

Like To Learn

Projekt webbapplikation

Grupp 4 United

Gruppmedlemmar

Namn	GitHub-alias	E-post	Tel
Fifi Johansson	fific	fifi@student.chalmers.se	070-076 76 81
Quang Luong	qualu	quangl@student.chalmers.se	076-902 06 39
Linn Leiulfsrud	linnlei	linnl@student.chalmers.se	070-829 16 44
Jenny Forsberg	jenfor	jenfor@student.chalmers.se	070-561 06 88

Start-commit

Jenny (github: jenfor) gjorde i början av projektet en commit av resource-mappar på ca 47 200 rader kod.

Vår applikation

Vår applikation är menat som en mini-version av en större tänkt web-applikation. Denna applikation kommer att användas för ett spelprojekt i undervisningssyften. Denna applikation hålla reda på kurser skapade av lärare. Dessa kurser innehåller flertalet moment som i sin tur innehåller flertalet frågor. Elever har koll på de kurser de har registrerat sig på för tillfället, och i den större versionen kommer eleven se sina framsteg och resultat på ett användarvänligt sett.

Logga ut

LIKE TO LEARN

Elever Kurser Statistik Konto Välkommen lärare1 Inloggad som lärare1

Kurser

Föregående Nästa

Kurskod	Kursnamn			
67	Gångertabell_2	Redigera kurs	Hantera moment	Ta bort kurs
68	Gångertabell_3	Redigera kurs	Hantera moment	Ta bort kurs
66	Gångertabell_1	Redigera kurs	Hantera moment	Ta bort kurs
256	Historia 1	Redigera kurs	Hantera moment	Ta bort kurs
255	Svenska 1	Redigera kurs	Hantera moment	Ta bort kurs

Skapa kurs

2016 Grupp 4 United

Idén är i den större web-applikationen att mer funktionalitet tilläggs så att kurser, moment, frågor mm, dyker upp i ett spel som uppdateras efter det innehåll som finns i databasen.

Spelet tänks även kommunicera via nätverk så att web-applikationen uppdateras efter händelser i spelet. (Dubbelriktad kommunikation)

Vi tänker oss alltså en läroplattform bestående av ett spel (som vi till stora delar redan utvecklat i en parallell kurs), som ska kunna styras av lärare via denna web-applikation som interface. Allt detta för att göra undervisningen, läxförhör, osv roligare för elever, samt underlätta lärares arbetssituation genom att minska tiden för rättning. Främst är spelet riktat till åldrar i låg-/mellanstadiet.

Nuvarande funktionalitet (2016-03-13)

I nuvarande version av web-applikationen går det att registrera konton med användarnamn, (person)namn, lösenord och roll. Dessa konton kan användas för att logga in (ingen säkerhet förutom att lösenord är korrekt), var på man dirigeras till lämplig sida beroende på om man som roll är lärare eller elev.

Första sidan för båda roller efter inloggning är "konto"-sidan (som ligger i index.xhtml) där kontots egenskaper visas.

Som elev kan man gå över till en flik där man ser sina registrerade kurser och lägga till fler bland urvalet av existerande kurser.

Som lärare kan man se, utöver sitt konto, samtliga elever, kurser och statistik. Listan för elever visar alla elever i databasen och lärare har fått dubbelroll som administratör, så att de tillåts redigera och ta bort elevkonton. Varje kurs innehåller en mängd av moment, och varje moment innehåller frågor som läraren lagt till.

