

| | | | | | |
|---------------------------|-------------|-----------------------------|----------------------|-------------|----------------------------|
| Kristoffer Skjutar | 920129-3371 | skjutar@student.chalmers.se | Lisa Stenberg | 900212-5244 | lisast@student.chalmers.se |
| Madeleine Appert | 891110-4845 | appertm@student.chalmers.se | Gustav Dahl | 920113-1811 | gdahl@student.chalmers.se |

Projekt PM
-för kursen DAT076

Projekt Ove

Group 12

<https://github.com/skjutar/OVE>
git@github.com:skjutar/OVE.git

| | | | |
|---------|---------------------|--------------|-----------------------------|
| skjutar | Kristoffer Skjutar, | 920129-3371, | skjutar@student.chalmers.se |
| malla | Madeleine Appert, | 891110-4845, | appertm@student.chalmers.se |
| gdahl92 | Gustav Dahl, | 920113-1811, | gdahl@student.chalmers.se |
| lisast | Lisa Stenberg | 900212-5244, | lisast@student.chalmers.se |

Kristoffer Skjutar 920129-3371
Madeleine Appert 891110-4845

skjutar@student.chalmers.se
appertm@student.chalmers.se

Lisa Stenberg
Gustav Dahl

900212-5244
920113-1811

lisast@student.chalmers.se
gdahl@student.chalmers.se

Ove

Chalmers

Läxhjälsprojekt drivs av
och sysselsätter ca 25

Chalmerister, som
bedriver läxhjälp på
högstadieskolor i

Göteborgsområdet. Ove

är en webapplikation

utformad främst för att
effektivisera arbetet för

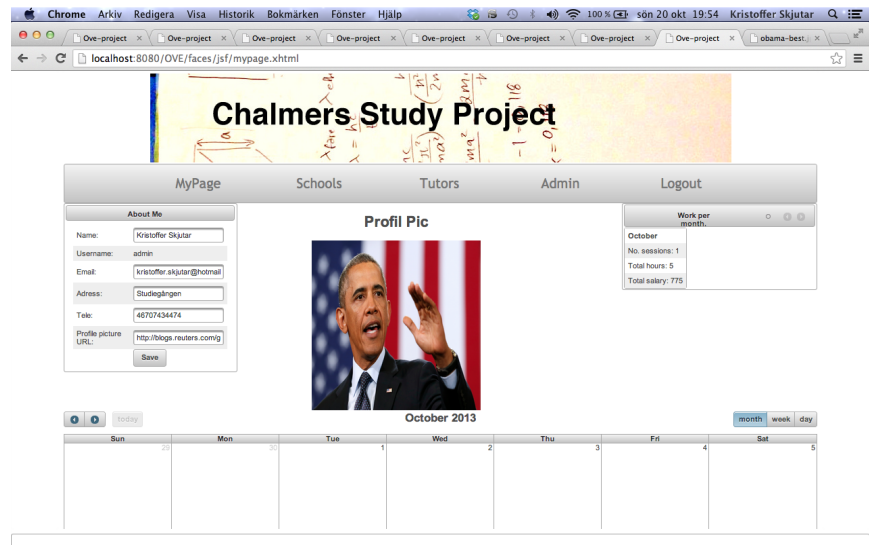
ledningen av projektet,

men även för att ge alla handledare en god översikt över deras respektive verksamhet.

Applikationen erbjuder ledningen en möjlighet att organisera kontaktinformation till både

handledare och kontaktpersoner, samt att hålla en god överblick över hur arbetsschemat både

har sett ut, samt ser ut i framtiden. För handledare, blir det möjligt att se sitt arbetsschema, samt
se hur mycket lön hen kan väntas få varje månad.



Som användare av applikationen behöver man ha ett användarkonto. Det finns två

användarroller, antingen är man administratör eller så är man handledare. Som administratör

har man en utökad funktionsmöjlighet, och redigeringsförmåga. See Use Cases nedan för mer

detaljerad information.

Use Cases

Både Admin och Handledare.

- Skapa konto - som administratör eller vanlig handledare.

Kontotyp väljs vid skapandet av kontot. Ett mail skickas från den emailadress som är förknippad med Glassfish servern och ber användaren att konfirmera kontot.

