Universitetet i Sørøst-Norge

EKSAMENSFORSIDE

Skriftlig eksamen uten tilsyn

Fylles ut av ansvarlig faglærer

Emnekode: OBJ2100R-1	Emnenavn: Objektorientert programmering 2	
Campus: Ringerike	Fakultet: Handelshøyskolen	
Ansvarlig faglærer: Trond Braadland		
Utlev. tidspkt: Tirsdag 5. juni kl. 9.00	Innlev. tidspkt: Tirsdag 12. juni kl. 14.00	Innlev. sted: WiseFlow
Skrifttype: Times New Roman	Skriftstørrelse: 12	Linjeavstand: 1,5
Antall sider innledning / metatekst :	Antall ord:	Maks antall sider ekskl. forside og vedlegg:
Kriterier / oppgave (ved for liten plass, forts. på ny side):		

Oppgave OBJ2100 Objektorientert Programmering 2 Eksamenscase våren 2018

Formelle krav, levering og muntlig høring

Se eget notat.

Oppgaven

Det skal lages en applikasjon for en kinosentral med et antall kinoer og kinosaler, for bestilling av kinobilletter. Applikasjonen skal lages i Java med database i MySQL. Det skal gå an å logge seg på databasen (med fulle rettigheter) med brukernavnet *Case* og med passord *Esac*. Det er laget et eget script som oppretter databasen.

Det legges vekt på at applikasjonen er ryddig, feilfri og robust i bruk med god dokumentasjon. Det legges også vekt på at det er benyttet anerkjente prinsipper for god kode.

Systemet skal ha tre hoveddeler:

1. **For kinosentralens planlegger:**
    a. **Administrasjonsdel** der planleggeren kan legge inn filmer og visninger
    b. **Rapportdel** der planleggeren får ut statistikk om billettsalget
2. **For kinobetjenten:**
    a. **Betalingsdel** der betjenten kan notere at en billett er betalt
    b. **Salgsdel** der betjenten kan notere direkte salg i kinoen
    c. **Avbestillingsdel** der betjenten for en bestemt visning kan slette alle bestillinger som ikke er betalt – denne starter betjenten selv 30 minutter før visningen starter
3. **For kunden (kinopublikum):**
    Bestillingsdel som kjøres på flere terminaler i kinolokalet og andre steder, der kunden kan bestille kinobilletter og få vist bestillingen

Når en kunde vil bestille billetter, starter de bestillingsdelen (uten pålogging) og får vist fremtidige visninger (der det er minst 30 minutter igjen til visningen starter) med pris og antall ledige plasser på visningen. De kan sortere dette etter film, eller etter tidspunkt. De må bestemme seg for en visning og får da vist de ledige enkeltplassene for denne visningen. Her kan de velge (og ombestemme) plasser. Hele tiden blir totalbeløpet og antall plasser vist og oppdatert. Det er ikke nødvendig at systemet kan foreslå plasser.

For å finne forestillinger der det er minst 30 minutter til den starter, må dere hente aktuell tid og presentere den i vinduet. Java har flere klasser for tid, men den beste i dette tilfellet er kanskje Date, som når et objekt opprettes vil inneholde aktuell tid. Date har metoden toString(), som returnerer tiden som en String. Date kan gjerne kombineres med DateFormat. Vi hadde om dette i høst, og dere finner mye ved å google.