Applikationens sidstruktur:

- **Hemsida** - Information om spelet
 - Skapa användare
 - Logga in
 - Lärare
 - Elever - Se elever, lägg till, redigera, ta bort
 - Kurser - Se kurser, lägg till, redigera, ta bort
 - Se moment, lägg till, redigera, ta bort
 - Se frågor, lägga till, redigera, ta bort
 - Statistik - Se antal elever, lärare, användare, kurser
 - Konto - Redigera egna kontots detaljer
 - Logga ut
 - Elev
 - Registrerade kurser - Se registrerade kurser, lägg till, ta bort
 - Se moment
 - Konto - Redigera egna kontots detaljer

- Logga ut

Use cases

- För att skapa nytt lärarkonto, klicka på registrera nytt konto och fyll i detaljer, välj Lärare
- För att skapa nytt elevkonto, klicka på registrera nytt konto och fyll i detaljer, välj Elev
- För att logga in, fyll i användarnamn och lösen och klicka på Logga in
- För att se alla elever, logga in som lärare, klicka på Elever
- För att redigera en elevs detaljer, logga in som lärare, klicka på Elever, klicka på Redigera elev för en elev, fyll i nya detaljer, klicka på Spara
- För att ta bort en elev, logga in som lärare, klicka på Elever, klicka på Ta bort elev för en elev, klicka på Ta bort, Spara
- För att se alla kurser i läroplanen, logga in som lärare, klicka på Kurser
- För att skapa kurs, logga in som lärare, klicka på Kurser, Skapa kurs, fyll i detaljer, Spara
- För att redigera kurs, logga in som lärare, klicka på Kurser, Redigera kurs, fyll i nya detaljer, Spara
- För att ta bort kurs, logga in som lärare, klicka på Kurser, klicka på Ta bort kurs för en kurs i listan, klicka på Ta bort
- För att lägga till moment i en kurs, logga in som lärare, Kurser, klicka på Hantera moment för en kurs i listan, Skapa moment, fyll i detaljer, Spara
- För att redigera ett moment, logga in som lärare, Kurser, Hantera moment, Redigera moment, fyll i nya detaljer, Spara
- För att ta bort moment, logga in som lärare, Kurser, Hantera moment för en kurs i listan, Ta bort moment för ett moment i listan, Ta bort.
- För att lägga till fråga, logga in som lärare, Kurser, Hantera moment för en kurs i listan, välj Hantera frågor för ett moment i listan, Skapa fråga, fyll i detaljer, spara
- För att redigera fråga, logga in som lärare, Kurser, Hantera moment för en kurs i listan, välj Hantera frågor för ett moment i listan, Redigera fråga, fyll i nya detaljer, spara
- För att ta bort fråga, logga in som lärare, Kurser, Hantera moment för en kurs i listan, välj Hantera frågor för ett moment i listan, Ta bort fråga, Ta bort
- För att se statistik, logga in som lärare, klicka på Statistik
- För att logga ut, klicka på Logga ut i övre högre hörnet
- För att registrera på existerande kurs, logga in som elev, klicka på Kurser, Lägg till kurs, fyll i en av kurskoderna i listan, klicka på Anslut till
- För att avregistrera från kurs, logga in som elev, klicka på Kurser, Lämna kurs för en av de registrerade kurserna
- För att se alla moment i en kurs, logga in som elev, klicka på Kurser, klicka på Se moment för en av de registrerade kurserna
- För att ändra namn/lösenord, logga in som valfri typ av användare, klicka på Konto, skriv in nytt namn/lösenord, Spara.

Teknisk design av systemet

Vi har använt oss av den komponent-baserade approachen.

I front end/presentationslagret har vi använt JSF (Facelets), XHTML, Bootstrap, CSS.

I backend/business-lagret har vi använt CDI-böjor och EJBs i för att ta hand om logik.

DAOs/listor implementeras i form av EJBs som hanterar databasen via persistens-lagret.

JPQL-förfrågor har också använts för att hämta ut data via persistens-lagret.

Persistens-lagret omhändertas av JPA. En egen modell/databas-struktur har planerats och skapats, i form av entity beans, för att få till önskad funktionalitet.

Projektets struktur

Web pages

Paket:

- resources: Innehåller bootstrap, css, bilder.
- student: Innehåller sidor menade för student.
- teacher: innehåller sidor menade för lärare.
- partials: Innehåller olika listors sidor med tillhörande partials.

