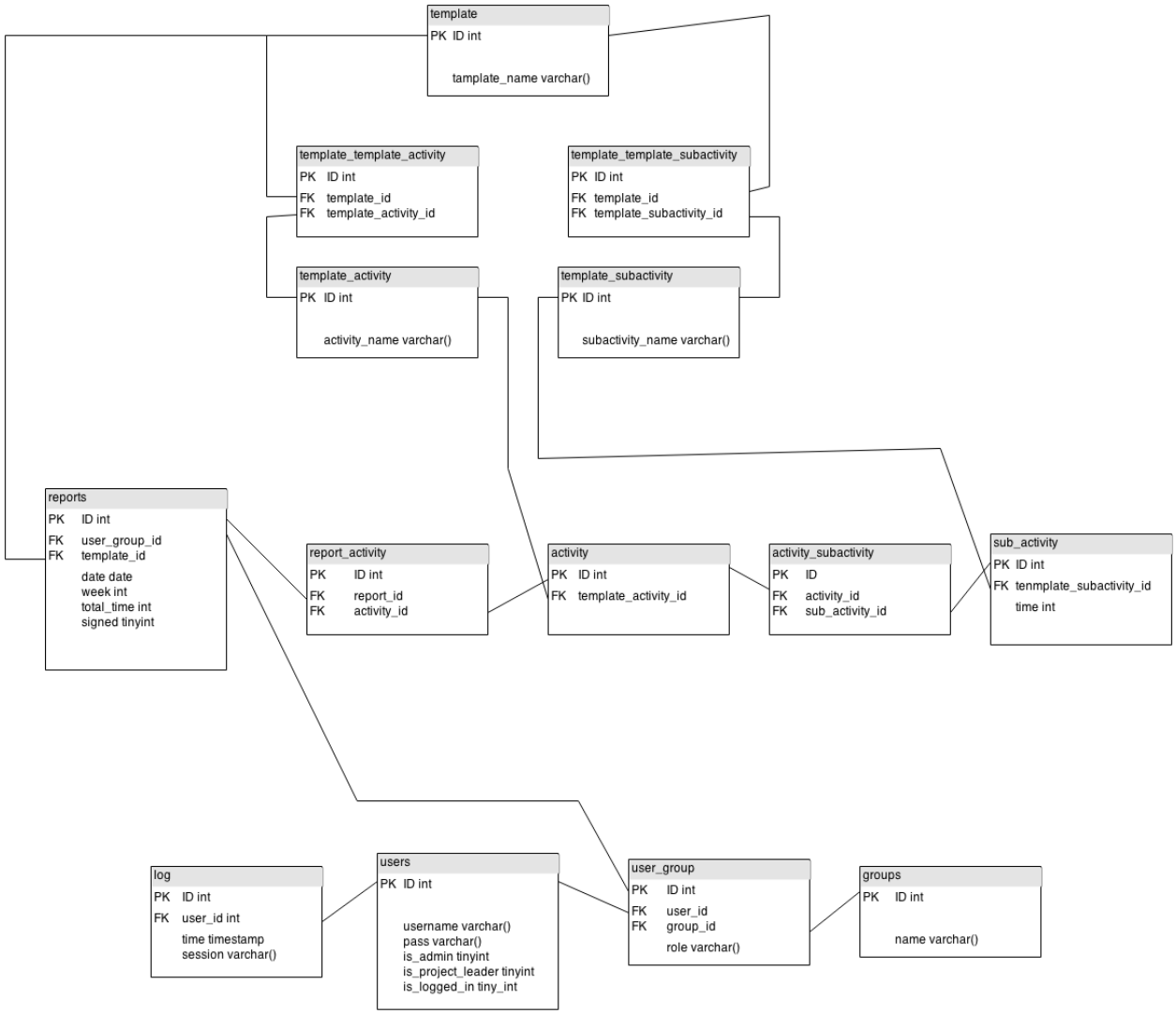
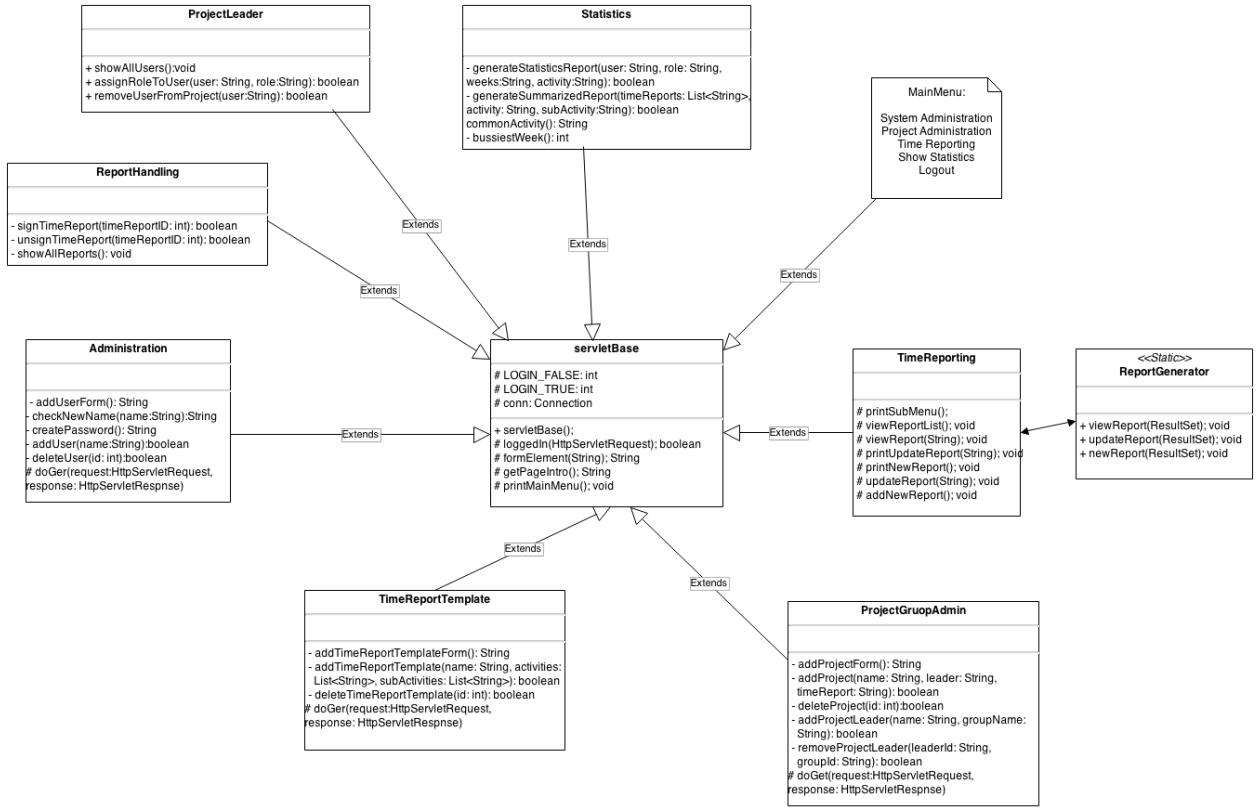