- Logga in (förutsatt att man har ett konto).
- Se information som rör den inloggande användaren genom att trycka på fliken My Page.

Så som:

| | | | | | |
|---------------------------|-------------|-----------------------------|----------------------|-------------|----------------------------|
| Kristoffer Skjutar | 920129-3371 | skjutar@student.chalmers.se | Lisa Stenberg | 900212-5244 | lisast@student.chalmers.se |
| Madeleine Appert | 891110-4845 | appertm@student.chalmers.se | Gustav Dahl | 920113-1811 | gdahl@student.chalmers.se |

- Personlig information: Namn, email, telefonnummer et.c
- Profilbild
- Lönespecifikation för olika månader.
- Schema över stundande månad. Genom att navigera sig med knapparna ovanför kan man se andra månader.
- Ändra information som rör den inloggande användaren. Så som:
 - Personlig information: Personnummer, email, telefonnummer et.c
 - Profilbild
- Se en lista på alla skolor genom att klicka på fliken Schools.
- Se detaljerad information i en ny vy, genom att klicka på en skola i listan.
- Lägga till en kontaktperson för en skola.
- Se detaljerad information beträffande en kontaktperson.
- Se en lista på alla handledare genom att trycka på fliken Tutors.
- Se detaljerad information om en handledare.
- Logga ut

Bara Admin

- Lägga till en skola i systemet, genom att klicka på Add school, och fylla i dialogrutan.
- Ändra skolans information.
- Ta bort en kontaktperson till en skola.
- Lägga till en handledare genom att klicka på Add New Tutor.
- Ta bort en handledare genom att trycka på Delete när man ser detaljerad information om en handledare.
- Se alla inlagda handledarpass i en kalender med vymöjligheten månad, vecka, dag genom att trycka på fliken Admin.
- Lägga till nya handledarpass genom att klicka på kalendern.
- Ändra information om ett handledarpass genom att trycka på det, ändra informationen och klicka på Save.
- Ta bort ett handledarpass genom att trycka på det och klicka på Delete.
- Se alla handledares löner genom att trycka på Salaries.
- Ändra en handledares lön genom att trycka på lönen, ändra den och sedan klicka någon annanstans på sidan.

