Group 25's problems with the Web Server and database structures.

We have got access to Michael's server through the application VNC Viewer and can access it from other location.

Through the viewer we installed XAMPP which includes Apache Web server, MySQL (with PHPMyAdmin) and some other applications and services that we have chosen not to use (such as Filezilla server, Mercury and Tomcat).
After the installation we had problems starting the Apache Web Server and after a lot of hours of frustration and reinstallation with errors like shutting down directly, problems with ports or giving us error codes (which was of no use and could not really be traced to the root cause) we found out that the control panel had to be started in Admin mode. This was the solution to all of it, even the port problem.

When we finally got the Web server working none of us had any experience in databases and didn't even know if we should use the PHPMyAdmin in the root folder or copy it to the "appsolutdistribute" folder in "htdocs". Since PHPMyAdmin didn't work well when copied into "htdocs", we created a new database "appdist" in the root and then started to create tables for users, groups and features. Because of the lack of experience in databases, Mattias talked to a friend of his and got some information in how to think regarding databases and the importance of defining the relations before tables are created. This was only sketched as an example on a sketchpad over a couple of beer in a bar.

As we knew more or less how databases worked, meaning normal tables for users, groups and features and junction tables (many-to-many) for users-to-groups (many users can be in many groups and groups can have many users) as well as between groups and features, we started to redesign the tables for users, groups and features.
To test if we could add data, we built a web page in PHP, HTML and CSS and PHP files including SQL Queries mainly taken from PHPMyAdmin in the beginning, but had to be configured to match the data we wanted to have. This worked as tests, to see that it worked to insert data in the database as well as retrieving data.

When we saw that it worked writing data and reading data, we started to find out how we could create the junction tables from examples on the internet, which we then modified in order to get the tables names we wanted, correct column names as well as foreign keys and relation to correct tables.

Then we have been adding additional PHP/MySQL queries to insert data and PHP, HTML, CSS for retrieving data.

We have now started to list all types of queries that our app require and have also decided our architecture which is described in a separate document.

Learnings:

1. We should always start the Web server in Admin mode.
2. That we can create a database with PHPMyAdmin in the root folder.
3. The web server can be accessed from internet directly (port 80)
    - http://95.80.48.213/appsolutdistribute/index.php
4. Learned PHP more.
5. Learned how SQL works with PHP.
6. How database tables are created and how many-to-many relations are managed with junction tables.
7. Learned how you can nest SQL queries in order to get information from junction tables and then fetch the complete data from another table.

Author: Mattias Isene 2012-10-16

- Example: Getting all users that are member of a specific group and then get all their information from the user table.

Further work:

- Need to create all SQL queries that we need for the Client ←→ Server ←→ Web server solution.
- For all queries specify the input parameters when inserting data into the database as well as when retrieving to get the correct data.

Future possible issues:

- As of this moment, we are not sure we can access the database and use “normal” SQL queries directly from Server Java Application or if we should use HTTP connection to access PHP/MySQL queries from the Web Server in order to write/read data.
    - If Server Java Application instead of via Web server, we are not sure if we should directly enter SQL queries from the server application or if we should write .sql files and access invoke them to write/read data.
- We are not sure about the security, if we might have any security issue and might have any intrusion in the Web Server or if it is secure enough.

Future possible improvements:

- Since we have tested a lot with PHP/MySQL, we actually are beginning to get some functionality that we could extend to a web portal and add the possibility for users to also login via web and collaborate with their groups from the web instead of from their Android Client.