

Projektrapport Webbapplikationer

Grupp 5, ADDE Inc.

Innehållsförteckning

[Innehållsförteckning](#)

[Gruppinformation](#)

[Beskrivning av projektet](#)

[Övergripande beskrivning](#)

[Skärmdump av applikationen](#)

[Användarroller](#)

[Use-cases](#)

[Teknisk design](#)

[Object Oriented model](#)

[Setup](#)

[Tekniska specifikationer](#)

[Backend](#)

[Frontend](#)

Gruppinformation

Eric Ahlberg

git: eahlberg

850829-4876

aheric@student.chalmers.se

Joakim Danielsson

git: JoakimDanielsson

900324-0398

joadan@student.chalmers.se

Gustav Davidsson

git: gdavidsson

921101-2951

gusdavi@student.chalmers.se

Hannes Elvemyr

git: ehannes

880917-5931

ehannes@student.chalmers.se

Gruppen nås vid behov via Joakim Danielsson på telefon 0706043524

Projektet finns på Github: <https://github.com/ehannes/webapp>

Beskrivning av projektet

Övergripande beskrivning

Projektet innebär att bygga en webbapplikation för projektarbeten, en projektplattform.

När ett projekt, vilket som helst, utförs innebär det förutom att utföra projekt målet även mycket arbete vid sidan av. Bland annat:

- Kommunikation mellan projektmedlemmar
- Sätta upp delmål och deadlines
- Skriva upp sådant som behöver göras
- Deligera arbetet till de olika personerna i projektet

Detta projekt har för avsikt att bygga ett verktyg för att underlätta underlätta ovanstående punkter under ett projektets gång och förse projektmedlemmarna dessa funktioner via ett webbgränssnitt. En webbapplikation är lättåtkomlig, användaren slipper installera något program vilket också innebär att plattformen är plattformsoberoende.

Tanken är att man kan registrera sig på en hemsida och sedan få tillgång till en plattform där projekt kan skapas. Användaren kan sedan bjuda in andra personer till projektet och de kan sätta igång att arbeta tillsammans.

I ett projekt ska man kunna skapa artiklar som i en liten wiki. Dessa har information om vilka som ändrat artikeln och när - denna information lagras på databasen som objekt av klassen ArticleEdit. Man ska också kunna skapa Todo's, som innehåller ett meddelande. De kan även innehålla en deadline och en prioritet (low, medium, high). Det finns också möjlighet att skapa wallposts, likt de på facebook. Dessa wallposts innehåller bara ett meddelande, men man kan lägga till posts som kommentarer på en wallpost. Posts innehåller också bara ett meddelande, likt wallposts, men man kan inte kommentera på en vanlig post.

Skärmdump av applikationen

Project Platform

[Home](#)[About](#)[Contact](#)[Sign up](#)[Log in](#)

Welcome!

Create projects and work together to get more productive!

Sign up!

Project Platform

[Home](#)[About](#)[Contact](#)[Sign up](#)[Log in](#)

Sign up

Email

Password

Repeat Password

Log in

Log in

☐ **Remember**{Project name} [All users](#) [All projects](#)

Options ▾

Successfully logged in!

{Project name} [All users](#) [All projects](#) [All articles](#) [All wallposts](#) [All todoposts](#)

Options ▾

Username	Email
user0	user0@user0.com
user1	user1@user1.com
user2	user2@user2.com
user3	user3@user3.com
user4	user4@user4.com
user5	user5@user5.com
user6	user6@user6.com
user7	user7@user7.com
user8	user8@user8.com
user9	user9@user9.com
gustav	gustav@adde.com

Show all projects

Project name	Id	Admin
projekt	2013-10-21T19:11:41.068+02:00	apan
projekt2	2013-10-21T19:11:47.420+02:00	apan

Show all articles

titel

hej

Last edited: 2013-10-21T19:16:08.598+02:00

Detta är ett test

Hejhej detta är en artikel. Den kan man redigera och annat roligt, och vem som ändrat artikeln och när sparas.

