Schedule Maker

Richard Lindberg, Annelie Dammander and Dennis Ålund
{tm01rlg, c01ajn, dva98dad}@cs.umu.se
Department of Computing Science, Umeå University, SE-901 87 Umeå, Sweden

Abstract An issue with the existing schedules for different courses is the fact that they are created with different templates and this leads to a set of different appearances, some of which are very hard to read. A web interface for easy creation and modification of unified schedules is needed. This leads to less work for the teachers creating schedules and simple and consistent schemas for students trying to read them.

1 Introduction

The idea of this project was to create a solution for simplifying the procedure of creating a schedule for a course. Also, we wanted to push for a good standard on schedules at course pages. Sometimes the present schedules are bad formatted, inconsistent and not up to date. All of these problems are treated and presented with a suggested solution in our system.

We have created a standard layout for schedule presentation, an assisted procedure of schedule creation and an interface for easy editing and maintaining of a schedule.

Our visions are that this system should be integrated with CS course pages on the web. Also, that login information were connected with each teachers CS user. This would simplify the maintaining of user information and password keeping.

If the system also could be connected to Husservice room booking system, the procedure of selecting class rooms would be more accurate.

Source code for the project is available in any of the group members home catalogs with the extension `edu/db2/tdbd15_vt05_g02/project`

2 The Database

The problem with a schedule is that it seems to consist of just a big table with rows and columns, when it in fact is built up on several different types of events and rather nested relations. Final table distribution can be found in the file `project.sql`

2.1 EER

We started out by designing an EER diagram for our project, depicted in figure 1 on the following page. The final result tables in the database does not correspond exactly to the model we first created. This due to complications regarding table implementation in the database.
Figure 1. EER diagram over Schedule Maker
2.2 Database Conclusions

We decided to categorize the different types of scheduled events into three types. Each row in the schedule is either a lecture, event or exam. An event is a type of entry that has no start- or end time, rather just a time of occurrence. For example, due dates on assignments.

A lecture can be held in zero or more class rooms and be held by one or more lecturers. This covers the situations of guest lecturers and activities held in more than one class room.

For each lecture there can be any number of additional topics, such as readings, links and similar. Each topic will be presented in the last column of the schedule at the web page.

All teachers participating in any relation of the schedule has to be represented in the teacher table. The drawback of this is of course that no external teachers, outside of CS institution, can be represented in the teaching relation. Instead this has to be solved with an additional topic for guest lecturers.

To facilitate the queries we created a view of lectures, events and exams merged into a table where all common fields are represented in the same columns. By doing this we could select every row in the, presented schedule, in a uniform way. There are also views for lecture- and course info. In the case where web user privileges can be configured, the views fill their purpose by preventing visitors from selecting directly from the tables, hence preventing some risk of unintended use of the database.

The login procedure is done by a stored procedure, simply returning true or false. The idea is that a anonymous web user shouldn’t have any privileges in the database, other than selecting from views and calling some procedures. This will hopefully avoid the risk of someone trying to extract user information from the database.

3 The Web Interface

Presentation of the result is the core of our project. The main objective was to create a good standard for schedules that is easy to read. We also wanted to make the schedules easy to maintain and thereby encourage teachers to keep the schedule up to date.

3.1 Making the Schedule

When a schedule is about to be created, certain basic information is required, such as which class room will mainly be used, what days of the week, time and so on. This is just for the creating procedure to get a rough idea of how the teacher will be holding lectures. A preliminary schedule will automatically be created with lectures placed between the course’ start- and end date.

The automatic creating procedure will try to assign the desired room for each lecture. Although if the room already is booked on a given time of lecture, this lecture will be assigned a “None” representation as assigned room. This can later be changed in the “edit” section.
As an assumption that no courses are held at late hours, we chose to restrict scheduling hours to be within the time span 08-20. Also it is only possible to make a preliminary schedule with lectures on Monday through Friday. Adding activities on Saturday or Sunday or events such as “exercise due 23.59” can be made in the “edit” section.

3.2 Editing the Schedule

When adding new lectures or events, the date has to lie within the boundaries of start- and end date of the course. Exams can be scheduled at any time, any year, for a course.

For the moment there are no restrictions in the web interface for entering invalid data. This is the result of the limited support, in PHP, for dynamic forms with interactive control of entered values. The restrictions are enforced in the database by constraints set to the tuples. An erroneous value will cause the inserted row to fail, an error message is displayed and the user is prompted to try again.

4 Future Features

For the future we have examined the possibilities of exporting schedules to PDA’s and mobile phones which is quite possible through formats like vCalendar and such. This would be a very nice feature that would enable students to receive their usual notifications and reminders automatically for lectures and events during a course without the hassle of pushing in the data in the regular way.

We would also like to integrate this system with "Husservice" and their room database to enable teachers to receive assistance in choosing available and suitable rooms for the course and number of participants. Another feature that might be interesting to some teachers would be the possibility to change the templates and styles for schedules, as such show some kind of priority of the different tasks, like obligatory versus optional lectures.

5 Summary

Implementation of the web interface consumed more time than expected. Some of the features that first was intended had to be left out. Other features depended on information we did not have, such as room sizes, courses and teacher information.

The system works as intended. One can create and change schedules in an easy way. The administration is not fully implemented, so with some modifications this system could easily be put into use.