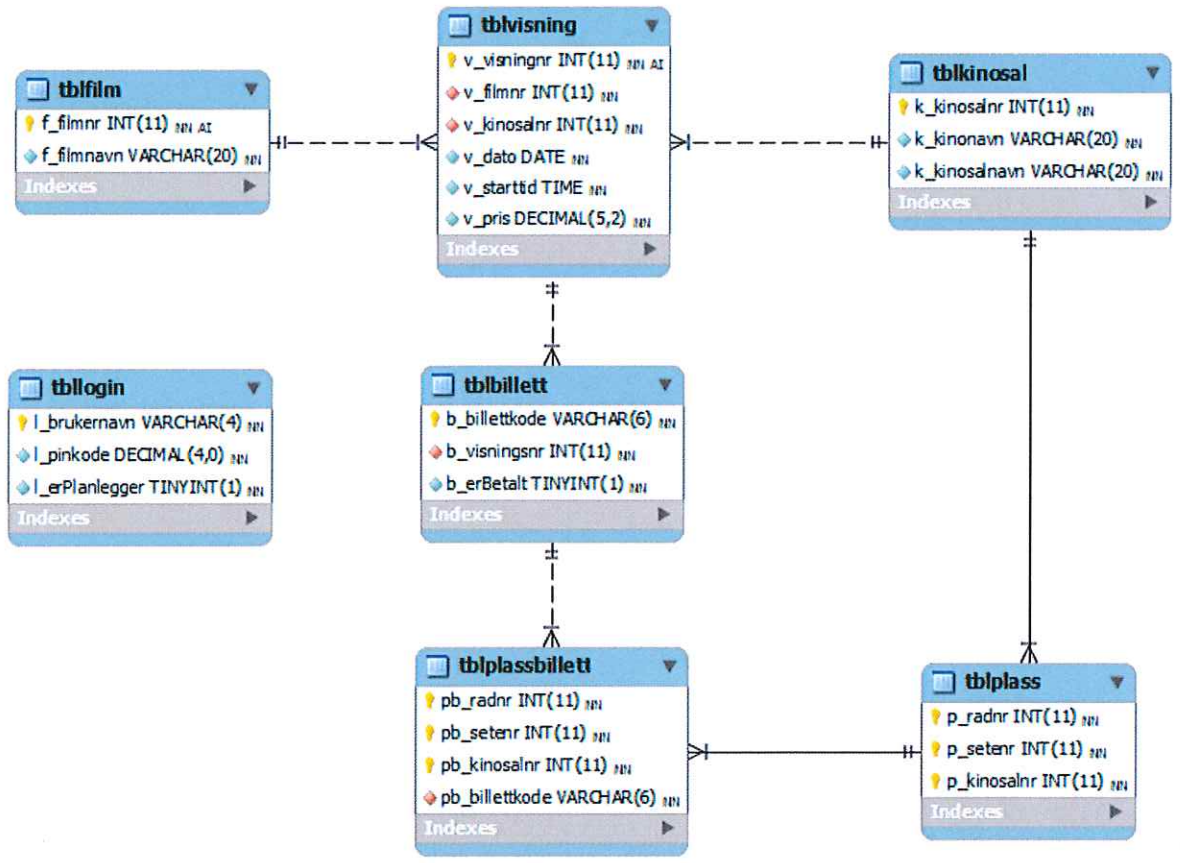
Når brukeren har bestemt seg for visning og plasser, vises et bekreftelsesskjema der alle opplysninger er angitt inkludert totalpris og billettkoden. Det står en tydelig beskjed om at billettene må hentes senest 30 minutter før forestillingen og at kunden da må oppgi den viste billettkoden. Det er ikke nødvendig å lage utskrift av billetten, men det skal være mulig å bekrefte bestillingen (bestillingen gjennomføres) eller avbryte (da settes plassene ledige igjen). I begge tilfelle bekreftes med en meldingsboks og åpningsskjemaet vises.

Når kundene kommer til kinoen for å betale og hente de bestilte billettene, må de oppgi den koden de fikk oppgitt ved bestillingen. Kinobetjenten tar imot betaling og noterer da i systemet at billettene er betalt.

Kinobetjenten selger – når som helst – ledige billetter direkte. De betales alltid med en gang. Kinobetjenten kan også – for en bestemt visning – slette alle bestillinger som ikke er betalt. Slettingen logges til en egen fil *"slettinger.dat"*. For å starte denne delen av systemet, må kinobetjentene logge seg på med brukernavn og PIN-kode (4 siffer). En av disse er "knut"/1234. Kinosentralens planlegger må også logge på med brukernavn og PIN-kode for å få kjørt planleggingssystemet. Én av planleggerne er "tunk"/4321. I denne systemdelen, kan planleggeren endre alle opplysninger om filmer og visninger som det ennå ikke er bestilt/solgt billetter til. Planleggeren skal også kunne få ut statistikk om billettbestilling og billettsalget. En statistikk gjelder én, bestemt film og skal vise utviklingen over tid for hvor mange som så filmen på hver enkelt visning (antall og i prosent av kinosalens kapasitet) og hvor mange bestillinger som ble slettet fordi de ikke ble hentet/betalt i tide. En annen statistikk planleggeren trenger, skal vise hvor populær en kinosal er. For en bestemt kinosal skal den vise hvor mange prosent av plassene som gjennomsnittlig ble solgt for visninger som har vært holdt (dvs. som er startet).

Databasen *kino* er laget. Den skal anses som ferdig og låst. Et skript som genererer databasen med noen data, er vedlagt. Dere må gjerne legge inn flere data men dere får ikke endre databasen. Dere får heller ikke lagre spørringer i databasen for å bruke dem i programmet.

Datamodellen ser slik ut [1]:



[1] MySQLs *TINYINT(1)* tilsvarer Javas boolean.

