

COMP3760 PROJECT S1 2008

SIMPLE ROSTERING SYSTEM

PROJECT REQUIREMENTS SPECIFICATION

BEN GRIFFITHS (4302927)

OVERVIEW

Simple Rostering System (SRS) is a web-based rostering system that is targeted toward simple rostering tasks such as odd jobs around the home, office or in community groups. It will provide a simple, flexible, and accessible web-based interface through which users can create & modify rosters (if authorised to do so), as well as view them and configure notifications. The system will feature email confirmation when a user is added to a roster, and private rosters will need to be logged in to using an authorised email address and associated password (as will roster administration and notification options).

BACKGROUND

ROSTERING SYSTEM

As rostering system is a tool for generating a schedule of activities which a group of individuals participate in on a regular basis. Participants in a roster will usually rotate or 'take turns' completing an activity, and the roster assists them in determining who is responsible for completing the activity whenever it is called for.

Roster: "a list or plan showing turns of duty or leave in an organization."

- Compact Oxford English Dictionary (via AskOxford)

WEB-BASED SYSTEM

A web-based system is a software system with which the end user interacts via web services (such as through a web browser) over the Internet. Web-based systems that are delivered via a web browser are accessible from any device that has a web browser and Internet connectivity. Web-based systems suffer from the limitations of common web browsers, which include very limited interactivity and constraints on layout & presentation. This results from the necessary use of standard mark-up such as HTML, albeit with some enhancements offered by languages such as CSS and JavaScript.

TECHNOLOGIES

HTML

Hypertext Markup Language (HTML) is a language for marking up the content of a text document so that a web browser can display it. HTML is specified by the World Wide Web Consortium (W3C). The specification that will be used for the SRS web interface is XHTML1.0.

<http://www.w3.org/TR/xhtml1/>

CSS

Cascading Style Sheet (CSS) is a specification for formatting documents that use mark-up languages such as HTML. It allows web developers to separate the presentation of their web documents from the content, and alter the presentation of a whole set of documents from a single file. Both of these features are highly desirable for the SRS web interface. CSS is specified by the World Wide Web Consortium (W3C). The specification that will be used for the SRS web interface is CSS 2.1.

<http://www.w3.org/TR/CSS21/>

PHP

Hypertext pre-processor (PHP) is a scripting language which is interpreted by web servers to generate web pages. It is a popular language for generating dynamic content in web pages. As well as basic dynamic content, PHP also facilitates interaction with a database server. Other common applications using PHP include scripts for sending out emails and user authentication.

As with any server-side scripting, there are security concerns that need to be addressed to protect the web and database servers.

<http://www.php.net/>

PHP is a flexible language, and has very little structure built into it. This will mean that a structure will have to be devised for the system that is modular, flexible and efficient. The *Joomla!* content management system will be studied as a basis for such a structure, as well as any security measures applicable to it. For information on why *Joomla!* has been selected for study, see the section entitled *Related Systems – Web-based System (PHP/SQL)*.

MYSQL

MySQL is an open source database software package that can be used in conjunction with PHP. Dynamic web page scripts can interface with a database server to run queries and store/retrieve data. Queries to the MySQL database(s) use the Structured Query Language (SQL) and can be used to request, modify, add or remove data. SQL queries can also be used to create and erase (or 'drop') database tables.

<http://www.mysql.com/>

These technologies will be combined in the SRS system to provide an efficient, reliable, and accessible rostering system delivered through a web interface and via email.

To illustrate how these technologies will interact, take the simple example of a user viewing a

roster. The PHP will receive the request and load the template for a roster web page. The PHP will then retrieve the details of the requested roster from the MySQL database and use them to fill out the template. The completed template will then form a HTML document which is passed to the user's browser. The HTML document will contain a reference to the applicable CSS file(s), which the browser will download and use to display the HTML document to the user as intended.

RELATED SYSTEMS

PAPER ROSTER

This rostering method is quite simple, and is used for similar purposes to the planned SRS system. Paper rosters can be generated quite efficiently with some spreadsheet applications, but are still limited in terms of accessibility. This is due to the necessity for participants in the roster to travel to and physically view the roster (wherever it may be posted). It also adds printing costs to roster production.

Rosters generated using these systems are usually either authored by hand or using spreadsheet software. Neither of these methods is ideal.

MANUAL EMAIL ROSTER

Rosters are sometimes not formally recorded (if at all) and participants are emailed by the person responsible for managing the roster when something is required of them. This method relies on the coordinator of the rostering remembering (or recording) who is scheduled to do what when, and for remembering to remind participants when it is their turn on the roster. Communication is limited, as participants can only view the roster if the supervisor provides it to them.

COMMERCIAL ROSTERING SOFTWARE

Most of these systems are very complex as they are intended for staff rostering, and have payroll and skill-level features. These systems are advanced and far more efficient than any other method for their purposes; however the complex interface means the system has a steep learning curve and makes the system inefficient when only simple rosters are required.

One such system is Oriador Rota (<http://oriador-staff-scheduling.com/>) which features much of the complexity described above. Another is VersaERS (<http://www.versadev.com/versaershome.aspx>), which features a web-based interface that mimics an application. This means that it doesn't follow standard web page conventions and is similar in complexity to Oriador.