template
PK ID int
tamplate_name varchar()
template_template_activity
PK ID int
FK template_id
FK template_activity_id
template_template_subactivity
PK ID int
FK template_id
FK template_subactivity_id
template_activity
PK ID int
activity_name varchar()
template_subactivity
PK ID int
subactivity_name varchar()
reports
PK ID int
FK user_group_id
FK template_id
date date
week int
total_time int
signed tinyint
report_activity
PK ID int
FK report_id
FK activity_id
activity
PK ID int
FK template_activity_id
activity_subactivity
PK ID
FK activity_id
FK sub_activity_id
sub_activity
PK ID int
FK tenmplate_subactivity_id
time int
log
PK ID int
FK user_id int
time timestamp
session varchar()
users
PK ID int
username varchar()
pass varchar()
is_admin tinyint
is_project_leader tinyint
is_logged_in tiny_int
user_group
PK ID int
FK user_id
FK group_id
role varchar()
groups
PK ID int
name varchar()

Databasen



Uml över klasserna

Klasser och metoder:

class Administration: se text i grundSTLDD.
Ändringar: deleteUser(String name) ska returnera en boolean och inparametern ändras till (int id).

class ProjectGroupAdmin: Den här servleten ärver från **ServletBase**. Först kontrollerar den att användaren har behörighet att besöka sidan. Om så är fallet presenteras en tabell med information om varje projektgrupp (namn, projektledare, tidrapportmall). I tabellen kan administratören välja att radera gruppen och att lägga till/radera en projektledare. Projektgruppsnamnet i tabellen länkar administratören till projektledarens ändringsmeny (för att kunna utföra samma ändringar som projektledaren kan utföra. Den servleten lämnar vi åt Jonas och Patrik). Nedanför tabellen finns en meny där administratören kan välja att lägga till nya projektgrupper (ny sida), gå tillbaka till startsidan för administration, gå tillbaka till den gemensamma första inloggningssidan samt logga ut.

/*
* Constructs a form for adding project groups.
* @return String - The html-code for constructing the form.
*/

```
private String addProjectForm()

/*
* Adds a new project group.
* @param name: The name of the new project group.
* @param leader: The name of the user that will be added as project leader to the project group.
* @param timeReport: The time report template which will be used for the project group.
* @return boolean: True if the group is added succesfully, else false.
*/
private boolean addProject(String name, String leader, String timeReport)

/*
* Deletes a project group.
* @param id: The id of the project group which will be deleted.
* @return boolean: True if the group is deleted succesfully, else false.
*/
private boolean deleteProject(int id)

/*
* Adds a project leader to a selected project group.
* @param leaderName: The name of the user which will be added as project leader.
* @param groupName: The project group to which the project leader will be added.
* @return boolean: True if the project leader is added succesfully, else false.
*/
private boolean addProjectLeader(String leaderName, String groupName)

/*
* Removes a project leader from a selected project group.
* @param leaderId: The id of the user which will be removed as project leader.
* @param groupId: The id of the project group from which the project leader will be removed.
* @return boolean: True if the project leader is removed succesfully, else false
*/
private boolean removeProjectLeader(String leaderId, String groupId)
```

