

# GroupFinder

Sofia Edström 901218-4124 sofia.edstrom@galaxen.se  
Hampus Forsvall 921122-2279 hampus.forsvall@gmail.com  
Tomas Hasselquist 920811-1097 tomas@hasselquist.se  
Patricia Paulsson 920101-5188 patricia.paulsson@gmail.com

## Syfte

GroupFinders syfte är att underlätta sökandet efter olika former av grupper inom diverse kurser på Chalmers. Sidan ska agera en portal för personer som inte har lyckats hitta en grupp på egen hand och därav söker likasinnade för att skapa egna grupper.

GroupFinder ska alltså inte ses som en tjänst för att hantera samtliga grupper i en kurs utan enbart ett hjälpmedel för studenter som inte redan har en etablerad grupp.

## Användare

Systemet stödjer två former av användare.

### Admin

Har full kontroll över skapandet av såväl kurser som grupper och användare. Denna kan fritt hantera och editera samtliga former av entiteter (Kurser, Grupper och Användare).

### User

En vanlig användare av GroupFinder. Kan skapa grupper i en given kurs och gå med i andra grupper.

## Use Cases

### Användare

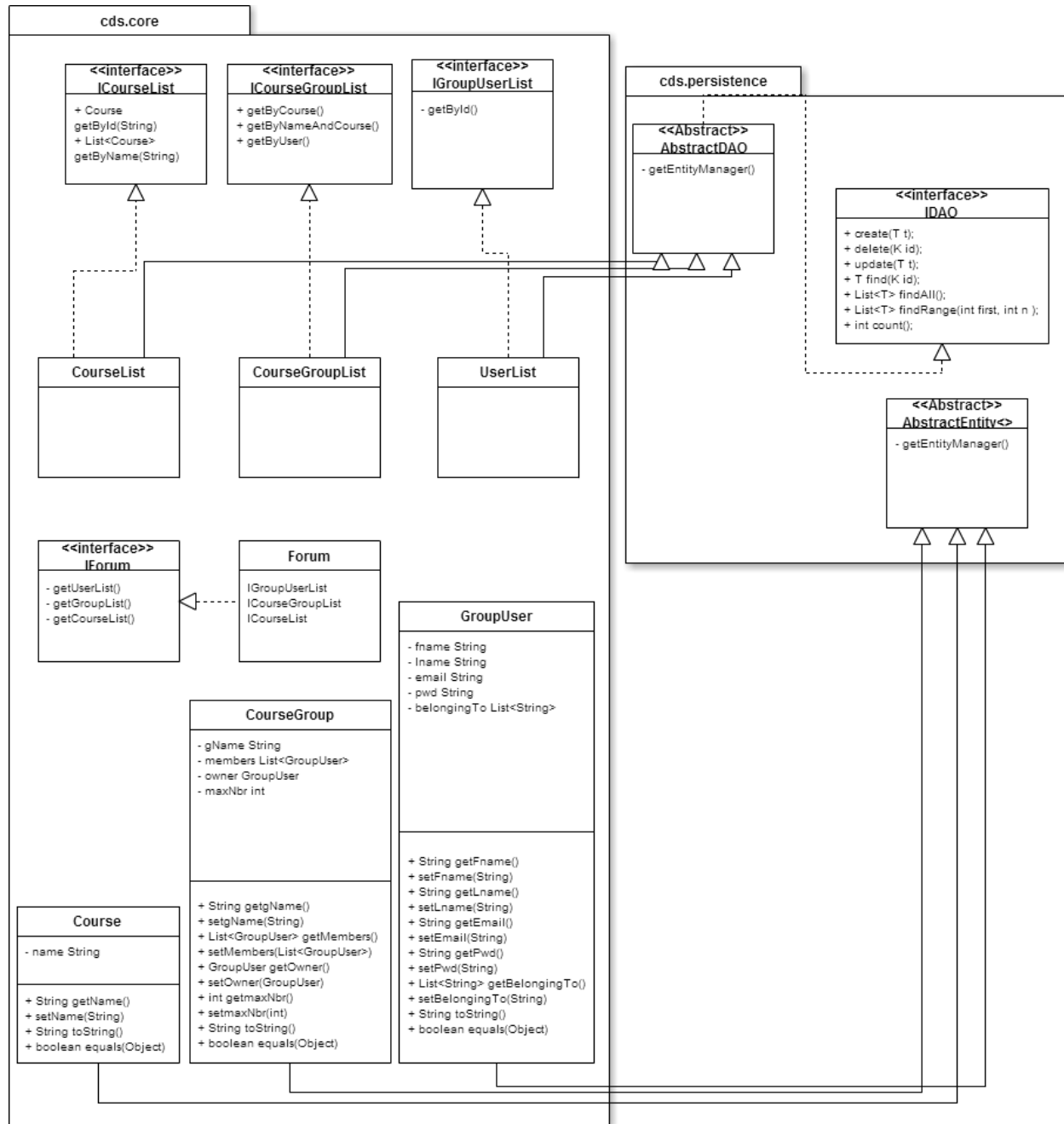
- Användare kan registrera sig
- Användare kan logga in/ut
- Användare kan besöka sin profil
- Användare kan ändra sina användaruppgifter utom nyckeln (användarnamn)
- Användare kan se om den är admin
- Användare kan söka efter samtliga kurser på Chalmers

