Plan - Kravspesifikasjon

**Oppgave: Forsikring**

**Gruppe: Awesomenauts**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Navn | Student nr. | Linje | Klasse | Tlf. nr. |
| Aleksander Haugsdal | S929681 | Dataingeniør | HINGDATA14HA | 99536664 |
| Steinar Øxseth | S236611 | Informasjonsteknologi | INFORMATIK14HA | 99539678 |
| Stian Hvidsten | S236619 | Informasjonsteknologi | INFORMATIK14HA | 90274285 |

Vedlegg: UML av klassehierarki (UML.zip)

**Spesifikasjoner**

Utviklingsverktøy som blir brukt er IntelliJ IDEA 14.1.1. GitHub blir brukt til å opprette prosjektet og samkjøre programkode via SourceTree. Arbeidsoppgaver som skal utføres blir opprettet og holdt orden på i waffle.io. Kommunikasjon går via facebook.com og programmet Slack.

Vi bruker JavaFX 8 som har java 8 update 40 som krav .

**Programmets funksjon:**

Programmet skal inneholde forsikringer, utbetalingslogikk, logikk for forsikringspremie og evt. bonus, statistikk for firmaet, register for kunder, ansatte og forsikringssaker.

Programmet skal registrere kunder og kunder skal kunne tegne nye forsikringer. Registrering av kundeforhold og forsikring skjer enten via en kundebehandler eller via brukergrensesnittet. Det kan registreres forsikringssaker med informasjon om hendelse, kunden, bilder og evt. vitner via kundebehandler eller brukergrensesnittet. Etter endt saksbehandling skal evt. utbetaling av erstatning bli utregnet og betalt ut.

**Brukerne av programmet**

Forsikringstakere og ansatte.

I utgangspunktet er det bare privatkunder og kundebehandlere som bruker programmet. Hvis vi ser at vi har god tid, vil vi etterhvert prøve å utvide programmet med bedriftskunder, selskapets direktør(er) og evt. avdelingsledere.

**Datastruktur:**

Ansattregister: Hashmap <ansattnummer, <ansatt>>

Kunderegister Hashmap <personnummer, <kunde>>

Forsikringsregister Hashmap <forsikringsnummer, <forsikring>>

Hendelsesregister Hashmap <hendelsesnummer, <hendelse>>

For andre små register vil vi i utgangspunktet bruke ArrayList.

**Klassehierarki:**

Abstrakte superklasser:

* Incident
* Insurance
* Person

Disse vil ha flere subklasser. Eksempel: (Abstrakt Super)Person --> (Sub)Customer og (Sub)Employee

Foruten om disse vil det være flere individuelle klasser.

(UML av hierarki: Viser til vedlegg)

**Brukergrensesnittet:**

Vi har et felles innloggings vindu. Når en logger inn vil en enten møte et ansatt eller et kunde grensesnitt. Hvilket grensesnitt en får opp avhenger om brukernavnet er knyttet til et kunde nr. eller ansatt nr. Brukergrensesnittet vil bli delt opp i moduler, som vil hentes fram etter hva slags type oppgave som skal løses. Modulene skal vi lage med FXML, mye ved hjelp av JavaFX' SceneBuilder, men også noe manuell FXML koding. Vi vil programmere en MasterController-klasse som styrer hvilke moduler som skal vises på skjermen

**Moduler (vinduer):**

Det skal være vinduer for registrering av kunder, ansatte (dette vil en direktør eller avdelingsleder gjøre) og hendelser. Tegning av forsikringer, vindu for selskapets statistikk og vindu(er) for søking etter kunder, ansatte, forsikringer og forsikringssaker.

**Kodestandard og kommentering:**

Koden skal være så selvforklarende som mulig, det vil resultere i lite kommentering. Kommentarer (Javadoc) skal hovedsakelig brukes til å vise parameterlister og en liten forklaring til metoder

Planlagte funksjoner vil bli kommentert ut som //TODO:( … )

**Fil-lagring**

Registrene vil bli lest til en fil ved hjelp a ObjectOutputStream.

**Fremdriftsplan:**

Planleggingsfasen: 27.03.15 - 13.04.15

Implementeringsfasen: 13.04.15 - 17.05.15

Innlevering: Onsdag 20. mai 2015 kl. 15:00

|  |  |  |  |
| --- | --- | --- | --- |
| Beskrivelse | Start Dato | Forventet Ferdig Dato | Ferdig Dato |
| Forprosjekt | 27.03.15 | 13.04.15 | 13.04.15 |
| Skjelett: 1. utkast GUI | 13.04.15 | 13.04.15 |  |
| Alpha versjon |  | 01.05.15 |  |
| Beta/feature freeze |  | 10.05.15 |  |
| Testing, feilretting og rapportskriving |  |  |  |
| Ferdigstilling |  | 19.05.15 |  |

**Arbeidsoppgaver og fordeling:**

Må ha med:

1. Registerklasser ( Stian )
2. Forsikringsklasser ( Aleksander )
3. Personklasser ( Aleksander
4. Utrekning/utbetalingsklasse ( Aleksander )
5. Statistikk ( Steinar )
6. Brukergrensesnitt – Hovedvindu -> KundeVindu eller AnsattVindu ( Steinar )
7. Hendelsesklasser Husbrann/tyveri, personskade etc ( Stian ) generering av sampledata\ "automatisk" test ( Steinar )
8. Fil-lagring ( Stian )
9. Bug fiksing ( Alle )
10. Rapportskriving ( Aleksander )

Hvis vi får tid:

1. Ansatte: Kundebehandler, Direktør, Saksbehandler
2. Ny kunde selvbetjening
3. Ny forsikring selvbetjening
4. Multithreading
5. CSS
6. Bedrifts Kunder