Kristoffer Skjutar 920129-3371
Madeleine Appert 891110-4845

skjutar@student.chalmers.se
appertm@student.chalmers.se

Lisa Stenberg
Gustav Dahl

900212-5244 920113-1811
lisast@student.chalmers.se
gdahl@student.chalmers.se

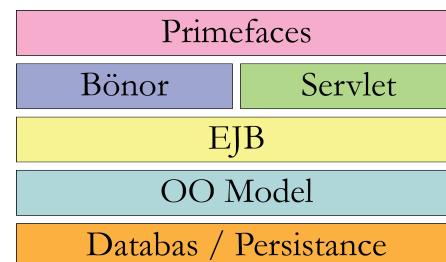
Teknisk Dokumentation

Mjukvarukomponenter

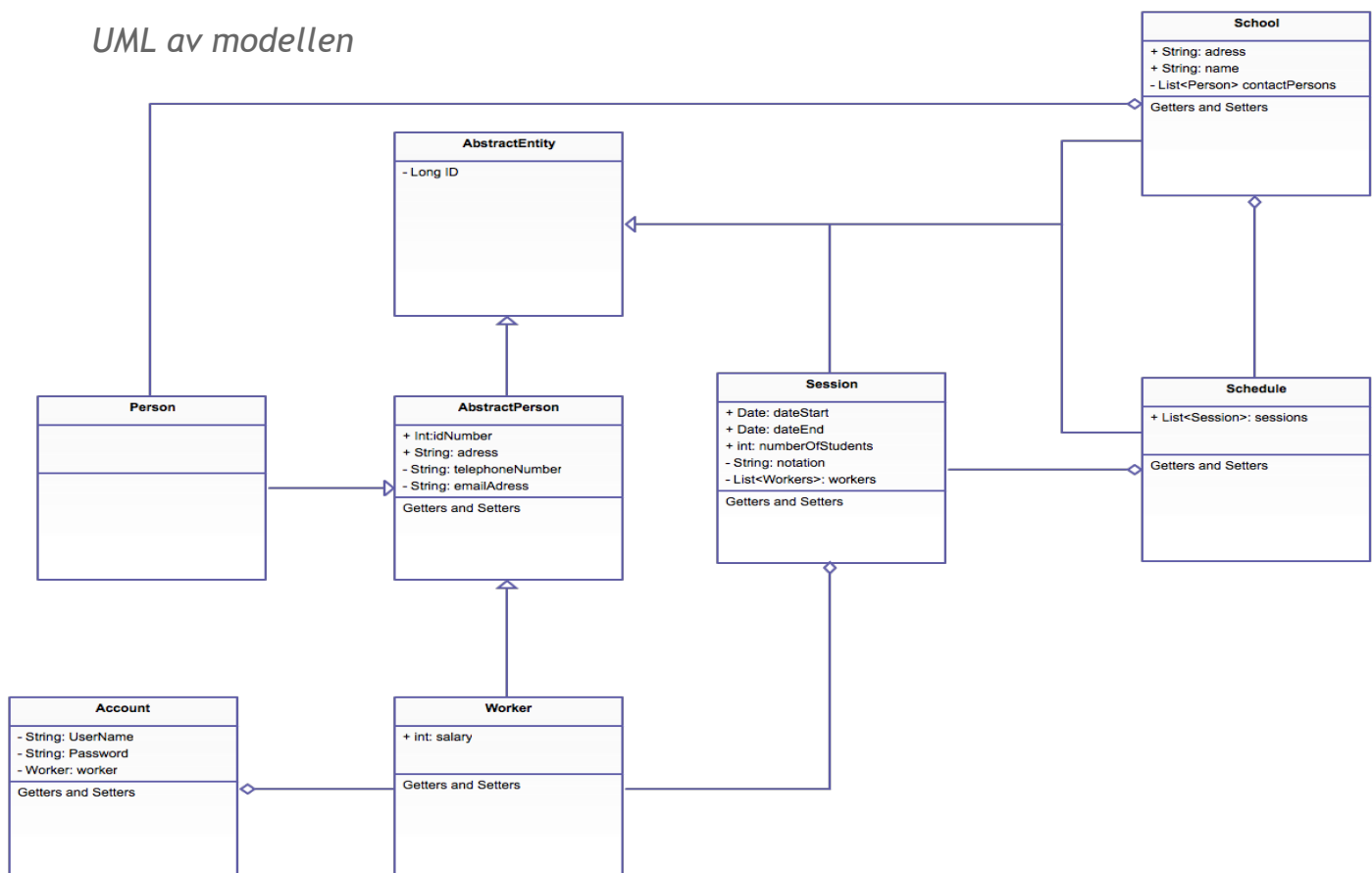
Vi har använt oss två komponenter, webbservern Glassfish och databasen Derby. Webbservern är ansvarig för hemsidan medan Derby är ansvarig för att lagra data. Webbservern består av av tre komponenter; frontenden, backenden samt Glassfish integrerande JavaMail Service.

Moduler

Frontenden består av JSF eller mer specifikt påbyggnaden PrimeFaces. De tekniker som används här är dels JSF som redan nämnts men också AJAX som är integrerat i Primefaces. Backenden består av "Component Based Approach" med bönor samt EJB emellan databasen och bönorna. Vi har också en ensam Servlet som lyssnar på aktiveringar av konton till hemsidan på en specifik url.



UML av modellen



| | | | | | |
|---------------------------|-------------|-----------------------------|----------------------|-------------|----------------------------|
| Kristoffer Skjutar | 920129-3371 | skjutar@student.chalmers.se | Lisa Stenberg | 900212-5244 | lisast@student.chalmers.se |
| Madeleine Appert | 891110-4845 | appertm@student.chalmers.se | Gustav Dahl | 920113-1811 | gdahl@student.chalmers.se |

Kommentarer och highlights.

Skolsidorna

När en användare vill se en detaljerad vy av en skola, är hen 'redirectad' till den sidan via en URL i metoden `onRowSelect` i `SchoolListBean` klassen. För applikationen att veta vilken skola den ska visa upp, läser `SchoolPageBean` från `URLens` fält 'school'. Vi kunde inte komma på en alternativ lösning till detta som fungerade, därför har vi hållit oss till detta.

