

Kravspesifikasjoner Prosjektoppgave i Programutvikling våren 2014

Studentnummer:

s198570 Henrik Fischer Bjelland

s198610 Peter Wilhelmsen

s180167 Haakon Winther

Alle går Informasjonsteknologi, 1.klasse

Funksjonalitet i programmet

Programmet vårt består i hovedsak av tre deler: Registrering, uthenting av informasjon og statistikk.

Under **REGISTRERING** skal det være mulig å gjøre følgende:

- Registrere leger. Herunder legens navn, arbeidssted, hvilke reseptgruppe legen kan skrive ut medisiner fra og endre legens bevilling.

- Registrere Pasienter. Herunder navn og fødselsdato. Pasienten får ved opprettelese en tom reseptliste.

- Registrere Resepter. Herunder dato(den dagen resepten er skrevet), data om pasienten resepten blir skrevet ut til, data om legen som skriver ut resepten, hvilken medisin resepten er på, mengden av medisin, hvilken kategori medisinen tilhører og legens anvisning om bruk av medisinen.

Under **INFORMASJON** skal det være mulig å gjøre følgende:

- Ut fra en leges navn skal det være mulig å skrive ut oversikt over resepter denne legen har skrevet ut og all data lagret om legen.

- Vise en liste over godkjente leger og en liste over hvilke leger som har bevilling innen en bestemt reseptgruppe.

- Skrive ut en liste over alle pasienter og leger som er tilknyttet en bestemt medisin ved hjelp av medisinen ATC-nr¹.

- Ut fra en pasients navn og fødselsdato skal man få en oversikt over alle resepter som er skrevet ut på denne pasienten.

Under **STATISTIKK** skal det være mulig å gjøre følgende:

- Få en grafisk fremstilling av medisinforbruket til en eller flere spesifikke medisiner i løpet av et år.

Alle data skal lagres på fil ved programslett og kunne leses fra fil når programmet startes opp.

- Hvis vi får tid ønsker vi å implementere flere av ønskene til arbeidsgiver

Prioriteringsrekkefølge:

- Vi vil prioritere å lage en GUI med felt og knapper til alle funksjonene vi ønsker at programmet skal inneholde, legge til en og en metode for å hele tiden kunne ha et fungerende program.

Programmet skal fungere mest mulig etter kravene til oppdragsgiver. Vi vil ha hovedfokus lage metoder for å:

- Registrer lege, registrer pasient, registrer resept.

- Vise info om: lege, pasient og resept.

- Så fort en enkel GUI og noen av metodene fungerer vil vi utvide GUI og funksjonalitet i programmet.

¹ anatomisk terapeutisk kjemisk register

Hva må gjøres?	Hva?	Husk
Programkode	Hele koden	Minnepenn, .Jar
Produktdokumentasjon	Hvordan det er bygget opp	Beskrive funksjonaliteten
Prosessdokumentasjon	Hvem har gjort hva?	Valg og hvorfor?
Bruerveiledning	Hvordan funker det	Er for en bruker, ikke koder
Testrapport	Kvalitetsikring	hvilke feil, hensyn osv

Brukergrensesnittet

Vi vet at vi kommer til å endre mye på brukergrensesnittet ettersom vi skal lære mer om dette i forelesningene framover. Under er enkle skisser over hvordan vi kan tenke oss at programmet vårt skal se ut.

Hovedvindu:

INFORMASJON	REGISTRERING
LEGE	LEGE
PASIENT	PASIENT
MEDISIN	RESEPT
RESEPT	

Registreringsvinduer:

REGISTRERING PASIENT	
NAVN	
FØDSELSDATO	

REGISTRERING LEGE

NAVN

ARBEIDSTED

RESEPT-BEVILLING

A

B

C

REGISTRERING RESEPT

DATO

PASIENT

LEGE

MEDISIN

MENGDE

KATEGORI

ANVISNING

Oppbygging av programmet

Vi har valgt å dele programmet opp i 2 deler. En del for registrering og en del for informasjon. Ved å trykke på de forskjellige knappene vil man bli sendt videre inn i programmet. For eksempel ved å registrere en lege velger man "registrer lege", så blir man sendt videre til en ny side der man fyller inn info om legen som skal registreres. Dette vil fungere bra da alle oppgavene til programmet er stort sett like. Det er ikke stor forskjell på sjekk lege/sjekk pasient/sjekk resept. Dette vil også se bedre ut enn å ha masse info på en side som tar opp masse plass på brukeren sin skjerm.