Last edited: 2013-10-21T19:26:01.694+02:00

Användarroller

Alla användare kan fritt skapa och ta bort egna projekt. När en användare skapar ett projekt blir han automatiskt *administratör* för det projektet. När ett projekt väl är skapat kan andra användare bjudas in. Dessa blir då *projektmedlemmar*. Det som skiljer administratören från de vanliga användarna är att denne kan ta bort projektet och även ge över administrationsrättigheten till någon annan.

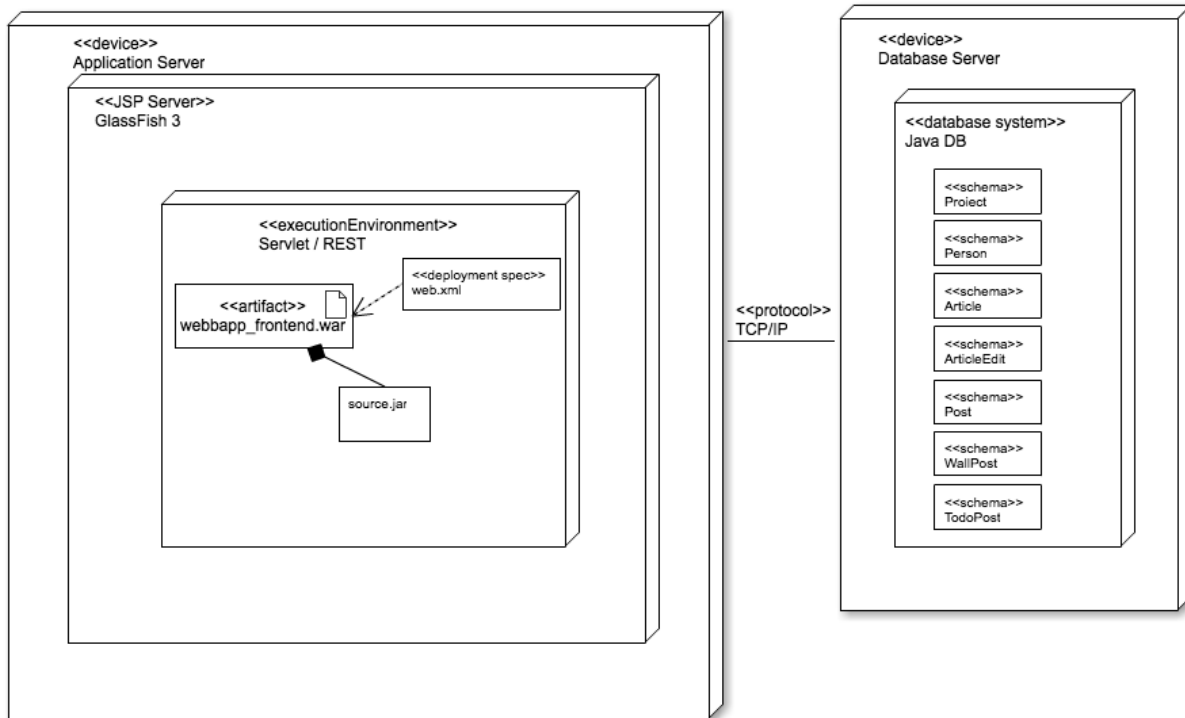
Use-cases

Tanken är att plattformen kan användas i projektarbeten. Vi tror oss ha byggt en bra och stabil grund för en projektplattform. Modellen stödjer många features och är relativt avancerad. Exempelvis stödjer den följande funktioner:

- En användare kan skapa ett projekt
- Projekt kan i sin tur innehålla artiklar (Wiki) och en “todo”-lista (att-göra-lista).
 - Artiklar finns till för att användare ska kunna samla viktig information om projektet
 - Artiklar sparar data om vem som gör ändringar i artikeln och även när.
 - Att-göra-listan gör att användarna kan skriva upp sådant som behöver göras i projektet. Varje todo kan ges till en specifik person/personer så att arbetet lätt kan distribueras. Dessa kan ha en deadline samt en prioritet.
- Projektet har också en sorts diskussionssida (wall, hämtat från Facebook wall). Här kan användarna lätt kommunicera med varandra. Man kan kommentera på ett inlägg i projektets wall.

