Task Management System – Full-Stack Web Technologies Project

Developed by “**Team Whatever**”

Team members:

Emiljano Gjiriti

Elio Kolici

Nikolay Zahariev

# **Project overview:**

Our team developed a service to help people, mainly students, organize their task schedule. By tasks we mean activities that the person in question takes part in and which are defined by one of the following categories: reading, practicing, problem solving, test preparation, subtask, or other. A user can take part in any number of these tasks and can even work in a group with other people on a given task.

The service supports the following functionalities

* Entry of personal user information such as first and last name, username, password, email, phone number, main field of study, institution. The password and username will be used for authentication into the system
* User can and must log in to the system in order to tinker with his schedule.
* When the user loges into his profile, he/she can then sign up for any number of different tasks as well as remove tasks currently in his schedule.
* The user can also send group invites to other users and join in groups with them based on same tasks undertaken by group members
* The user can also filter tasks he wants to join or drop based on several criteria such as task name, its duration, priority, and so on.
* User can also filter out other users to pick only those he wants to invite to a group

The Back-End of our application was written in PHP while the Front-End was done with a combination of HTML, CSS, JavaScript, and AngularJS. The Back-End testing was done with PHPUnit but were largely unsuccessful due to the reasons either of technical difficulty or poor knowledge of PHP testing.

Furthermore our team made use of the Agile method of software development. This means we maintained a high-level of communication between team members with a team meeting at least ones per week (during which all team members attended).

All the project files were uploaded to a shared GitHub repository. Our team also used Travis CI for our continuous integration. However, since, as mentioned above, the unit tests for our program don’t function correctly the Travis integration gives off a failure.

# **SQL Database Schema:**

For our project we will need several tables in our database:

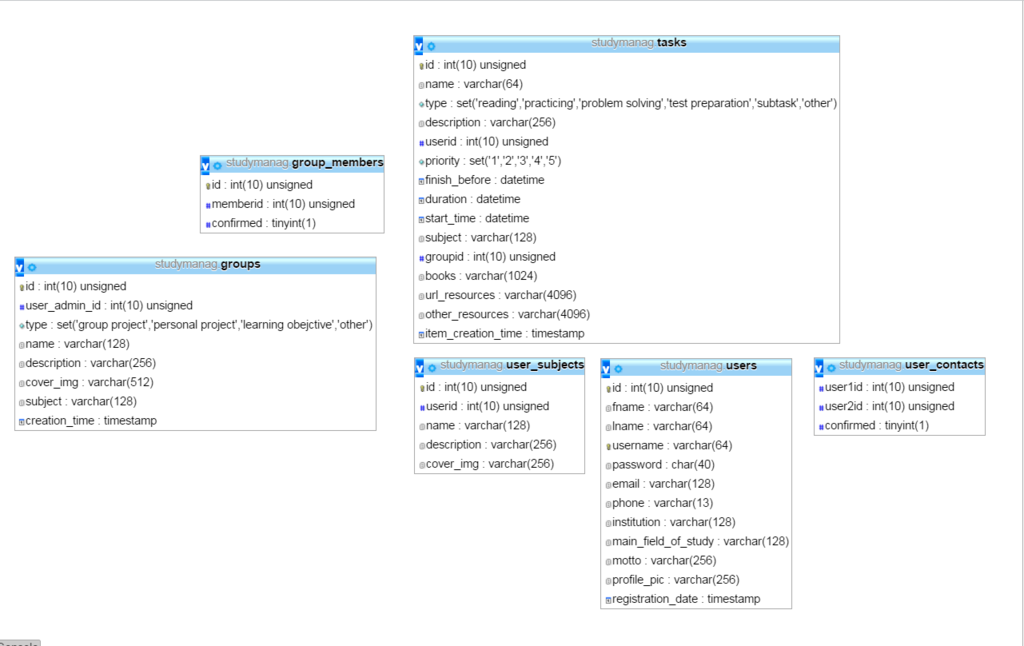
- First of we will one table to contain all our user information: his name, ID, email, password, username, phone number, institution name, main field of study, and registration date.

- Next we will need a table to hold info on the diferent tasks a user can take: their name, priority, duration, description, type, subject, and so on. This table will be linked to the user table through the users’ ID.

- Then we need one table to store data on the groups a user can be part in: the ID of group, its type, name, description, and the admin of that group (denoted by user ID).

- We will also need a table to hold info on user contacts with other users: the ID of the given contact, the ID of the first user, and the ID of the second user, and whether both users agreed to keep this contact.

- Another table that is required is the “Subjects” table which tasks (denoted by the task ID) that are part of a particular subject. This table will also be linked to the user table (by the user ID).



# **Work distribution:**

All team members contributed equally to all parts of the project. Apart from the regular team meetings during which every team member was present, we divided the work for the project into three main parts:

- Nikolay Zahariev was mainly responsible for creating the SQL database (not its design), as well as writing the PHP tests with PHPUnit.

- Emiljano Gjiriti was generally responsible for creating the Front-End of our application as well as writing some unit tests for it.

- Elio Kolici was mainly responsible for creating the Back-End of the service and largely coming up with the general idea of our project.

All team members contributed to the creation of the project idea, the creation of the database ER diagram and schema, the testing, as well as in the creation of the documentation.