WEB-BASED SYSTEM (PHP/SQL)

These systems feature dynamic web pages generated by PHP scripts (with access to one or more databases using SQL queries). Desirable capabilities offered by PHP/SQL systems include user accounts and the ability to store & retrieve rosters. Typical applications include forums, news websites, and e-commerce websites.

An example of this type of system is *Joomla!* (<http://www.joomla.org/>). Joomla! is a flexible open source Content Management System which uses a highly extensible platform built with PHP and SQL. As a community developed system that has been through numerous revisions, Joomla! is a

good example of a flexible, modular, secure and efficient PHP/SQL system. More detail regarding the good elements of Joomla!'s design and how they will be applied in the SRS will be available in the Design section of the final Project Report.

SRS will be used for similar applications to the paper roster system, but will combine some of the functionality of commercial rostering systems such as Oriador with the technology of PHP/SQL web-based platforms such as Joomla! The result will be simple, portable, flexible, and accessible rostering system.

REQUIREMENTS (SCOPE)

CORE FEATURES

The following features constitute the core functionality of the SRS:

- User accounts based on email addresses
- Ability for authorised users to:
 - Create and store custom rosters
 - Invite participants to rosters via email
 - Modify rosters that they have created
 - Set repeat interval of rosters and alter individual instances
 - Optionally roster activities at particular times
- Ability for all users to:
 - View rosters they participate in and/or administer
 - Submit availability for a particular roster in which they participate to its administrator
 - View and modify account information such as display name and password

DESIRABLE FEATURES

The following features will be added to the system as the time frame permits:

- Advanced roster scheduling (first Thursday of every second month, etc)
- Advanced print formatting (using CSS) and PDF output
- Automatic roster recommendation based on availability
- Multiple tasks per roster (ability to assign administrators to individual tasks)
- Ability for participants to request swaps with other participants
- Ability to generate roster calendars over any time period (eg. weekly roster for the year)
- Ability to create rosters with no time or date associated with the activities
- Multiple participants assigned to a single instance of an activity

SCHEDULE

OVERVIEW

Week	2	3	4	5	6	7	8	9	10	11	12	13	14
Tasks													
Project/Supervisor Selection	Red												
Contract Finalised		Red											
Initial Report & Presentation			Red										
Project Report			Red	Red	Red	Red	Red	Red	Red	Red	Red	Red	Red
Prototyping				Green	Green	Green	Green	Green					
Database Design/Modelling				Blue	Blue								
PHP Code Design/Modelling					Blue	Blue							
Interface Design						Blue	Blue						
Prototype & Report Outline								Red					
Database Implementation						Green	Green	Green					
PHP Code Implementation							Green	Green	Green				
Interface Implementation								Green	Green	Green			
Testing & Evaluation						Blue	Blue	Blue	Blue	Blue	Blue	Blue	Blue
Finalize DB/Code/Interface											Green	Green	
Final Presentation												Red	
Final Submission													Red
Tasks													
Week	2	3	4	5	6	7	8	9	10	11	12	13	14

DUE DATES (FROM COMP8700 WEBSITE)

Week 4: Initial Presentations - includes overview of project topic and project plan timetable

Week 9: Mid project result (including prototype, outline of final report) due to supervisor(s).

Week 13: Final Presentations - includes demonstration where applicable

Projects due on **4pm Friday 13 June** (2 hardcopies of final report plus softcopy of report / final presentation slides / project artefacts to projects over-coordinator)

REFERENCES

V4 Solutions Limited, 2007, *STAFF SCHEDULING SOFTWARE, ORIADOR STAFF ROTA*, Victoria Place, Carlisle, UK, viewed 20 March 2008,
<<http://oriador-staff-scheduling.com/>>.

VersaDev, 2008, :: *VERSADEV.COM : HELP DESK, CRM, E-GOVERNMENT AND INTRANET SOFTWARE SOLUTIONS* ::, Gilbert Street, Adelaide, AUSTRALIA, viewed 20 March 2008,
< <http://www.versadev.com/versaershome.aspx/>>.

OpenSourceMatters Inc., 2008, *JOOMLA! – WHAT IS JOOMLA! ?*, New York, NY, USA, viewed 20 March 2008,
<<http://www.joomla.org/>>.

W3C, 2002, *XHTML 1.0: THE EXTENSIBLE HYPERTEXT MARKUP LANGUAGE (SECOND EDITION)*, Cambridge, MA, USA, viewed 20 March 2008,
< <http://www.w3.org/TR/xhtml1/> >.

W3C, 2007, *CASCADING STYLE SHEETS LEVEL 2 REVISION 1 (CSS 2.1) SPECIFICATION*, Cambridge, MA, USA, viewed 20 March 2008,
< <http://www.w3.org/TR/CSS21/>>.

The PHP Group, 2008, *PHP: HYPERTEXT PREPROCESSOR*, viewed 20 March 2008,
<<http://www.php.net/>>.

MySQL AB, 2008, *MYSQL :: THE WORLD'S MOST POPULAR OPEN SOURCE DATABASE*, Cupertino, CA, USA, viewed 20 March 2008,
<<http://www.mysql.com/>>.

Oxford University Press, 2008, *ASKOXFORD: ROSTER*, Great Clarendon Street, Oxford, UK, viewed 20 March 2008,
< http://www.askoxford.com/concise_oed/roster?view=uk>.