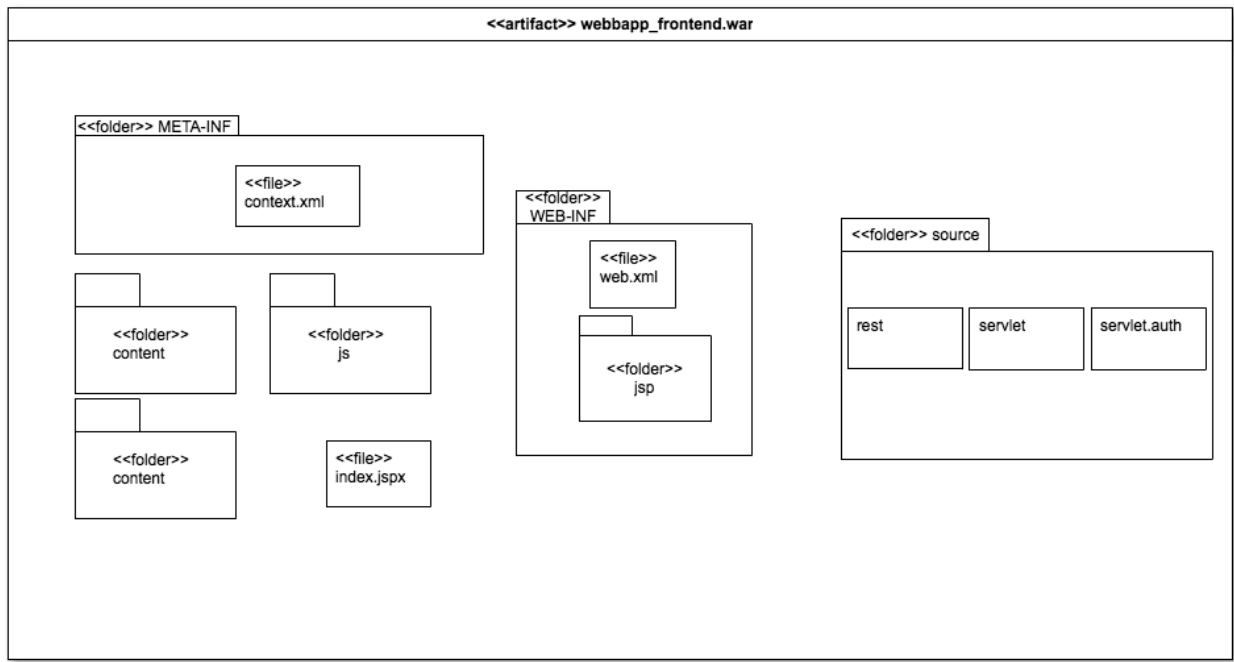
Teknisk design

Fysisk setup



Tekniska specifikationer

Participating software components



Moduler

Backend

Backenden av applikationen heter `webbapp_model`. Den består av två moduler.

Denna ena modulen kallas `model.entity` och innehåller klasserna som ska sparas på databasen.

Den andra kallas `model.dao` och ansvarar för kommunikationen med databasen. Detta sker via DAO:s (Data Access Object), eller kataloger som de heter i detta projekt. Varje entity (klass som ska sparas på databasen) har en motsvarande katalog. Katalogerna anropar databasen och hämtar/sparar det som användare önskar.

Frontend

Front-enden består av 3 moduler. Två av dem är `servlet`, `frontend.servlet` och `frontend.servlet.auth`. Den första hanterar de sidorna en användare som inte är inloggad kan se, såsom förstasida och inloggningssida. Dessa sidor körs via Java Servlet och JSPX-filer. `servlet.auth` sköter inloggningen. Det var även tänkt att ett filter skulle filtrera anropen som går till den delen av hemsidan som man måste vara inloggad på, men tyvärr har vi inte lyckats lösa så att det fungerar. Dock går det bra att registrera sig och logga in på sidan.

