase.
 - Risiko: **Middels**
 - Gjenværende part av gruppen vurderer i lag med veileder hvorvidt det er hensiktsmessig å fortsette, eventuelle kompromisser taes for å kompensere for situasjonen.
- Det lykkes ikke å hente ut data på ønsket og hensiktsmessig måte fra medisinske databaser.
 - Risiko: **Lav**
 - Kommunikasjon med driftere av de største medisinske databasene, og søk etter mulige api'er o.l. som muliggjør elektronisk respons og programmatisk søk.

⁸Printgrensesnitt og ordbokfunksjonalitet kan utsettes til fordel for de øvrige.

⁹Se seksjon A.2.2 Avgrensninger.

A.6 Plan for gjennomføring

A.6.1 Forklaring til gantt-skjema

Prosjektet er inndelt etter den inkrementelle modellen. Det er lagt opp til fem målbare inkremitter som skal leveres utover prosjektperioden. For å imøtekomme eventuelle nye krav og optimalisering underveis, kjøres det en iterativ prosess under hvert av inkrementene. Antall iterasjoner vil variere etter behov, for eksempel vil inkrement 1 (Søk) og inkrement 3 (Etterbehandling av søk) ha større behov for flere iterasjoner da de omhandler brukergrensesnitt for kjernefunksjonalitet. Aktivitetene *Revisjon av kravspesifikasjon* og *Revisjon av design* viser til den iterative tankegangen i inkrementene.

Testing vil inngå i de iterative syklusene.

Gantt-skjemaet har en egen aktivitet som heter *Revisjonsfase*. Revisjonsfasen er ment å være en fase der applikasjonen som en helhet revideres og testes. Det arbeides her kun mot eksisterende moduler og funksjoner, ingen nye.

Nedenfor følger en oversikt over de leverbare milepælene:

1. Søk (13.03)
2. Brukerbehandling og autentisering (24.03)
3. Etterbehandling av søk (08.04)
4. Printgrensesnitt (15.04)
5. Ordbokfunksjonalitet (22.04)

Grunnstrukturen er ikke ansett som et eget leverbart inkrement da det ikke er målbart av arbeidsgiver, e.g. inget brukergrensesnitt.

Det gjøres oppmerksom på at tidsdisposisjonen for de enkelte aktivitetene er estimerer og vil oppdateres fortløpende i prosjektet. Gantt-skjemaet vil holdes oppdatert på prosjektets hjemmesider¹⁰.

¹⁰Prosjektwebsiden: <http://hovedprosjekter.hig.no/v2009/imt/in/picobase/>

Tillegg B

Statusrapporter

B.1 Status

B.1.1 Planlegging

Henviser til Gantt-skjema lagt ut på prosjektets hjemmeside: hovedprosjekter.hig.no/v2009/imt/in/picobase.

B.1.2 Organisering av gruppens arbeid og ansvarsområder

Design- og arkitekturmessige avgjørelser blir tatt mellom gruppens to medlemmer, men forslag til utkast blir utarbeidet både sammen og individuelt, for så å diskuteres over etterpå. Det er i størst mulig grad et samarbeidsprosjekt, særskilt i den første tredjedelen av prosjektets utførelse. Arild har ansvaret for server, database og prosjektets egen webside. Dette på bakgrunn av tidligere ekspertise på feltet. Mens Øyvind har dokument- og rapportansvar, i tillegg til felles ansvar for applikasjonsutvikling på begge grensesnitt. Selv om det på papiret er delt opp i ansvarsområder har begge et like stort ansvar for at arbeidet skal gjennomføres og gjøres på en best mulig måte. SVN er en sentral kilde for samarbeid mellom prosjektets medlemmer, det er også blitt tatt i bruk mye telefon. Det er blitt lagt vekt på visualisering og bruk av både elektroniske (Enterprise Architect, Illustrator) og fysiske hjelpemidler (tavle, whiteboard) for å få frem viktige problemstillinger og løse disse på felles grunnlag.

B.1.3 Klargjøring av problemstilling

Problemstillingen sett under et er tro mot det som ble definert under forprosjektrapporten. Men ettersom prosjektet får mer håndfast så vil det bli lagt mer vekt på den grafiske delen av applikasjonsorienteringen, dette i samarbeid med oppdragsgiver. Da dette er en sentral del av prosjektet, det skal øke kunnskapsbasert praksis og bruk av dette ved et lettfattelig og intuitivt brukergrensesnitt, samtidig som avanserte søk kan utføres.

