Databaseprosjekt DB2

TDT4145

VÅR 2021

Gruppe 177

Fremming, Daniel
Meo, August Inge Lønning
Simhan, Viveka Priya

Innhold i innleveringen

1. Java-filer:
   - LogInCtrl.java
   - PostCtrl.java
   - AnswerCtrl.java
   - SearchCtrl.java
   - Statistic.java

2. SQL-filer:
   - createDatabase.sql som lager tabellene
   - createContent.sql som generer innhold til tabellene for å teste usecasene

A. Oversikt over klassene i programmet og hvilke oppgaver de løser.

ConnectorClass

Klassen muliggjør en tilkobling til databasen i MySQL for de andre klassene i prosjektet. Alle klasser med Ctrl i navnet arver fra ConnectorClass.

LogInCtrl

Klassen brukes til usecase 1, og sjekker at at gitt username eksisterer allerede i databasen, og at gitt username og passord korresponderer med hva som er i databasen for denne kombinasjonen.

PostCtrl

Klassen brukes til usecase 2, og setter inn ny instans av post i post-tabellen. I tillegg brukes det hjelpefunksjoner for å kontrollere at om anonymitet er tillatt for emnet(course), samt innsetting av post i riktig folder(Exam).

AnswerCtrl

Klassen brukes som kontroller for usecase 3. Klassen setter inn ny instans i InstructorAnswer- og AnsweredByInstructor-tabell. I tillegg inneholder den hjelpefunksjoner for å oppdatere fargekoden i post-tabellen for besvart post, for å sjekke om postID finnes samt om posten er blitt svart på tidligere.

SearchCtrl

Klassen brukes som kontroller for usecase 4, og gjennomfører søk for input søkeord. Printer ut en liste med postID som inneholder søkeordet.

StatisticCtrl

Klassen brukes til usecase 5, og printer ut en liste med statistikk på formatet: username, number read, number posted. Listen er sortert etter number read i synkende rekkefølge.

Piazza

Piazza-klassen samler kontrollene på et sted, og det er her de ulike usecasene(1-5) i programmet leser inn input fra bruker.

Main

Main-klassen er skallet som samler alle usecasene i en switch, der bruker kan velge hvilke usecase som skal kjøres. Det er i denne klassen bruker interagerer med programmet.

B. Oversikt over usecasene og hvordan disse er implementert.

Bruker får tilgang til alle usecasene i main-metoden til klassen Main. Bruker blir først bedt om å logge inn og kan så velge mellom de andre usecasene ved hjelp av en switch-meny.

Usecase 1: Innlogging
Brukerinput håndteres i metoden logIn i klassen piazza. Bruker blir bedt om å angi epost og passord. Dersom innloggingsforsøket feiler får bruker også mulighet til flere forsøk.

Input blir gitt til en instans av klassen LogInCtrl som håndterer spørring mot databasen. Denne tar inn to strenger: epost og passord. Først sjekkes det om eposten finnes i databasen. Dersom den finnes utføres det en spørring for å hente tilhørende passord og rolle. Passordene sammenlignes og det returneres en boolsk verdi på om de er like.
Dersom passordene stemmer overens settes attributtene username og role i klassen piazza og attributtet loggedIn settes lik true. Disse brukes senere i klassen main for å håndtere tilgang til usecasene.

Usecase 2: Bruker lager en post av typen “Question” i mappen “Exam”
Tilgang til denne forutsetter at bruker har klart å logge seg inn. Brukerinput håndteres i metoden makePost i klassen piazza. Bruker blir bedt om å angi tittel og innhold til posten, samt ønsket post-type og mappe. Dersom emnet tillater anonym posting blir bruker også spurt om han/hun ønsker dette. Inputet gis videre til en instans av klassen PostCtrl som håndterer innsetting og spørring mot databasen samt tilhørende logikk.

I PostCtrl benyttes hjelpemetoder for å finne riktig FolderID, om emnet tillater anonym posting og for lage nye rader i tabellene Post og PostInFolder. Her benyttes input fra bruker til å sette tittel, innhold, post-type og mappe. Author atributtet settes basert på om anonym posting er tillatt og ønsket. Email, date og ColorCode settes automatisk til brukerens epost, dagens dato og “red”. Sql filen som oppretter databasen er endret fra første innlevering slik at postID nå automatisk inkrementeres.

Usecase 3: en instruktør svarer på en post
Tilgang til denne forutsetter at bruker er innlogget og har rollen “Instructor”. Brukerinput håndteres i metoden answerPostInstructor i Piazza. Bruker blir spurt om å angi postID til posten han/hun ønsker å svare på, og innholdet i svaret. Dette gis videre til en instans av klassen AnswerCtrl som håndterer innsetting i databasen og tilhørende logikk.

AnswerCtrl sjekker om posten er blitt besvart av en instruktør tidligere, og hvilken instruktør det i så fall er som har svart, og håndterer tre ulike tilfeller:

1. Posten er ikke besvart fra før av.
   Da opprettes det helt nye rader i både answeredByInstructor og InstructorsAnswer tabellene med brukers epost, postID, innholdet og datoen. I tillegg endres fargekoden til posten i Post tabellen til “yellow”.
2. Posten er blitt svart på av en annen instruktør.
   Det tidligere svaret hentes, og raden i InstructorsAnswer oppdateres ved at det nye svaret legges til det gamle. En ny rad opprettes i answeredByInstructor med postId, brukers epost og dato.
3. Posten er blitt svart på av bruker tidligere
   Da må radene i både InstructorsAnswer og answeredByInstructor oppdateres. InstructorsAnswer oppdateres igjen ved å legge det nye svaret til det gamle. I answeredByInstructor oppdateres date attributtet i raden med brukers epost og postID til dagens dato.

Usecase 4: Bruker søker etter innhold i poster
Brukerinput håndteres av searchForKeyword metoden i klassen Piazza. Her blir bruker bedt om å skrive inn hvilket innhold det skal søkes etter i poster. Input blir behandlet av en instans tilhørende klassen SearchCtrl, der metoden executeSearch utfører spørring mot databasen og leter etter søkeordet i tabellene post, follow-up og comment som tilsammen utgjør en thread.

Usecase 5: En instruktør undersøker statistikken over brukere
Tilgang til denne forutsetter at bruker er innlogget med rollen “Instructor”. Metoden checkStatistics i klassen Piazza kontrollerer om bruker har instruktør-privilegier med getRole-funksjonen. Hvis bruker har instruktør-privilegier kalles showStatistics metoden som tilhører StatisticCtrl-klassen. Det utføres spørringer mot post-, hasRead-, enrolled- og user-tabellene for å hente informasjon om antall poster en bruker har lest og postet. Metoden henter kun statistikk om brukere som er studenter og som er påmeldt emnet. Metoden printer ut en liste på formatet username/read/posted som er sortert etter antall leste poster i synkende rekkefølge. Listen inneholder også informasjon om brukere som hverken har lest eller postet noe.