Då en användare försöker logga in skickas anropet till en `authentication-servlet` som verifierar användarnamn och lösenord. Finns användaren och lösenordet är korrekt kan användaren logga

in.

Då användaren väl är inloggad används en REST-service. Denna modul kallas frontend.rest. I mappen "content/rs" finns alla resurser som en användare kan nyttja.

För GUI används JavaScript. Denna del av projektet hann pga våra problem med EntityManager tyvärr inte bli färdigställd, så den är inte som vi hoppades att den skulle vara.

Applikationens olika lager

GUI: xhtml, JavaScript, Java Server Pages

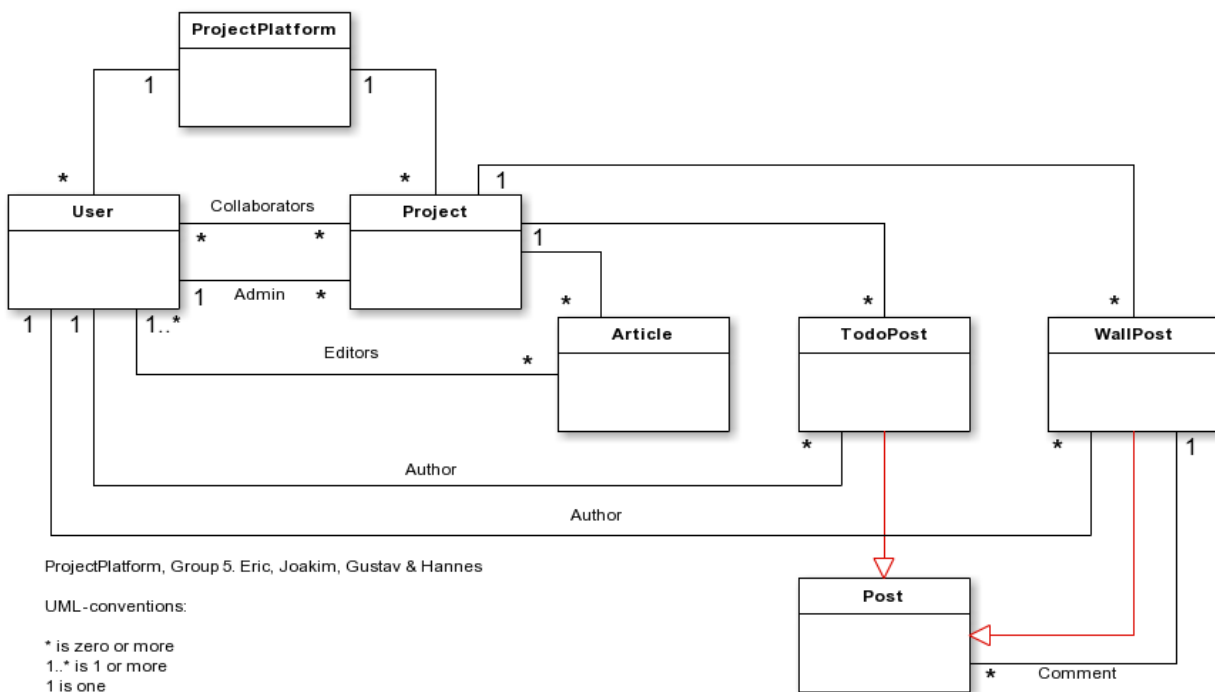
Applikationslager: Servlet

Services: REST

Modell: Java

Persistence: JPA

Object Oriented model



UML-diagram över modellen. Efter anpassning av modellen till databasen togs bas-klassen ProjectPlatform då den inte behövdes längre. Den finns dock med i front-end för att koppla ihop JavaScript med varandra. Därav finns den fortfarande med i diagrammet.

Klasser

Koden är kommenterad där det anses behövas. Viktiga klasser har klasskommentarer.