B.1.4 Løsningsmetode

Det benyttes et bredt spekter av både utviklingsverktøy, designverktøy og teknologier og metoder for å utvikle de to applikasjonene som sammen skal utgjøre en applikasjon for håndholdt enhet og et webgrensesnitt. Status for dette er at etter avtale med oppdragsgiver og en del kommunikasjon frem og tilbake, valgt å ta i bruk Qt/C++ som programmeringsspråk for den håndholdte enheten, og PHP / MySQL på webgrensesnittet. Kommunikasjon mellom de to vil foregå gjennom HTTP protokollen (vha. REST-metodikk). Frem til nå har det gjennom den iterative og inkrementelle metoden blitt satt stor fokus på å få til et godt program- og arkitekturdesign, en god og gjennomtenkt grunnstruktur forhindrer mange problemer i etterkant. Det er også blitt satt fokus på å programmere testkode for separate funksjonalitetsprinsipper i forbindelse med designet, for å avdekke feil, og teste ut designkonsepter. Bl.a. en fullverdig klient / tjener applikasjon i Qt/C++, og flertrådsprogramlogikk for å teste synkroniseringsprinsipper. Noe som har til nå hatt avgjørende effekt på designvalg.

B.1.5 Rapportskriving

Det er ikke blitt utført noe direkte arbeid på rapporten utover disposisjonsforslag innad i gruppen, men vi har satt opp en egen weblog (fungerer som en blogg), hvor vi poster forskjellige prosjektspesifikke tanker, problemstillinger og diverse. F.eks. en egen post rundt forutsetninger (kravspesifikasjons revisjoner), og en rundt GUI-avgjørelser. Som nevnt under løsningsmetode og organisering, så har vi brukt mye tid på visualisering av forskjellige deler rundt applikasjonene, dette skaper mange grafiske modeller som vi akter å sentrere mye av rapporten rundt, da de er av sentral art for prosjektet. Det er derfor valgt en mer itererende metode rundt utformingen av rapporten, og satt av tid til intensiv skriving og utforming mot slutten av prosjektet.

B.2 Muligheter, trusler og farer

I prosjektet så har vi tatt på oss å bruke utviklingsverktøy vi har lite eller ingen kjennskap til, dette medfører alltid en litt større risiko for at vi i etterkant kan finne ut at dette har mangler vi ikke har kunnet forutset på forhånd. Men en mulighet er også at det gjør arbeidet lettere og har egenskaper som gjør at vi kan løse ting bedre og raskere enn antatt. Qt-rammeverket har vist seg å gi oss mer enn vi hadde trodd på forhånd, og har ført til et foreløpig bedre resultat enn antatt. Det gir også muligheter til å gi et bedre produkt til oppdragsgiver, og har gitt oss tid til å bruke tid på nevnt grunnstruktur, enn å lage "blind kode". Vår største utfordring ligger i synkroniseringsproblematikken, vha. mye tid og diskusjon rundt temaet har vi nå et visualisert løsningsutkast til dette, men implementasjonen kan forårsake problemer, og utfordringen ligger i å holde databasen fri for redundante og feile data. En trussel er da at vi feilberegner designet og kan komme til et punkt hvor vi må rekonstruere mye kode på bakgrunn av dette.

B.3 Avsluttet arbeid

Prosjektet arbeider etter en iterativ og inkrementell metode, dette gjør at selv om inkrement leveres til rett tid, så er det bare første utkast, med basic funksjonalitet. Dette kommer til å gradvis (iterativt) forbedres under hele den resterende prosessen. Per dags dato så har vi oppe fritekstsøk mot pubmed, gjennom et ferdiglagd interface (utfordringen lå i å modellere det til vår utviklingsmodell / språk). Vi har også kommunikasjon med et script på webgrensesnittet, som tar imot en xmlfeed. Dette aksesspunktet skal brukes til å overføre data som skal lagres sentralt på server (kliniske spørsmål med dens søkekomposisjon). Dette er et kritisk punkt og kommer til å iterativt forbedres mye gjennom den nærmeste tiden.

B.4 Gjenstående arbeid

Webgrensesnittet er ikke påbegynt, da det fokuseres på å utvikle funksjonalitet på den håndholdte enheten først, denne må opp før webgrensesnittets database og server kan benyttes for fullt. Databasedesignet avhenger også sterkt av hvordan den håndholdte applikasjonen fungerer. Det henvises her også til fremdriftsplanen på prosjektets hjemmeside, og første inkrement. I tillegg så er mesteparten av implementasjonen utover basic funksjonalitet gjenstående på håndholdt enhet, men grunnstrukturen rundt er ferdig implementert og gjennomarbeidet.