Klasser

Vi ønsker å starte med klassene oppdelt etter lege/pasient/resept, vi vil videre ha egne klasser for funksjonene til klassene. En egen for legefunksjoner osv. Dette vil medføre en enklere kode for en utenforstående i tillegg til at det ikke blir veldig lange klasser. Vi må også dele opp GUI i egne klasser da det kommer til å bli litt forskjellige vinduer.

Oppgaveteksten stiller krav om å bruke subklassen, vi ønsker følgende klassehierarki:

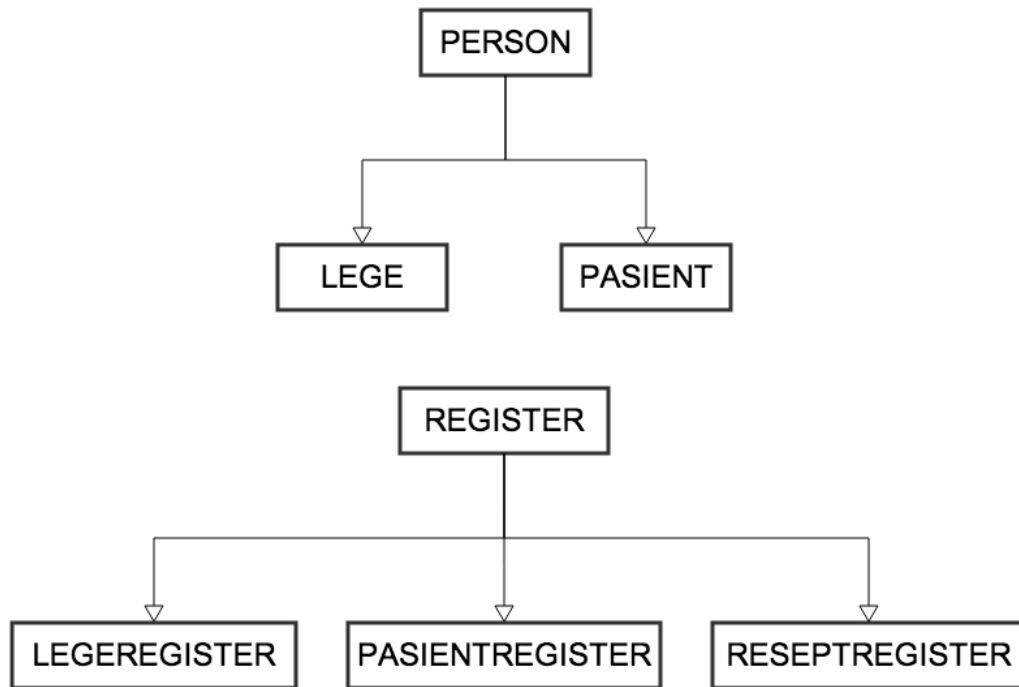
Superklasse av **PERSON**, subklasse **LEGE** og **PASIENT**.

Superklasse av **REGISTER**, subklasse **LEGEREGISTER**, **PASIENTREGISTER** og **RESEPTREGISTER**.

Vi har foreløpig ikke sett behovet for flere super-subklasser under planleggingsfasen. Forslag til noen av klassene:

Person, Lege, Pasient, Resept, Medisin, Legeliste, Pasientliste, Reseptliste, Medisinliste, Register, Legeregister, Pasientregister, Reseptregister, MainGUI, InfoGUI, StatistikkGUI, LegeSystem, PasientSystem, ReseptSystem.

Beskrivelse av klassehierarkiet



Datastruktur

Vi ønsker å bruke javas Collection-rammeverk. Vi vil vurdere hvilke av Collections-klassene vi kommer til å bruke utifra hvilke oppgaver vi skal gjennomføre. Vi har ikke bestemt oss for nøyaktig hvilke datastrukturer vi ønsker å bruke derfor har vi ingen figur av dette.

