

PUSP1742##

v.##

#DOKUMENTMALL#
ETSF20 GRUPP 2

Ansvarig Grupp: #(PG) Projekt Grupp#
Uppgjord Av: #(PG) Projekt Grupp#

1 mars 2017

Innehåll

1 Introduktion

Detta dokument beskriver högnivådesignen för ett tidrapporteringsystem, baserat på "BaseBlockSystem", som är ett system med inloggnings-funktionalitet på en webbserver. Användare i detta tidrapporteringsystem kan rapportera sin arbetade tid för respektive projekt de ingår i. Alla tider som rapporteras sparas i databasen och kan sedan sammanställas av en projektledare.

Avsikten med systemet är att projektledarna ska kunna få en överblick över arbetad tid i projektet."KOLLA UPP"

2 Referensdokument

- SRS PUSP174212 version: 0.2
- BaseBlockSystem STLDD: PUSS1204 version: 1.0 gäller för alla punkter om det inte är specificerat i respektive underrubrik att det som står i PUSS412004 version: 1.0 utgår för specifik del.

3 Terminologi

4 Översikt

Systemet är utvecklad för applikationsservern Apache Tomcat, där Java Servlet, JavaServer Pages (JSP) samt Java Expression Language (JSP EL)-teknologier tillämpas enligt JavaEE:s 2:a MVC-modellarkitektur. Systemets huvudfunktion är att fungera som ett tidrapporterings-system.

4.1 Vynivå: JSP (JavaServer Pages)

Nedan förklaras vad varje fil för JSP:n har för betydelse.

4.1.1 login.jsp

Denna fil innehåller vy-design som representerar inloggningsformuläret på förgrunds nivå, d.v.s. för den webbsida som är publikt tillgänglig utan inloggning. Denna webbsida används för att komma åt bakgrundsnivån för tidrapporteringsapplikationen. Designen består utav en inloggningsfält för användarnamn, ett för lösenord samt en rullista över val av projektgrupper. Kontrollen för denna vy är **class LoginServlet**.

4.1.2 groupmanagement.jsp

Filen innehåller vy-design som representerar projektgrupphanteringen för webbapplikationen som används av administratören. Här finns ett formulärfält för att lägga till en ny projektgrupp, samt en lista över vilka projektgrupper som existerar i systemet. I denna listan finns funktionen att ta bort aktuell projektgrupp från systemet. Kontrollen för denna vy är **class GroupManagementServlet**.

4.1.3 usermanagement.jsp

Vy-designen för användarhantering representeras av denna fil. Vyn används av administratör och projektledare där vyn anpassas utifrån användarens behörigheter utifrån dess roll i systemet. Kontrollen för denna vy är **class UserManagementServlet**.

4.1.4 reportmanagement.jsp

Filen innehåller vy-designen som representerar hantering av veckorapporter. Endast projektledaren har tillgång till denna webbsida. Kontrollen för denna vy är **class ReportManagementServlet**.

4.1.5 report.jsp

Filen innehåller vy-designen som representerar skapandet av veckorapporter. Projektledare och användare kommer använda denna webbsida för veckorapportering. Automatiserad signering av projektledarens veckorapportering sker på modellnivå när veckorapporten skickas in till servern. Kontrollen för denna vy är **class ReportServlet**.

4.1.6 dashboard.jsp

Vy-design som representerar sammanställning av tidrapporter som renderas ut på sidan. I denna vy väljs även hur sammanställningen ska presenteras genom att välja olika renderingsformer av sammanställningen. Kontrollen för denna vy är **class DashBoardServlet**.

4.1.7 user.jsp

I denna vy-design ser användaren sin personliga information, och kan ändra sitt lösenord. Kontrollen för denna vy är **class UserServlet**.

4.2 Kontrollernivå: Servlet

Klasserna som specificeras nedan agerar som kontroller i systemet.

4.2.1 LoginServlet

Denna kontrollern tar hand om kommunikationen mellan **login.jsp** och modellen **LoginBean**, som innehåller information om en given projektgrupp. Denna **LoginBean** skickas som en lista till **login.jsp** för att rendera ut valen av projektgrupp vid inloggning.