B.5 Tidsfrister

Jf. fremdriftsplan så ser det ut til at første inkrement overskrides med én uke, dette på bakgrunn av nevnt fokus på å få til en godt design og en god grunnstruktur. Det anses ikke som kritisk, da det i prosjektplanen er satt av tid i henhold til dette. Det ligger også et sikkerhetsnett i at de to siste inkrementene (ordlistefunksjonalitet og printfunksjonalitet) kan strykes og avhenger av at vi blir ferdig i god tid med den kritiske delen av prosjektet.

B.6 Motivasjon

Samarbeidet mellom prosjektets medlemmer har fungert over all forventning, begge er selvstendige og arbeider godt både sammen og på egenhånd og inkluderer den

andre parten i alt arbeid. Det har også oppstått mange diskusjoner og kritiske vurderinger av hverandres arbeid og meninger, særskilt i henhold til arbeidet med design og arkitektur. Dette har vært sunt på alle mulige måter og bidratt til økt motivasjon både for prosjektet og for den enkeltes arbeid. Det er veldig oppmuntrende å delta i en prosjektgruppe hvor alle tar ansvar for å bære prosjektet frem. Arbeidsformen har på bakgrunn av dette variert mellom individuelt arbeid hver for seg (med jevnlig telefonkontakt) og arbeid på “kontoret” (A018).

B.7 Veileder

På bakgrunn av at prosjektets teknologi for den håndholdte enheten ikke er veileders spesialfelt, så har det vært lite kontakt utover fastsatte statusmøter (hver 14.dag). Dette vil nok kunne øke når arbeidet på webgrensesnittet tar fatt, da dette ligger nærmere veileders kompetanseområdet. Det har også vært lite håndfast å få vurdering på, grunnet mye fokus på utforming av programdesign og innlæring av ny teknologi. Selve møtene har vært oppfattet som litt ensidige og gruppen savner mer kritiske meninger rundt prosjektet, men tar også selvkritikk for å ha kommet med lite konkret informasjon. Det har imidlertid vært lite kritiske situasjoner som gruppen ikke har følt de ikke har kunnet løse på egenhånd, mye på bakgrunn av meget god tone og godt samarbeid mellom gruppens medlemmer.

Tillegg C

Møtereferat og timelogg

C.0.1 Møtereferat

Vi har i løpet av prosjektperioden hatt en rekke møter med både veileder og oppdragsgiver. Det meste av kommunikasjon har foregått uformelt per telefon og mail, men i starten hadde vi noen formelle møter. Her vises et eksempel på innkalling og referat fra et slikt møte.

Innkalling

SAKSLISTE: PICObase - Statusmøte med veileder: #2-2009.

Sted: Utenfor kråkereirot, K-bygget på HiG.

Dato: Mandag 09.02.2009

Tid: Kl. 1430.

SAKSLISTE:

1. Forprosjektet
2. Utviklingsmiljø (update)
3. Kravspesifikasjon: Use Case + Domenemodell (Se vedlegg)
4. Design og arkitektur (Se Vedlegg)
5. Statusrapport

6. Fremdriftsplan

7. Eventuelt

Møtereferat

Møtereferat: PICObase - Statusmøte med veileder: #1-2009.

Sted: Utenfor kråkereirot, K-bygget på HiG.

Dato: Mandag 22.01.2009

Tid: Kl. 1415 - 1500.

Tilstede: Øivind Kolloen (prosjektveileder), Arild Plassen, Øyvind Skattum Vesteng.

SAKSLISTE:

1. Utviklingsmiljø
2. Oppgavens status og moduler
3. Hyppighet av møter med veileder
4. Systemmodell

Referat:

1. Tre muligheter vi har sett på, native (Windows mobile 5 SDK), Qt og Java. Java følte vi var utelukket grunnet vanskeligheter med JVM i forhold til embedded programmering o.l. Veileder var helt enig og sa at det burde utelukkes da "write once, run anywhere"ideologien ikke fungerer i praksis med Java, og ville ført til at vi måtte ha reprogrammert alt for hver enhet.

Qt var det på møtets tidspunkt usikkerhet rundt i forhold til om oppdragsgiver er villig til å betale for lisensiering, en open source lisens vil sette føringer i etterkant i forhold til distribusjon på bakgrunn av GPL sitt krav om åpen og tilgjengelig kildekode.

Native begrenser seg til kun den ene plattformen, og til dels bare én versjon av den, og er sånn sett et sistealternativ, dette var veileder enig i.

Veileder kom da med forslag om en fullstendig nettbasert løsning, som da kan løse problematikken i forhold til portabilitet ved å ta i bruk ajax-teknologi. Endel mulige problemer og fordeler ved dette ble diskutert som.