Utviklingsverktøy

Fram til nå har vi jobbet med javakoding i Sublime Text 2, men vil nå gå over til å programmere i nyeste versjon av Netbeans: NetBeans IDE 8.0. Vi har valgt NetBeans for å ha mulighet til å debugge.

Github er et flott versjonskontrollsystem for å samarbeide om kode. Her er det mulig å splitte og sammenslå filer som flere har jobbet med samtidig. Alle på gruppen ble enige om at å bruke dette.

Vi bruker Trello, som er et prosjekthåndteringsverktøy, for å holde oversikt over hvilke oppgaver vi har foran oss og hvem som jobber med de forskjellige oppgavene til en hver tid.

Systemkrav²

Vi ønsker å benytte oss av java-versjon 7. For å kunne bruke denne versjonen stiller vi følgende systemkrav til maskinen:

Windows

- Windows 8 (Desktop)
- Windows 7
- Windows Vista SP2
- Windows XP SP3 (32-bit); Windows XP SP2 (64-bit)
- Windows Server 2008
- Windows Server 2012 (64-bit)
- RAM: 128 MB
- Disk space: 124 MB

Mac OS X

- Intel-based Mac med Mac OS X 10.7.3 (Lion) or newer
- Administrator privileges for installation
- 64-bit browser

Linux

- Oracle Linux 5.5+
- Oracle Linux 6.x (32-bit)*, 6.x (64-bit)**
- Red Hat Enterprise Linux 5.5+, 6.x (32-bit)*, 6.x (64-bit)**
- Ubuntu Linux* 10.04 and above
- Suse Linux Enterprise Server* 10 SP2, 11.x
- RAM: 64 MB
- Disk space: 58 MB
- Browsers: All OS that support versions Firefox 3.6 and above

²Hentet fra <http://www.java.com/en/download/help/sysreq.xml>

Java-versjon

Vi har valgt å bruke java-versjon 7.0 fordi vi ønsker å benytte oss av Collections-rammeverket og generiske datatyper.

Planlagt arbeidsfordeling.

Vi ønsker å ha følgende ansvarfordeling:

Peter:

Produktdokumentasjon

Funksjonskode

Haakon:

GUI/design

Prosessdokumentasjon

Henrik:

GitHub ansvarlig

Funksjonskode

Dette er ment som en veiledende arbeidsfordeling, det er ikke ment at en person skal ha alt ansvaret for en ting.

Fremdriftsplan med milepæler

Mars

Uke 12 17 - 23.mars

Velge oppgave, starte på kravspec. Bruke god tid på milepælsplanen.

Uke 13 24 - 30.mars

Starte å tegne opp GUI-kladd, slik at vi kan starte med skjelettet senere.

Klasseoppsett

Fikse videre på kravspecen

30. Mars - Vi leverer inn **Kravspec'en**

April

Uke 14 31- 6.mars/april

1.April - Starter med koding, starte på skjelettet til oppgaven.

2 April - **Kravspec'en** MÅ være levert inn.

4-6 April - Hyttetur, sosial workshop

Uke 15 7 -13.april

Skjelett GUI, de fleste metoder skal være på plass, men uten innhold.

Bygget opp et skjelett som begynner å bli klar til koding av metoder.

Starte på programmering av metoder back-end.

Uke 16 14 - 20.april

Programmering av metoder back-end og GUI.

Dokumentere underveis.

Uke 17 21 - 27.april

Programmering av metoder back-end og GUI.

Dokumentere underveis.

Mai

Uke 18 28 - 4.april/mai

Vi ønsker å få programmet vårt ferdig denne uken.

Jobbing med dokumentasjon.

Uke 19 5 - 11. mai

Gjennomgang av alle kommentarer i programmet.

Jobbing med dokumentasjon.

Uke 20 12 - 16. mai

Jobbing med dokumentasjon.

Lage en .jar fil av programmet vårt.

16. Innlevering