URL-pattern: ./

4.2.2 GroupManagementServlet

Tar hand om kommunikationen mellan **groupmanagement.jsp** och bönan **GroupManagementBean**.

URL-pattern: ./management/groups

4.2.3 UserManagementServlet

Tar hand om kommunikationen mellan **usermanagement.jsp** och bönan **UserManagementBean**.

URL-pattern: ./management/users

4.2.4 ReportManagementServlet

Tar hand om kommunikationen mellan **reportmanagement.jsp** och bönan **ReportManagementBean**.

URL-pattern: ./management/reports

4.2.5 ReportServlet

Tar hand om kommunikationen mellan **report.jsp** och bönan **ReportBean**.

URL-pattern: ./reports

4.2.6 DashboardServlet

Tar hand om kommunikation mellan **dashboard.jsp** och bönan **DashboardBean**.

URL-pattern: ./dashboard

4.2.7 UserServlet

Tar hand om kommunikationen mellan **user.jsp** och bönan **UserBean**.

URL-pattern: ./settings/user

4.3 Modellnivå: JavaBeans

Klasserna som specificeras nedan kommer skickas till vynivån i systemet.

4.3.1 LoginBean

En get/set-klass innehållandes data som krävs av **login.jsp** för att rendera vyn.

4.3.2 GroupManagementBean

En get/set-klass innehållandes data som krävs av **groupmanagement.jsp** för att rendera vyn.

4.3.3 UserManagementBean

En get/set-klass innehållandes data som krävs av **usermanagement.jsp** för att rendera vyn.

4.3.4 ReportManagementBean

En get/set-klass innehållandes data som krävs av **reportmanagement.jsp** för att rendera vyn.

4.3.5 ReportBean

En get/set-klass innehållandes data som krävs av **report.jsp** för att rendera vyn.

4.3.6 DashboardBean

En get/set-klass innehållandes data som krävs av **dashboard.jsp** för att rendera vyn.

4.3.7 UserBean

En get/set-klass innehållandes data som krävs av **user.jsp** för att rendera vyn.

4.4 Övriga klasser på modellnivå

Klasser som specificeras nedan skickas aldrig till vynivån.

4.4.1 Database

Hanterar databasuppkopplingen genom designmönstret singleton.

4.4.2 DatabaseHandler

Hanterar SQL-förfrågningar direkt mot databasen.

4.4.3 BeanFactory

Matar in information från databasen till bönorna.

4.4.4 BeanUtilities

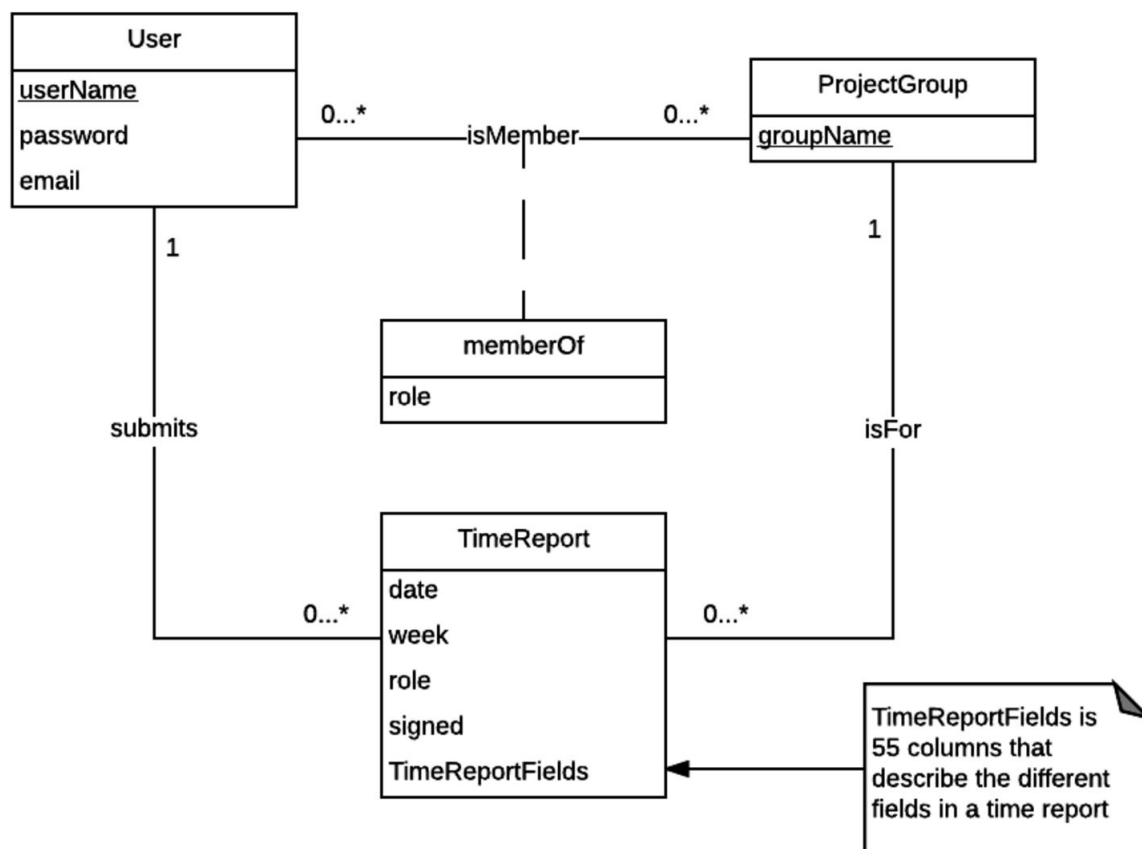
Matar in information från HTTP-förfrågan ("request") till bönorna.

