OBJ2100 - PROSJEKTRAPPORT

17/06/2021

Kandidater deltatt på eksamen:

Gruppenummer 13: 7104, 7074, 7085, 7088, 7079

Innhold

Organisering:

Gruppen startet utleveringsdagen, 14.06.2021 med å lese, kartlegge og få en oversikt over eksamensoppgaven. Deretter fordelte vi oppgavene jevnt utover de fem gruppemedlemmene. Klasser, attributter, standardmetoder ble satt opp i fellesskap med oversikt over en storskjerm på et grupperom på USN. Vi opprettet et grensesnitt for å bli enig om hvilke metoder kontroll-klassen minimum skulle ha med tilhørende parameter. Utover eksamensperioden jobbet alle sammen over GitHub, hvor alle forandringer ble pushet og pullet som vi utviklet oppgaven videre.

Deltakelse:

Kandidat 7104 gjorde oppgave 2a som gikk ut på betalingsdelen og notere ned salg av billetter. Vedkommende jobbet også med GUI relatert til hovedsiden og oppsett av prosjektrapporten. Kandidat 7074 jobbet med oppgave 2c som handlet om avbestillingsdelen hos kinobetjenten. Det ble gjort mulig å slette alle ubetalte bestillinger. Vedkommende hadde også hovedansvaret for å teste og feilsøke applikasjonen via sin pc på en storskjerm. Kandidat 7085 gjorde oppgave 1b om rapportering og statistikk om billettsalget, kandidaten jobbet også med design og utvikling av GUI. Kandidat 7088 jobbet med oppgave 3 som handlet om bestillingsdelen for kunden, som gjør det mulig å bestille i kinolokalet og overalt. Kandidaten hjalp også gruppen med å sette opp og forklare oss hvordan man kan jobbe i fellesskapet over GitHub. Kandidat 7079 hadde oppgave 1a som handlet om administrasjonsdelen for kinosentralens planlegger. Det ble mulig å legge til film og visning i databasen. Vedkommende hjalp også gruppen med testdata til databasen, som gjorde det lettere å utføre oppgaver.

Vanskeligheter og problemløsning:

Gruppen hadde vanskeligheter med å sette opp GitHub for alle gruppemedlemmene på starten av perioden. Da alle hadde forskjellige operativ systemer (Windows, Mac IOS, og Linux) dukket det opp forskjellige problemstillinger. Vi hjalp hverandre med å få det til å fungere, det vi ikke klarte å hjelpe hverandre med, ble søkt opp på google for løsninger.

Et gruppemedlem hadde vanskeligheter med å sette opp dato og tid i tabellen, vedkommende fikk hjelp til dette fra andre gruppemedlemmer. Utfordringen handlet blant annet fra tabellen tblvisning hvor dato og tidspunkt var splittet i to kolonner. En annen utfordringen var formatet dato og starttid fikk fra SQL-spørringene. De samsvarte ikke med ikke lokale tider i Java. Derfor måtte dette formatteres. Vi endte med å gjøre mye av testing på dato/tid i applikasjonen kontra å gjøre alt i MySQL. Noe av grunnen er blant annet at det ble lettere å holde oversikt over opprinnelig data fra database og hva som er endret i applikasjonen. Vi splittet dette i to lister.

Vi hadde vanskeligheter i forbindelse med presentering i tableview. Grunnen var fordi man typisk setter verdiene med en setItems-metode som bruker en Observablelist som kun, ifølge Java dokumentasjon, kun at det legges inn verdier for en objekttype. Det betyr at observablelist<Billett, Plassbillett> ikke vil fungere. I tilfeller hvor vi måtte presentere verdier fra forskjellige lister eller eksempelvis ved statistikk hvor det må lages ny kolonner som ikke finnes, måtte vi bruke label. Her la vi alle verdiene i en Label som godtar string med alle verdier.

Anerkjente prinsipper:

- CamelCase syntaks
- Splittet applikasjonen i flere lag (Domene, grensesnitt og kontroll). Kontroll tar seg av all aktivitet mot database og vedlikehold av lister. Grensesnitt står for GUI kommunikasjon mot klient/sluttbruker og domeneklasser deklarerer hvordan de forskjellene objektene skal se ut.
- Vi har brukt PreparedStatement
- Prinsipp om minst mulig rettigheter. Variabler er deklarert privat og kan kun manipuleres gjennom gettere/settere
- Ellers er strukturen basert på Trond sitt JavaFX kompendium og informasjon fra forelesninger

- Informasjonen hentes fra databasen én gang og lagres i minnet (ObservableList). Dersom applikasjonen enten avsluttes ved å velge “avslutt”-knappen eller klikke på “X” vil vi slette alt innholdet i databasen. Deretter vil listene, som inneholder gammel informasjon pluss ny oppdatert info, bli lagt inn i databasen. På den måten sikrer vi at det unngås duplisering og nyeste informasjon er lagret
- Databaseapplikasjonen lukkes
- Implementering av try/catch