Opplysningene i *tblLogin*, *tblKinosal* og *tblPlass* er relativt faste og legges inn direkte i MySQL. De andre opplysningene skal kunne endres ved hjelp av den applikasjonen dere lager. Hver plass i en kinosal har et radnummer (fra 1) og et setenummer (fra 1 for hver rad). Attributtet *b_billettkode* er en systemgenerert, unik, tilfeldig streng med fire store bokstaver og to siffer. (Den kunne vært generert av MySQL, men det er ikke lagt inn. Dere må altså lage det i programmet. Da den er satt som primærnøkkel, vil MySQL gi feilmelding hvis den ikke blir unik. Da må programmet generere en annen billettkode uten å gi brukeren feilmelding.)
Tabellen *tblPlassbillett* inneholder bare de plassene, for hver visning, som er bestilt og/eller betalt.

Krav til designen av systemet

Dere har stor frihet når det gjelder utformingen av brukergrensesnittet, men noen felles krav er det. Applikasjonen skal ha:

- Et hovedvindu der man velger hvilken del av systemet som skal brukes
- Et dialogvindu for hver av de tre hoveddelene av systemet
- Rapportdelen skal vise statistikk om billettsalget i en JTable
- Når man søker på forestillinger (jfr. foran), skal aktuell tid vises i vinduet.

Applikasjonen skal ved oppstart lese data fra databasen og legge dem i egnet datastruktur i minnet (jeg foreslår ArrayList'er). Oppdatering av databasen skal først skje når programmet avsluttes. Dere trenger altså en kontrollklasse for å håndtere dette.

Formelle krav

Oppgaven utleveres **tirsdag 5. juni kl. 9.00 i WiseFlow**. Klokken 1200 samme dag arrangeres et møte med faglærer for avklaring av uklarheter og feil i oppgaveteksten.

Oppgaven skal løses i grupper på maks fem studenter. (Det er ikke noe minimumsantall, men erfaringsmessig bør gruppen være på minst tre. Det har vist seg vanskelig for én eller to studenter å klare tilstrekkelig.) Studentene setter selv sammen gruppene, som meldes til faglærer (gjennom diskusjonsforum i kurset i Fronter) sammen med et selvvalgt gruppenavn senest **tirsdag 12. juni kl. 14.00 i WiseFlow**.

Alle hjelpemidler er tillatt. Det er også tillatt å søke hjelp til løsninger hos fagansatte ved høyskolen hvis det er noe gruppen ikke får til. (Her regnes ikke generell veiledning, men hjelp til å skrive kode i løsningen.) Hvis gruppen har fått slik hjelp, skal dette uttrykkelig angis både i prosjektrapporten og der i systemet det er relevant (f.eks. i programkoden eller system-dokumentasjonen). Både hvem som ga hjelpen og hvordan det synes i løsningen skal dokumenteres.

Levering/krav

Besvarelsen leveres som zip-fil av prosjektet og annen dokumentasjon i WiseFlow ***senest tirsdag 12. juni kl. 14.00****. Fristen er absolutt. Det betyr at dere leverer det dere har klart.*

Følgende skal leveres:

1. Fullt kjørbart system. Det er ikke nødvendig å kompilere, det gjør jeg selv. Pass på at MySQL-driveren ligger i en mappe under prosjektet. Jeg har selv min utgave av databasen. Kontroller for all del at alt kommer med i zip-filen.
2. Dokumentasjon, bestående av:
    a. **Systemdokumentasjon**. Det inkluderer komplett, kommentert kildekode der det også fremgår hvem som har laget koden og hvem som har testet/godkjent den. Bruk JavaDoc til dette. Det er ikke nødvendig å kopiere kildekoden til andre dokumenter – levér kildekoden slik den er (som .java-filer).
    b. **En test med JUnit**. Dere lager her en test for når kinobetjenten oppgir brukernavn og pinkode.
    c. **Komplett brukerveiledning** som en pdf-fil. Her viser dere skjermbilder og forklarer kort hva de enkelte deler av programmet gjør, og hvordan de brukes.
3. **Enkel prosjektrapport** der det fremgår
    a. navnet på gruppen
    b. hvem som var med i gruppen (fullt navn og studentnummer),
    c. hvordan prosjektarbeidet ble organisert og
    d. detaljert hva hver deltaker har gjort
    e. spesielle vanskeligheter gruppen har møtt underveis og hvordan de ble løst og hjelp som gruppen har fått

Alt leveres samlet i elektronisk form i én, enkelt zip-fil gjennom mappen "Innlevering" i Fronter av én av gruppemedlemmene. Pakkingen skal gjøres til standard zip-format.