Application

Paket:

- com.united.auth - självständigt paket med entitet, kontroller mm. för inloggning.
- com.united.core - innehåller alla entiteter samt DAOs i databasen (förutom de som rör användare, som alltså ligger i föregående paket).
- com.united.ctrl - innehåller kontroller för samtliga entiteter (förutom login).
- com.united.data - innehåller exempeldata och servlet som körs vid uppstart.
- com.united.persistence - innehåller DAO-mall (basklass) samt IDAO (Interface för DAO).
- com.united.view - har undermappar som håller backing beans för varje entitet

Libraries:

- eclipselink
- javaee-web-api
- javax.persistence

Lager och deras komponenter

JSF pages:

För vår webb-del använder vi oss av tre olika templates och en mängd partials.

- **template.xhtml**: Används innan inloggning. Har enbart header och footer.
- **studentTemplate.xhtml**: Denna template används på sidor som visas för inloggade elever. Inklusive: registrationList, addRegistration, deleteRegistration.
- **teacherTemplate.xhtml**: Denna template används när man loggar in med ett lärarkonto. Denna template använder sig av alla följande partials uppdelade i olika paket utefter vad de behandlar. Dessa paket är:
 - **courses**: Innehåller sidorna som berör kurser
 - **moments**: Innehåller sidorna som berör kursmoment
 - **questions**: Innehåller sidorna som berör frågor
 - **students** : Innehåller sidorna som berör studenter

Varje sådant paket innehåller ett antal jsf-sidor, med en master som är en lista över det som paketet behandlar, samt ett antal details. Details varierar mellan paket, men de vanligaste är: add (för att lägga till en rad i databasen), delete (för att ta bort en rad ur databasen), samt edit (för att ändra en rad i databasen).

Backing beans:

Våra backing beans ligger i paketen med prefixet com.united.view. Varje paket innehåller 3-4 st backing beans: Add, Delete, Edit och List. De ansvarar var och en för de motsvarande jsf-sidorna som nämndes i JSF pages.

- **com.united.view.courses**: CourseListBB, AddCourseBB, DeleteCourseBB, EditCourseBB
- **com.united.view.moments**: MomentListBB, AddMomentBB, DeleteMomentBB, EditMomentBB
- **com.united.view.questions**: QuestionListBB, AddQuestionBB, DeleteQuestionBB, EditQuestionBB
- **com.united.view.registrations**: RegistrationListBB, AddRegistrationBB, DeleteRegistrationBB
- **com.united.view.users**: UserListBB, AddUserBB, DeleteUserBB, EditUserBB

Support beans:

- **School**: Application scoped. Är container för alla DAO, som i vår applikation är: UserList, CourseList, MomentList, QuestionList, och RegistrationList. Alla control beans kommunicerar med School för att hämta dessa DAO och sedan kunna skriva till dem.

Control layer (control beans):

Våra control beans ligger i paketet `com.united.ctrl`, och är:

- **CourseController**: Hanterar kurser
- **MomentController**: Hanterar kursmoment
- **QuestionController**: Hanterar frågor i moment
- **RegistrationController**: Hanterar användares registreringar på kurser
- **UserController**: Hanterar användarkonton

Vi har även en control bean för login-funktionen, som ligger i paketet `com.united.auth`:

- **LoginBean**: Hanterar in- och utloggning.

Model:

I paketet `com.united.core` ligger alla DAO, som extender `AbstractDAO`. Våra DAO är:

- `UserList`
- `CourseList`
- `MomentList`
- `QuestionList`
- `RegistrationList`

De hanterar var för sig de entiteter som deras namn indikerar.

Persistence layer:

Persistence-delen av detta lager ligger i paketet `com.united.persistence`, och innehåller:

- **AbstractDAO**: basklass för alla persistenta DAO
- **IDAo**: Interface för `AbstractDAO` och därmed alla DAO.