- Användare kan se alla grupper inom en kurs
- Användare kan se alla medlemmar, antal medlemmar och max antal medlemmar i en grupp
- Användare kan skapa en ny grupp
- Användare kan gå med i en specifikt vald grupp
- Användare kan gå med i en slumpmässigt (av systemet) vald grupp tillhörande kurs man är inne på
- Användare kan lämna en grupp
- Användare kan se alla grupper den är med i

## Admin

- Admin kan se en lista på alla användare, kurser respektive grupper och söka efter dessa i de tre olika listorna.
- Admin kan lägga till/ta bort kurser
- Admin kan lägga till/ta bort användare
- Admin kan lägga till/ta bort grupper
- Admin kan göra andra användare till admin/ta bort adminrättigheter
- Admin kan redigera alla kurser förutom nyckeln (kurskod)
- Admin kan redigera alla grupper förutom nyckeln (gruppnummer)
- Admin kan redigera alla användare förutom nyckeln (användarnamn)

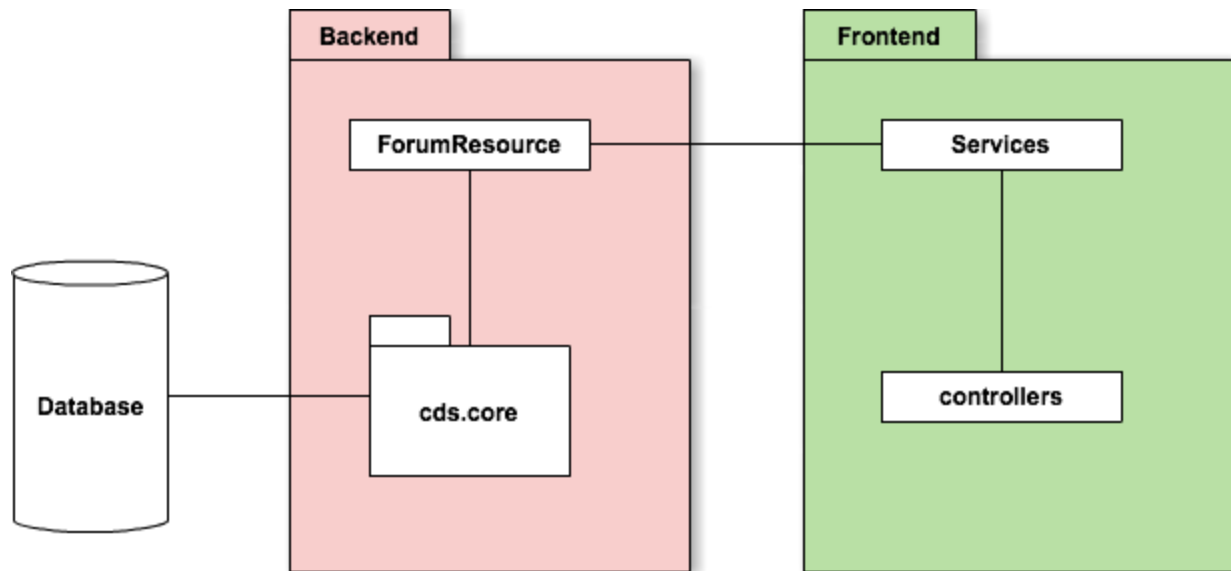
# Modell



## Selected approach

Vi har valt REST-ful, alltså service based approach som förekom i workshop 2 med Angularjs.

## Tiers



HTML-sidorna är inte med i diagrammet ovan, men självklart är dessa kopplade till kontrollern.

## Software components

### Front-end

- **JQuery:** Används för att enklare hantera DOMen och samtidigt underlätta AngularJS.
- **AngularJS:** Upprethåller en MVC-struktur för javascript. Underlättar strukturering av frontend
- **Bootstrap:** css/js-bibliotek för grafisk frontend design.

### Back-end

- **JavaDB:** Minimerar behovet av ren SQL och underlättar kommunikation med databasen genom att generera databastabeller utifrån entitetobjekt som utgörs av javaklasser.
- **Arquillian:** Underlättar testar av JavaDB-relaterad karaktär.

# Paket

## Front-end:

- **css:**  
Stylingen ligger här i en egen modul för att inte kladda ner mark-upen.
- **img:**  
De bilder som kommer att användas i utseendet. Just nu endast en hårdkodad bild för att visa kommande funktionalitet.
- **js:**  
Här ligger javascript-filerna. Dessa sköter navigeringen, REST-anrop, hämtning av Chalmers kurser och kontroller.
- **partials:**  
HTML-sidor, i olika paket beroende på område. Vi hoppas att namnen är självförklarande.

## Back-end

- **cds:**  
Mellansteg mellan services.js och resource-filen för att REST-anrop ska fungera, samt själva resource-filen..
- **cds.core:**  
De tre Entity-klasserna som ska sparas i databasen, samt övriga klasser och interface som behövs.
- **cd.persistence:**  
Superklasser för klasserna i core som tar hand om kontakten till databasen.

# Skikt vyer

## Model

Modellen består av våra entity klasser som används för att bygga upp databasen. Detta innehåller även alla delar i cds.core paketet.

## Persistence

Det är här JPA ramverket används för att koppla samman Modellen med databasen.

## **Service**

Vårt service layer består av services.js, ApplicationConfig.java och ForumResource.java då dessa används för kommunikation mellan controller på front-end och hanteringen av databasen på back-end.

## **Control**

Vår kontroller är javascriptet controller.js som används för hantering av ändringar på front-end.

## **View**

Här är alla html filer som används för att visa upp applikationens frontend.