Abstract

In any project that involves multiple people, coordination among its members may very well be lacking. And by the end of the project it is not always clear how much time have been spent by each individual and on what task. This project aims to solve these problems by introducing a Project Managing System to help its users to share information involving the project and give a clear view of what they have ahead of them in the coming days. The final result is a functional web-based product enabling easy access and planning.

1 Introduction

The purpose of this project is to create an application where you can keep track of meetings, deadlines and other things in one or more projects. Today when meetings are planned and you get to know the time and place for the meeting, you may fill in your own personal calendar, and so must everybody else in the project. You might use applications such as Microsoft Outlook or other vendor dependent software suites.

The Project Management System solves this with a project calendar that contains all this info and is available for all members in the group. It’s available to everyone because it’s web-based and all you need is a web browser to access the Project Management System. When something is planned, either a meeting or a deadline is decided and any user may enter this into the application.

The application will also contain the ability to report hours worked on different projects and even keep a work log with info about what you have done during that time. This information may be interesting when invoices are produced.
The Project Management System will hopefully give project groups a better coordination of its members and give the users a clear overview of what to achieve during a project. The ability to have a shared calendar among the users in a project will also facilitate the handling of setting up meetings.

2 Approach

In the planning phase of the project it was decided that the database would be built in PostgreSQL\(^1\) and the user interface would be made in HTML using PHP\(^2\) to fetch and process information from the database. After the planning was done, it was time to design the database and web interface.

To illustrate the database design, an EER diagram (Extended Entity Relationship diagram) was created. The final version of this diagram can be seen in figure 1. Note that this is a simplified version without the attributes to avoid a cluttered image. The main parts of the database are the users, projects and calendar. Users are persons working on one or more projects, where each user has a calendar to put his/her work hours in. A project however, uses the calendar to store the important dates of the project. As can be seen in the figure, the calendar is a central part of the system with several specializations such as the time report which is the work log of the user.

When designing the web interface, sketches were drawn for each page that would be in the system. These pages were tied together into a kind of flow chart so that it would be easy to know what kind of data to expect and show in each page. The two types of users for the system would be the regular user, who wants to know what happens in his projects, and the administrator, who would be able to create projects and assign users to them. Since these two types wouldn’t have any functions in common, the interface had to be split into two parts. This division would make the implementation easier as well, since the work could be more easily divided between people. The choice was made to have a single page for logging into the system since this would be more user friendly than having separate pages for users and administrators.

3 Results

The result is fully functional web-based software supporting multiple users working on any number of projects at the same time. It gives the regular user an easy to follow overview of the current and following days, showing a

---

\(^1\)PostgreSQL website: http://www.postgresql.org

\(^2\)PHP website: http://www.php.net
The system also gives the user an easy way to report how many hours he/she has worked on any particular project. Any user can easily add/modify/remove events, such as meetings or demonstrations, as they come up in any project they are working on, reflecting these changes on to all of his/her co-workers.

The system administrator has an equally easy to use interface where he/she can add new users and projects to the system and assign users to the projects they are working on or are going to work on.
4 Discussion

The big problem when designing the database was how the calendar was going to be represented. Things like what the calendar parent table and all its children should and should not contain, what should be in the project table and what should be in the calendar and other logical issues took some time to figure out. Once the database design was set, the implementation of it was pretty straight-forward, with some small changes to correct some small misses in the design.

When it came to implementing the web interface, most of the members of the group were familiar with PHP already, and for those who weren’t, it was easy to grasp since its syntax is similar to that of C. PHP has a lot of useful functions like date and time conversions, string manipulation and number conversion functions that were used quite a lot in the implementation. One problem while coding the interface was that it was difficult keeping the HTML and PHP code separately. The HTML files ended up having almost as much
PHP code as HTML in them, which wasn’t really what was desired.

Some things had to have less priority in this project. One of the original features was that invoices could be created based on the number of hours the users had worked on a project. This feature had to be left out though, since the other content took more time than expected. Another thing that was planned was an extra feature allowing for categorization of users based on their job description, such as coders or project managers. This feature also had to be postponed for a future version of the system. The security of the system have not been analyzed and there are probably some loopholes that may have to be fixed.

To sum up, the database is well designed, so that most of the improvements of the system will only have to do with the interface.

5 Conclusions

We now have a functional project management system that has most of the features we set out for it to have. With a little further work we think our solution could come to some practical use in the right environment. During its development we’ve experienced how easy it is to use PHP to work as a front-end to a database. We’ve also learned how to apply databases to a real life problem, something which is useful in many different areas.