När användaren lägger till en ny skola till applikationen och listan, körs ett javascript på `uponcomplete`, som bland annat ber sidan att uppdatera sig själv. Det finns alternativa lösningar till detta, men i detta läge var det en enkel lösning som uppfyllde allt vad vi ville åstadkomma efter tillägget.

Vi hade problem med att sätta ett lämpligt scope på `SchoolPage`. Till en början hade vi satt `Sessionscoped`, och alla våra funktioner fungerade fint. Men vi upptäckte att det då inte gick att byta till en detaljerad vy av en annan skola, då informationen om den gamla stannade kvar. Att byta till `RequestScoped` var inte heller ett alternativ då det inte gick att uppdatera informationen om skolan och många funktioner gick i kras. Vi försökte, men lyckades inte implementera `ConversationScoped`. Därför har vi hållit oss till `SessionScoped`, och kringgått de problem vi hade med att visa fel skola, genom en tom sträng 'check' som vid dess getter, kontrollerar att den aktiva skolan stämmer överens med `URLen`.

MailAktivering

MailAktiveringen startas när användaren lägger till sitt konto genom `CreateAccountBean`. Bönan ber då `MailBean` som instansieras via EJB att genom användarens nu helt unika ID från databasen samt det nya kontots mailadress att skicka ett mail till användaren. Genom att utnyttja Glassfish inbyggda `JavaMail Service` kan vi då genom `MailBean` skicka iväg ett mail till den angivna adressen med det unika ID:t som token i URL:en som användaren får via mailet. När användaren klickar på den givna URL:en i mailet fångar vår `MailServlet` upp requestet till sidan och aktiverar kontot i databasen med det token som användaren anlände med. För grafisk visning, se flowcharts längre bak i dokumentet.

Modellklasser för Primefaces

Då komponenter i Primefaces använder sig av klasser från sitt egna API för visning av data i dess komponenter var vi tvungna att implementera egna klasser som passade våra behov. Ett exempel är `ScheduleEvent` som representerar ett event (i vårt fall en session) i den grafiska representationen.

| | | | | | |
|---------------------------|-------------|-----------------------------|----------------------|-------------|----------------------------|
| Kristoffer Skjutar | 920129-3371 | skjutar@student.chalmers.se | Lisa Stenberg | 900212-5244 | lisast@student.chalmers.se |
| Madeleine Appert | 891110-4845 | appertm@student.chalmers.se | Gustav Dahl | 920113-1811 | gdahl@student.chalmers.se |

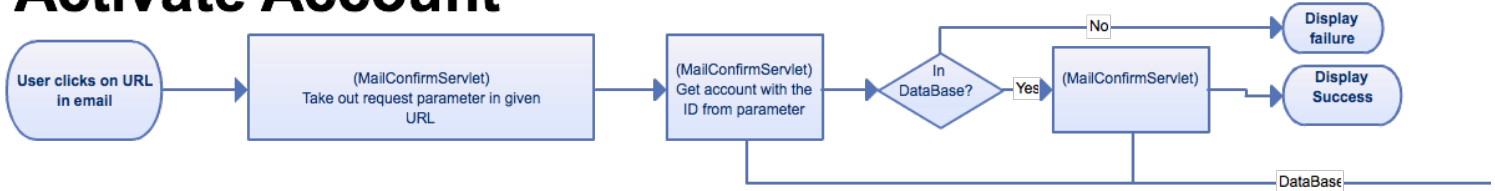
AuthorizationListener

Denna PhaseListener använder vi som säkerhet för vår sida. Det möjliggör vi genom att hela tiden hålla koll på den sessionen som pågår samt att undersöka vart användaren är på väg. På grund av att vi sparar information i sessionen kan vi genomföra kontroller om användaren ska ha tillgång till administrationsfunktioner. Vi förhindrar också användaren att kringgå loginfunktionen på detta vis. För grafisk representation, se flowcharts i slutet av dokumentet.