Core-delen av detta lager ligger i paketet `com.united.core`, samt `com.united.auth` för funktionalitet relaterad till användare och login. Paketen innehåller:

Entity-klasser:

- `Course`: Kurser. Har `courseId` och ett namn
- `FinishedMoment`: En relation mellan användare samt de kursmoment de slutfört.
- `Moment`: Moment i en kurs. Håller frågor (`Questions`)
- `Question`: Frågor som finns i ett kursmoment. Har en fråga samt ett korrekt svar.
- `Registration`: En relation mellan användare samt kurser de är registrerade på.
- `User`: En användare. Har ett unikt login, ett lösenord, samt en användartyp (lärare eller elev)

DAO: Varje DAO hanterar den sorts Entity som dess namn indikerar

- `CourseList`
- `FinishedMomentList`
- `MomentList`
- `QuestionList`
- `RegistrationList`
- `UserList`

Database:

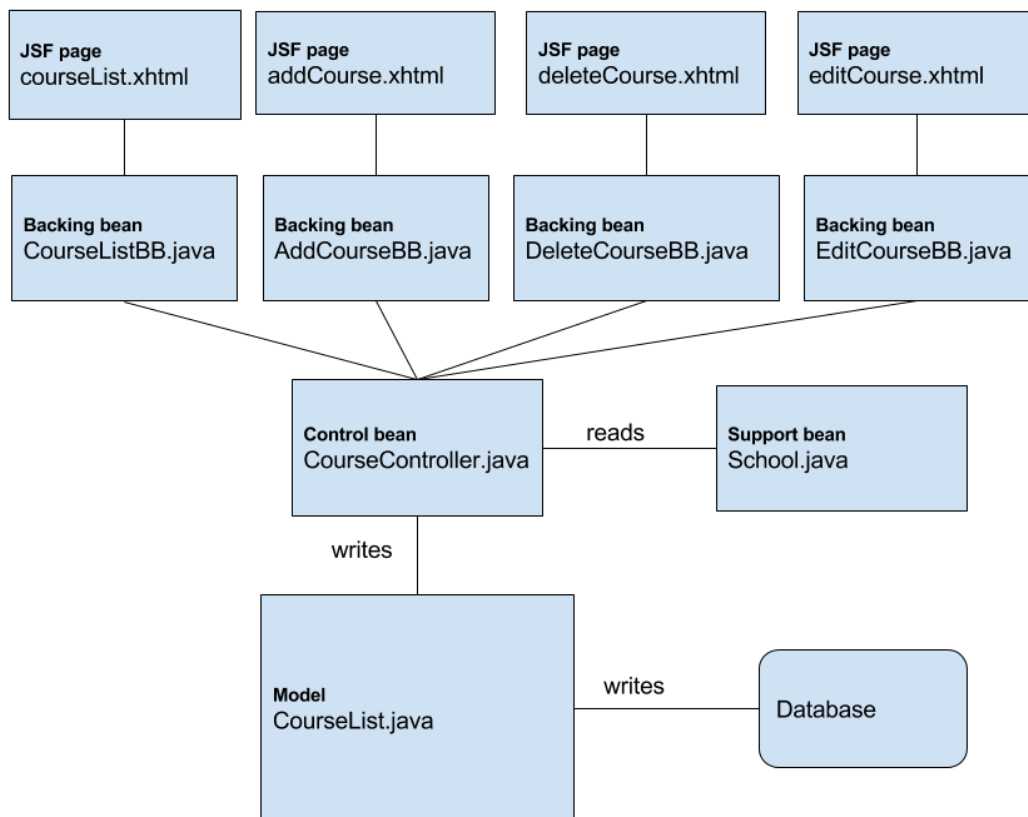
Vår databas innehåller tables för samtliga entities, samt för följande relationer:

- `COURSES_MOMENTS`: Kurs innehåller moment (one to many)

