Room Booking System

Jonas Sidborn, Daniel Wiberg, Jan Öhman, and Per Öhman

Department of Computer Science
Umeå University, Sweden
{c01jsn, c00dwg, c01jon, c01pro}@cs.umu.se

Abstract. The management and booking of rooms in large buildings is a tedious and complicated task. The combination of different rooms with different sizes and equipment, the possibility to choose between different dates and times is the reason why a good booking system is needed to make this task as easy as possible. A system that supports reservations of rooms in conference buildings or universities or other large buildings. This article shows how a complicated human task like reservations can be solved by using a database management system as the key tool in the development. The prototype shows an example how such a system could look like.

1 Introduction

The management and booking of rooms in large buildings is a tedious and complicated task. The combination of different rooms with different sizes and equipment, the possibility to choose between different dates and times is the reason why a good booking system is needed to make this task as easy as possible.

The basic functionality of the system is to keep track of rooms in different buildings, rooms and their equipment, reservations of rooms and different types of users. Reservations of the rooms should be visualized to the user in a simple and intuitive way. There should be a simple way of getting an overview of the reservations. The system should be interoperable with any platform and OS.

A Booker should be able to reserve and cancel his/her own bookings and should be able to search for a room with a certain property like size, price, number of seats and equipment. There should be an administrator that can administrate the whole system through the interface without writing any SQL-queries. This includes the management of add, edit and delete functions for every entity in the system including new users and their privilege management. One type of user should only be able to look at existing reservations and not be able to reserve rooms.

The system should be a complete content management system for handling the reservations of rooms, buildings, equipment and users. The interoperability of the system should make it easy to access and use from different platforms.
2 Approach

The work began by contacting the janitors at “Hus service” and an appointment was scheduled for an evaluation of their existing system. The janitors showed how their existing system worked and the process of reserving a room for a particular course at an institution. They got a schedule from the secretary of the institution where all the lecture hours were filled in and they showed how they chose rooms for the whole schedule.

With the evaluation of the existing system several sketches was drawn of the interface and basic functionality and user scenarios. The sketches together with the requirements of the system was an excellent background to make the EER-diagram of the database that was needed to perform this complicated task. A relational schema was then created using the EER-diagram that can be seen in figure 1.

The focus on the system is on the rooms and the users which makes the reservation of rooms. To get interoperability it was decided that the system should be web-based. Because of this, HTML together with an embedded scripting language PHP\(^1\) was used. The back-end of the system was chosen to be the

\(^1\) http://www.php.net

Fig. 1. The figure shows the EER-diagram of the database.
relational database system named MySQL\(^2\). This database system was chosen because our web-hotel, b-one\(^3\) only supported MySQL.

A distribution of all the work and an agreement on how to put the system together was done and all code was put in a repository at the web-hotel using a common FTP-server. Then the implementation phase began with a lot of testing.

Authentication to the system is done through a log on procedure where the user writes his/her user name and password. The user name and password is then verified against the database. If the log on succeeded the user will receive a privilege that the system has given to the user. There are three different types of privileges which are viewer, booker and admin. The viewer can look at reservations and search for rooms, the booker can reserve and cancel his/her own reservations and the administrator can manage the whole system. The privilege that the user got determine what kind of access to the system it will have. The management of the log on session is handle by the PHP built in functions.

3 Result

The system that has been created is a very good prototype, almost a complete room booking system. It has functions for handling users with different privileges, it shows the reservation of rooms in an intuitive way and has a fully administrative web-interface. The schedule over the reservations (week-view) can be seen in figure 2 which is an easy to use interface for managing room reservations. The basic idea is that a gray color is shown when a room is occupied and an orange color is shown if the reservation is made by the current logged on user. And to make a reservation the user just fills in the check boxes for the actual time and date to reserve and then pushes the book button to confirm the reservation.

A user that don’t know which room to reserve can use the room search function. This search function can search for rooms in all buildings or in a specific building. The user then specifies what requirements he/she has on the room e.g. maximum price, minimum number of seats or what kind of equipment the room must have. The resulting rooms from the search can then directly be connected to the week-view making it easy for the user to make a reservation on that room.

The system has an complete administrative interface. The administrator can manipulate the whole database from the web-interface. Functions for adding, deleting and editing existing or new tuples for each table exists. These functions makes it easy for an administrator to manage the system. The administrator can also reserve and cancel reservations for any user that has the privilege to reserver rooms. As an administrator it is also possible to get information about the reservations done by other users through the reservation search function. One limitation of this search function is that it is not possible to search for reservations over a period of days, only on a specific day. This limitation is due to lack of time and could easily be fixed.

\(^2\) http://www.mysql.com
\(^3\) http://www.b-one.net
4 Discussion

The design and requirement analysis phase of the development was a very important foundation to start the implementation on. This made it very easy to create the user interface from the sketches and distribute work among group members which lead to a quite good prototype.

We have had some trouble to get the week-view to work in different browsers. It does not work in IE but works fine in Mozilla/Firefox and Safari. It was a limitation in our relational database MySQL. The version we used did not support foreign key constraints. This limitation can depend on the version that we used did not have the innoDB function activated. Another thing that we noticed while developing was that it is quite hard to separate the HTML code from the PHP code. The built in function library in PHP contained functions for session and database management that we found very useful.

One thing that was not prioritised, and because of that not implemented was the notification by email to the user confirming the reservations made by that user. Even though we used MySQL as the database system it would be fairly easy to use another system like PostgreSQL\footnote{http://www.postgresql.org}.

Some future work with the system is to extend the capabilities to present more statistics to the users and to handle accounting information for invoices. This could be solved by connecting a customer register to the reservations. A
history of all the reservations can also be added to give the administrator better control over the system. This can easily be added by appending $TST$ and $TET$ attributes to the reservation table. Of course the last thing that needs to be done is to let real users evaluate the system and then improve the system according to their opinions.

5 Conclusions

We have shown how to build a room booking system with a database as the back-end and a front-end in HTML embedded with PHP. It is easy to make room reservations in the graphical user interface and the week schema gives a good overview of the reservations for the room. The combinations of the search functions, the ability to reserve for any user and the ability to manage the whole database from the web-site made this system a very powerful administrative room booking system. The security model used based on different privilege for different users also makes the system a real-life application.