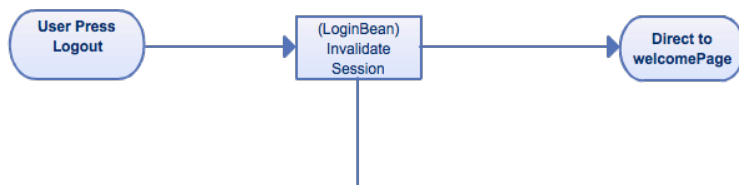
TestBean

Vi hade lite problem att skapa testklasser för våra databasregister där vi försökte på båda det sättet som länkades på kurshemsidan. Men i mån av tid, valde vi att göra en lite mer primitiv lösning. Vi gjorde en klass som vi kallade TestBean där vi testar add, remove och update för de flesta register, där en räknare ökas om den klarar testet och jämförs med antal test den testas med. För att köra dessa test gjorde vi en xHTML sida som inte kan nås såvida du vet dess url, där en knapp finns som kör alla test. Där finns också en progressbar som visar hur många procent av testen som klarades.

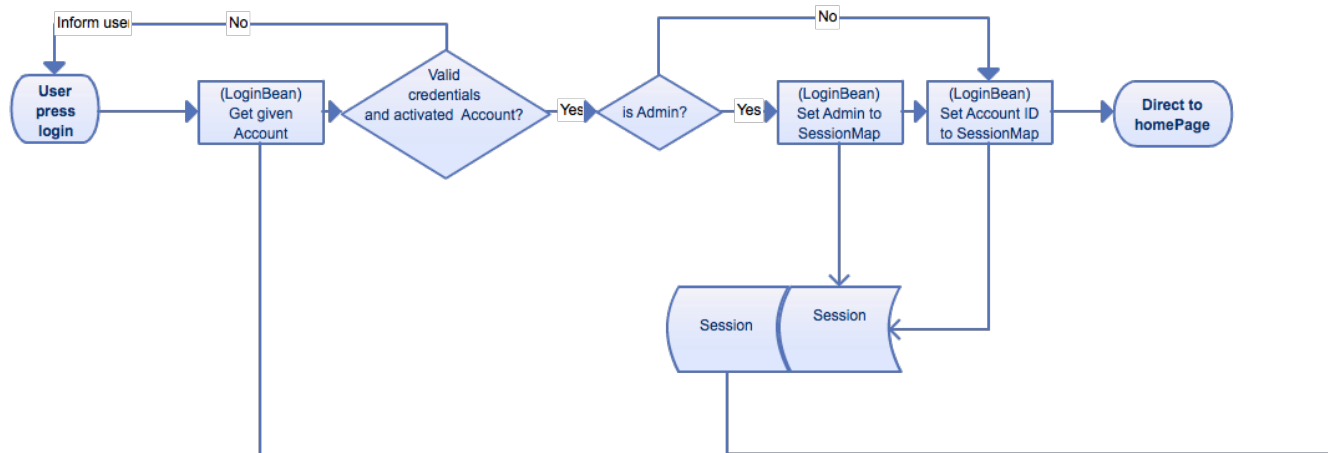
Activate Account



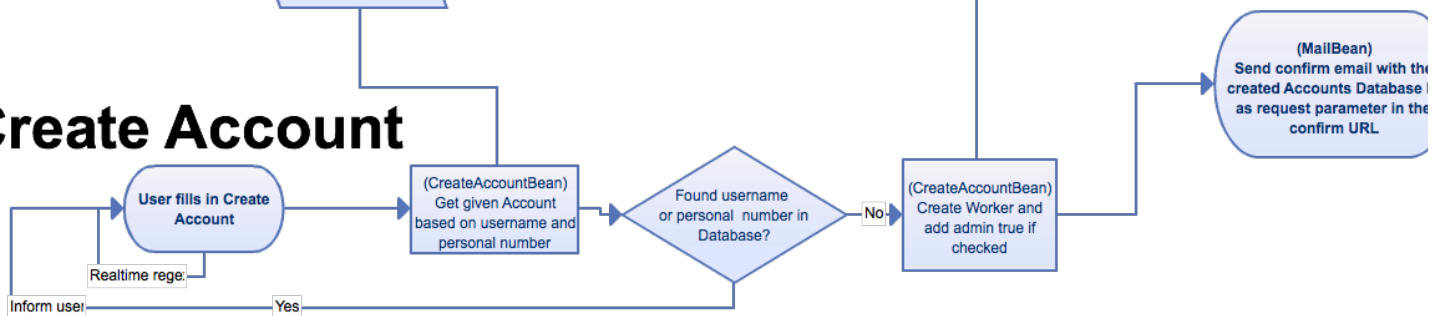
Logout



Login



Create Account



Page Navigation