4.4.5 BeanTransaction

Extraherar bönans information och utför databaskommunikation genom **class DatabaseHandler**.

5 Databas

Databasen består av tre huvudsakliga tabeller Users, ProjectGroups och TimeReports. Den har även en relationstabell memberOf som beskriver relationen mellan Users och ProjectGroups.



5.1 Users

Users(userName, password, email) Users är en tabell som innehåller användare som är inlagda i systemet. Den består av tre kolumner: userName, password och email. userName är primary key vilket innebär att det endast kan finnas en användare med det användarnamnet. Tabellen har följande utseende:

Field	Type	Null	Key	Default	Extra
userName	varchar(10)	NO	PRI	NULL	
password	varchar(6)	NO		NULL	
email	varchar(100)	NO		NULL	

Tabellen kan konstrueras med följande SQL-satser:
 mysql> CREATE TABLE Users (

```
-> userName varChar(10) NOT NULL,
-> password varChar(6) NOT NULL,
-> email varChar(100) NOT NULL,
-> PRIMARY KEY (userName));
```

5.2 ProjectGroups

ProjectGroups(groupName)

ProjectGroups är en tabell som innehåller de olika projektgrupperna i systemet. Den innehåller endast en kolumn, groupName, som är satt som primary key"vilket innebär att alla projektgrupper måste ha unika namn.

Tabellen har följande utseende:

Field	Type	Null	Key	Default	Extra
groupName	varchar(100)	NO	PRI	NULL	

Tabellen kan konstrueras med följande SQL-satser:

```
mysql> CREATE TABLE ProjectGroups (
-> groupName varChar(100),
-> PRIMARY KEY (groupName));
```

5.3 TimeReports

TimeReports är en tabell som innehåller all information som rör tidrapporterna. Då tabellen har många kolumner (65 st) visas den inte här. För att få ut relevant information ur tidrapporterna så finns det en hjälp-procedure som tar fram relevant information för tillfällen då detta kan kräva många rader kod. För enklare fall så räcker det med en vanlig SELECT. Det finns även två triggers, en för när man lägger in en ny rad, och en när man uppdaterar en existerande rad, som väljer rätt roll för användaren, uppdaterar de totala värdena för aktiviteterna med subaktiviteter (11-19), uppdaterar den totala tiden spenderad för subaktiviteterna (d, i, f och r) och den totala tiden rapporten avser.

5.4 memberOf

memberOf(groupName, member, role)

memberOf är en relationstabell som beskriver relationen mellan användare och de olika projektgrupperna. Den har tre kolumner: groupName, member och role. Kolumnerna groupName och member är tillsammans satta som UNIQUE, vilket resulterar i att en medlem endast kan vara medlem i samma grupp en gång (en roll per projektgrupp). groupName är en foreign keyssom refererar till **ProjectGroup(groupName)** och member är en foreign keyssom refererar till **Users(userName)**.