+ Lik kode på webgrensesnittet som for mobile enheter

- Mindre batterkapasitet grunnet må være på nett, evt. mulighet for frakoblet modus.

-

Vi ble enige om å se nærmere på dette etter møtet.

2. Status ved møtets avholdelse var at det forelå mye usikkerhet rundt kunnskapsbasert praksis og applikasjonen generelt, både funksjonelle / operasjonelle krav og miljøet rundt. Veileder var enig og antydte at vi skulle ta kontakt så fort som mulig, noe vi allerede hadde gjort og ventet svar på.

Angående forprosjektrapport og modulinndeling, så foreslo veileder at vi skulle overgeneralisere og sette opp veldig vide moduler som f.eks. Gui og Søk, og så ta det derfra gjennom kravspesifikasjon og design, når vi vet mer.

3. Møter med veileder ble avtalt til annenhver mandag, klokka 1500. Eventuelt materiale som veileder skal se på før møtet, må være han i hende innen klokka 1000 på møtets dag. Hastespørsmål o.l. taes fortløpende.

4. Veileder mener Scrum er en god systemmodell å bruke, noe vi er enig i, men vi skulle se på inkrementell og evolusjonær også, evt. kombinere deler av de to.

Neste møte: mandag 1.februar.

Timelogg

Timelogg vedlegges på den medfølgende CD'en i sin helhet.

Tillegg D

XML-Strukturer

D.1 Lagringsformat for kliniske spørsmål

D.1.1 Håndholdt enhet

XML-format

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <clinicalQuestion>
3   <cqid></cqid>
4   <title></title>
5   <description></description>
6   <searchCompositions>
7     <searchComposition>
8       <type></type>
9       <scid></scid>
10      <displayString></displayString>
11      <cqid></cqid>
12      <resultCount></resultCount>
13      <field>
14        <name></name>
15        <value></value>
16      </field>
17    </searchComposition>
18  </searchCompositions>
19 </clinicalQuestion>
```

Listing D.1: XML-format på lagret klinisk spørsmål

Valideringsskjema

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
3   <xs:simpleType name="scidType">
4     <xs:restriction base="xs:integer">
5       <xs:pattern value="(\d+)?"/>
6     </xs:restriction>
7   </xs:simpleType>
8   <xs:complexType name="fieldType">
9     <xs:sequence>
10      <xs:element name="name" type="xs:string"/>
11      <xs:element name="value" type="xs:string"/>
12    </xs:sequence>
13  </xs:complexType>
14  <xs:complexType name="searchCompositionType">
15    <xs:sequence>
16      <xs:element name="type" type="xs:string"/>
17      <xs:element name="scid" type="scidType"/>
18      <xs:element name="cqid" type="xs:integer"/>
19      <xs:element name="displayString" type="xs:string"/>
20      <xs:element name="resultCount" type="xs:integer"/>
21      <xs:element name="field" type="fieldType" maxOccurs="unbounded"/>
22    </xs:sequence>
23  </xs:complexType>
24  <xs:complexType name="searchCompositionsType">
25    <xs:sequence>
26      <xs:element name="searchComposition"
27        type="searchCompositionType" minOccurs="0"
28        maxOccurs="unbounded"/>
29    </xs:sequence>
30  </xs:complexType>
31  <xs:complexType name="clinicalQuestionType">
32    <xs:sequence>
33      <xs:element name="cqid" type="xs:integer"/>
34      <xs:element name="uid" type="xs:integer"/>
35      <xs:element name="title" type="xs:string"/>
36      <xs:element name="description" type="xs:string"/>
37      <xs:element name="dateCreated" type="xs:string"/>
38      <xs:element name="searchCompositions"
39        type="searchCompositionsType"/>
40    </xs:sequence>
41  </xs:complexType>
42  <xs:element name="clinicalQuestion" type="clinicalQuestionType"/>
43 </xs:schema>
```

