

# Prosjektrapport

## Reseptregistret

Her kan du finne og vise ulik info om resepter, leger og pasienter. Under søk etter kan du velge om du vil søke etter resept, lege eller pasient. Du bestemmer søkekriteriene ved å fylle ut feltene under, tomme felter blir ikke tatt med i søket

Søk etter:

☒ Resepter

☐ Pasienter

☐ Doktorer

Pasient navn/personnummer:

Doktor navn/personnummer:

Medikament navn:

Medikament kategori:

Forskrevet medikament i gruppe:

☒ Gruppe A ☒ Gruppe B ☒ Gruppe C

Søk

Legg til en ny doktor

Vis statistikk

Reseptnummer	Skrevet ut	Hentet	Pasient	Lege	Medisin navn	Medisin kategori	Medisin gruppe
--------------	------------	--------	---------	------	--------------	------------------	----------------

Tilbake

Logg ut

Vis resept

Fram

Sondre Husevold (s198755) og Magnus Tønsager (s198761)  
Mai 2014

## Oppstart, planlegging og valg av oppgave

En tomanns gruppe var valgt med stor hensikt lenge før oppgaven var delt ut, vi følte dette var det perfekte antallet personer å ha i en gruppe uansett hva slags oppgave vi kom til å få. Det var enighet mellom fem personer at vi kom til å dele oss i grupper på to personer og vi delte opp gruppen i hvem som kom til å jobbe best sammen med tanke på arbeidsmetoder og ferdigheter.

Det var en del planlegging som allerede hadde blitt gjort før oppgaven ble gitt slik som mye av strukturen og arbeidsfordelingen. Begge to hadde stor arbeidsvilje for å lage et ordentlig program og mange ideer fløt rundt blant annet om struktur av brukergrensesnittet og andre nyttige funksjoner som var generelt for alle programmer.

Valget av oppgave var enstemmig og reflekterte hva vi tenkte var oppnåelig og som det var mulig å bygge mange ekstra funksjoner rundt som ikke kun var gitt i oppgaven. Det var også et pluss at Magnus hadde leger i familien og kunne dermed få input fra de som faktisk hadde brukt programmet.

Dyp planlegging og kravspesifikasjon ble laget mandag etter oppgaven ble lagt ut og ferdiggjort på dagen. Programmet ble påbegynt en uke senere.

## Utviklerverktøy og utviklermiljø

Under kravspesifikasjonen ble vi enige om å hovedsakelig bruke Eclipse som utviklerverktøy og JDK8 som basis. Dette ble endret til JDK7 etter litt problemer. Sondre brukte alltid Eclipse, mens Magnus byttet på mellom TextPad og Eclipse som utviklerverktøy.

Det ble ikke brukt versjonshåndtering under prosjektet og prosjektet ble påbygd ved bruk av Dropbox og versjonsmapper. Dette er noe vi angreir på og vil heller bruke Git i senere prosjekter.

Programmet ble for det meste bygget individuelt og mesteparten av kommunikasjonen skjedde over Facebook og eventuelle timer på skolen. Programmeringen skjedde for det meste hjemme med møter på høyskolen en gang i blant.

Programmet ble utviklet på to forskjellige operativsystemer, hvor Sondre brukte hovedsaklig OS X 10.9.2 og Magnus brukte Windows 8.

## Arbeidsfordeling

Fordelingen var gjort slik at begge skulle ta litt med lister, litt brukergrensesnitt og litt andre parter av programmet. Vi prøvde å dele opp oppgaven slik at vi fikk gjort omtrent like mye og vi er svært fornøyde med resultatet, da begge to har gjort nesten identisk mye arbeid.

Magnus tok hånd om resepter, hovedregistret og brukergrensesnitt, mens Sondre tok hånd om personer, lagring og brukergrensesnitt. Grunnen til at begge fikk gjøre brukergrensesnitt er fordi dette var noe begge ville være en del av. Det var også en av de større delene av programmet.

Mot slutten av prosjektet fikk Magnus oppgaven å sette sammen programmet og se etter eventuelle bugs når programmet kom sammen til en entitet. Sondre skrev dokumentasjon, rapport og andre ting imens dette foregikk. Vi testet også alle våre egenproduserte klasser for oss selv for å forsikre oss om at det ble minst mulig bugs når programmet ble satt sammen.

## Datastruktur og andre hierarkier

Vi ble anbefalt sterkt av faglærer å bruke TreeSet for personer og lister generelt for å få struktur og orden på listene. Dette ble tatt hensyn til og vi brukte dermed TreeSet på alle lister vi hadde, inkludert referanselister. Vi fant også ut at TreeSet var en av de mer effektive listene å søke i og dette var også en av grunnene til at vi valgte denne listetypen.

Å ha en felles registerklasse var valgt tidlig i prosjektet og ble startet på umiddelbart av Magnus etter listene var kommet opp. Denne var brukt til å lagre Sondres subregister som doktorregistret og pasientregisteret. Dette gjorde det lettere å ha mye av koden på en plass ettersom mye av brukergrensesnittet bruker de samme metodene hele tiden. Dermed synes vi at det var best å sentralisere mye av mekanikken.

## Litt om lagring

Det ble valgt at vi lagret hovedregister som et objekt istedet for å ha det som en tekstfil. Dette var gjort fordi det var lettere å lese filen og at ikke all informasjon sto i plain tekst. Det er mer krevende for programmet å opprette et helt nytt register enn å lese det, noe som også var en grunn til å la hele hovedregisteret bli lest opp.

Originalt hadde vi tre forskjellige filer som skulle lagres i data mappen fordi vi følte det var mer naturlig å separere dem. Dette skulle være legene, pasientene og reseptene da disse var de som skulle lagres i programmet. Dermed kunne man slette eventuell korrupt data uten å slette alt og alle. Dette ble omgjort til hovedregisteret fordi det oppsto store feil ved endring av person objekter og reseptene.

## Innlogging

Dette var et valg vi tok tidlig i prosessen hvor vi ville dele hele brukergrensesnittet opp i tre deler. En for administrator, en for leger og en for apotek. Dermed synes vi innlogging var den beste måten å dele dette på.

Vi ville at passordene skulle være kryptert på et eller annet vis, begge to prøvde seg på dette, men det var en vanskelig prosess og vi ble til slutt enig om obfuskering istedet.

## Brukergrensesnitt

I alle brukergrensesnittene blir det brukte TModel, en egen tabellklasse som ble laget for mer tilpasning hvordan vi selv ville at tabellene skulle være. Dette ga oss også muligheten til å gjøre alle cellene i tabellen umulig å endre som blir brukt i personsøk brukergrensesnittet. Grunnen til å ha dette i personsøk er fordi det ga en mer oversiktlig og brukervennlig opplevelse for legene og det var lettere å programmere. Administrator har muligheten til å endre i cellene for å gjøre programmet litt mer avansert, men med mange funksjoner for administrator. Spesielt med tanke på administrator skal endre mange personer på en gang, mens legene skal kun tenke på en pasient om gangen.

Vi valgte å ha frem og tilbake knapper i programmet for å gjøre programmet mer brukervennlig istedet for å ha faner. Vi ville at hele programmet skulle være fokusert på akkurat det panelet man er på, og for å gå frem og tilbake følte vi det var bra å ha en frem og tilbake funksjon som lagret hvordan brukergrensesnittet så et par ganger tilbake slik at det er lett å gå frem og tilbake uten å miste informasjon. Ved å trykkelogg ut tømmer den all informasjon om frem og tilbake vinduene slik at man ikke kan bruke dette til å komme igjennom innlogging.