Tabellen har följande utseende:

Field	Type	Null	Key	Default	Extra
groupName	varchar(100)	YES	MUL	NULL	
userName	varchar(10)	YES	MUL	NULL	
role	varchar(10)	YES		NULL	

Tabellen kan konstrueras med följande SQL-satser:

```
mysql> CREATE TABLE memberOf (  
-> groupName varChar(100),  
-> member varChar(10),  
-> role varChar(10),  
-> UNIQUE (groupName, member),  
-> FOREIGN KEY (groupName) references ProjectGroups(groupName) ON UPDATE CASCA-  
DE  
-> ON DELETE CASCADE,  
-> FOREIGN KEY (member) REFERENCES users(userName) ON UPDATE CASCADE ON  
DELETE CASCADE);
```

6 Klassdiagram

Ett klassdiagram för Java-klasserna hittas i Appendix A.

7 Information som lagras i sessionen

Under varje session lagras följande attribut i sessionen utöver det som redan beskrivs i STLDD:n för BaseBlockSystem:

Long lastActive: används för att hålla reda på när användaren senast var aktiv i servern.

Följande gäller:

lastActive => 30: Användaren har inte varit aktiv i servern under de senaste 30 minuterna och blir därmed utloggad.

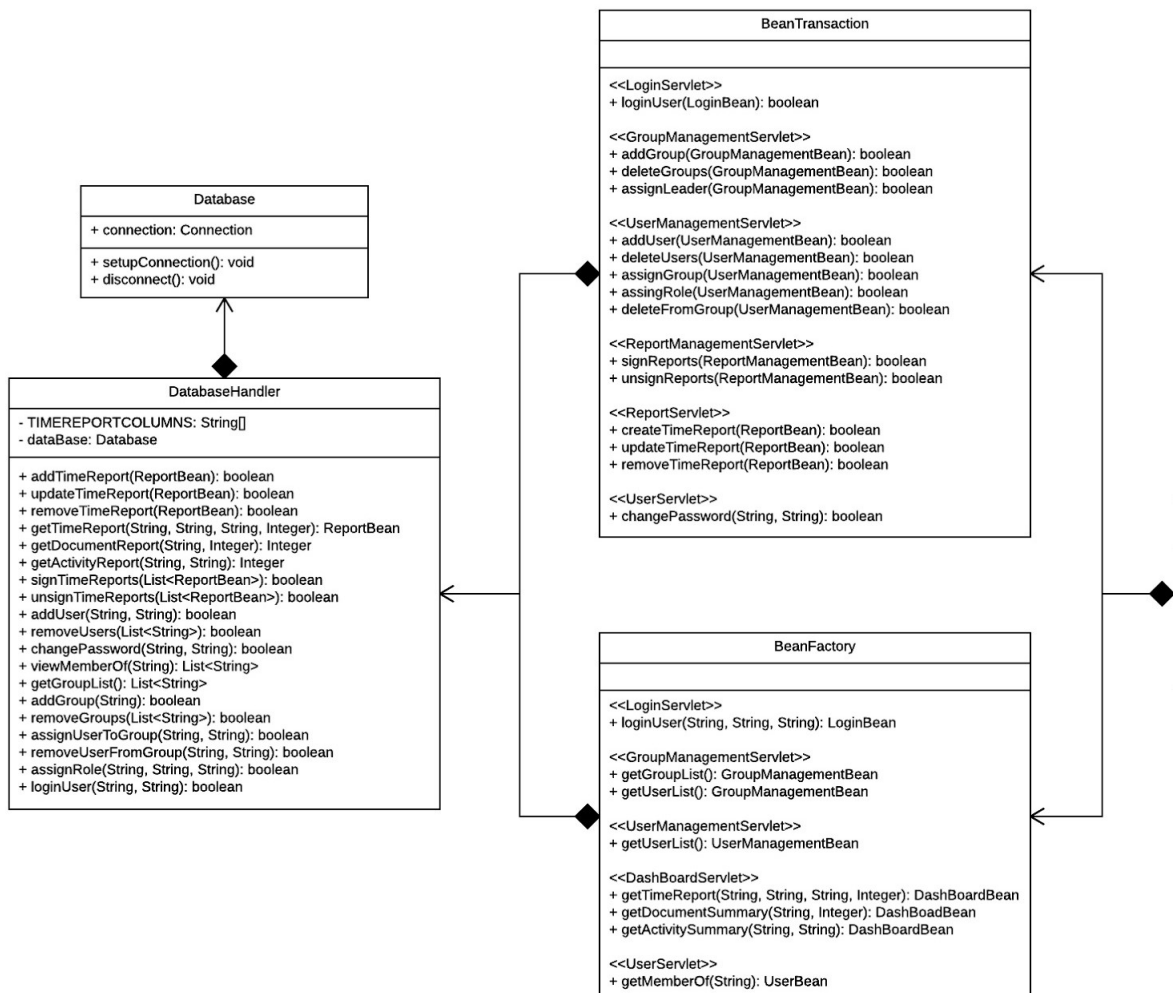
String group: Gruppnamnet för gruppen användaren valt vid inloggning, tex. 'Grupp 2'