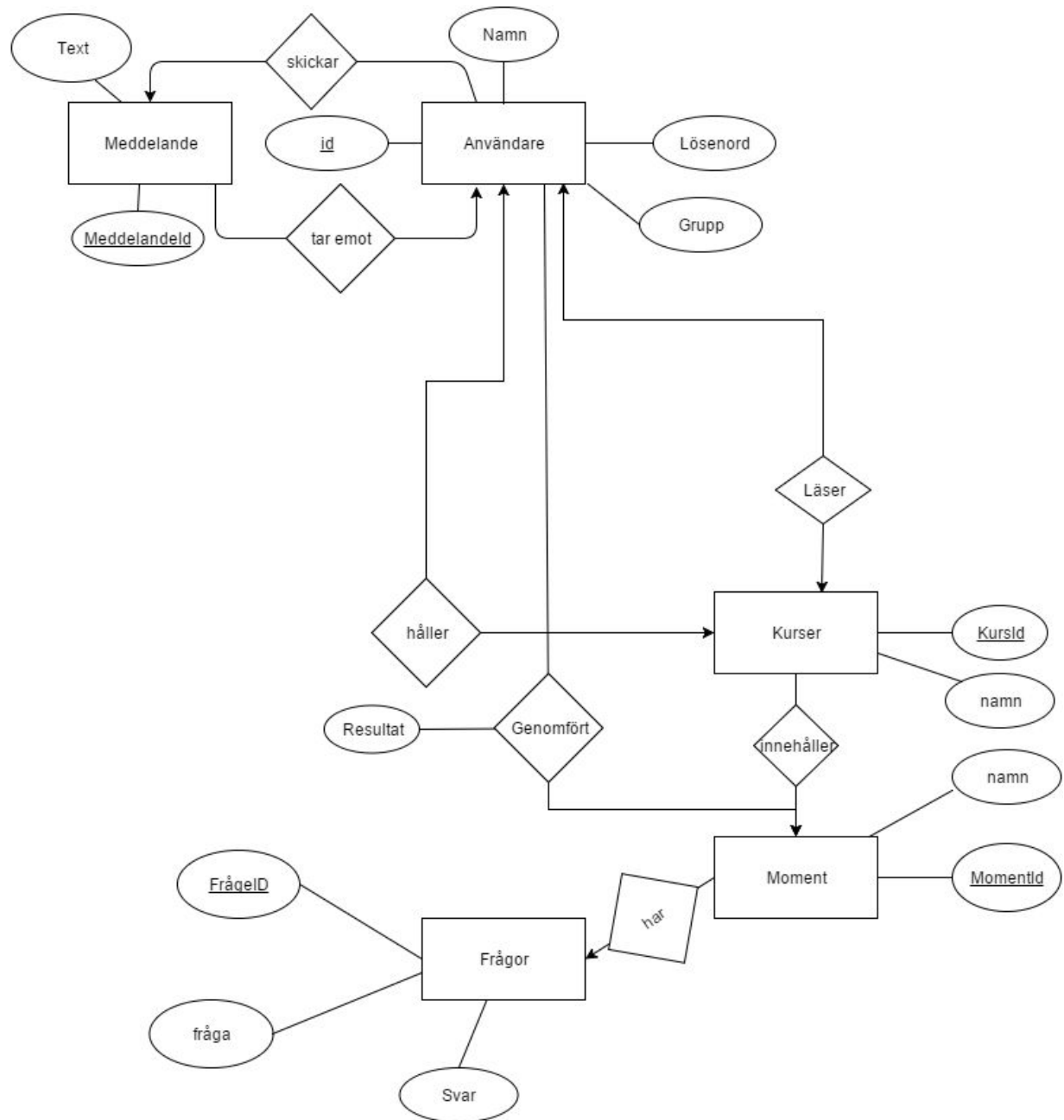
- FINISHEDMOMENTS: Användare har avklarat moment (many to many)
- MOMENTS_QUESTIONS: Moment innehåller frågor (one to many)
- REGISTRATIONS: Användare är registrerad på en kurs (many to many)
- USERS_GROUPS: Användare tillhör viss grupp (many to many)

För ER-diagram över databasen, se Bilaga 1.

Följande är ett exempel på hur våra komponenter kommunicerar. I detta exempel visas de komponenter som hanterar kurser.



Bilaga 1 ER-diagram



Notera att meddelanden ej används, inte heller relationen "genomfört" (men finns skapade i projektet). Relationen "håller" är ej skapad.

Bilaga 2 UML-diagram