Listing D.2: Valideringsskjema for lagret klinisk spørsmål

Eksempel i bruk

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <clinicalQuestion>
3   <cqid>0</cqid>
4   <title>Warfarin vs aspirin hos eldre med Af</title>
5   <description>Effekt med hensyn til cerebrale
6     insulter.</description>
7   <searchCompositions>
8     <searchComposition>
9       <type>PICO</type>
10      <scid>0</scid>
11      <displayString>nr1</displayString>
12      <cqid>0</cqid>
13      <resultCount>0</resultCount>
14      <field>
15        <name>patient</name>
16        <value>elderly</value>
17      </field>
18      <field>
19        <name>intervention</name>
20        <value>warfarin</value>
21      </field>
22      <field>
23        <name>control</name>
24        <value>placebo</value>
25      </field>
26      <field>
27        <name>outcome</name>
28        <value>sideeffects</value>
29      </field>
30    </searchComposition>
31    <searchComposition>
32      <type>PICO</type>
33      <scid>0</scid>
34      <displayString>nr2</displayString>
35      <cqid>0</cqid>
36      <resultCount>0</resultCount>
37      <field>
38        <name>patient</name>
39        <value>elderly</value>
40      </field>
41      <field>
42        <name>intervention</name>
43        <value>warfarin</value>
44      </field>
45      <field>
46        <name>control</name>
47        <value>aspirin</value>
```

```
47     </field>
48     <field>
49         <name>outcome</name>
50         <value>sideeffects, death</value>
51     </field>
52 </searchComposition>
53 <searchComposition>
54     <type>FreeText</type>
55     <scid>0</scid>
56     <displayString>nr1</displayString>
57     <cqid>0</cqid>
58     <resultCount>0</resultCount>
59     <field>
60         <name>querystring</name>
61         <value>elderly sideeffects warfarin placebo</value>
62     </field>
63 </searchComposition>
64 <searchComposition>
65     <type>FreeText</type>
66     <scid>0</scid>
67     <displayString>nr2</displayString>
68     <cqid>0</cqid>
69     <resultCount>0</resultCount>
70     <field>
71         <name>querystring</name>
72         <value>elderly death, sideeffects aspirin
73             placebo</value>
74     </field>
75 </searchComposition>
76 <searchComposition>
77     <type>FreeText</type>
78     <scid>0</scid>
79     <displayString>nr3</displayString>
80     <cqid>0</cqid>
81     <resultCount>0</resultCount>
82     <field>
83         <name>querystring</name>
84         <value>elderly death warfarin aspirin</value>
85     </field>
86 </searchComposition>
87 </searchCompositions>
</clinicalQuestion>
```

Listing D.3: Eksempel på lagret klinisk spørsmål

D.2 Format for indeksering av kliniske spørsmål

D.2.1 Håndholdt enhet

XML-format

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <drafts>
3   <draft>
4     <title></title>
5     <path></path>
6   </draft>
7 </drafts>
```

Listing D.4: XML-format for indeksering av kliniske spørsmål på håndholdt

Eksempel i bruk

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <drafts>
3   <draft>
4     <title>Warfarin vs aspirin hos eldre med Af</title>
5     <path>cq.xml</path>
6   </draft>
7 </drafts>
```

Listing D.5: Eksempel på indekseringsfil i bruk

D.2.2 PICObaseWeb

XML-format

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <clinicalquestionList>
3   <clinicalQuestion>
4     <cqid></cqid>
5     <uid></uid>
6     <title></title>
7     <description></description>
8     <date_created></date_created>
9   </clinicalQuestion>
10 </clinicalquestionList>
```

Listing D.6: XML-format for indeksliste av kliniske spørsmål fra web.

Eksempel i bruk

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <clinicalquestionList>
3   <clinicalQuestion>
4     <cqid>22</cqid>
5     <uid>1</uid>
6     <title>My only, single clinical question</title>
7     <description>With a single description</description>
8     <date_created>2009-04-19 03:11:37</date_created>
9   </clinicalQuestion>
10  <clinicalQuestion>
11    <cqid>24</cqid>
12    <uid>1</uid>
13    <title>Poststep. glomerulonefritt</title>
14    <description>Effekt av penicillin hos asymptotiske barn med
15      poststrep.glomerulonefritt</description>
16    <date_created>2009-05-01 17:24:54</date_created>
17  </clinicalQuestion>
18  <clinicalQuestion>
19    <cqid>25</cqid>
20    <uid>1</uid>
21    <title>Warfarin vs aspirin hos eldre med AF</title>
22    <description>Effekt med hensyn til cerebrale
23      insulter.</description>
24    <date_created>2009-05-01 17:56:09</date_created>
25  </clinicalQuestion>
26  <clinicalQuestion>
27    <cqid>26</cqid>
```

```
26 <uid>1</uid>
27 <title>Malign acites</title>
28 <description>hva er effekten av tapping?</description>
29 <date_created>2009-05-08 11:08:12</date_created>
30 </clinicalQuestion>
31 </clinicalquestionList>
```

Listing D.7: Eksempel på indekseringsfil fra web i bruk.

Tillegg E

CD'ens innhold

Oversikt over den medfølgende CD'ens innhold og katalogstruktur finnes i filen `index.pdf` på cd'ens rot.