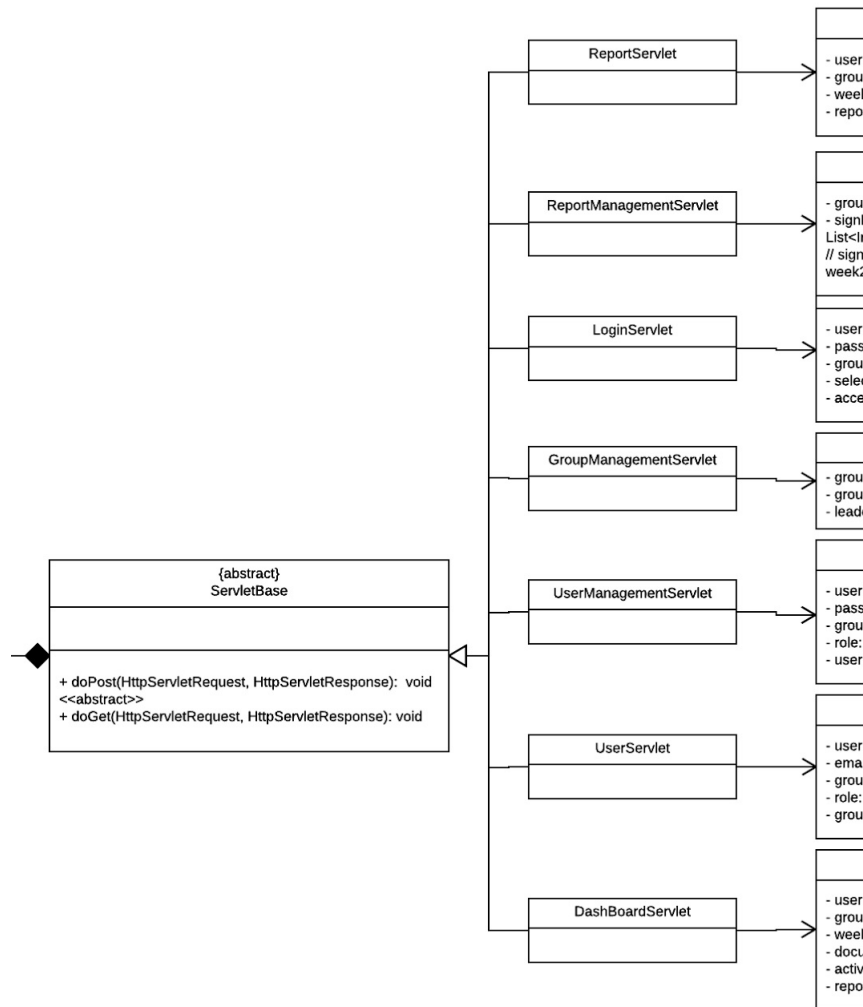
8 Sekvensdiagram

9 Routing

10 Appendix A

Klassdiagrammet för Java-klasser. Diagrammet är delat på mitten, så slutstrecket i fig. A.1 fortsätter i fig. A.2.





Figur A.1. Första halvan i klassdiagrammet.

Figur A.2. Andra halvan i klassdiagrammet.