class TimeReportTemplate: Den här servleten ärver från **ServletBase**. Först kontrollerar den att användaren har behörighet att besöka sidan. Om så är fallet presenteras en tabell med information om befintliga tidrapportmallar (namn, vilka projektgrupper som använder mallen). I tabellen kan administratören välja att radera tidrapportmallen. Nedanför tabellen finns en meny där administratören kan välja att lägga till nya tidrapportmallar (ny sida), gå tillbaka till startsidan för administration, gå tillbaka till den gemensamma första inloggningssidan samt logga ut.

```
/*
* Construcs a form for adding a new time report template.
```

```
* @return String: The html-code for constructing the form.
*/
private String addTimeReportTemplateForm()

/*
* Adds a new time report template.
* @param name: The name of the time report template.
* @param activities: A list which contains the activities in the time report template.
* @param subActivities: A list which contains the subactivities in the time report template.
* @return boolean: True if the time report template is added succesfully, else false.
*/
private boolean addTimeReportTemplate(String name, List<String> activities, List<String> subActivities)


/*
* Deletes a time report template.
* @param id: The id of the time report template which will be removed.
* @return boolean: True if the time report template is deleted succesfully, else false.
*/
private boolean deleteTimeReportTemplate(Int id)
```

public class servletBase extends HTTPServlet

```
/*
* Prints out the main menu including the options:
* System Administration
* Project Administration
* Time Reporting
* Show Statistics
* Logout
*/
protected void printMainMenu()
```

TimeReporting

```
/*
* Prints out a sub menu including the following option:
* View Reports
* New Report
*/
protected void printSubMenu()

/*
* Prints out a list of the user's own reports.
*/
```

```
protected void viewReportList()

/*
* Fetches the data for the report specified by the String parameter and passes the data on to the static method viewReport in the static ReportGenerator class.
*/
protected void viewReport(String)

/*
* Fetches the data for the report specified by the String parameter and passes the data on to the static method updateReport in the static ReportGenerator class.
*/
protected void printUpdateReport(String)

/*
* Calls the static method newReport in the static ReportGenerator class.
*/
protected void printNewReport()

/*
* Updates the data stored in the database for the report specified by the String parameter.
*/
protected void updateReport(String)

/*
* Inserts the data from the new report form into the database.
*/
protected void addNewReport()
```

ReportGenerator

```
/*
* Prints out a report in the right format and with the data specified by the ResultSet parameter.
*/
public static void viewReport(ResultSet)

/*
* Prints out a html-form in the right format and with the data specified by the ResultSet parameter.
*/
public static void updateReport(ResultSet)

/*
* Prints out a html-form with the right format specified by the ResultSet parameter.
*/
public static void newReport(ResultSet)
```

ProjectLeader: Visar automatiskt en lista över alla användare. Här ifrån kan projektledaren antingen välja att antera användaren i listan (lägga till, ta bort ur grupp) eller välja andra funktioner.

```
/*
* Shows a list of all the users in the system
*/
public void showAllUsers()


/*
* Adds the user with the String user parameter to the project with the String role parameter. Returns *
True if the user was successfully added, else False.
*/
public boolean assignRoleToUser(String user, String role)


/*
* Removes a user from the project group with the String user parameter. Returns True if the user was *
successfully removed, else False
*/
public boolean removeUserFromProject(String user)
```

ReportHandling : Visas om man väljer att hantera tidrapporter. En lista med signerade och en lista med osignerade. Kan här signera eller avsignera rapporter.

```
/*
* Signs the time report with the int timeReportID parameter. Returns True if the time report was *
succesfully signed, else False.
*/
private boolean signTimeReport(int timeReportID)

/*
* Unsigns the time report with the int timeReportID. Returns True if the time report was successfully
*unsigned, else False
*/
private boolean unsignTimeReport(int timeReportID)

/*
* Shows a list of all the time reports
*/
private void showAllReports()
```

Statistics: Visas om man väljer att generera statistik. Här kan vissa funktioner väljas för att visa t.ex.

vilken vecka som mest tid har gått åt under.

```
/*
* Shows a time report based on the parameters requested by the user. Returns True if the report was
* succesfully generated and shown, else False.
*/
private boolean generateStatisticsReport(String user, String role, String weeks, String activity)


/*
* Shows a time report with the summarized activity from several time reports. Returns True if the time *
report was succesfully generated, else False.
*/
private boolean generateSummarizedReport(List<String> timeReports, String activity, String subactivity)

/*
* Returns a String of the activity that has the most combined minutes reported by the users in the        *
project group.
*/
private String commonActivity()

/*
* returns the week with the most combined minutes reported by the users in the project group.
*/
private int busiestWeek()
```