

# HiOA

TDK

Ingeniørfag – data

DATS1600 – Programutvikling

Eva Hadler Vihovde

Prosjektoppgaven 2015

- Prosessdokumentasjon -

Alternativ 1

- Forsikring -

Gruppe #14

Studentnavn	Studentnummer
Marius Alexander Skjolden	s114143
Hans Christian Nenseth	s236334
Hans Petter Osvold	s929913

# Innholdsfortegnelse

Introduksjon	2
Kritiske valg	3
Arbeidsmetode	4
Konkrete ansvarsområder	4
Verktøy	5
Konklusjon	5

## Introduksjon

Dette dokumentet er produsert i forbindelse med en prosjektoppgave i kurset DATS1600 ved Høgskolen i Oslo og Akershus. Dokumentet skal gi et inntrykk av hvordan vi har arbeidet med utviklingen av applikasjonen. Vi kunne velge mellom tre ulike oppgaver, og vi ble raskt enige om å utvikle alternativ 1: Forsikring.

# Kritiske valg

## Programmet hovedoppgave

Vi valgte som nevnt tidligere å utvikle et forsikrings-program. Det største valget vi gjorde for å avgrense oppgaven var å bestemme at dette programmet kun er et utsnitt av en større system, hvor fokuset er på forsikringsagenten. Dermed har vi ikke utviklet en kundeportal eller et fler-bruker program.

Dette har vi valgt på grunn av oppgavens lengde og krav til bruk av teknologier. Vi hadde i underkant av seks uker som skulle inkludere planlegging, implementasjon, testing og dokumentasjon. Samtidig så var hovedkravet til oppgaven at programmet ble utviklet som en Java desktopp applikasjon med Java sine standard klassebiblioteker.

Vi føler at hvis dette skulle vært en reel applikasjon så ville det vært naturlig at kundeportalen ble utviklet som et nettsted i for eksempel Java Spring eller Java Play, eller en serie av mobilapplikasjoner.

Vi ønsker i tillegg å påpeke at dette er et første års prosjekt og at vi ikke har lært noe om datasikkerhet enda. Derfor ville det vært ugunstig å implementere en fler-bruker applikasjon med tanke på hvordan vi skulle håndtere sensitiv informasjon som passord på en god måte.

Fokuset vårt har vært på forsikringsagenten og vi tar det for gitt at eventuelle kundeportaler og innlogging-systemer vil være på plass. Derfor har vi satt fokuset på implementasjon av en agnostisk datastruktur, adaptere, fulltekst søk, og statistikk visning. Legg spesielt merke til bruken av Lambda til å filtrere søk og hvordan man enkelt kan ta med seg søkeresultatene og vise statistikk på nettopp dette settet. Utover dette har benyttet kjente design patterns der det har vært hensiktsmessig. Vi føler derfor at resultatet gir oss et modulært program som til en viss grad vil skalere godt med tanke utvidelser av modell eller et større antall objekter.

## Java FX

Samtlige deltagere hadde et ønske om å utvikle en Java FX applikasjon. Dette hadde både noe fordeler og ulemper. En av fordelene er at Java FX vil være et mer fremtidsrettet valg. Derfor var det kjekt å kunne lære denne teknologien. En annen fordel er at Java FX er mere objektorientert av natur, enn for eksempel Swing. En av ulempene er at dette er en teknologi under utvikling, og derfor er det en del sentrale komponenter som ikke er helt standard enda, for eksempel Dialog og Widgetbokser. Derfor har vi valgt å ikke implementere disse komponentene i vår applikasjon.

# Arbeidsmetode

I de første 2 / 3 delene av prosjektet har vi møtes omtrentlig fem dager i uka, hvor vi har hatt et grupperom fra 08.00 til 13.00.

Under implementasjonen av datastruktur testet vi ut en utviklingsmetode som kalles for «ekstrem programmering» som i korte trekk går ut på at vi hadde med skjerm og tastatur og satt alle tre å skrev kode på samme datamaskin. Dette funket bra, men tok også mye tid så vi valgte derfor å ikke gå videre med denne typen utvikling.

Vi fortsatte i alle fall å sitte sammen fem timer om dagen slik at vi kunne samarbeide godt, og mer eller mindre skrive koden i fellesskap.

Mot slutten av prosjektet har vi fordelt oppgavene slik at testing, rapportskriving og siste fin justering av programmet ble gjort samtidig.

## Konkrete ansvarsområder

Vi har jobbet mye med de samme filene men for å gi noe mer konkret svar på hvilke klasser som er utviklet av hvem så kan dette være til hjelp:

Marius Alexander Skjolden har fungert som en senior utvikler, og hatt av ansvaret for utviklingen av programmets arkitektur.

Hans Christian Nenseth har tatt hånd om det grafiske brukergrensesnittet.

Hans Petter Osvold har tatt hånd om oppgaver som faller litt i mellom disse to ytterpunktene.

Vi vil enda en gang gjenta at dette prosjektet er en gruppeprestasjon, og vi opplever at vi har oppnådd en grei fordeling av arbeidsoppgaver. Denne konkrete listen er kun en oversikt over hvem som har hatt overordnet ansvar for de ulike delene av programmet. Samtlige medlemmer har bidratt med kode i hele systemet.

## Verktøy

Vi har brukt en rekke utviklingsverktøy for å forenkle utviklingen av programmet. Samtlige deltagere har brukt IntelliJ Idea, et integrert utviklingsmiljø.

Vi hadde ambisjoner om å bruke et nettbasert gjøremål og milepæl system som heter Redmine, men det ble for mye styr og vi endte opp med å skrive gjøremål inn i en tekstfil som ligger i versjon kontroll området vårt.

Utover dette brukte vi versjons kontroll til å samarbeide. Dette muliggjorde det å kunne jobbe mer selvstendig hjemme. En effekt av nettopp denne bruken av versjons kontroll er at det ikke er mulig på en ærlig og rettferdig måte å si at prosjekt deltager x har forfatter klasse y. Grunnen til dette er at flere medlemmer har jobbet med ulike versjoner av samme fil.

Vi har gjennom hele implementasjonsfasen brukt unit tester for å være sikre på integriteten til program logikken. Rammeverket vi har brukt til testing er JUnit med Hamcrest. En snedig fordel av å kjøre disse testene er at de har generert en tilfeldig datafil for oss.

## Konklusjon

Vi gikk inn i prosjektet med ulik kunnskapsnivå, det var med andre ord grobunn for både konflikter og frustrasjon. Men prosjektet har gått bra, og vi er alle enige om at dette har vært en veldig lærerik prosess. Både med tanke på programmering og mellommenneskelige kunnskaper. Med andre ord har dette prosjektet vært en stor suksess.